Summer 8-2020

# Towards Dynamic Vehicular Clouds

Aida Ghazizadeh
*Old Dominion University*, aghaz004@odu.edu

# TOWARDS DYNAMIC VEHICULAR CLOUDS

by

Aida Ghazizadeh
M.S. December 2014, Old Dominion University

A Dissertation Submitted to the Faculty of
Old Dominion University in Partial Fulfillment of the
Requirements for the Degree of

DOCTOR OF PHILOSOPHY

COMPUTER SCIENCE

OLD DOMINION UNIVERSITY
August 2020

Approved by:

Stephan Olariu (Director)

Ravi Mukkamala (Member)

Michele C. Weigle (Member)

Dimitrie C. Popescu (Member)

## ABSTRACT

# TOWARDS DYNAMIC VEHICULAR CLOUDS

Aida Ghazizadeh
Old Dominion University, 2020
Director: Dr. Stephan Olariu

Motivated by the success of the conventional cloud computing, Vehicular Clouds were introduced as a group of vehicles whose corporate computing, sensing, communication, and physical resources can be coordinated and dynamically allocated to authorized users. One of the attributes that set Vehicular Clouds apart from conventional clouds is resource volatility. As vehicles enter and leave the cloud, new computing resources become available while others depart, creating a volatile environment where the task of reasoning about fundamental performance metrics becomes very challenging. The goal of this thesis is to design an architecture and model for a dynamic Vehicular Cloud built on top of moving vehicles on highways. We present our envisioned architecture for dynamic Vehicular Cloud, consisting of vehicles moving on the highways and multiple communication stations installed along the highway, and investigate the feasibility of such systems. The dynamic Vehicular Cloud is based on two-way communications between vehicles and the stations. We provide a communication protocol for vehicle-to-infrastructure communications enabling a dynamic Vehicular Cloud. We explain the structure of the proposed protocol in detail and then provide analytical predictions and simulation results to investigate the accuracy of our design and predictions. Just as in conventional clouds, job completion time ranks high among the fundamental quantitative performance figures of merit. In general, predicting job completion time requires full knowledge of the probability distributions of the intervening random variables. More often than not, however, the datacenter manager does not know these distribution functions. Instead, using accumulated empirical data, she may be able to estimate the first moments of these random variables. Yet, getting a handle on the expected job completion time is a very important problem that must be addressed. With this in mind, another contribution of this thesis is to offer easy-to-compute approximations of job completion time in a dynamic Vehicular Cloud involving vehicles on a highway. We assume estimates of the first moment of the time it takes the job to execute without any overhead attributable to the working of the Vehicular Cloud. A comprehensive set of simulations have shown that our approximations are very accurate. As mentioned, a major difference

between the conventional cloud and the Vehicular Cloud is the availability of the computational nodes. The vehicles, which are the Vehicular Cloud's computational resources, arrive and depart at random times, and as a result, this characteristic may cause failure in executing jobs and interruptions in the ongoing services. To handle these interruptions, once a vehicle is ready to leave the Vehicular Cloud, if the vehicle is running a job, the job and all intermediate data stored by the departing vehicle must be migrated to an available vehicle in the Vehicular Cloud.

*To my parents, for their everlasting love and support.*

# ACKNOWLEDGEMENTS

First and foremost, I would like to express my deepest gratitude to my advisor Dr. Stephan Olariu for his invaluable guidance, support, and inspiration. He spent countless hours with me discussing research ideas and provided direction and encouragement throughout the program. I consider myself to be extremely fortunate to have had him as my advisor, and I will forever treasure everything that I have learned from him.

I would also like to express my appreciation for the members of my dissertation committee, Dr. Ravi Mukkamala, Dr. Michele C. Weigle, and Dr. Dimitrie C. Popescu, for their valuable time, careful review, and constructive feedback to enhance my dissertation. Beyond providing invaluable input on my dissertation, they have also been incredibly supportive throughout the program.

Finally, and most importantly, I cannot find words to describe my gratitude to my family. My academic journey would not be possible without their constant love, encouragement, sacrifice, and support.

# TABLE OF CONTENTS

# LIST OF TABLES

# LIST OF FIGURES

# CHAPTER 1

# INTRODUCTION

## 1.1 MOTIVATION

Back in the 1980s, visionaries predicted that the ability to network vehicles together would be a game-changer, redefining the fashion we use our vehicles and roads and provide safer and more efficient transportation. [1].

This idea has inspired scientists to extend the concept of Mobile Ad-hoc Networks (MANET) to roads, highways, and parking lots and create the new notion of *vehicular networks*. Vehicular Network uses many forms of communication, such as Vehicle-to-Vehicle (V2V) and Vehicle-to-Infrastructure (V2I) communications in order to enhance the drivers' awareness of impending traffic conditions.

It has been noticed [2] that under the present-day state of the practice, the vehicles on our roads and city streets are mere spectators that witness traffic-related events without being able to participate in the mitigation of their effect. Recently, it has been shown that in such situations the vehicles have the potential to cooperate with various authorities to solve problems that otherwise would either take an inordinate amount of time to solve (e.g., routing traffic around congested areas) or could not be solved for lack of adequate resources [3, 4].

Back in 2010. Abuelela and Olariu [5] called on the vehicular networking community to "take vehicular networks to the clouds" to use the vehicular resources available to improve safety, increase efficiency, and help the economy.

Vehicular Cloud (VC) was inspired by the success of cloud computing (CC). Cloud computing, a catchy metaphor for utility computing, implemented through the provisioning of various types of hosted services over the Internet, has seen a phenomenal growth in the past decade [6]. A few years ago, inspired by the success and promise of conventional CC, a number of papers have introduced the concept of a VC, a non-trivial extension of the conventional CC paradigm [7].

VCs were motivated by the realization that present-day vehicles are endowed with powerful on-board computers, powerful transceivers, and an impressive array of sensing devices.

Most of the time, the computing, storage, and communication resources available in present-day vehicles are chronically under-utilized. Putting these resources to work in a meaningful way is poised to have a significant societal impact [8].

Back in 2010, Eltoweissy *et al.* [9] have argued that in the near future the vehicles on our roads and city streets will self-organize into VCs utilizing their corporate resources on-demand and largely in real-time for resolving critical problems that may occur unexpectedly. Although [9] envisioned from the start VCs set up on moving vehicles; it is interesting to note that the first papers discussing VCs considered the simpler case of parked vehicles [7,8].

It has become customary in the VC literature to distinguish between *static* VCs involving stationary (e.g., parked) vehicles and *dynamic* VCs that are harnessing the compute resources of moving vehicles. In this thesis, we focus on dynamic VCs.

## 1.2 OUR IDEA AND CONTRIBUTION

In this thesis, our goal is to propose an architecture for a dynamic VC to harness the computing power of the vehicles moving on highways and investigate the feasibility of such systems.

We propose a dynamic VC that mainly consists of vehicles moving on a highway and pre-installed Access Points (APs) deployed along the highway.

In a dynamic VC, the owners of vehicles or the drivers can rent their excess computational resources in a similar approach to that of large companies and enterprises which rent their excess resources for motives such as financial benefits.

In our proposed dynamic VC, vehicles establish two-way communications with the road-side APs for data transmission between vehicles and APs on the roads. We provide a communication protocol for V2I communications enabling a dynamic VC. We explain the structure of the proposed protocol in detail and then provide analytical predictions and simulation results to investigate our design and analytical predictions' accuracy.

To further investigate the feasibility of our proposed dynamic VC, we look into the job completion time, which, just as in conventional clouds, ranks high among the fundamental quantitative performance figures of merit. In general, predicting job completion time requires full knowledge of the probability distributions of the intervening random variables. In practice, however, the datacenter manager does not know these distribution functions. Instead, using accumulated empirical data, she may be able to estimate the first moments of these random variables. Yet, predicting job completion time is a very important problem that must be addressed.

The main difference between conventional cloud and VC is the dynamic nature of resources as the vehicles, which are the computation nodes, arrive and depart unexpectedly at various times, which makes VC a dynamic environment in terms of computation resources. This characteristic makes the task of reasoning about job completion time, very challenging.

In this thesis, we offer easy-to-compute approximations of job completion time for the proposed dynamic VC. We assume estimates of the first moment of the time it takes the job to execute in the absence of any overhead attributable to the working of the VC. Our extensive simulations have shown the accuracy of our approximations.

As mentioned, a major difference between the conventional cloud and the Vehicular Cloud is the availability of the computational nodes. The vehicles, which are the VC's computational resources, arrive and depart at random times, and as a result, this characteristic may cause failure in executing jobs and interruptions in the ongoing services. To handle these interruptions, once a vehicle is ready to leave the VC, if the vehicle is running a job, the job and all intermediate data stored by the departing vehicle must be migrated to an available vehicle in the VC. In this thesis, we also look into and provide strategies for job migrations in dynamic VCs.

## 1.3 ROADMAP AND ORGANIZATION

The remainder of this dissertation is organized as follows. In Chapter 2, we provide the necessary background and related work.

In Chapter 3, we present and propose an architecture for a dynamic VC established on vehicles on highways and provide a communication protocol for dynamic VCs. We offer a theoretical analysis of the probability of successful slot allocation in the dynamic VC and the present analytical and simulation results.

In Chapter 4, we investigate the job completion time in the dynamic VC for cases in which download of a job can be completed under one AP. We offer a theoretical analysis of the expected job completion time and present analytical and simulation results for the job completion time.

In Chapter 5 we investigate the job completion time in the dynamic VC for cases in which download of a job may be completed under several APs. We offer a theoretical analysis of the expected job completion time and present analytical and simulation results for the job completion time. We also investigate job migration and present job assignment strategies and simulation results for the job completion time in case of job migration.

In Chapter we put the work in perspective, and finally in Chapter 7, we offer concluding remarks, and highlight directions for future research.

# CHAPTER 2

# BACKGROUND AND RELATED WORK

## 2.1 CLOUD COMPUTING

Professor McCarthy, an MIT computer scientist, and cognitive scientist was the first to suggest the idea of *utility computing* in 1961. He envisioned that in future computing power and applications may one day be organized as a public utility and could be sold through a utility business model, like water or electricity. *"Computing may someday be organized as a public utility just as the telephone system is a public utility. Each subscriber needs to pay only for the capacity he actually uses, but he has access to all programming languages characteristic of a very large system,"* Professor McCarthy said at MIT's centennial celebration in 1961.

Cloud computing (CC) is a catchy metaphor for *utility computing* implemented through the provisioning of various types of *hosted services* over the Internet [10]. Indeed, it has been argued [11] that CC was inspired by the concept of utility computing and was enabled by the availability of infrastructure providers whose surplus computational resources and storage that can be rented out to users. In this context, a user may purchase the amount of compute services they need at the moment. As their IT needs, grow and as their services and customer base expands, the users will be in the market for more and more cloud services and more diversified computational and storage resources. As a result, the underlying business model of CC is the familiar *pay-as-you-go* model of metered services, where a user pays for whatever he/she uses and no more, and where additional demand for service can be met in real time [10, 12]. This helps users to not worry about over-acquiring of resources and services that might not meet their needs in future and protects them from being obligated to pay for what they do not use.

The term "Cloud" started to gain popularity when Google's CEO Eric Schmidt introduced the term Cloud Computing at an industry conference in 2006. Weeks later, Amazon used the term *Cloud* when launching Amazon Elastic Compute Cloud (EC2).

Jumping on the bandwagon, the National Institute of Standards [13] defined CC as *"A model for enabling convenient, on-demand network access to a shared pool of configurable* *computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction"*.

The past decade has witnessed the amazing success of CC, a paradigm shift adopted by information technology (IT) companies with a large installed infrastructure base that often goes under-utilized [10, 14–16]. The unmistakable appeal of CC is that it provides scalable access to computing resources and to a multitude of IT services. Not surprisingly, CC and cloud IT services have seen and continue to see a phenomenal adoption rate around the world [17]. In order to achieve almost unbounded scalability, availability and security, CC requires substantial architectural support ranging from virtualization, to server consolidation, to file system support, to memory hierarchy design [15, 18]. With the help of distributed and parallel computing, virtualization and high-speed Internet that is available for affordable prices, users easily can rent software and infrastructures required for their business [19].

### 2.1.1 A REVIEW OF CLOUD COMPUTING BENEFITS AND SERVICES

CC has been associated with *hosted services* over the past decade. The idea that IT and specialized services and storage and computational resources are available on-demand creates the opportunity for business to invest in what they need at the moment and acquire more services as their business grow. Similarly, software developers can easily test their applications and develop their ideas on platforms that are offered by CC.

There are many benefits and opportunities that are provided by CC, including the followings:

- It provides the illusion that infinite computing and storage resources and services are available therefore, it helps users to save time on planning ahead and also reduces any potential risks of acquiring excess services.
- It reduces the upfront financial commitments and unnecessary investments for the users and companies and it helps businesses to begin their plans as small as they want and develop and expand it as the number of their customers grows and as their needs escalate.
- It allows users to purchase computing, storage and IT services and resources on a short-term basis, for example, for months, days or hours. It also gives users the ability to release these resources as needed, therefore providing more flexibility and also encouraging conservancy [20].

There are three basic types of conventional cloud services:

**Infrastructure as a Service (IaaS):**

IaaS is the most seasoned and also the most popularly utilized type of cloud service that is available [21]. In this type of service, the cloud provider offers its customers with fundamental computing, network, and storage resources via a network and interface. In other words, an abstraction of such physical resources is provided to users with the aid of virtualization, which is undoubtedly one of the most critical elements in the practicality of IaaS. A good example of IaaS is Amazon Web Services (AWS), where Amazon provides its customers with computing resources through its Elastic Compute Cloud (EC2) service and storage service through both Simple Storage Service (S3) and Elastic Book Store (EBS) [22];

**Platform as a Service (PaaS):**

PaaS solutions are usually intended towards the developers rather than the traditional end users. In this type of service, the cloud provider offers development platforms, environments, and tools for the creation and execution of specific software and applications. PaaS users can build and execute programs without the need to install the required tools and packages on their local machines or without the need to have knowledge of specialized systems administration tools and instead utilize the available services on the cloud [23]. Google AppEngine [24] and Microsoft Azure [25] are good examples of this category;

**Software as a Service (SaaS):**

SaaS is a type of cloud computing service utilized mostly by the end-user. With SaaS, the cloud provider licenses cloud-based software or application to the user in an on-demand fashion, through subscription, and *pay-as-you-go* models. The service is accessed via a network over the web or an API. This allows customers to use expensive software and application services that they require, without the hassle of installing and maintaining such software. GoogleAppEngine and IBM are good examples of this category [26].

## 2.2 VEHICULAR CLOUDS – HOW IT ALL STARTED

In 2010, inspired by the success and promise of conventional clouds, a number of papers have introduced the concept of a *Vehicular Cloud*, (VC, for short), a non-trivial extension, along several dimensions, of the conventional CC paradigm. VCs were motivated by understanding that the current and future vehicles are equipped with powerful computers, transceivers and sensing devices. These resources are usually underutilized and employing them in a meaningful way has compelling economic and environmental impacts. Arguably, the first researchers that introduced the concept of a VC were Eltoweissy *et al.* [9], Olariu *et al.* [27] and Olariu *et al.* [28]. Their early papers defined various possible versions of VCs, their various applications and research challenges. Specifically, *autonomous clouds*, a precursor of VCs, were proposed for the first time by Olariu *et al.* [28] where they also provided an overview of a number of important applications and research challenges. Ghazizadeh [29], one of the pioneer researchers in the field of VC, introduced a resource allocation model in his Ph.D. thesis. Later, Florin *et al.* [30] have extended the VC model to accommodate the computational needs of military deployments and tactical missions.

The insight that led to VC was that by providing financial opportunities and encouragements, the drivers and owners of vehicles can rent their excess computational and storage resources to customers. This approach is similar to the approach of large companies and corporations that rent out their excess resources in exchange for financial benefits. For example, Arif *et al.* [31] suggested that in the near future air travelers will park and plug their vehicles in airport long-term parking lots. In return for free parking and other perks, they will allow their vehicles to participate, during their absence, in the airport-run datacenter.

A number of researchers have pointed out that even under the current state of the practice, many implementations of VCs are both technologically feasible and economically viable [8, 27, 28, 32]. Given their large array of applications, it is reasonable to expect that, once adopted, VCs will be the next paradigm shift, with a lasting technological and societal impact [2, 33]. It is somewhat surprising that, in spite of a good deal of theoretical papers, no credible implementation attempts of VCs have been reported in the literature. The only notable exceptions are the work of Lu *et al.* [34] and Florin *et al.* [35].

One of the fundamental ways in which VCs differ from conventional clouds is the ownership of resources. In VCs, the ownership of the computational resources is scattered over several owners as opposed to a single owner as is the case of conventional clouds run by companies such as Amazon, Google and IBM. A corollary of this is that the resources of the VC are highly dynamic. As vehicles enter the VC, new resources become available while others depart, often unexpectedly, creating a volatile environment.

Gu *et al.* [36] published a survey paper where they reviewed key issues in VC mostly concerning its architecture, inherent features, service taxonomy and potential applications. More recently, Whaiduzzaman *et al.* [8] offered an updated perspective of various research topics in VCs.

Arif *et al.* [31] were the first to look at the feasibility of datacenters running on top of the vehicles parked at a major airport. Given time-varying arrival and departure rates, they proposed a stochastic model predicting occupancy in the parking lot. Specifically, they provided a closed form for the distribution of parking lot occupancy as a function of time, for the variance and number of vehicles. In addition to analytical results, they have obtained a series of empirical results that confirm the accuracy of their analytical derivations and the feasibility of these datacenters.

Further motivation for investigating VCs and their applications was provided by Olariu *et al.* [2] where it was suggested that VCs are the next paradigm shift, taking vehicular networks to the next level of relevance and innovation. The authors of [2] have also hinted to an entire array of possible applications of VCs. Since the publication of [2], numerous papers have proposed possible solutions for many problems of interest in VCs such as in security and privacy. Some of these recent papers will be discussed in this chapter.

## 2.3 ROAD AUTOMOBILE COMMUNICATION SYSTEMS (RACS)

In 1992, Goes *et al.* [37] proposed and investigated a stochastic model for a communication system built for thruways in Japan which is a subsystem of the Japanese RACS (Road Automobile Communication System) (Figure 1).

The purpose of the RACS is mainly to provide drivers on the road with updated information about the traffic congestions, possible road hazards, optimal rerouting, driving directions, and updated detailed maps of regions.

This communication system consists of vehicles moving on the thruway and multiple fixed communication stations, which are installed with a certain equal distance from each other along the thruway. The communication stations are connected to a host computer system through a communication network, and provide a time-division multiplexed channel with fixed length frames.

This communication system is based on two-way communications between vehicles and the stations, meaning not only stations can transmit data to the vehicles, but also they can receive data from the vehicles. Vehicles or customers can subscribe to this system to transmit or receive messages.

Goes *et al.* [37] proposed a communication protocol for this communication system, to manage possible collisions and access to the time-division multiplexed frame and also investigated the effect of the number of stations, and the distance between the stations on the probability of the successful completion of communication between stations and vehicles.

What makes this system compelling is the fact that it has been built, studied and tested in Japan. It is worth noting that this communication system inspires some of our work in Chapter 3.

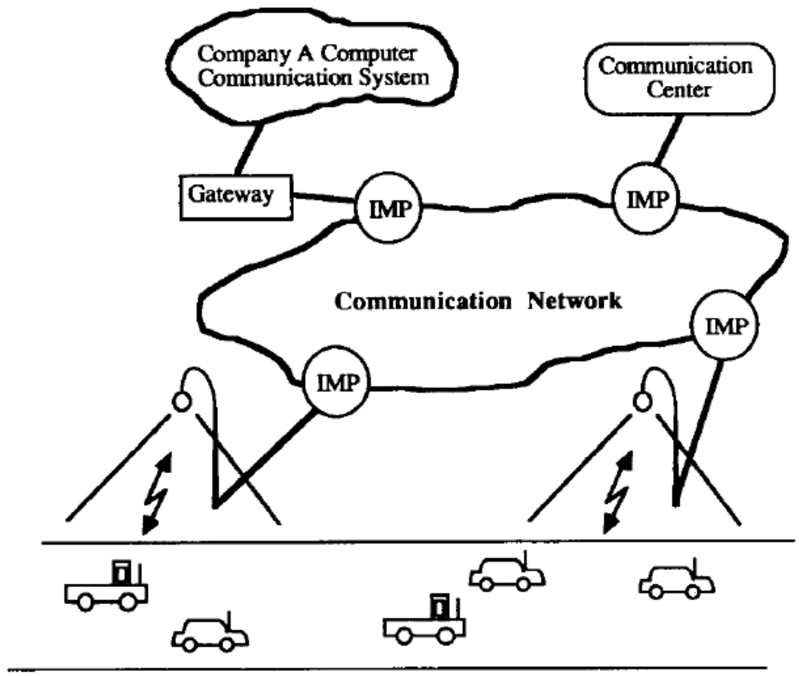


Fig. 1: Illustration of the the thruway communication system from [37]. Copyright © 1992, IEEE.

## 2.4 DRIVE-THRU INTERNET SYSTEMS

Ott *et al.* [38] have investigated the idea of using IEEE 802.11 protocol and the wireless local area network (WLAN) in a drive-thru fashion for users in vehicles which are moving on roads, particularly on highways and are in coverage of APs placed close to the roadside with a coverage area of approximately 200 meters. Authors have found that the connectivity span even at high speeds is reasonable, for example, at the 180 km/h, the connectivity under the coverage of an AP is about ten seconds. It is shown that the connection stability is low around the edges of the AP coverage area, with increased transmission delay and packet loss. Authors have also investigated the transmission of large amounts of data using UDP and TCP protocols in such drive-thru systems, concluding that the performance of the TCP is relatively superior to UDP.

Ott *et al.* [39] have proposed a Persistent Connection Management Protocol (PCMP) for Drive-thru Internet systems utilized by Drive-thru proxies and Drive-thru clients. This protocol supports the perseverance of the application sessions in the face of interruptions in the connection and helps users to leverage even short periods of connectivity whenever it is accessible. The proposed PCMP protocol is based on wireless local-area network (WLAN) technology networks accessible for users in moving vehicles at the roadside, resting areas, or gas stations. Such technology is especially important as the number of public hotspots is steadily increasing, allowing users to obtain Internet access in a varied number of locations.

Tan *et al.* [40] investigated the data communication performance of Drive-thru Internet systems, in which the users in vehicles moving on the road can connect to the roadside APs and obtain Internet connectivity while in the coverage area of an AP. In this study, authors have taken into account that there are other vehicles in the coverage area of an AP, competing with each other to receive the AP's bandwidth. This work combines vehicular traffic parameters and wireless network properties, such as traffic density, vehicle speed, service penetration rate, AP's coverage, transmission rate, and the amount of data downloaded by a vehicle into an analytical model to analyze the data communication performance of the vehicle.

Hadaller *et al.* [41] argue that the future vehicular networking protocols need to be more aware of the operating environment and demonstrate that an increased understanding of the environment can improve the overall throughput of the vehicular connection. Authors used the model by Ott *et al.* [38] to divide the connection to the AP into three phases, the entry phase beginning at 640 m before the AP, in which the connection quality is low, and connection setup occurs, the production phase between -320 m and +320 m with good connection quality, and the exit phase ending at 640 m past the AP in which the connection quality decreases. Authors suggest requiring the mobile devices to avoid the edge areas of the AP altogether and while leaving the coverage area of an AP and argue that the mobile device should not try to use an AP until the beginning of the production phase to reduce protocol timeouts and decreases in the throughput.

## 2.5 OTHER RELATED WORK

Yu *et al.* [42] proposed a Cloud-based, Vehicular Network system in which the vehicles can share computing, storage, and network resources. The main idea of the proposed system is that vehicles have limited computing, storage, and network resources, which is not sufficient for advanced applications that require plenty of such resources. The users of this system can choose between central cloud, roadside cloud, or vehicular cloud based on the type of service that they require. Authors also study virtual machine migration and theoretical solutions for optimizing resource allocation in the cloud.

In 2006, Bychkovsky *et al.* [43] explored the idea of having an *open Wi-Fi*, in which participants give access to their 802.11 (Wi-Fi) APs to mobile users in a contained approach. This study utilized vehicles driving normally and under real traffic conditions around the Boston metropolitan area and Seattle. When a vehicle is in the coverage area of an AP, it attempts to establish a connection to the AP. The authors researched the performance of such networks for mobile users traveling in urban and suburban locations, considering the challenge of the highly mobile nature of the vehicles. They have investigated the distribution of the duration of connectivity per AP and the coverage region of an AP, the distribution of packet loss and data transfer rates, and the role of the vehicle speed in such measures. This research shows that the median duration of link-layer connectivity at vehicular speeds is 13 seconds, and the upload experiments had a median throughput of approximately 30 Kbytes per second, which, at the time, was comparable to the typical uplink speeds of home broadband links in the US.

Banerjee *et al.* [44] have investigated enhancing the performance of the mobile networks by adding supporting infrastructure, such as APs, base stations, meshes, and relays, by conducting experiments and comparing the benefits of each kind of such hybrid mobile networks. They have concluded that in some cases utilizing a small number of infrastructures is much more beneficial than having a large number of mobile nodes able to route data to one another.

Jang *et al.* [45] have discussed a software-defined vehicular cloud (SDVC) architecture consisting of roadside units (RSUs) and resource provider vehicles and identified the open research challenges of such systems such as managing the connections between vehicles and VC controllers, the lack of user-friendly APIs comparable to the ones existing for the mobile platforms, security, and privacy concerns, the uncertainty of the performance considering the highly mobile nature of the vehicle and the lack of realistic performance measures.

Kumar *et al.* [46] have used Bayesian Coalition Game (BCG) models for the problem of reliable data forwarding in Vehicular Mobile Cloud. In this approach, learning automata are deployed on vehicles and are used to manage different operations such as data dissemination and local vehicle density. The vehicles that are in a VC are divided into clusters based on the location and region. Information is shared between the automatons in different clusters. The environment rewards or penalizes the actions of the automatons, which then affects the future actions of the automatons. The benefit of using learning automata in VC is primarily the ability of learning and improvement of actions based on the feedback from the environment.

Mershad *et al.* [47] discussed the important role of Vehicular Ad hoc Networks (VANETs) and Vehicular Clouds and proposed a system composing of Roadside Units (RSUs) and vehicles to detect nearby mobile cloud resources.

Jiang, Zhiyuan, *et al.* [48] have proposed a task replication method in which a task is replicated and assigned to multiple vehicles to be executed, which in return lowers the deadline violation probability and reduces the delay.

## 2.6 VIRTUALIZATION, THE WORKHORSE OF VCS

The past decade has witnessed the large-scale adoption of different virtualization techniques such as hypervisor-based and container-based virtualization. In this section, we describe how to utilize such virtualization techniques in VCs and provide an overview about the capabilities of vehicles, virtualization, Virtual Machine (VM) migration and strategies necessary to support the execution of a large variety of user jobs. A high-level description of a possible VC architecture will be presented in Section 2.7.

### 2.6.1 HYPERVISOR-BASED VIRTUALIZATION

Hypervisor-based virtualization, which is used in VMs allows users to create virtual hardware with desired resources like processor and memory, and install and run an operating system (OS), such as Linux on top of the virtual hardware.

The Virtual Machine abstraction is one of the pillars of conventional CC [15, 19]. The same concept remains fundamental in VC. While the type of service provided by the VC is largely immaterial, to fix the ideas, in the following discussion we assume that the VC offers IaaS cloud services. It is well known that in IaaS, the users can acquire virtualized computing resources. They are able to choose an operating system and receive a VM with an installed instance of their desired OS.

We assume that, as illustrated in Figure 2, each vehicle has a virtualizable on-board computer and has been pre-loaded with a suitable Virtual Machine Monitor (VMM).

As illustrated in Figure 2, the VC offers the user a virtualized instance of the desired hardware platform and operating system bundled as a VM and guest OS.

### 2.6.2 CONTAINER-BASED VIRTUALIZATION

Container-based virtualization, which has gained popularity in recent years, is a lightweight and agile alternative to the traditional hypervisor-based virtualization [6]. Containers are standard units of software that run on top of the host operating system and are isolated from

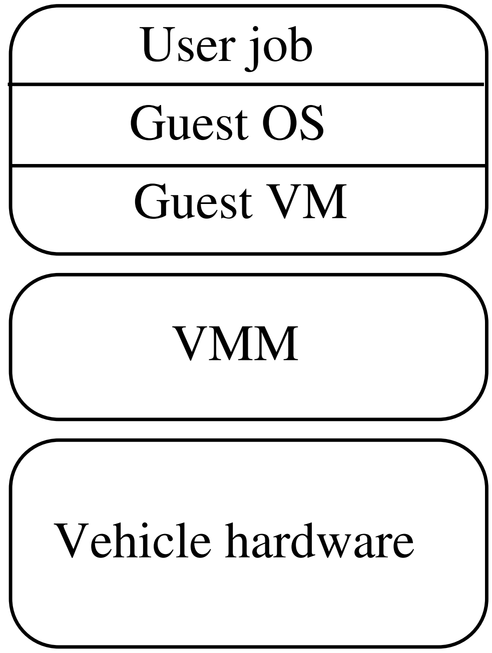


Fig. 2: Illustrating a virtualization model for VCs.

each other [49]. An executable application and its dependencies and everything needed to run the application such as the code, runtime, system tools, system libraries and settings are packaged into a lightweight executable package of software called the container image [50].

In a container image, layers or intermediate images can be created on top of the existing images by adding a layer that contains the filesystem differences between the images [6]. The basic image on which layers are added to create a final image is called a base image. See Figure 3.

In this model, vehicles are assumed to be preloaded with the Linux operating system and a container engine such as LXC (Linux Container) and container images can be preloaded on the vehicles or downloaded as vehicles enter the VC.

### 2.6.3 VM MIGRATION

Once a vehicle is ready to leave the VC, if the vehicle is hosting a guest VM, the VM and all intermediate data stored by the departing vehicle must be *migrated* to an available vehicle in the VC. There are several known strategies for VM migration [51–54]. In spite of its fundamental importance, somewhat surprisingly, only a few authors addressed VM migration in VCs such as pioneering work done by Ghazizadeh *et al.* [29, 55], Baron *et*

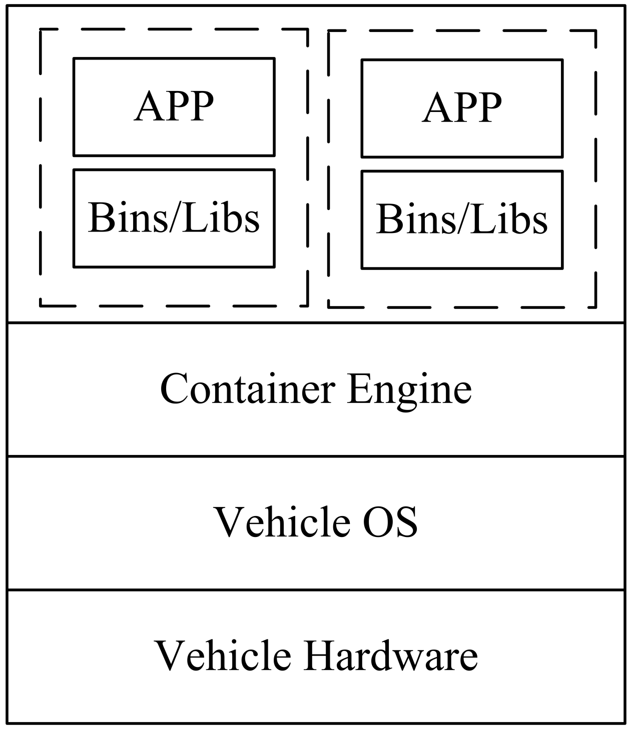


Fig. 3: Container based virtualization model for VCs

*al.* [51], Refaat *et al.* [54] and Florin *et al.* [35].

Refaat *et al.* [54] proposed different schemes and algorithms for virtual machine migration in VCs and evaluated the performance of such algorithms with the help of simulations.

Baron *et al.* [56] have studied the virtual machine migration in a vehicular network conducting simulations using available data of real traces of a bus transit system. Authors have found that hundreds of Megabytes of data can be transferred in a virtual machine migration via V2V communications, without having to rely on cellular networks.

## 2.7 A HIGH-LEVEL DEFINITION OF A VC ARCHITECTURE

The main goal of this section is to present a bird-eye's view of a possible VC architecture, similar to the one proposed by Florin *et al.* [35].

Throughout this section we deal with a datacenter supported either by a *static VC* built on top of vehicles parked in a adequately large parking lot, or else by a *dynamic* VC involving vehicles in on a highway or, perhaps, moving vehicles in the downtown area of a smart city where they have full-time connectivity [57–59]. We assume that a sufficient number of vehicles participate in the VC that it is always possible to find one or more vehicles that can be assigned to an incoming user job.

The participating vehicles share their computational and storage resources to create a VC-based datacenter that then helps companies and corporations to make the most out of their underutilized resourced and potentially save thousands of dollars.

Conceptually, the architecture of the VC is almost identical to the architecture of a conventional cloud [10,15], with the important difference that the VC is far more dynamic than a conventional cloud. As an illustration, assume that the VC offers only IaaS cloud services. Recall that in IaaS, the users request a hardware platform and specify their preferred OS support. Referring back to Figure 2, the VC offers a virtualized version of an OS based on user preferences and then the assigned jobs can run on the provided VM and guest OS on top of the vehicles. To reduce the rate of unsuccessful jobs and improve reliability, each job is assigned to several vehicles. When the VM running the user job in a specific vehicle finishes execution and the job is done, then the result is uploaded to the datacenter. In this scenario, the datacenter waits for the prescribed number of instances of the user job to terminate and makes a final determination by using, for example, a quorum-based algorithm or voting mechanism [60–63].

The datacenter is equipped with a system that identifies idle vehicles in the VC. In the simplest form, this can be implemented by assigning a status bit to each vehicle in the VC. Upon joining the VC, the status bit is set to available. Upon receiving a job, the datacenter updates the bit to unavailable. For purposes of fault-tolerance, the datacenter is able to assign multiple vehicles to a user job; this assignment utilizes some criteria based on locality. Upon a vehicle departure, the datacenter notifies all other vehicles assigned to the same job.

When a vehicle enters the VC, it will initiate communication with the *Resource Manager* component of the datacenter manager. The Resource Manager identifies and keeps track of the location of the vehicle. Finally, the *Job Scheduler* identifies and picks the available vehicles for the assignment of jobs.

To get an idea of the type of processing that is going on, consider a user job submitted for execution by the datacenter and refer to Figure 4 that offers a functional view of the datacenter manager module. The requirements of the submitted user job, in terms of the hardware platform that needs to be emulated, the requested OS and the user-specified input data are evaluated by the *Job Admission* daemon. If these requirements can be met by the datacenter, the job is admitted and gets inserted into a queue of jobs awaiting execution. Otherwise, the job is rejected. It would be of interest to mimic the Amazon AWS paradigm offering support for different types of jobs, described by the quality and quantity of system resources they need.

Once the job makes it to the front of the queue, control passes to the *Virtualization Agent* that bundles the resources specified by the user job into a VM, specific OS, and the input data to the VM. Finally, the Virtualization Agent passes control to the *Job Scheduler* that identifies one vehicle (or a group of vehicles) on which the job is to be executed.

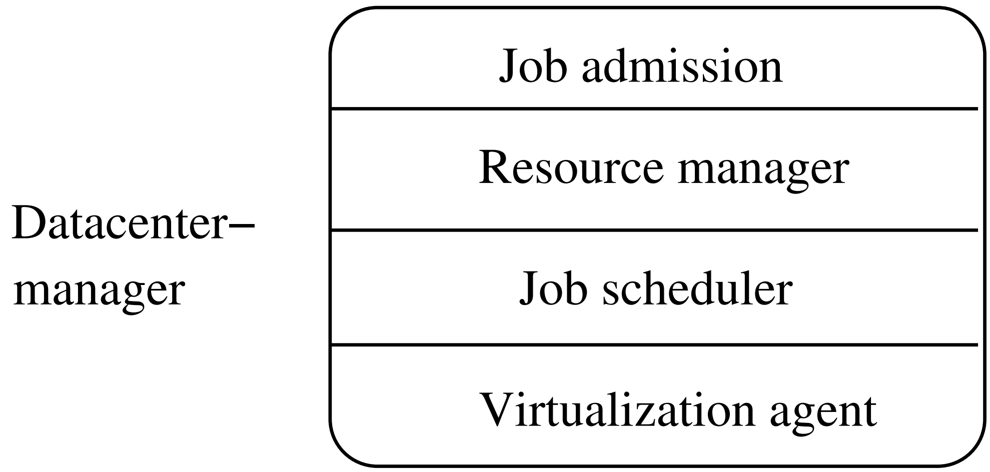


Fig. 4: Illustrating the functional view of the datacenter manager.

Figure 5 presents the logical view of the communication structure of the datacenter. This communication structure is, conceptually, very similar to that of a standard datacenter (see Barroso *et al.* [10]). The analogy is better understood if one assumes a static VC involving the vehicles in a parking lot. Indeed, one can think of a vehicle in the parking lot as a server in a rack of servers. Accordingly, the parking lot is partitioned logically into clusters of parking spots, regions of clusters and so on. The communication in each cluster is under the control of a switch called the *Cluster Controller*. Similarly, the communication in a region is under the control of a switch called a *Region Controller*. Referring to Figure 5, the parking lot is partitioned into $N$ regions. Each such region consists of $k$ clusters. The datacenter manager discussed above is in control of assigning user jobs to various vehicles in the parking lot.

The datacenter supplies a wired or wireless connection to give each vehicle access to a network fabric connecting each vehicle in the parking lot [31, 64, 65]. Also connected is data storage where data from each job is saved when the job is completed. When a vehicle to which a job was assigned leaves the VC prematurely, the datacenter notifies the other vehicles assigned to the same job. When the need arises, the datacenter can assign, using some suitably lightweight criteria, an available vehicles to the job [10].

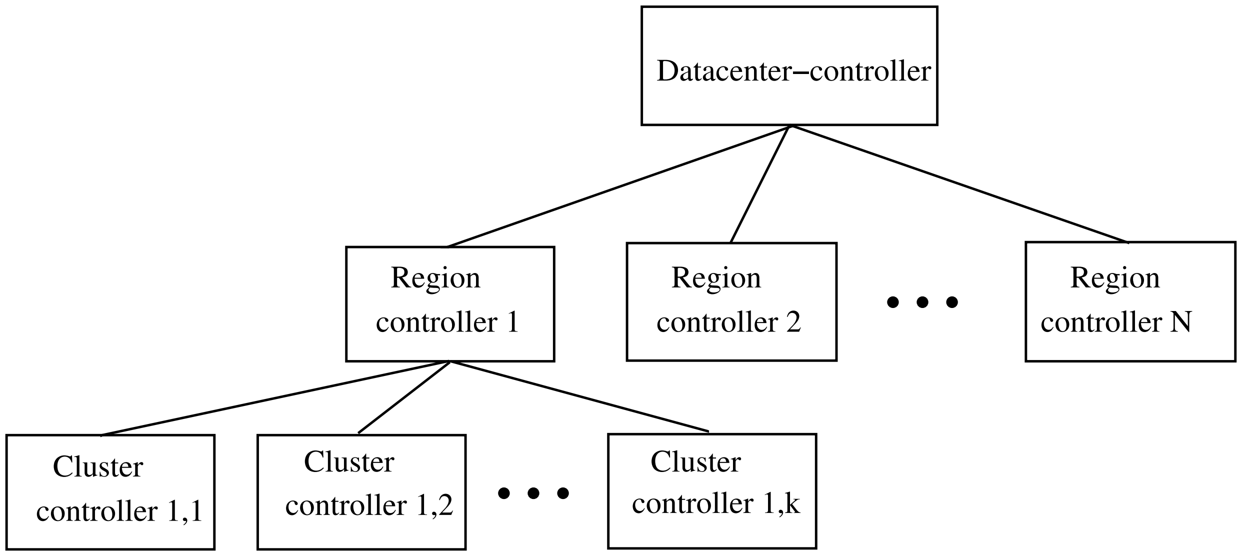


Fig. 5: Illustrating the logical view of the datacenter communications.

## 2.8 VEHICLE RESIDENCY TIMES IN THE VC, WHY DOES IT MATTER?

Although very few authors seem to pay much attention to the residency time of vehicles in the VC, it is of fundamental importance to distinguish between vehicle residency times in the VC and VM lifetime. Vehicle residency time shows the amount of time a vehicle spends in the VC. If a vehicle leaves the VC before finishing the job, meaning the VM that is running the job has not terminated, then we can say that the vehicle has left *prematurely*.

The VC literature is full of papers discussing fanciful VC models implemented on top of moving vehicles and supported by ubiquitous and pervasive roadside infrastructure. Those authors do not seem to be concerned with the obvious fact that moving vehicles' residency times in the VC may, indeed, be very short-lived and therefore so is their contribution to the amount of useful work performed. Worse yet, should a vehicle running a VM leave the VC prematurely the amount of work performed by that vehicle may be lost. Unless, of course, special precautions are being taken. Such precautionary measures involve either some flavor of checkpointing [66,67] or else enforcing some form of redundant job assignment [29,68–70]. Both approaches have consequences in terms of overhead and impact the mean time to failure [69, 71, 72] and job completion time [73, 74].

### 2.8.1 A TAXONOMY OF RESIDENCY TIMES

Given the importance of residency times, it is useful to offer a more general look at VCs in terms of the assumed vehicular residency times in the VC.

**Long residency times:**

At one end of the spectrum, we find VCs characterized by long vehicular residency times. These can be further partitioned depending on whether or not the vehicles are assumed to be parked or mobile. For example, Arif *et al.* [31] and other researchers have assumed a VC is developed using the vehicles in the *long-term* parking lot of an international airport where residency times are days or maybe even weeks [69]. Intuitively, it is clear that the users of such a parking lot may share their transportation plans with the datacenter at the airport and so the residency times become almost deterministic.

As another example of a VC characterized by long and stable vehicular residency times consider a plant that operates 24 hours a day, seven days a week. The patrons of the parking lot are working at the plant in eight-hour shifts, that creates a pool of vehicles that can serve as the substructure for a datacenter with the help of a communication model proposed by Ghazizadeh *et al.* [75]. We assume that the vehicles in the parking lot are plugged into a standard power outlet and are provided Ethernet connection to the datacenter.

Recently, Florin *et al.* [30] proposed Military Vehicular Clouds (MVC) an instance of VCs with (relatively) long vehicular residency times. In their vision, a MVC comprises either parked military vehicles or vehicles that move in formation so that, for all practical purposes, the distance between adjacent vehicles remains the same. This controlled mobility regimen applies to military units deployed in support of a tactical mission. The vehicles participating in the MVC have a wireless connection to a Mobile Access Point (MAP) which may be connected to a center of command and control. There are two major differences between MVCs and conventional clouds. First, unlike conventional clouds that are mostly public, MVCs are private, catering to the special needs of the military. Second, because of mobility and of the fact that military vehicles join and leave the MVC in ways that best support the tactical mission at hand, the MVCs are characterized by unpredictable availability of compute resources.

**Medium residency times:**

Some VCs are characterized by shorter vehicle residency times. Such is the case, for example, of VCs build on top of the vehicles in the parking lot of a shopping mall. Observe that the vehicular residency times at the mall are correlated with the average time a standard American spends at the shopping mall in one visit. While vehicle residency times at shopping malls are not directly available, there is a profusion of data concerning shopping pattern statistics – see, for example, [76–79]. This statistical data indicates that, during normal business hours, when the datacenter services are required, the shopping center parking lot contains a sufficient number of vehicles which makes the task of finding available vehicles, for job assignment, feasible. Also, according to these statistics, the average American shopper spends about *two hours* per visit at the shopping mall. Of course, the exact residency time of any one individual is a random variable and so any form of reasoning here must be probabilistic.

The challenge facing the implementation of a datacenter built on medium residency vehicles is to maintain high availability and reliability in the face of the dynamically changing resources.

**Short residency times:**

Finally, one can conceive of instances of VCs involving vehicles with a short residency times. These are vehicles that happen to be in an area that offers free Internet connectivity, such a gas stations or restaurants at highway interchanges, where they can be assigned jobs to process. The vehicles will process their jobs while moving (and not being connected) and will upload their results at the next interchange.

As in the case of medium residency times, VC involving short-term vehicular residency times are very challenging to deal with and, to the best or our knowledge, no papers in the literature have reported on promoting high system availability and reliability in the face of the dynamically changing resources.

## 2.9 SERVICES SUPPORTED BY VCS

In [2] Olariu *et al.* have introduced three types of services that will be made possible by VCs: Network as a Service (NaaS), Storage as a Service (STaaS), and Cooperation as a Service (CaaS). Since then, other services were proposed. These services will be discussed briefly below.

### 2.9.1 NETWORK AS A SERVICE (NAAS)

It is clear that, at the moment, not every vehicle on the road has Internet connection. Therefore the vehicles that have access to Internet can share their excess Internet with other drivers that request it. Many drivers have limited or unlimited Internet connection through 4G or recently 5G cellular network and these resources can be under-utilized because, for example, not all drivers are constantly downloading from the Internet. A driver that wants to share the Internet connectivity will advertise it to all the vehicles nearby. Considering the fact that the vehicles are moving on the same direction and with relative speeds, this system can be viewed as a traditional VANET, constructed of a set of static APs and mobile nodes that are moving at low speeds.

For example vehicle a, and c have 4G and vehicle d has WiFi connectivity through an AP. Vehicles a, d and c broadcast packets and inform other drivers about their intention to share their networks. If any vehicle is interested to rent these Internet services, the requests will be sent to a selected vehicle with a stable Internet connection. There are several factors such as reliability of the network, expected connection time, and speed of the network that should be considered for selecting a candidate.

### 2.9.2 STORAGE AS A SERVICE (STAAS)

Many vehicles have plenty of on-board storage resources and capabilities, however there are also some vehicles that need extra storage for running complex applications. In addition parked vehicles in the shopping malls or any large or medium-size parking lot can rent out their storage resources to the management of such places. Another example is using this excess storage in backup purpose, peer-to-peer applications and various types of content such as multimedia contents that are larger in size and require more storage support. STaaS in VCs has limitations related to mobility of the vehicles. The vehicles are not static forever; hence the users are able to use storage resources for an extended period of time. This limitation is an obstacle against renting storage as a service in VCs.

### 2.9.3 COOPERATION AS A SERVICE (CAAS)

Mousannif *et al.* [80] have introduced Cooperation as a Service, as a new architecture that extends the two types of services in VCs, Network as a Service (NaaS) and Storage as a Service (SaaS). It aims to provide vehicles with a set of services for free, and without the need for increased roadside infrastructure. Vehicular networks can be help in many situations to improve the safety of transportation, decrease traffic congestion, and provide accident warnings, road condition and weather information, parking availability and advertisement [1]. ITS and available 3G and 4G networks can be useful for offering these services. In CaaS, which can be considered as a new form of community service, drivers can obtain a set of services using minimal roadside infrastructure. If no infrastructure is available, these vehicles can take advantage of V2V communication. V2V communications are made possible via DSRC or WAVE (Wireless Access in a Vehicular Environment) [81].

CaaS uses a hybrid publish/subscribe mechanism where the driver (or subscriber) informs the system about his wish to acquire a service and the vehicles/drivers that have subscribed to the same service will cooperate to provide the driver with the necessary information regarding the service, by publishing this information in the network. An example of a cooperative vehicular safety application is the Electronic Brake Warning (EBW) application. When a vehicle breaks sharply, it broadcasts an event-based message, alerting the surrounding vehicles of the braking incident. This information is especially useful in situations that the view of the break light is blocked and it will allow drivers to take necessary precautions. Another example is the Vehicle Stability Warning (VSW) application. In VSW, preceding vehicles alert following vehicles of upcoming potential hazardous conditions such as icy or slippery roads. VSW is an event-based application, similar to EBW. These applications require constant reliable communication between vehicles since they are not effective and reliable if the sent emergency messages are not received.

### 2.9.4 COMPUTING AS A SERVICE (COMPAS)

Statistics show that most vehicles are parked for several hours, every day, in parking lots, parking garages or driveways. The computing resources of these vehicles are an untapped and under-utilized and wasted under current state of the practice. The owner of the vehicles can use this opportunity and rent out their on-board capabilities on-demand for hours, days or weeks to customers. For example, in airport long-term parking lots, travelers will plug their vehicles and they will allow users or managements to use the computation power of their vehicles in return for free parking and other benefits. Another example is when vehicles are stuck in traffic jams, they are willing to allow municipal traffic management centers to use their resources and run designed simulations to find a solution for the traffic and alleviate the effects.

### 2.9.5 INFORMATION-AS-A-SERVICE (INAAS) AND TRAFFIC INFORMATION-AS-A-SERVICE (TIAAS)

Drivers need different types of information, for safety improvements, advance warnings, news about the sudden crashes or emergency situations. These services are recognized as Information-as-a-Service (INaaS). In [32], Hussain *et al.* have used a framework of VANET-based clouds namely VuC (VANET using Clouds) and defined another layer, named TIaaS (Traffic Information as a Service). TIaaS provides vehicles on the road with traffic information. Vehicles and drivers share their gathered information about the traffic situations, congestions, road conditions and etc with the other vehicles and also with the cloud infrastructure. Users that want to get notifications about such information can subscribe to the TIaaS and receive the timely information accordingly.

### 2.9.6 ENTERTAINMENT-AS-A-SERVICE (ENAAS)

Nowadays, travelers seek entertainment to make their trips more enjoyable. Movies, commercials and games can be played on the vehicle's screens and make the travel more entertaining and comfortable. These types of services are recognized as Entertainment-as-a-Service (ENaaS).

## 2.10 APPLICATIONS OF VCS

In this section we review some of the applications of VCs.

### 2.10.1 DISASTER MANAGEMENT USING VCS

The importance of emergency response systems and disaster management systems cannot be diminished due to the sudden nature of disasters and the damage, loss and destruction that it brings to human lives and as well as properties. In the past decades, we have seen many disasters such as the Earthquake in Haiti, in 2010 that caused over 200,000 deaths, left two million homeless and three million in need of emergency assistance.

At the time of disaster, VCs plays a very important role in helping with removing people from disastrous and damaged areas and transfer them to safe areas and therefore save many lives and also valuables including information. Raw *et al.* [82] proposed an Intelligent Vehicular Cloud Disaster Management System (IVCDMS) model and have analyzed the intelligent disaster management through VC network communication modes with the existing hybrid communication protocols. The architecture of this system consists of three interacting service layers named vehicular cloud infrastructure as a service, intelligence layer service and system interface service. The smart interface service helps in transportation of data from one place to another and it acquires data from various sources such as roadside units, smart cell phones, social network etc. The VC infrastructure as a service provides base platform and environment for the IVCDMS. The intelligence layer service provides the necessary computational models, algorithms and simulations; both stochastic and deterministic that then provides the emergency response strategies by processing the available data from various resources.

Raw *et al.* [82] have implemented the communication V-P (vehicle-to-pedestrian) protocol which uses cellular phones and wireless network to improve the safety of travelers, pedestrians and drivers. This protocol helps drivers and pedestrians to get informed of one another and have enough time to take proper action and avoid probable accidents and hazards.

### 2.10.2 VC DATA COLLECTION FOR TRAFFIC MANAGEMENT

VCs can help the ITS community by improving traffic and accident information sharing. Chaqfeh *et al.* [83] have proposed an on-demand and pull-based data collection model for better information sharing between vehicles. When a vehicle wants to get information about a specific rout, it will create a route request (RREQ) message that contains details about the destination and the route that the vehicle is interested to get more information about. The requesting vehicle then broadcasts this message to its neighbors. The RREQ message is then rerouted to vehicles in the identified destination. The vehicles at the destination that receive the message then publish their sensor data and gathered traffic information to the network by forming a dynamic VC. Finally, a vehicle is selected as the broker to collect the information and transfer and communicate it with the infrastructure. Chaqfeh *et al.* [83] have showed via simulation that their VC-based approach can effectively support ITS with real-time data, even with not too many participating vehicles and concluded that the participation of small number of vehicles in a dynamic VC is enough for data transferring and data collection. However, these authors have ignored fundamental VC problems including whether VMs will be set up and how system failures can be handled in case vehicles leave the VC unexpectedly.

Sood *et al.* [84] have developed a smart traffic management system based on IoT, Edge Computing, and cloud computing, to predict the traffic inflow, such as the traffic at intersections. This system leverages the benefits of cloud computing such as storage and computing power and improves the communication overhead by utilizing Edge Computing for time-sensitive services, and when low latency computing is critical. This system can be used to manage the traffic movement and reduce congestion by rerouting vehicles to the roads with less congestion and balancing the traffic load's distribution, hence optimizing the travel speed and navigating the impatient driver to the destination faster.

### 2.10.3 SHARING REAL-TIME VISUAL TRAFFIC INFORMATION VIA VCS

With advances in technology, vehicles are equipped with more sophisticated sensing devices such as cameras and these features give vehicles the ability to act as mobile sensors that provide useful traffic information.

With advances in technology, vehicles are having better quality and improved cameras and sensing devices which can be used to collect traffic and road information.

Users can send or receive accurate and up-to-dated information about the road or traffic conditions by using VC services. Kwak *et al.* [85] have presented an architecture for a possible collaborative traffic image-sharing system which allows drivers in the VC to report and share visual traffic information called NaviTweets. NaviTweets are selected and transformed into an informative snapshot summary of the route. This will provide reliable information about various details of the road condition, traffic, congestion, weather and any data that is useful for drivers and is beneficial for enhancing the safety and speed of the transportation.

### 2.10.4 OTHER APPLICATIONS OF VCS

There are some other research papers that investigate the appropriate applications of VCs. For instance, Saleem *et al.* [86] have discussed the relevance of Vehicular Cloud Computing (VCC) and reviewed the technology, applications, challenges and possible solutions. They have argued that the applications of VCC, include but not limited to intelligent parking management, traffic light management, vehicle maintenance updates, safety, congestion control, accident alerts, planned evacuations, and road condition sharing. The authors identified issues such inaccurate GPS signals and localization, authentication and privacy issues, confidentiality and communication security issues and briefly discussed possible solutions for such issues.

## 2.11 A LARGE-SCALE IMPLEMENTATION OF VCS

Researchers often desire convenient, cost-effective and reconfigurable testing environments. Developing and testing VC applications is challenging and expensive. One solution to this problem is creating miniature testbeds using robots instead of real vehicles. Recently, this approach was explored by Lu *et al.* [34] who have attempted to build a VC testbed with different types of mobile robots. Their testbed is constructed from robot vehicles on mini cloud, network, remote cloud and management servers. The VC-bots can be managed and controlled with the help of a user interface (UI) that was built using java. Using this interface, users can see and track vehicles on a map. The interface is a graph-based model that shows straight roads as edges and intersections as nodes. The width of an edge shows the width of the road and the position or coordinate of each node, shows the location of the intersection. Parking lots and open spaces are shown as special nodes connected to the road network. Users can move the robot vehicles from one location to another, using the UI. An advantage of this design is the configurable map and road network.

The testbed contains 19 robot vehicles of four types: VC-trucks, VC-van, VC-sedan, and VC compacts. The types of robot vehicles vary in sensing devices, computing resources, and battery life and an independent WiFi network was used for the purpose of the robot vehicle management. Each robot vehicle is equipped with WiFi interface and also LTE modem, or the purpose of better network connection and in case the robot is out of the range of the WiFi network. In addition, several WiFi routers are deployed as roadside units (RSU).

In this testbed, the cloud resources contain robot vehicles on board mini cloud and remote cloud in the data center. The remote cloud consists of Linux servers managed by OpenStack and provides a cloud management user interface, centralized control and storage. A Kernel-based Virtual Machine (KVM) was installed on the robot vehicles to provide basic virtualization environment. Users can create virtual machines (VM) or import existing VM images into mini cloud, and run their programs and applications on these virtual machines. Live migration is also allowed from a robot vehicle to another.

The testbed of Lu *et al.* [34] is a work-in-progress and proof-of-concept prototype for VC-bots, an evolving platform for testing vehicular network and VC applications. To the best of our knowledge this is the first time researchers have implemented a testbed for VCs and it is a proof that this area of research should be explored and studied further.

## 2.12 BIG DATA IN VCS

One of the significant research challenges in VCs is to identify conditions under which VCs can support Big Data applications. It is apparent that Big Data applications, with stringent data processing requirements, cannot be supported by ephemeral VCs, where the residency time of vehicles in the cloud is too short for supporting VM setup and migration.

Quite recently Florin *et al.* [35] have identified sufficient conditions under which Big Data applications can be effectively supported by datacenters built on top of vehicles in a parking lot. This is the first time researchers are looking at evaluating the feasibility of the VC concept and its suitability for supporting Big Data applications. The main findings of [35] are that

- if the residency times of the vehicles are sufficiently long, and
- if the interconnection fabric has a sufficient amount of bandwidth, then Big Data applications can be supported effectively by VCs.

In spite of this result, a lot more work is needed to understand what it takes for VC to be able to support, in a graceful way, data- and processing-intensive applications.

## 2.13 SECURITY AND PRIVACY IN VCS

When users are allowed to share pools of resources in a shared network, security and privacy questions and issues arise. In the past few years, many researchers have suggested solutions for such issues.

The first authors to investigate security and privacy issues in VC were Yan *et al.* [87] and Yan *et al.* [88]. They have shown that many of the insecurities found in conventional CC carry over to VCs. In addition, several VC-specific security challenges were identified and also preliminarily solutions were proposed. They have categorized the main targets of attacks into attacks related to confidentiality, integrity and availability. Examples for such attacks are finding the identities of users, personal and sensitive data, code and documents stored on the VC. Such attacks can be done in many ways, for example attackers pretend to be the user that is requesting the service or they can discover a bug or flaw in the system and get access to the sensitive and hidden data that they normally do not have the permission or access to. Yan *et al.* have argued that the security authentication in VC is challenging due to the mobile nature of the moving vehicles. Specially, authentication of messages with location context is not easy since the location of the vehicles is changing with time.

Another vulnerability area in VC is that often because of the legal reasons, the vehicle identity and information is pinned to its owner's identity. However most VC applications use the location information of the vehicle and tracking the location of the vehicle violates the owner's privacy. Pseudonymization has been suggested as a possible solution. In this approach the vehicle's identity is replaced by a pseudonym to protect the driver's privacy.

Huang *et al.* [89] have proposed a vehicular cloud computing system called PTVC for improving privacy and trust based verification communication. The purpose of this system is to provide a better solution for selection of credible and trustworthy vehicles for forming a VC. The proposed PTVC scheme is composed of a few stages: system setup, privacy-preserving trust-based vehicle selection protocol, privacy-preserving verifiable computing protocol and trust management. In this scheme, a trust authority is responsible for execution and maintenance the the whole system and it generates public and private keys for vehicles and road side units (RSU). When a vehicle wants to join or form a VC, it will try to find the nearest available vehicles with the highest reputation.

Participating vehicles that want to transfer data without worrying about leaking privacy, should first encrypt their data, which is done with the help of privacy preserving verifiable computing protocols. Each participating vehicle will receive feedback on the performance and participation which then helps in determining the reputation value of the vehicle. The authors have concluded that the security analysis shows that this proposed scheme is practical and effective against various security attacks and can identify the untrustworthy vehicles and blocks them from entering the VC.

# CHAPTER 3

# VEHICULAR CLOUD MODEL

In this chapter, we present and propose an architecture for a *dynamic* VC established on vehicles on highways and provide a communication protocol for dynamic VCs.

## 3.1 ARCHITECTURE

In this thesis, we envision a *dynamic* VC that is harnessing the compute power of vehicles moving on a highway. In order to implement this idea, the VC controller is connected by optical fiber to pre-installed APs deployed along the highway. Referring to Figure 6, the access points are placed $d$ meters apart along the highway and are numbered consecutively as $AP_0, AP_1, \cdots, AP_n, \cdots$. As illustrated in Figure 6, each AP has a radio coverage area of $c$ meters [90].

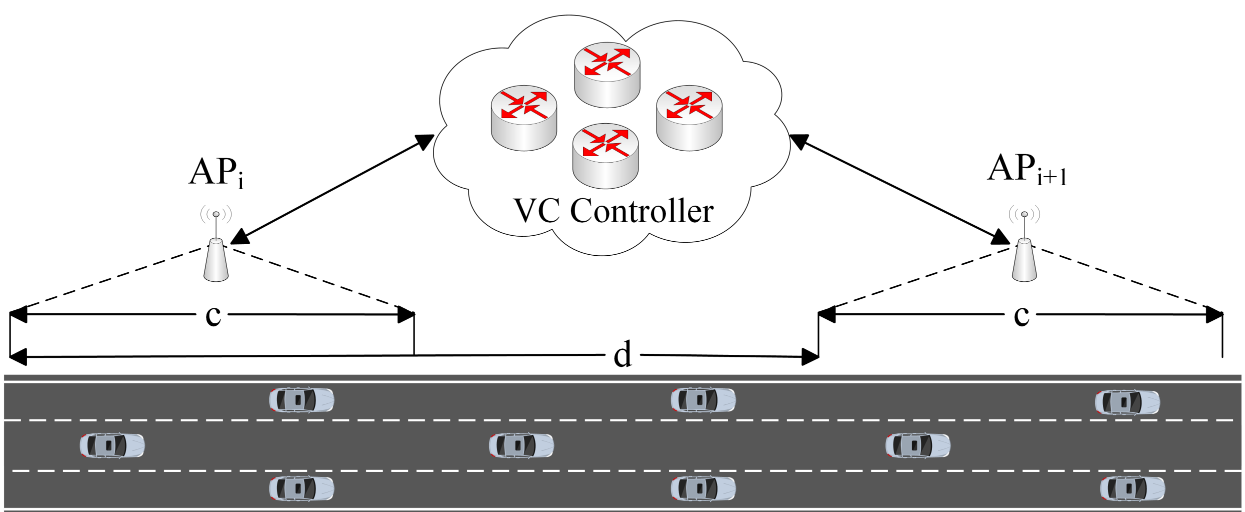


Fig. 6: Illustrating two consecutive APs and their coverage areas.

A vehicle can communicate with an AP only when it is in its coverage area. Upon entering the coverage area of an $AP$, each vehicle is assigned a job for processing. Consider a vehicle that just entered the highway at $AP_i$. The vehicle informs $AP_i$ of the Access Point $AP_j$ at which it will exit. Given this information, and the average speed on the highway between $AP_i$ and $AP_j$, the VC controller can estimate the amount of time the vehicle will spend on the highway. This helps determine the workload that can be allocated to the vehicle for processing.

In our proposed VC, vehicles are assumed to be preloaded with the Linux operating system and a container engine such as LXC (Linux Container) or Docker. Images can be preloaded on the vehicles or downloaded from the APs.

Jobs are encapsulated as container images. The vehicle will begin by downloading the corresponding container image, will execute the job and, upon termination, will upload the results to the first available $AP$. In case the vehicle leaves the highway before completing job execution, the corresponding container will have to be migrated to another vehicle, using one of several migration strategies.

### 3.2 SPEED-DENSITY RELATIONSHIP

In this section, we present the speed-density function that we have used in our analytical predictions and simulations. We assume $l$ lanes of traffic and the traffic density along the highway is constant in the coverage area of each AP. We assume that the length of the coverage area under each AP is $c$ meters. We refer to the corresponding density in 1 km as $\rho_{k+1}$, $(k \geq 0)$. It follows that

$$\rho_{k+1} = \frac{(k+1)10^3}{cl}.$$

As in [91], we define the speed $v_{k+1}$ of a vehicle (in m/s) as a function of the traffic density $\rho_{k+1}$ corresponding to the presence of $k+1$ vehicles in the coverage area of an AP.

$$v_{k+1} = v_b + \frac{v_f - v_b}{(1 + \exp(\frac{\rho_{k+1} - k_t}{\theta_1}))^{\theta_2}} \tag{1}$$

In (1), $v_f$ is the free-flow speed, $v_b$ is the average travel speed under stop-and-go condition, $k_t$ is the turning point where the speed-density curve makes the transition from free-flow to congested flow, $\theta_1$ is a scale parameter, and $\theta_2$ is a parameter which controls the lopsidedness of the curve. The values of these parameters are presented in Table 1. Table 2 shows the values of $v_{k+1}$ for different $k$.

### 3.3 COMMUNICATION MODEL

In this section we discuss a communication protocol and frame structure for V2I communications in dynamic VCs which is inspired by and based on a communication system that was used in 1992 in the Japanese RACS (Road Automobile Communication System) and that has been studied and tested in Japan [37]. The goal of the RACS system was to

TABLE 1: Speed-Density parameters

| **Symbol and Description** | **Value** |
|---|---|
| $l$ (number of lanes) | 3 |
| $c$ (access point coverage range) | 100 m |
| $v_{k+1}$ (vehicle's speed when k other vehicles are in the area) | (10, 30) m/s $\approx$ (36, 108) kph |
| $v_b$ (average travel speed at stop and go condition) | 2.5 m/s $\approx$ 9 kph |
| $v_f$ (free flow speed) | 29.8 m/s $\approx$ 107.44 kph |
| $k_t$ (turning point for the speed-density curve) | 17.53 |
| $\theta_1$ (scale parameter for speed-density function) | 1.8768 |
| $\theta_2$ (parameter which controls the lopsidedness of the curve) | 0.0871 |

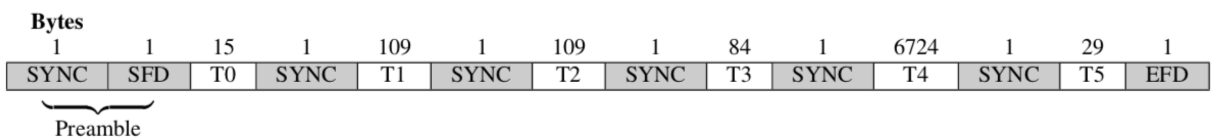


Fig. 7: Frame structure for V2I communication.

provide route guidance and navigation by allowing two-way communications between cars and APs. The APs could be connected through the network to a computer system which is responsible for processing the received messages.

### 3.3.1 FRAME CONFIGURATION

Figure 7 shows the frame structure which consists of multiple fields with fixed durations. The frame begins with a synchronization byte (SYNC) which is used for informing the vehicles that a new packet is arriving as well as synchronizing the receiver's clock with the transmitter's clock. Synchronization sequences are also used in between fields due to the dynamic characteristics of the vehicular environments. The start and the end of a frame are indicated by a start of frame delimiter (SFD) and an end of frame delimiter (EFD), respectively. The remaining fields are explained in detail in the next subsections.

### 3.3.2 T0: OPEN COMMUNICATION PERIOD

TABLE 2: $v_{k+1}$ as a function of $k$.

| **k** | $\mathbf{v_{k+1}}$ (m/s) |
|---|---|
| 1 | 29.83718427 |
| 2 | 29.80179124 |
| 3 | 29.60369025 |
| 4 | 28.70343391 |
| 5 | 26.38356621 |
| 6 | 23.30765748 |
| 7 | 20.38255661 |
| 8 | 17.82803541 |
| 9 | 15.63239015 |
| 10 | 13.75037811 |
| 11 | 12.13795097 |
| 12 | 10.7566017 |

| **bits** 22 | 24 | 3 | 40 | 11 | 4 | 8 |
|---|---|---|---|---|---|---|
| AP ID | Frame number | Number of fields | Length of each field | Beginning of T1 | Number of slots in T1 | Bidding sequence |

Fig. 8: Structure of T0.

This field is used for broadcasting the AP identification number and the layout of the frame to all vehicles that are in the coverage area of the AP. Figure 8 shows the structure of T0.

In this section, we explain in detail, the reasoning for the structure and size of each field in the T0 period. It is clear that the number of bits needed to represent a positive integer $n > 0$ is obtained by $\lfloor(\log_2 n + 1)\rfloor$.

## AP ID

The AP identification number consists of two components, the position of the AP along the highway and the highway number which is based on the current policy on numbering and designating US highways. For instance, 11-191 indicates an AP with a position of 11 miles along the US191 highway. Assuming that we have approximately 830 highways and the maximum length of a highway is 3365 miles, we can conclude that the size of this

| | |
|---|---|
| **bits** | |
| 12 | 10 |
| **Position of AP** | **Highway number** |
| 3237 | 191 |

Fig. 9: Structure of AP ID.

field should be 22 bits. Vehicles use this information to later construct a unique Vehicle identification number to communicate with the APs along the road. Figure 9 shows the structure of the AP ID.

**Frame number**

Vehicles that request to participate in the VC should be aware of the structure of the communication frame, therefore it is necessary that the AP broadcasts the basic information about the layout of the frame, such as the current frame number. To find the maximum number of frames $f_{max}$ that one AP produces in a 24 hour period we should first find the length of a frame $f_l$ in terms of seconds which can be calculated by dividing the frame length ($T$) by the available bandwidth ($b$) that is provided for Dedicated short-range communications (DSRC). $f_l$ and $f_{max}$ then can be calculated as follows:

$$f_l = \frac{T\ (bits)}{b\ (bps)} = \frac{56624}{27 \times 10^6} \approx 0.002\ s$$

$$f_{max} = \frac{24 \times 3600\ (s)}{f_l\ (s)} \approx 43200000$$

Therefore the maximum number of frames can be encoded using approximately 26 bits.

**Number of fields**

Vehicles should be aware of the number of fields in each frame. Since our frame consists of 6 fields, we can encode this information with approximately 3 bits.

| | | | |
|---|---|---|---|
| **bits** | | | |
| 11 | 10 | 16 | 8 |
| **Size of T1 & T2 in bits** | **Size of T3 in bits** | **Size of T4 in bits** | **Size of T5 in bits** |
| 1744 | 672 | 53792 | 232 |

Fig. 10: Length of fields.

**Length of fields**

Vehicles should be aware of the length of each field in bits so that they can calculate the beginning of each field and slot (Figure 10). Having the length of each period, the number of bits that we need to be able to encode the length of the fields ($l_f$) can be calculated using the following formula:

$$\begin{aligned} l_f &= \lfloor(\log_2(T1+T2)+1)\rfloor + \lfloor(\log_2 T3+1)\rfloor + \lfloor(\log_2 T4+1)\rfloor + \lfloor(\log_2 T5+1)\rfloor \\ &= \lfloor(\log_2 1744+1)\rfloor + \lfloor(\log_2 672+1)\rfloor + \lfloor(\log_2 53792+1)\rfloor + \lfloor(\log_2 232+1)\rfloor \\ &= 45\ bits \end{aligned}$$

**Beginning of T1**

Vehicles should be aware of the beginning of $T1$ period. Having this information as well as other information such as number of fields and length of fields, vehicles can then calculate the beginning of each period precisely which helps them to communicate reasonably with the AP. We allocate 11 bits to encode the beginning of $T1$.

**Number of slots in T1**

The number of slots that are available in each establishment periods $T1$ and $T2$ should be transmitted to the vehicles in the coverage area, in order for the vehicles to be able to randomly select a communication slot and compete with other vehicles to receive a communication slot. Since the number of slots in these two periods are the same, the AP only needs to send the information regarding one of the establishment periods. The number of slots is assumed to be 20 in each transmission period which can be encoded in 5 bits. The preferred number of slots in the establishment period depends on other factors such as the number of lanes, the average distance between two vehicles and the average length of the vehicle. In our system, we assume that we have 3 lanes, the average distance between two vehicles is 10 meters and the length of each car is on average 5 meters.

**Bidding sequence**

The bidding sequence is one byte information, which is an indicator of the bidding policy in the establishment period which should be revealed to the vehicles participating in the VC which will be discussed in detail in the next subsection.

### 3.3.3 T1 AND T2: ESTABLISHMENT PERIOD

After receiving the initial signal and the frame layout from the AP, each vehicle that wants to communicate should compete for a slot based on the bidding sequence received in $T0$. There are two main bidding policies that are available to vehicles and one of these policies is selected based on the congestion level of the network. The probability of success for obtaining a slot for communication given that there are $k$ other vehicles competing is defined as follows:

$$s(k)\text{: } \Pr(the\ vehicle\ obtains\ a\ slot\ for\ communication | \\ k\ other\ vehicles\ are\ competing\ for\ a\ slot)$$

**Bidding policy 1** A vehicle that wants to communicate with the AP selects a random slot number from 1 to $M$ to transmit information in that slot in $T1$. If no other vehicle picks the same number then the vehicle can transfer information in that slot, otherwise, if another vehicle picks the same number then a collision occurs and neither of the vehicles that picked the same slot number can transmit in that slot. The vehicles get another chance to compete for a slot in $T2$. In this case, the probability of success is the probability that at least one of the two attempts to obtain a slot in one of the $M$ slots in either recognition periods $T1$ or $T2$ is successful, which can be computed as:

$$\begin{aligned} s(k) &: 1 - (1 - (1 - \tfrac{1}{M})^k)^2 \\ &= 2\left(1 - \tfrac{1}{M}\right)^k - \left(1 - \tfrac{1}{M}\right)^{2k} \end{aligned} \tag{2}$$

*Proof.* Let $A$ be the event that our distinguished car is successful at receiving a communication slot at some Access Point $A_n$.

Let $B_m$ be the event that $m$ cars ($m \geq 0$) are in the coverage area of $A_n$.

$$\Pr[\bigcup_{m\geq 0} B_m] = 1$$

Let $C_{m,k}$ be the event that $k$ of the $m$ vehicles in the coverage area of $A_n$ attempt to communicate.

$$\Pr[\bigcup_{k=0}^{m} C_{m,k}] = 1$$

Further let $T1$ be the event that our distinguished car will compete in $T1$ and let $T2$ be the event that our distinguished car will compete in $T2$. We assume that $\Pr[T1] = Pr[T2] = 1$.

$\Pr[A \mid T1 \cap C_{m,k} \cap B_m]$ is the conditional probability of success when a specific vehicle competes in $T1$ along with $k$ other vehicles selected from the $m$ vehicles that happen to be under the coverage area of $A_n$.

$$\begin{aligned}\Pr[A \mid T1 \cap C_{m,k} \cap B_m] &= \binom{M}{1}\frac{1}{M}(1-\frac{1}{M})^k \\ &= (1-\frac{1}{M})^k\end{aligned}$$

Similarly:

$$\Pr[A \mid T2 \cap C_{m,k} \cap B_m] = (1-\frac{1}{M})^k$$

$\Pr[A' \mid T1 \cap C_{m,k} \cap B_m]$ is the conditional probability of failure when a specific vehicle competes in $T1$ along with $k$ other vehicles selected from the $m$ vehicles that happen to be under the coverage area of $A_n$.

$$\Pr[A' \mid T1 \cap C_{m,k} \cap B_m] = 1-(1-\frac{1}{M})^k$$

Similarly:

$$\Pr[A' \mid T2 \cap C_{m,k} \cap B_m] = 1-(1-\frac{1}{M})^k$$

$\Pr[A' \mid C_{m,k} \cap B_m]$ is the conditional probability of failure when a specific vehicle competes in $T1$ and $T2$ along with $k$ other vehicles selected from the $m$ vehicles that happen to be under the coverage area of $A_n$.

$$\begin{aligned}\Pr[A' \mid C_{m,k} \cap B_m] &= \Pr[A' \mid T1 \cap C_{m,k} \cap B_m] \cdot \Pr[A' \mid T2 \cap C_{m,k} \cap B_m] \\ &= (1 - (1 - \frac{1}{M})^k)^2\end{aligned}$$

We then obtain $\Pr[A \mid C_{m,k} \cap B_m]$, the conditional probability of success when a specific vehicle competes in $T1$ and $T2$ along with $k$ other vehicles selected from the $m$ vehicles that happen to be under the coverage area of $A_n$:

$$\begin{aligned}\Pr[A \mid C_{m,k} \cap B_m] &= 1 - \Pr[A' \mid C_{m,k} \cap B_m] \\ &= 1 - (1 - (1 - \frac{1}{M})^k)^2 \\ &= 2(1 - \frac{1}{M})^k - (1 - \frac{1}{M})^{2k}\end{aligned}$$

$\square$

**Evaluation**

As described in the previous subsection, the probability of successful slot allocation as a function of $M$ (number of available slots in the establishment period) and $k$ (number of other vehicles in the coverage area that are competing for a slot), can be computed using Equation (2). To evaluate the accuracy of this prediction we have simulated a similar condition. In our simulation, a vehicle randomly selects a slot from $M$ slots and simultaneously $k$ other vehicles each randomly select a slot from $M$ slots, then the vehicles that picked unique slots are declared as successful vehicles in that period, however if multiple vehicles pick the same number then collision occurs and neither of the vehicles that picked the same slot is successful in that period. The vehicles again will repeat this process one more time and the ones that picked a unique slot number in first or second try will be declared as successful vehicles. Figure 11 shows our analytical predictions versus our simulation results which were averaged over $10^3$ runs.

**Bidding policy 2** A vehicle that wants to communicate with the AP, gets only one chance per frame to select $T1$ or $T2$ randomly and then select a random slot number from 1 to $M$

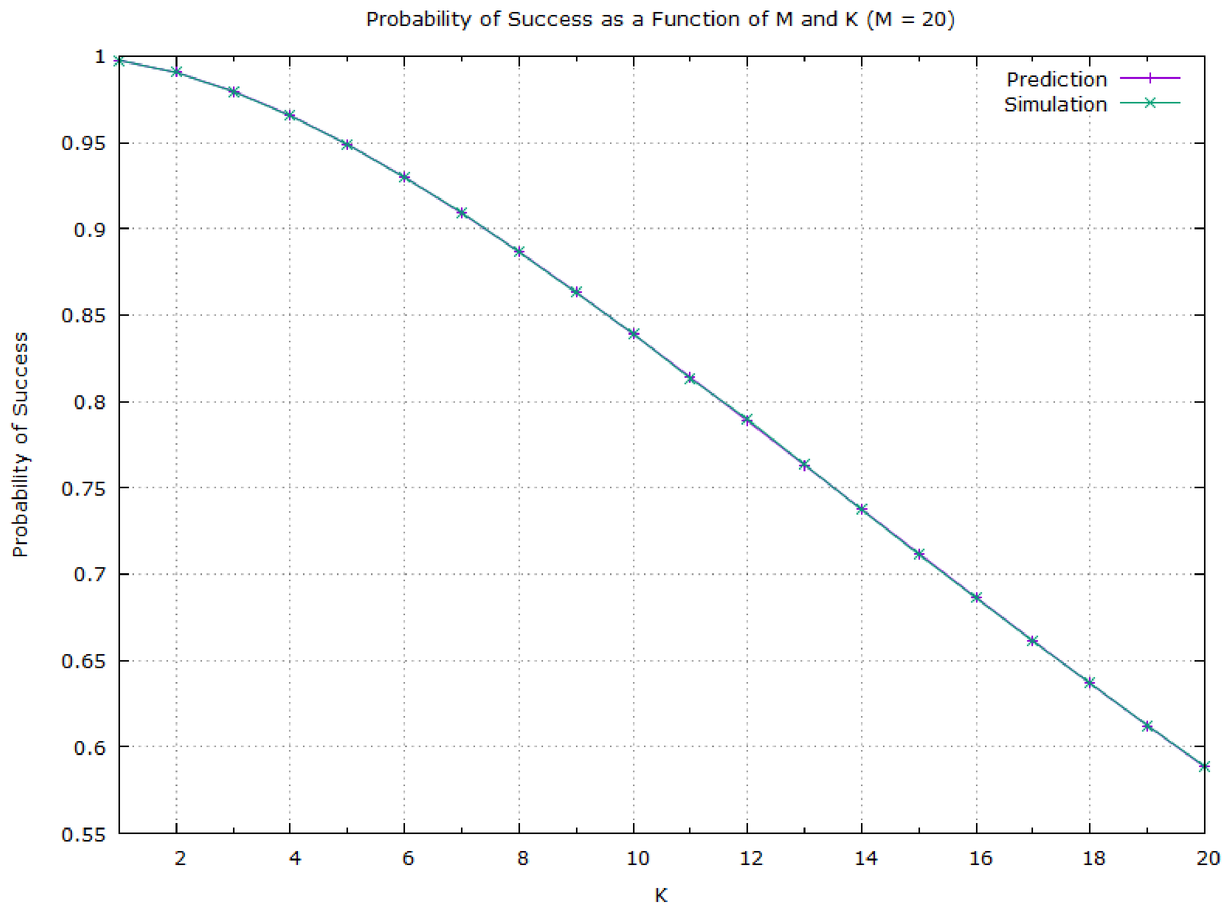


Fig. 11: Probability of successful slot allocation as a function of M and K for the first bidding policy.

and transmits in that slot. Similar to the previous bidding policy, if another vehicle picks the same number then a collision occurs and neither of the vehicles that picked the same slot number can transmit in that slot. In this case, the probability of success can be computed as:

$$s(k) : \left(1 - \tfrac{1}{2M}\right)^k \tag{3}$$

*Proof.* Let $A$ be the event that our distinguished car is successful at receiving a communication slot at some Access Point $A_n$.

Let $B_m$ be the event that $m$ cars ($m \geq 0$) are in the coverage area of $A_n$.

$$\Pr[\bigcup_{m\geq 0} B_m] = 1$$

Let $C_{m,k}$ be the event that $k$ of the $m$ vehicles in the coverage area of $A_n$ attempt to communicate.

$$\Pr[\bigcup_{k=0}^{m} C_{m,k}] = 1$$

Let $D_{k,k1}$ be the event that $k_1$ out of the $k$ vehicles that attempt to communicate with $A_n$ compete in $T1$, clearly:

$$\Pr[\bigcup_{k_1=0}^{k} D_{k,k_1}] = 1$$

We evaluate $\Pr[A \mid C_{m,k} \cap B_m]$ as follows:

$$\begin{aligned}
\Pr[A \mid C_{m,k} \cap B_m] &= \Pr[A \cap \bigcup_{k_1=0}^{k} D_{k,k_1} \mid C_{m,k} \cap B_m] \\
&= \Pr[\bigcup_{k_1=0}^{k} (A \cap D_{k,k_1}) \mid C_{m,k} \cap B_m] \\
&= \sum_{k_1=0}^{k} \Pr[A \cap D_{k,k_1} \mid C_{m,k} \cap B_m] \\
&= \sum_{k_1=0}^{k} \Pr[A \mid D_{k,k_1} \cap C_{m,k} \cap B_m] \Pr[D_{k,k_1} \mid C_{m,k} \cap B_m]
\end{aligned} \tag{4}$$

Using the binomial distribution we have:

$$\begin{aligned}
\Pr[D_{k,k_1} \mid C_{m,k} \cap B_m] &= \binom{k}{k_1} (\frac{1}{2})^{k_1} (1 - \frac{1}{2})^{k-k_1} \\
&= \binom{k}{k_1} (\frac{1}{2})^{k}
\end{aligned} \tag{5}$$

(4) and (5) yield:

$$\Pr[A \mid C_{m,k} \cap B_m] = \sum_{k_1=0}^{k} \binom{k}{k_1} (\frac{1}{2})^{k} \Pr[A \mid D_{k,k_1} \cap C_{m,k} \cap B_m] \tag{6}$$

Further let $T1$ be the event that our distinguished car will compete in $T1$ and let $T2$ be the event that our distinguished car will compete in $T2$. We assume that $\Pr[T1] = Pr[T2] =$

$\frac{1}{2}$. Therefore:

$$\begin{aligned}
\Pr[A \mid D_{k,k_1} \cap C_{m,k} \cap B_m] &= \Pr[A \cap (T1 \cup T2) \mid D_{k,k_1} \cap C_{m,k} \cap B_m] \\
&= \Pr[A \cap T1 \mid D_{k,k_1} \cap C_{m,k} \cap B_m] \\
&+ \Pr[A \cap T2 \mid D_{k,k_1} \cap C_{m,k} \cap B_m] \\
&= \Pr[A \mid T1 \cap D_{k,k_1} \cap C_{m,k} \cap B_m] \Pr[T1 \mid D_{k,k_1} \cap C_{m,k} \cap B_m] \\
&+ \Pr[A \mid T1 \cap D_{k,k_1} \cap C_{m,k} \cap B_m] \Pr[T1 \mid D_{k,k_1} \cap C_{m,k} \cap B_m]
\end{aligned}$$

Let $\Pr[T1 \mid D_{k,k_1} \cap C_{m,k} \cap B_m] = \frac{1}{2}$.
$\Pr[A \mid T1 \cap D_{k,k_1} \cap C_{m,k} \cap B_m]$ is the conditional probability of success when a specific vehicle competes in $T1$ along with $k_1$ other vehicles selected from the $m$ vehicles that happen to be under the coverage area of $A_n$.

$$\begin{aligned}
\Pr[A \mid T1 \cap D_{k,k_1} \cap C_{m,k} \cap B_m] &= \binom{M}{1} \frac{1}{M} (1 - \frac{1}{M})^{k_1} \\
&= (1 - \frac{1}{M})^{k_1}
\end{aligned}$$

Similarly:

$$\Pr[A \mid T2 \cap D_{k,k_1} \cap C_{m,k} \cap B_m] = (1 - \frac{1}{M})^{k-k_1}$$

It follows that:

$$\Pr[A \mid D_{k,k_1} \cap C_{m,k} \cap B_m] = \frac{1}{2}(1 - \frac{1}{M})^{k_1} + \frac{1}{2}(1 - \frac{1}{M})^{k-k_1} \tag{7}$$

**Theorem 1** (Newton's Binomial Theorem). *Let $\alpha$ be a real (or even complex) number. Then:*

$$(x+y)^\alpha = \sum_{n=0}^{\infty} \binom{\alpha}{n} x^n y^{\alpha - n}$$

By (6), (7) and Newton's Binomial Theorem (Theorem 1) we then obtain $\Pr[A \mid C_{m,k} \cap B_m]$, the conditional probability of success when a specific vehicle competes in $T1$ or $T2$ along with $k_1$ other vehicles selected from the $m$ vehicles that happen to be under the coverage area of $A_n$:

$$
\begin{aligned}
\Pr[A \mid C_{m,k} \cap B_m] &= \sum_{k_1=0}^{k} \binom{k}{k_1} (\frac{1}{2})^k [\frac{1}{2}((1-\frac{1}{M})^{k_1} + (1-\frac{1}{M})^{k-k_1})] \\
&= \frac{1}{2^{k+1}} [\sum_{k_1=0}^{k} \binom{k}{k_1} (1-\frac{1}{M})^{k_1} + \sum_{k_1=0}^{k} \binom{k}{k_1} (1-\frac{1}{M})^{k-k_1}] \\
&= \frac{1}{2^{k+1}} [\sum_{k_1=0}^{k} \binom{k}{k_1} (1-\frac{1}{M})^{k_1} 1^{k-k_1} + \sum_{k_1=0}^{k} \binom{k}{k_1} 1^{k_1} (1-\frac{1}{M})^{k-k_1}] \\
&= \frac{1}{2^{k+1}} [((1-\frac{1}{M}) + 1)^k + (1 + (1-\frac{1}{M}))^k] \\
&= \frac{1}{2^{k+1}} [(1+1-\frac{1}{M})^k + (1+1-\frac{1}{M})^k] \\
&= \frac{1}{2^{k+1}} \cdot 2 \cdot (2-\frac{1}{M})^k \\
&= \frac{1}{2^k} \cdot (2-\frac{1}{M})^k \\
&= (1-\frac{1}{2M})^k
\end{aligned}
$$

$\square$

**Evaluation**

As described in the previous subsection, the probability of successful slot allocation as a function of $M$ and $k$ can be computed using Equation (3). To evaluate the accuracy of our predictions we have simulated a similar condition. In our simulation, a vehicle randomly selects whether it wants to compete in $T1$ or $T2$. It then selects a slot from $M$ slots and simultaneously $k$ other vehicles each randomly select $T1$ or $T2$ and then select a slot from $M$ slots, then the vehicles that picked a unique slot are declared as successful vehicles in that period. Figure 12 shows our analytical predictions versus our simulation results which were averaged over $10^3$ runs.

**Transmitted Information**

The transmitted information in $T1$ and $T2$ differs based on the status of the vehicle and can be classified as follows:

1. Initial request: A vehicle that contacts the AP for the first time to receive a job does not have an ID and therefore in $T1$ the vehicle transmits an estimated number of

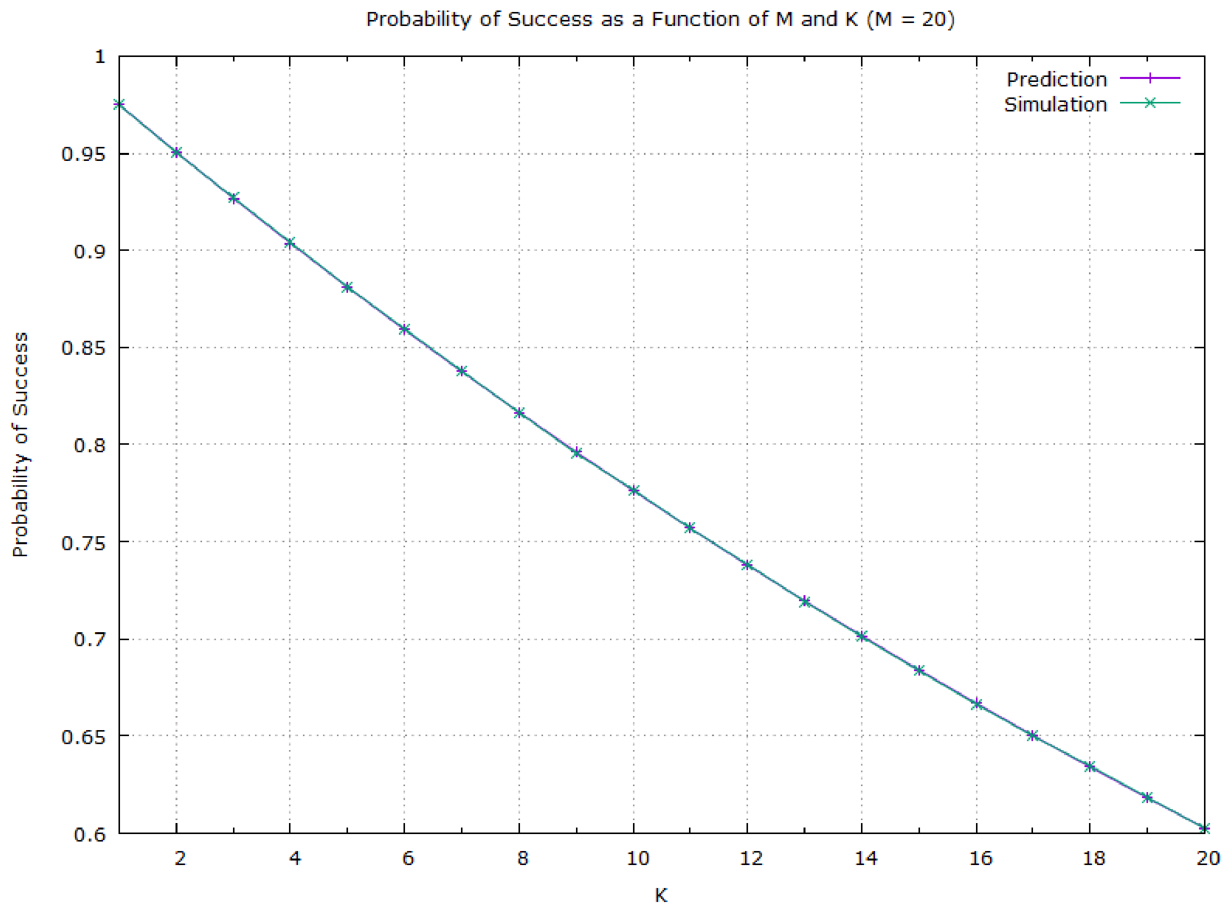


Fig. 12: Probability of successful slot allocation as a function of M and K for the second bidding policy.

miles that it will be on the highway and the slot number that it will select in $T2$. In $T2$, the vehicle sends the estimated number of miles that it will be on the highway, and the slot number selected in $T1$. This will help the system to identify the vehicles that successfully obtained a slot in both transmission periods.

2. Job download request: Vehicles that are requesting to download an assigned job should compete for a slot and therefore send the vehicle ID and a sequence indicating a request to download the job.
3. Input data request: Vehicles that successfully downloaded the assigned job and need to download the intermediate input data send the vehicle ID and a sequence indicating a request to download the input data.
4. Job submission and migration request: Vehicles that complete the assigned job and

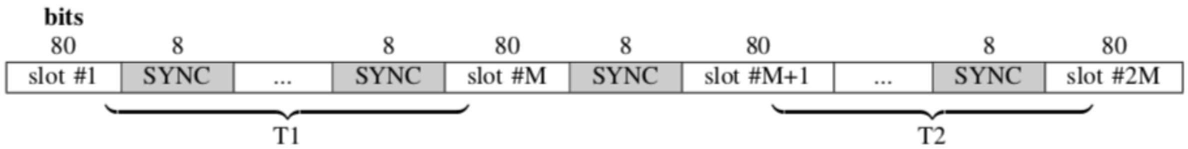


Fig. 13: Structure of T1 and T2.

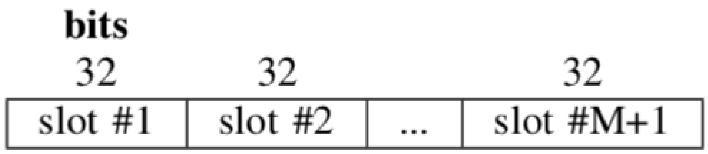


Fig. 14: Structure of T3.

want to upload the results or want to migrate an incomplete job should send their vehicle ID, along with a sequence indicating the request.

Figure 13 shows the structure of T1 and T2.

### 3.3.4 T3: ID AND AVAILABILITY ACKNOWLEDGMENT

This period has $2M + 1$ slots and the purpose of it is to send acknowledgments to the vehicles that successfully obtained a slot in the establishment period. As a general rule, each vehicle that obtained the slot number $k$ in the establishment period should listen to the slot number $k$ to in this period to receive the acknowledgment. Each acknowledgment is based on the messages that vehicles transmitted in the establishment period. The last slot contains information regarding the length of each slot and the number of vehicles that are guaranteed a slot in the transmission period. Figure 14 shows the structure of T3.

### 3.3.5 T4 AND T5: TRANSMISSION PERIOD AND TRANSMISSION ACKNOWLEDGMENT

In this period the acknowledged vehicles can transmit or receive information in the assigned slot. The number of slots in this period $(N)$ and the size of each slot is adjusted based on the number of vehicles that were guaranteed a transmission slot. In $T5$, vehicles that received a message in $T4$ send the acknowledgments to the AP. Figure 15 shows the

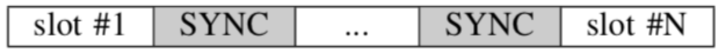


Fig. 15: Structure of T4.

structure of T4 which is identical to the structure of T5.

Table 3 shows the frame parameters.

TABLE 3: Frame parameters

| **Symbol and Description** | **Value** |
|---|---|
| $f_l$ (length of a frame) | 0.002 s |
| $f_{max}$ (maximum number of frames produced by an AP in 24 hours) | 41203900 |
| $T$ (one frame length) | 56624 bits |
| $T0$ (open communication period) | 119 bits |
| $T1 + T2$ (recognition period) | 1744 bits |
| $T3$ (ID and availability acknowledgment period) | 672 bits |
| $T4$ (transmission period) | 53792 bits |
| $T5$ (transmission acknowledgment period) | 231 bits |
| $M$ (number of slots in T1 and T2) | 20 |

# CHAPTER 4

# JOB DOWNLOAD UNDER ONE ACCESS POINT

In this chapter, we explore a VC in which download of a job can be completed under one AP. We provide the workload of interest, examples of practical applications and finally present the analytical predictions and simulation results of job completion time for such cases.

## 4.1 REVIEW OF THE COMMUNICATION MODEL

As explained in Chapter 3 the communication between APs and passing vehicles is supported by a variant of DSRC. The Dedicated Short Range Communications (DSRC) is the wireless communication standard for vehicular communications designed to facilitate V2I and V2V communications. DSRC provides high data transfer rates (i.e. 27 Mbps) with minimized latency, which is convenient for the highly mobile nature of vehicles and transportation systems [81]. Since each coverage area may contain several vehicles, the communication between the APs and the vehicles in their coverage area is, necessarily, contention-based. Specifically, the APs continuously send out frames of size $F$ bits containing a fixed length payload of $b$ bits as illustrated in Figure 16. In each frame, vehicles that are under the coverage area and wish to communicate with the AP compete with each other to secure a communication slot in that frame. As in [37], we assume that each frame has two adjacent contention periods of $M$ minislots each. In each contention slot a vehicle that wishes to communicate with the AP picks a random number between 1 and $M$ and transmits its identity in the corresponding minislot. This is repeated in the second contention period. If two vehicles transmit in the same minislot a collision occurs and the AP receives a garbled message. Vehicles whose message gets to the AP ungarbled in at least one contention period are considered successful and are allocated one data slot in the frame [92, 93].

In the Chapter 3, we discussed two main bidding policies for vehicles that want to communicate with the AP and compete with other vehicles to receive a communication slot and provided the probability of successful slot allocation for each of these bidding policies (see Equations (2) and (3)). Since the success probability in the first bidding policy is higher for the values of $M$ that are of practical interest (i.e. $M \leq 20$) we will adopt the first bidding scheme.

We let $p_{k+1}$ denote the conditional probability of successful slot allocation given that the vehicle competes with $k$ other vehicles in a given frame. We have computed $p_{k+1}$ in Equation (2).

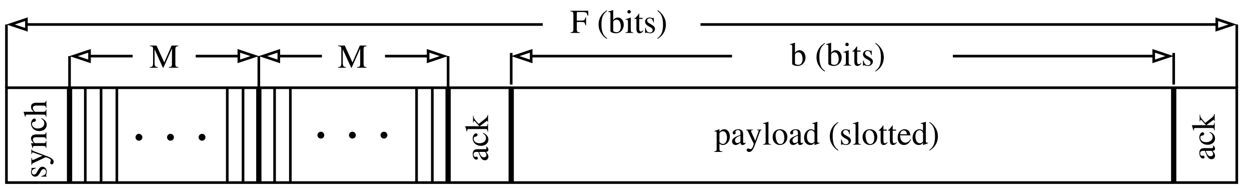


Fig. 16: General structure of a communication frame.

From the standpoint of a given vehicle, a frame is *successful* if the vehicle has secured a communication slot in that frame. In each frame, the payload of $b$ bits is partitioned evenly among the successful vehicles in that frame. We assume a communication bandwidth of $B = 27$Mbps which is the maximum data transmission rate in DSRC.

### 4.2 WORKLOADS OF INTEREST

The workloads contemplated for the VC in which the download of a job can be completed under one AP have a number of defining features in common:

- they are all delay-tolerant, not constrained by real-time requirements;
- the input size is relatively small, say, a few MBs;
- the processing time is relatively short;
- they are, as a rule, CPU-bound and do not require inter-processor communication.

Examples of such workloads include:

- *Machine translation* of short documents, say, from English to Spanish or vice-versa. These types of documents include all sorts of memoranda, personal and business letters, job applications, newsletters, sundry legal documents, resumes, and the like. As pointed out by [94], context-based machine translation (CBMT) is a new paradigm for corpus-based translation that requires no parallel text. CBMT relies on a light-weight translation model utilizing a full-form bilingual dictionary and a sophisticated decoder using long-range context via long n-grams and cascaded overlapping;
- *$N$-body simulation* of dynamic systems. An $N$-body simulation approximates the motion of particles that are known to interact with one another through some type of (electro)-mechanical forces. The types of particles that can be simulated using $N$-body methods are quite varied, ranging from vehicles in dense traffic, to trajectories

of particles in Brownian motion applications, to predicting the interaction of celestial bodies, to predicting the diffusion of individual atoms in an enclosure filled with gas, among many similar ones [95];

- *Molecular simulations:* Molecular mechanics is a growing area of interdisciplinary research and it involves the use of classical mechanics to describe the physical basis molecular models. These models typically describe atoms as point charges with an associated mass [96]. The interactions between neighboring atoms are described by spring-like interactions [97];
- *Monte Carlo simulations:* Monte Carlo simulations are very important in computational physics, chemistry and related applied fields. Monte Carlo simulations are also very widely used in engineering for sensitivity analysis and in the study of various "what if" scenarios [98];
- *Strategic Planning for Emergency Evacuations:* As pointed out by [99,100] and others, traditional evacuation schemes utilize predetermined routes that work reasonably well for hurricanes and other similar planned events. However, these routes are likely infeasible for unplanned evacuation such as in the wake of a terrorist attack or a HAZMAT spill. Raw *et al.* [82] argued that dynamic VCs will be able to suggest dynamic clustering of vehicles over large areas to improve the efficiency of unplanned evacuations such as earthquakes and other similar natural or man-made disasters;
- *Vehicular crowdsourcing applications:* It has been suggested that there is a need to push cloud computation towards the edge of the network, closer to where the data sources are [101]. The vehicles on our highways and city streets are witnessing all sorts of traffic-related events and can be tasked with logging and reporting such events [7].

## 4.3 NOTATION AND TERMINOLOGY

The characteristics of the workloads above motivate us to make the following assumptions about the VC.

- The size of the downloaded job is (roughly) equal to the size of the uploaded result;
- The job and the results of the job processing can be downloaded under the coverage area of one AP;
- The results of the job processing can be uploaded under the coverage area of one or at most two consecutive APs;
- A job can be completed by one vehicle and there is no need for job migration. meaning that the residency time of a vehicle is larger than job completion time.

Let ${N^*}_{k+1}$ denote the total number of frames that a generic vehicle entering the coverage area of an arbitrary AP will see while under the coverage area. ${N^*}_{k+1}$ is counting all the frames, including complete and incomplete ones as illustrated in the Figure 17.

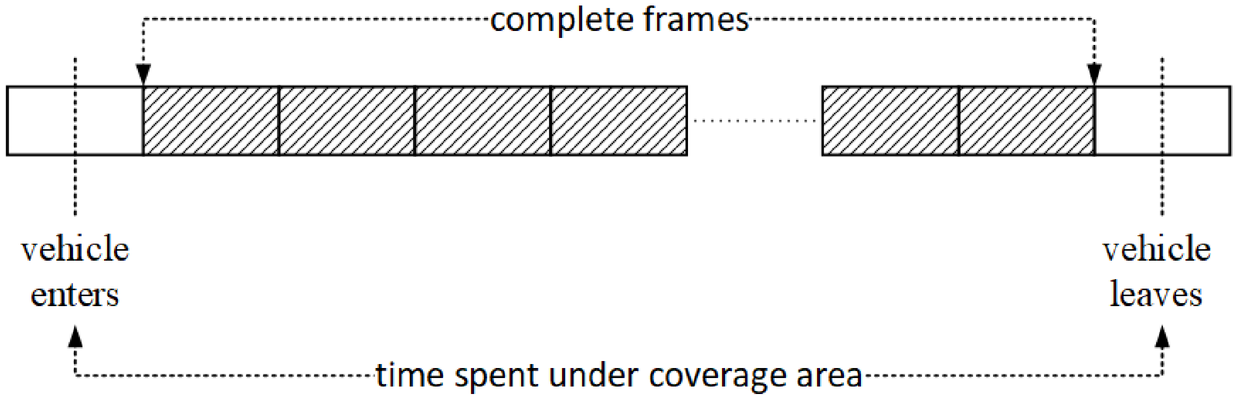


Fig. 17: Illustrating complete and incomplete frames a generic vehicle entering the coverage area of an arbitrary AP will see while under the coverage area.

Let $N_{k+1}$ denote the number of complete frames that a vehicle sees while in the coverage area of an arbitrary AP.

Let $B$ be the available bandwidth in *bps*, and $c$ is the size of the AP's coverage area in *meters*, $F$ is the length of each frame in *bits*, and $v_{k+1}$ is the vehicle's speed in *meters per second* given a density of $k+1$ vehicles, we can then compute $N_{k+1}$ as,

$$N_{k+1} = \left\lfloor \frac{c}{v_{k+1}} \cdot \frac{B}{F} \right\rfloor = \left\lfloor \frac{cB}{v_{k+1}F} \right\rfloor .$$

We are interested in $E[N_{k+1}]$. We instantiate the results in the Appendix 7.2 with $x = \frac{F}{B}$ and $Z = \frac{c}{v_{k+1}}$. It is easy to confirm that,

$$E[N_{k+1}] = \frac{cB}{Fv_{k+1}} - 1, \tag{8}$$

the derivation of which is,

$$
\begin{aligned}
E[N_{k+1}] &= \int_{u=0}^{\frac{F}{B}\left\lfloor \frac{\frac{c}{v_{k+1}}}{\frac{F}{B}} \right\rfloor + \frac{F}{B} - \frac{c}{v_{k+1}}} \left( \left\lfloor \frac{\frac{c}{v_{k+1}}}{\frac{F}{B}} \right\rfloor - 1 \right) \frac{du}{\frac{F}{B}} + \int_{u=\frac{F}{B}\left\lfloor \frac{\frac{c}{v_{k+1}}}{\frac{F}{B}} \right\rfloor + \frac{F}{B} - \frac{c}{v_{k+1}}}^{\frac{F}{B}} \left\lfloor \frac{\frac{c}{v_{k+1}}}{\frac{F}{B}} \right\rfloor \frac{du}{\frac{F}{B}} \\
&= \frac{1}{\frac{F}{B}} \left[ \left( \frac{F}{B} \left\lfloor \frac{\frac{c}{v_{k+1}}}{\frac{F}{B}} \right\rfloor + \frac{F}{B} - \frac{c}{v_{k+1}} \right) \left( \left\lfloor \frac{\frac{c}{v_{k+1}}}{\frac{F}{B}} \right\rfloor - 1 \right) + \left( \frac{c}{v_{k+1}} - \frac{F}{B} \left\lfloor \frac{\frac{c}{v_{k+1}}}{\frac{F}{B}} \right\rfloor \right) \left\lfloor \frac{\frac{c}{v_{k+1}}}{\frac{F}{B}} \right\rfloor \right] \\
&= \frac{1}{\frac{F}{B}} \left[ \frac{F}{B} \left\lfloor \frac{\frac{c}{v_{k+1}}}{\frac{F}{B}} \right\rfloor^2 - \frac{F}{B} \left\lfloor \frac{\frac{c}{v_{k+1}}}{\frac{F}{B}} \right\rfloor + \frac{F}{B} \left\lfloor \frac{\frac{c}{v_{k+1}}}{\frac{F}{B}} \right\rfloor \right. \\
&\quad \left. - \frac{F}{B} - \frac{c}{v_{k+1}} \left\lfloor \frac{\frac{c}{v_{k+1}}}{\frac{F}{B}} \right\rfloor + \frac{c}{v_{k+1}} + \frac{c}{v_{k+1}} \left\lfloor \frac{\frac{c}{v_{k+1}}}{\frac{F}{B}} \right\rfloor - \frac{F}{B} \left\lfloor \frac{\frac{c}{v_{k+1}}}{\frac{F}{B}} \right\rfloor^2 \right] \\
&= \frac{1}{\frac{F}{B}} \left( \frac{c}{v_{k+1}} - \frac{F}{B} \right) \\
&= \frac{\frac{c}{v_{k+1}}}{\frac{F}{B}} - 1 \\
&= \frac{cB}{Fv_{k+1}} - 1.
\end{aligned}
$$

Let $W$ be the size of the job or the container image in *bits* and let the random variable $r_{k+1}$ denote the number of successful frames necessary to download (or upload) the corresponding job. Recalling $b$ is the available payload per frame in *bits*, and $p_{k+1}$ is the conditional probability of successful slot allocation given that the vehicle competes with $k$ other vehicles in a given frame.

We can compute $r_{k+1}$ as the size of the job in *bits* ($W$), divided by the available payload for each successful vehicle per frame.

It is clear that to find the available payload for each successful vehicle per frame, we divide the available payload per frame in *bits* ($b$) by the number of potential successful vehicles in the area which is computed by the multiplication of $(k + 1)$ and the conditional probability of successful slot allocation given that the vehicle competes with $k$ other vehicles in a given frame ($p_{k+1}$).

Clearly,

$$r_{k+1} = \frac{W}{\frac{b}{(k+1)p_{k+1}}} = \frac{(k+1)Wp_{k+1}}{b}. \tag{9}$$

For all $k$, $(k \geq 0)$, let the random variables $D_{k+1}$ (resp. $U_{k+1}$) represent the *total number of frames* in which the vehicle has to compete, in order to complete the download (resp. upload) of a job, given that in each frame $k$ other vehicles are also competing for a slot.

It is easy to see that $D_{k+1}$ and $U_{k+1}$ have a negative binomial distribution with parameters $r_{k+1}$ and $p_{k+1}$.

To clarify we briefly review the properties of the negative binomial distribution.

**Theorem 2** (Negative Binomial Distribution)**.** *In probability theory and statistics, the Pascal distribution also known as the negative binomial distribution is a discrete probability distribution used to model the number of failures before the nth success in a sequence of independent and identically distributed Bernoulli trials, where the probability of success at each trial is $p$ and the probability of failure is $q = 1 - p$, [102].*

*The probability mass function of the negative binomial distribution where the probability of success at each trial is $p$ and the probability of failure is $q = 1 - p$ and $X$ is the number of $n$ trials that occur for a given number of $r$ successes can be expressed as,*

$$f(n; r, p) \equiv \Pr[X = n] = \binom{n-1}{r-1} p^r (1-p)^{n-r}.$$

*The expected value of $X$ is obtained as,*

$$E[X] = \frac{r}{p} \tag{10}$$

It follows (see, for example, [103] p. 61) that for an arbitrary non-negative integer $m$,

$$\Pr[D_{k+1} = m] = \binom{m-1}{r_{k+1}-1} p_{k+1}^{r_{k+1}} (1 - p_{k+1})^{m - r_{k+1}}. \tag{11}$$

Since $D_{k+1}$ and $U_{k+1}$ are identically distributed, $\Pr[U_{k+1} = m]$ has the same expression. By (10), (11) and (9) it follows that the expectations $E[D_{k+1}]$ and $E[U_{k+1}]$ are

$$E[D_{k+1}] = E[U_{k+1}] = \frac{r_{k+1}}{p_{k+1}} = \frac{W(k+1)}{b}. \tag{12}$$

It is interesting to note that $E[D_{k+1}]$ is independent of the success probability $p_{k+1}$ and only depends on the size $W$ of the job to download (or upload), the payload $b$, and the traffic density $k + 1$.

### 4.3.1 PROBABILITY THAT A JOB CANNOT BE DOWNLOADED UNDER THE COVERAGE OF A SINGLE AP

In this subsection, we present a solution to determine the probability that a job cannot be downloaded under the coverage of a single AP.

We let $\Pr[Y]$ denote the probability that a job cannot be downloaded under the coverage of a single AP. This probability an be expressed as,

$$\Pr[Y] = \sum_{j=1}^{r_{k+1}-1} \binom{N_{k+1}}{j} {p_{k+1}}^{j}(1-p_{k+1})^{N_{k+1}-j}. \tag{13}$$

Recall that the random variable $r_{k+1}$ denotes the number of successful frames necessary to download (or upload) the corresponding job; $N_{k+1}$ denotes the number of complete frames that a vehicle sees while in the coverage area of an arbitrary AP; $p_{k+1}$ denotes the conditional probability of successful slot allocation given that the vehicle competes with $k$ other vehicles in a given frame.

An approximation that is appropriate and used for the accurate evaluation of the factorials of large numbers is the Stirling's Approximation (or Stirling's formula).

**Theorem 3** (Stirling's Approximation)**.** *For two real sequences $x_n$ and $y_n$, we write $x_n \sim y_n$ if $\lim_{n\to\infty} \frac{x_n}{y_n} = 1$ [104]. Stirling's approximation to the factorial is typically written as,*

$$n! \sim \sqrt{2\pi n}(\frac{n}{e})^{n}.$$

*Using the Stirling's Approximation for all factorials in $\binom{n}{k}$ gives,*

$$\begin{aligned}
\binom{n}{k} &= \frac{n!}{k!(n-k)!} \\
&= \frac{\sqrt{2\pi n}(\frac{n}{e})^{n}}{\sqrt{2\pi k}(\frac{k}{e})^{k} \cdot \sqrt{2\pi(n-k)}(\frac{n-k}{e})^{n-k}} \\
&= \sqrt{\frac{n}{2\pi k(n-k)}}(\frac{n}{k})^{k}(\frac{n}{n-k})^{n-k}.
\end{aligned}$$

By applying the Theorem 14 to Equation (13) we then get,

$$\Pr[Y] = \sum_{j=1}^{r_{k+1}-1} \sqrt{\frac{N_{k+1}}{2\pi j(N_{k+1}-j)}} \cdot (\frac{p_{k+1}N_{k+1}}{j})^{j} \cdot (\frac{(1-p_{k+1})N_{k+1}}{N_{k+1}-j})^{N_{k+1}-j}.$$

Table 4 summarizes the values of $\Pr[Y]$ for various values of $k$ by using the Equation (14). It is easy to see that, for the workloads discussed in this chapter and assuming a contained image of 1.5MB, the probability that the job cannot be downloaded under the coverage of a single AP is 0 for all values of k from 1 to 12.

## 4.4 APPROXIMATING JOB COMPLETION TIME

TABLE 4: Probability that a job cannot be downloaded under the coverage of a single AP (for job size 1.5 MB), using the Equation (14).

| **k** | $\mathbf{Pr}[\boldsymbol{Y}]$ |
|---|---|
| 1 | 0 |
| 2 | 0 |
| 3 | 0 |
| 4 | 0 |
| 5 | 0 |
| 6 | 0 |
| 7 | 0 |
| 8 | 0 |
| 9 | 0 |
| 10 | 0 |
| 11 | 0 |
| 12 | 0 |

The goal of this section is to investigate the job completion time in dynamic VCs for the cases in which download of a job can be completed under one AP.

We begin by summarizing, for the readers' convenience the notation and terminology used throughout the remainder of this thesis.

Let $T$ be the random variable that keeps track of the execution time of the user job in the absence of any overhead attributable to the VC. We do not assume knowledge of the probability distribution of $T$.

Similarly, let $N$ be the random variable that keeps track of the subscript of the AP at which the job has completed, in other words, the results of the job execution have finished uploading. If results are uploaded under two APs then these APs are consecutive meaning that the vehicle does not wait longer than necessary to upload the result. To evaluate the job completion time, we distinguish between the three cases below.

**Case 1:** *Job execution terminates under the coverage of some AP but the results finish uploading under the coverage of the next AP.*

Referring to Figure 18, assume that the job execution terminates under the coverage of $AP_{N-1}$ and that the upload of the results begin in the same coverage area but finishes under

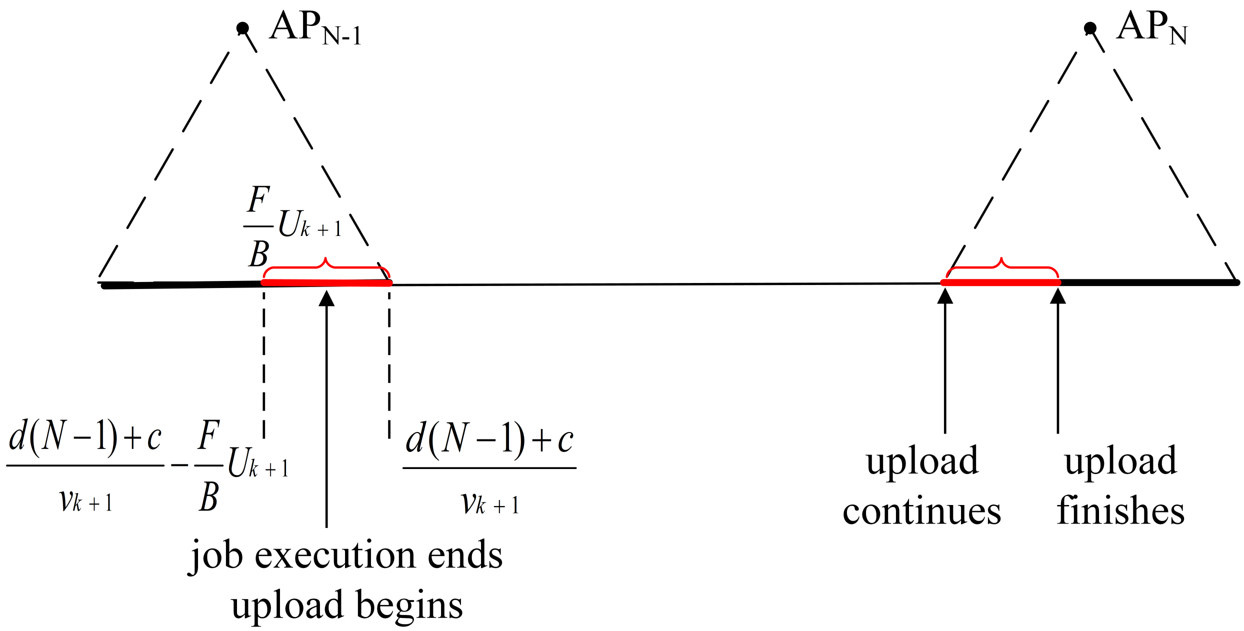


Fig. 18: Illustrating Case 1

the coverage of $AP_N$. In fact, the upload is interrupted when the vehicles leaves the coverage area of $AP_{N-1}$ and will resume when the vehicle enters the coverage area of $AP_N$.

In this case, it is natural to define the job completion time $J_1$ as:

$$J_1 = \frac{F}{B}(D_{k+1} + U_{k+1}) + T + \frac{d-c}{v_{k+1}}. \tag{14}$$

To justify the Equation (14), observe that $\frac{F}{B}D_{k+1} + T$ is the time it takes the vehicle to download the job and to execute the job, $\frac{d-c}{v_{k+1}}$ is the time it takes the vehicle to move between the coverage area of $AP_{N-1}$ and $AP_N$, and $\frac{F}{B}U_{k+1}$ is the combined time to upload the result.

Applying the expectation operator to the Equation (14), and using the Equation (12) and the linearity of expectation yields

$$E[J_1] = E[T] + \frac{2F}{B}E[D_{k+1}] + \frac{d-c}{v_{k+1}}. \tag{15}$$

**Case 2:** *Job execution terminates between the coverage areas of two adjacent APs.*

Referring to Figure 19, the job execution terminates after leaving the coverage area of $AP_{N-1}$ but before entering the coverage area of $AP_N$. The vehicle starts uploading the

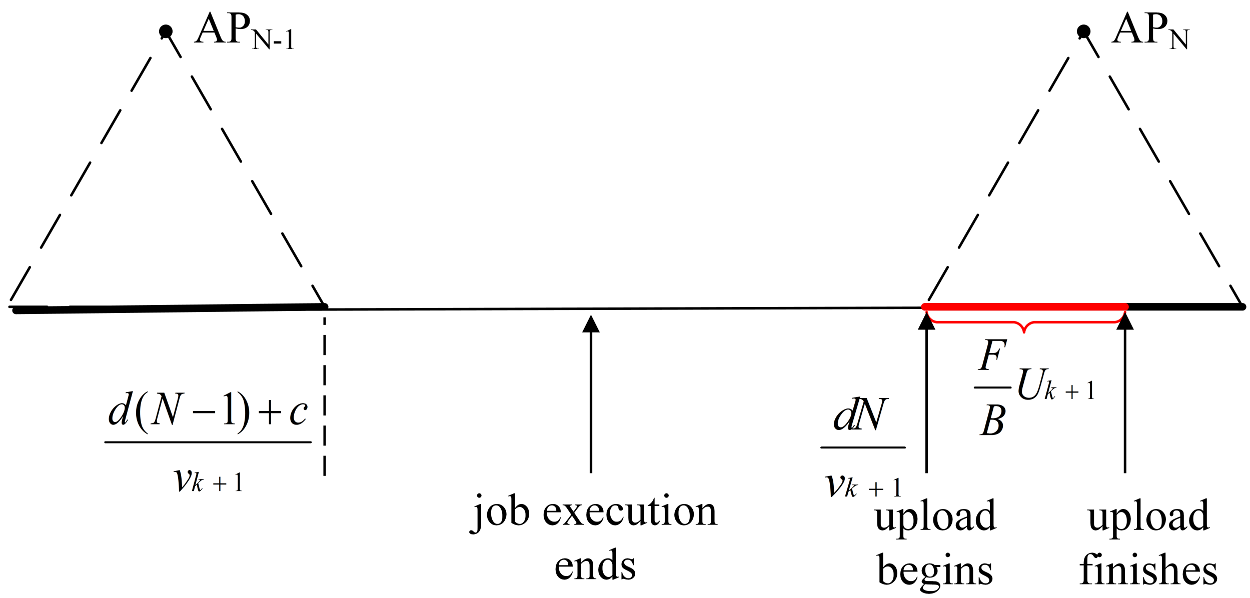


Fig. 19: Illustrating Case 2

results upon entering the coverage area of $\mathrm{AP_N}$. In this case, it is natural to define the job completion time $J_2$ as:

$$J_2 = \frac{dN}{v_{k+1}} + \frac{F}{B}U_{k+1}. \tag{16}$$

To justify (16) notice that the vehicle must first physically reach the coverage area of $\mathrm{AP_N}$ before it can start uploading the results. The former time is $\frac{dN}{v_{k+1}}$, while the latter is $\frac{F}{B}U_{k+1}$. Here, $N$ is the *unique* natural number for which job download and execution terminates strictly between the coverage areas of $\mathrm{AP_{N\text{-}1}}$ and that of $\mathrm{AP_N}$. In other words, $N$ satisfies

$$\frac{(N-1)d+c}{v_{k+1}} < \frac{F}{B}D_{k+1} + T \le \frac{Nd}{v_{k+1}}. \tag{17}$$

By applying the expectation operator to (16), we obtain

$$E[J_2] = \frac{dE[N]}{v_{k+1}} + \frac{F}{B}E[U_{k+1}]. \tag{18}$$

In order to obtain an expression for $E[J_2]$ we proceed as follows: From (17) by simple algebra we obtain, in stages,

$$\frac{F}{B}D_{k+1} + T < \frac{dN}{v_{k+1}} \le \frac{F}{B}D_{k+1} + T + \frac{d-c}{v_{k+1}}. \tag{19}$$

Applying the expectation operator to (19) yields

$$\frac{F}{B}E[D_{k+1}] + E[T] \leq \frac{dE[N]}{v_{k+1}} \leq \frac{F}{B}E[D_{k+1}] + E[T] + \frac{d-c}{v_{k+1}}. \tag{20}$$

After adding $\frac{F}{B}E[U_{k+1}]$ throughout in (20) and recalling (12) and (16), we write

$$E[T] + \frac{2F}{B}E[D_{k+1}] \leq E[J_2] \leq E[T] + \frac{2F}{B}E[D_{k+1}] + \frac{d-c}{v_{k+1}},$$

which yields the following approximation for $E[J_2]$ that turns out to be quite accurate and the accuracy of which will be assessed in detail.

$$E[J_2] \approx E[T] + \frac{2F}{B}E[D_{k+1}] + \frac{d-c}{2v_{k+1}}. \tag{21}$$

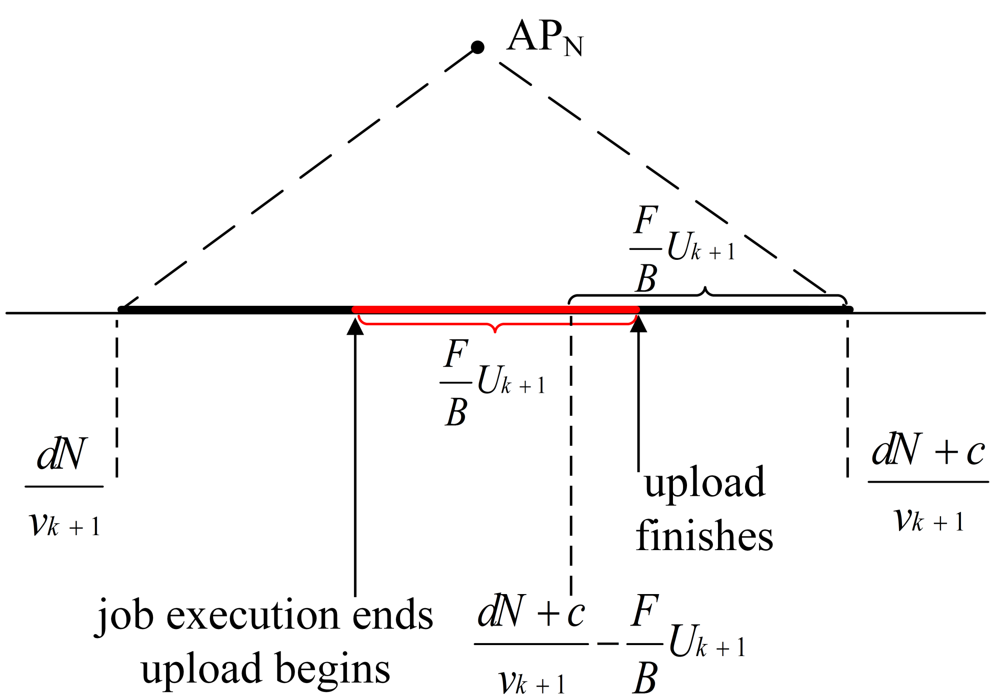


Fig. 20: Illustrating Case 3

**Case 3:** *Job execution terminates under the coverage of some AP and the results finish uploading under the coverage of the same AP.*

Referring to Figure 20, job execution finishes under the coverage area of $\mathrm{AP_N}$ and the results are uploaded under the coverage of $\mathrm{AP_N}$. In this case, it is natural to define the job completion time $J_3$ as

$$J_3 = \frac{F}{B}\left[D_{k+1} + U_{k+1}\right] + T. \tag{22}$$

Upon applying the expectation operator to (22) and using (12) and the linearity of expectation we obtain:

$$E[J_3] = E[T] + \frac{2F}{B} E[D_{k+1}]. \tag{23}$$

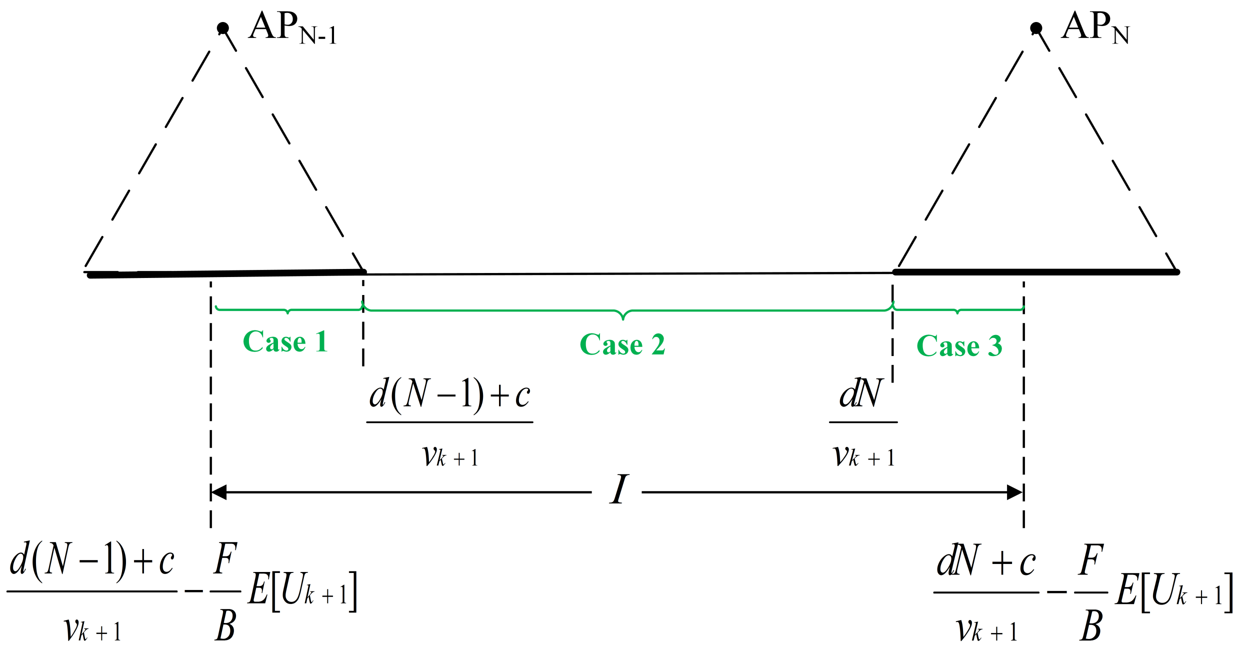


Fig. 21: Illustrating the evaluation of $E[J]$.

### 4.4.1 COMPLETING THE APPROXIMATION

The goal of this subsection is to combine the three cases discussed above into a coherent approximation of the job completion time. To compute the expected job completion time, we use the well-known *Law of Total Expectation* (Theorem 4) [105].

**Theorem 4** (Law of Total Expectation)**.** *Let $A_1, ..., A_n$ be a partition of a sample space, with $P(A_i) > 0$ for all $i$, and let $Y$ be a random variable on this sample space. Then*

$$E[X] = \sum_{i=1}^{n} E[X \mid A_i] P(A_i).$$

Let $\pi_1$, $\pi_2$ and $\pi_3$ be, respectively, the *limiting* probabilities of Case 1, Case 2 and Case 3 occurring. Using the Law of Total Expectation (Theorem 4), the expectation $E[J]$ of job completion time can be computed as

$$E[J] = \pi_1 E[J_1] + \pi_2 E[J_2] + \pi_3 E[J_3]. \tag{24}$$

To evaluate the limiting probabilities $\pi_1, \pi_2, \pi_3$, consider the time interval $I$ of length $\frac{d}{v_{k+1}}$ and refer to Figure 21.

$$I = \left[ \frac{d(N-1)+c}{v_{k+1}} - \frac{F}{B}E[U_{k+1}], \frac{dN+c}{v_{k+1}} - \frac{F}{B}E[U_{k+1}] \right] \tag{25}$$

Since the probability distribution of $T$ is not known, to a first approximation, we assume that job execution terminates, uniformly at random in the time interval $I$. In turn, this assumption implies that $\pi_1, \pi_2, \pi_3$ are given by the expressions

$$\pi_1 = \frac{FE[U_{k+1}]v_{k+1}}{Bd}; \ \pi_2 = \frac{d-c}{d}; \ \pi_3 = \frac{c}{d} - \pi_1. \tag{26}$$

Upon replacing the expressions of $\pi_1, \pi_2, \pi_3$ obtained in (26) back into (24) and recalling (15), (21) and (23), we obtain our approximation of the job completion time:

$$E[J] = E[T] + \frac{(3d-c)FE[D_{k+1}]}{Bd} + \frac{(d-c)^2}{2dv_{k+1}}.$$

## 4.5 EVALUATION OF JOB COMPLETION TIME

The main goal of this section is to discuss the details of our simulation model that we use to validate the theoretical results obtained in the previous sections.

We have simulated a three-lane highway with APs placed every 2000 meters. Each AP has a coverage area of 100 meters in which the vehicles driving along the highway can transmit or receive messages. The APs continuously send out frames of length 56624 bits with a payload of 53792 bits. We have implemented the different necessary fields in the frame, such as the Start of frame (SOF), end of frame (EOF), communication period, recognition period, transmission period and acknowledge (ACK). The speed of each vehicle is determined by the traffic density. In our simulation, we have used the five-parameter logistic speed-density function described in Chapter 3.2 to determine the vehicle's speed based on the number of the vehicles in the coverage area, the values of which are available in Table 2. When a vehicle enters the coverage area of an AP and receives the beginning of the frame, it competes with the other vehicles in the same coverage area. For this purpose, it chooses at random one of the 20 slots in the first contention period and the same procedure is repeated in the second contention period. Vehicles that select a unique slot in either contention periods are *successful*. The available payload is then divided equally among the successful vehicles. A vehicle that contacts the AP for the first time is assigned a job of size 1 MB with a processing time exponentially distributed with a common parameter $\lambda$. We vary $\lambda$ between $\frac{1}{1200}$ and $\frac{1}{1800}$, corresponding to an average processing time between 20 and 30 minutes. The vehicle then starts the download of the job and continues to compete in the next frames until the job is fully downloaded. The job execution starts immediately after the download of the job and once the job execution is completed, the vehicle immediately attempts to upload the results. The process of uploading of the results is similar to download, in terms of competing for transmission slots. If a vehicle is not under the coverage area at the time that the job execution is completed, then it attempts to upload the results at the next AP. We record the job completion time from the moment that the job is assigned to a vehicle until the results are uploaded. We have assumed that the residency time of a vehicle is larger than the job completion time. We also run our simulations for job with processing times that are normally distributed with mean $\lambda$ and uniformly distributed on the interval from $\lambda - a$ to $\lambda + a$. Table 6 shows the value of each parameter. Our simulations were developed in house, using the Java programming language, with each experiment repeated $10^4$ times. In the following subsection, the simulated and predicted job completion times for different $k$ and different job processing times are compared.

### 4.5.1 EVALUATION RESULTS

The simulation results for job completion times for $k$ values of 1 to 12 are shown in Figures 22 to 33. The job completion time values from simulations for exponentially distributed, uniformly distributed and normally distributed job processing times are plotted against the predicted values. The maximum relative error is less than 0.24%, with an average of 0.05% for uniform distribution, and less than 1.96%, with an average of 0.1% for exponential distribution, and less than 5.07%, with an average of 1.67% for normal distribution.

Our simulations have shown that the probabilities $\pi_1, \pi_2, \pi_3$ of (26) are respectively 0.0093, 0.95, and 0.0407. These values match closely the predicted probability values explained in Section 5.3.

Figure 34 shows a side-by-side view of the predicted and simulated job completion times for exponentially distributed, uniformly distributed, and normally distributed job processing times given that $k$ other vehicles are competing to receive a communication slot for $k$ values of 1 to 12.

TABLE 5: A summary of notation and terminology

| Symbol | Description |
|---|---|
| $l$ | number of lanes of traffic |
| $B$ | available bandwidth in bps |
| $W$ | size of the container image encapsulating the user job in bits |
| $b$ | payload per frame in bits |
| $F$ | frame length in bits |
| $M$ | number of available slots per contention period |
| $p_{k+1}$ | conditional probability of successful slot allocation given that the vehicle competes with $k$ other vehicles in a given frame |
| $N_{k+1}$ | number of (complete) frames a vehicle sees while in the coverage area of an arbitrary AP, given that $k$ other vehicles are also competing for slots |
| $c$ | size of access point coverage area in meters |
| $d$ | distance between two consecutive APs in meters |
| $v_{k+1}$ | vehicle's speed given a density of $k+1$ vehicles per coverage area |
| $T$ | execution time time of a job in the absence of any overhead attributable to the VC. Only $E[T]$ is assumed known |
| $J$ | job completion time, including all overhead attributable to the VC |
| $r_{k+1}$ | number of successful frames necessary to download the job given a density of $k+1$ vehicles per coverage area |
| $D_{k+1}$ | total number of frames necessary to download the job given a density of $k+1$ vehicles per coverage area |
| $U_{k+1}$ | total number of frames necessary to upload the results given a density of $k+1$ vehicles per coverage area |

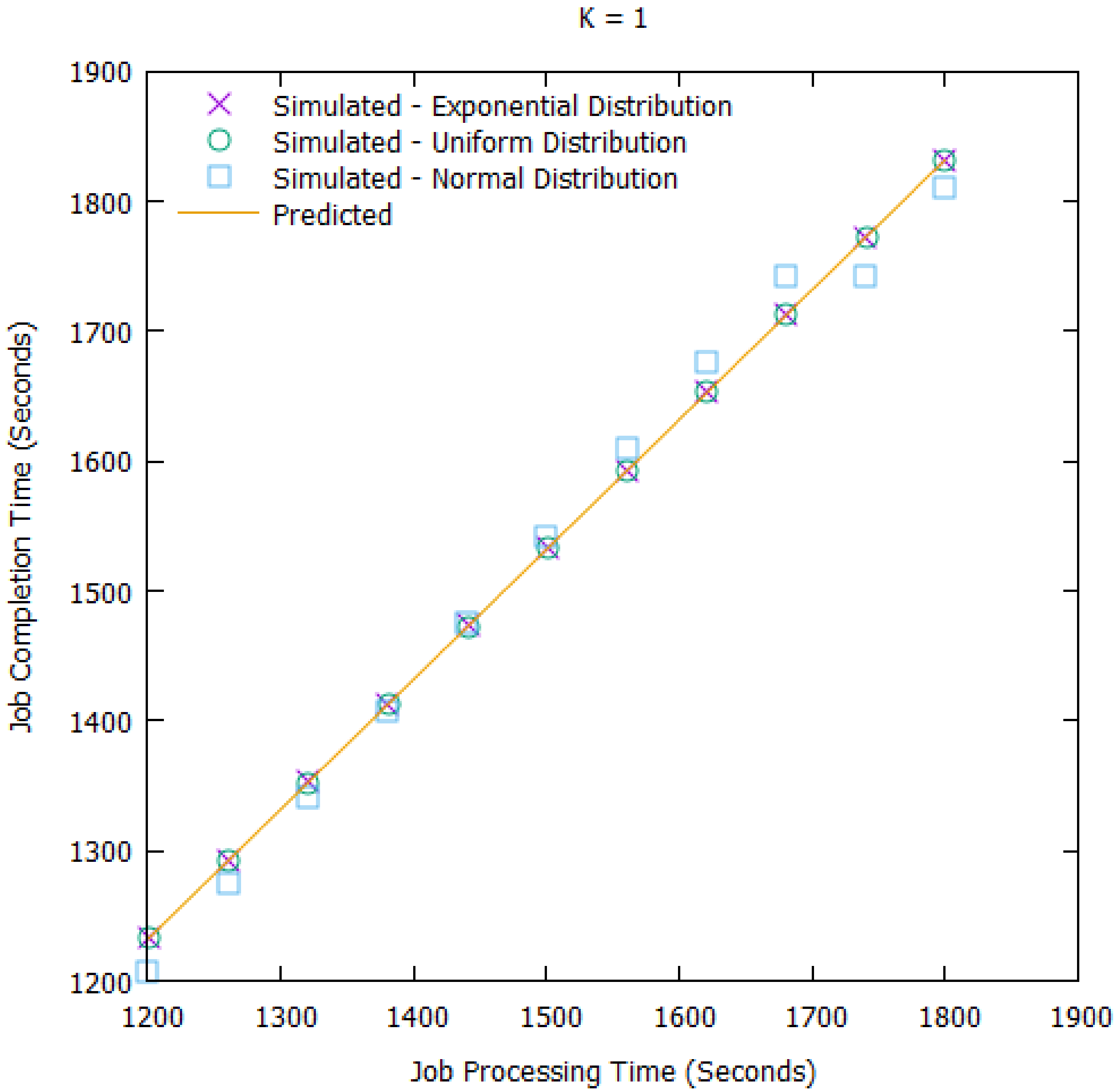


Fig. 22: Predicted and simulated job completion times for exponentially distributed, uniformly distributed and normally distributed job processing times given that one other vehicle is competing to receive a communication slot.

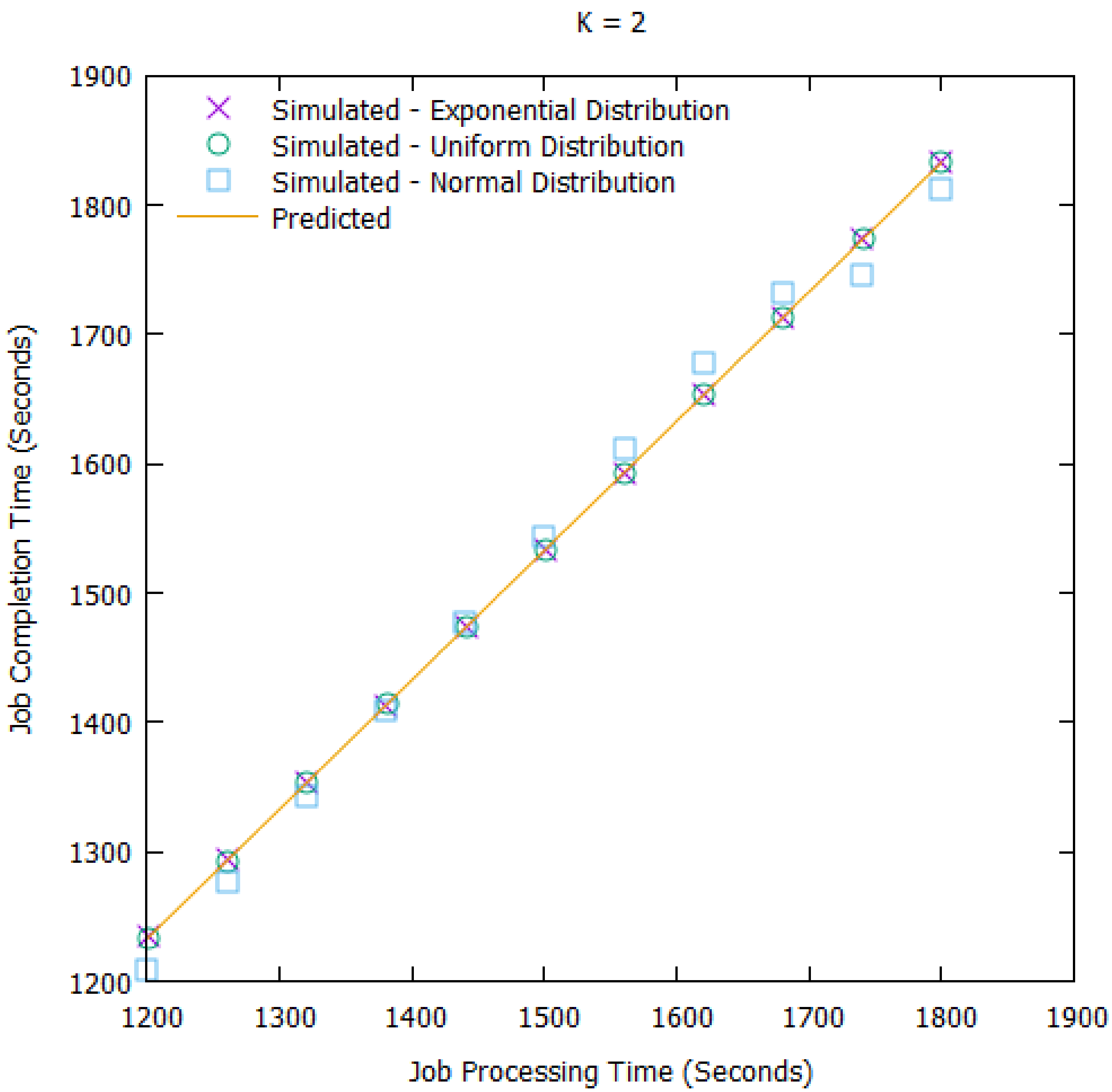


Fig. 23: Predicted and simulated job completion times for exponentially distributed, uniformly distributed and normally distributed job processing times given that two other vehicles are competing to receive a communication slot.

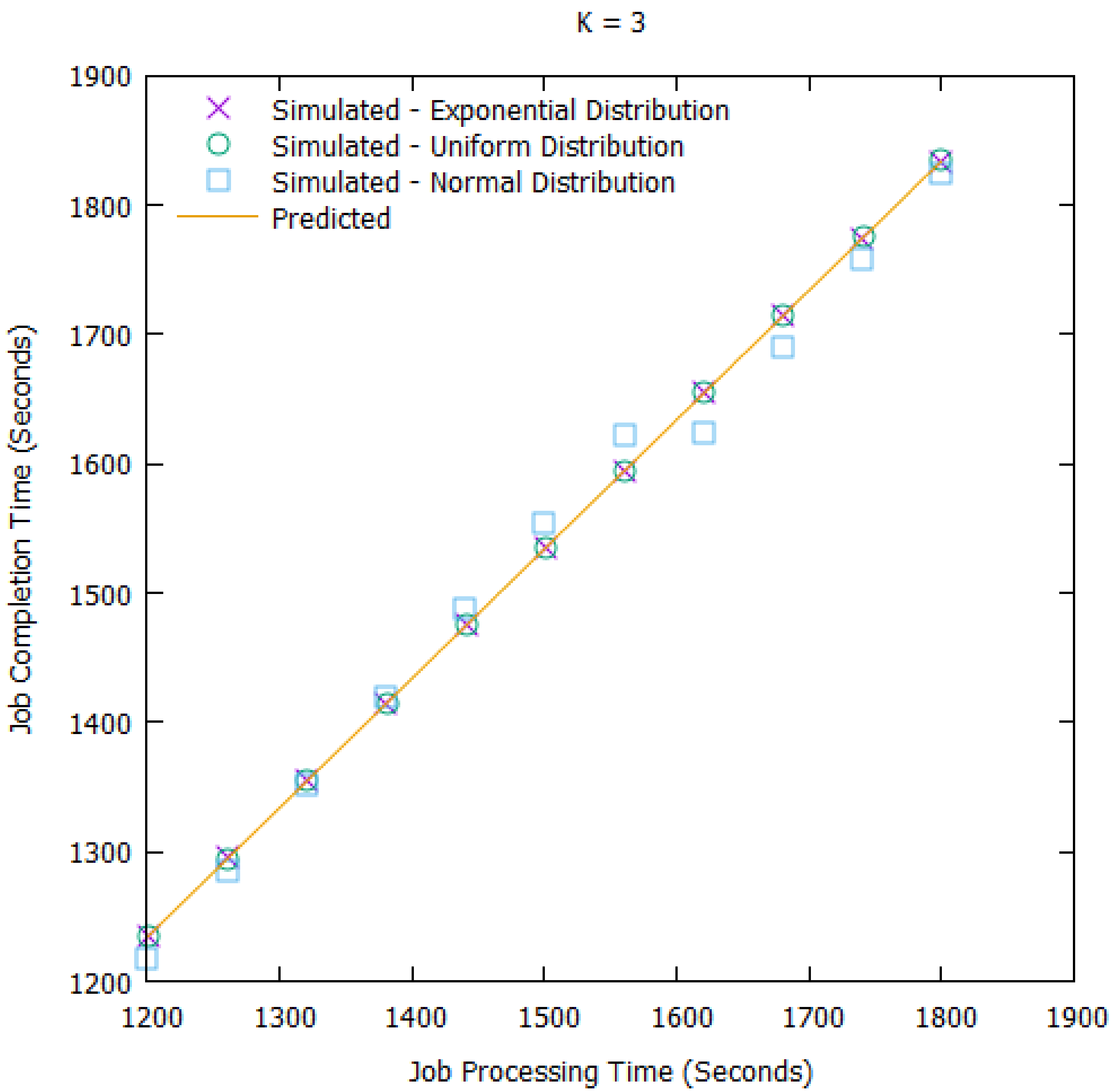


Fig. 24: Predicted and simulated job completion times for exponentially distributed, uniformly distributed and normally distributed job processing times given that three other vehicles are competing to receive a communication slot.

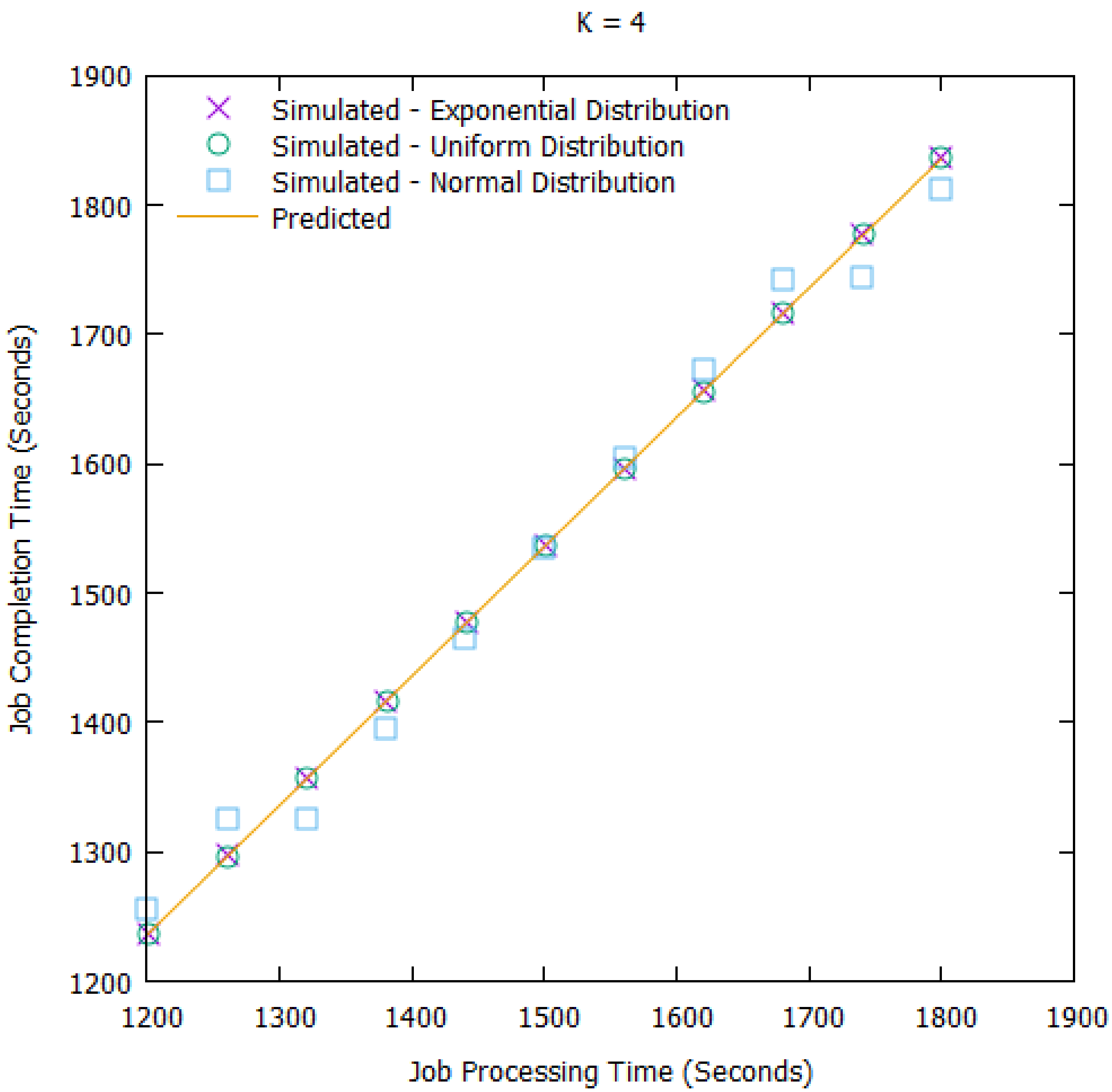


Fig. 25: Predicted and simulated job completion times for exponentially distributed, uniformly distributed and normally distributed job processing times given that four other vehicles are competing to receive a communication slot.

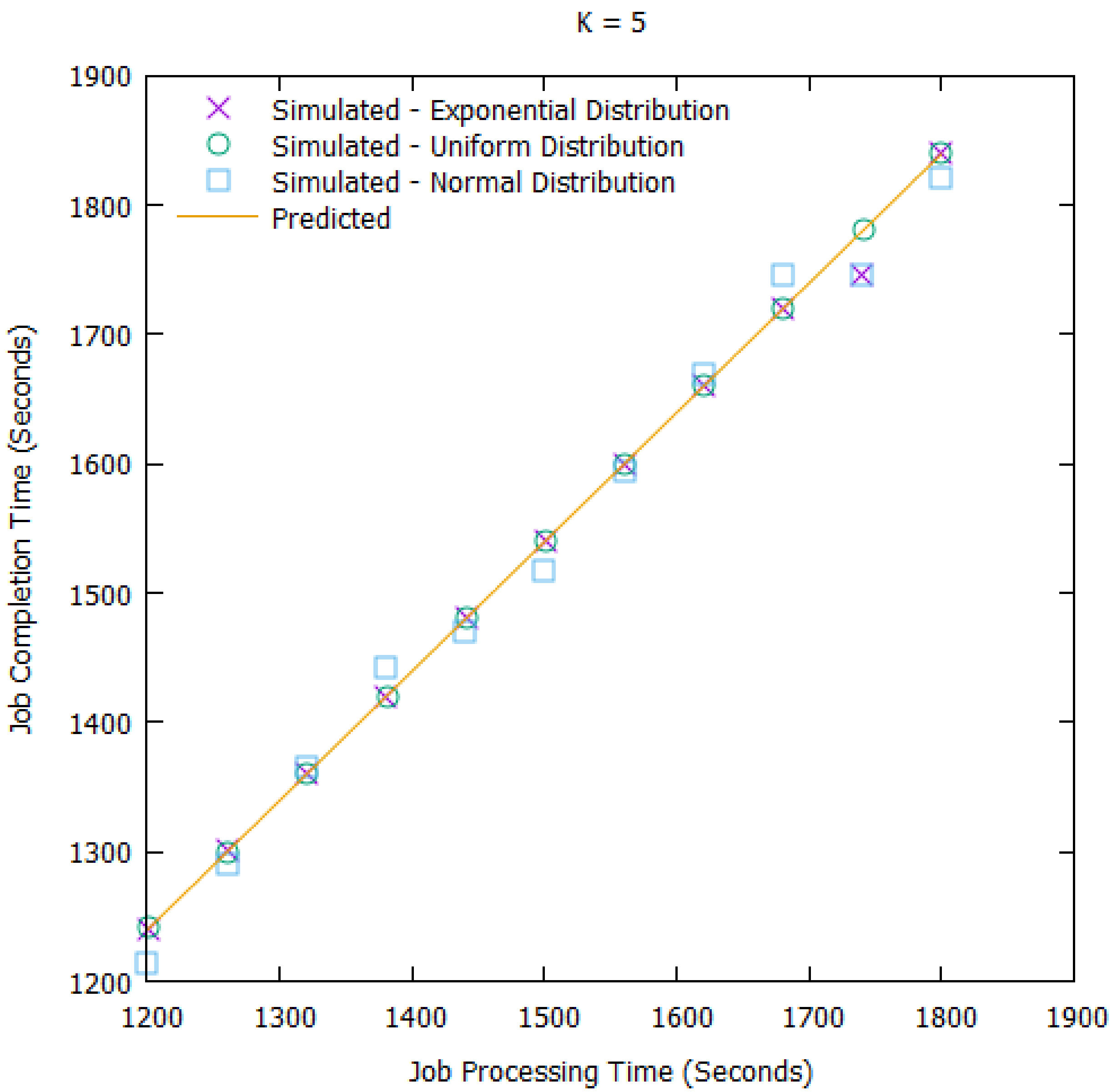


Fig. 26: Predicted and simulated job completion times for exponentially distributed, uniformly distributed and normally distributed job processing times given that five other vehicles are competing to receive a communication slot.

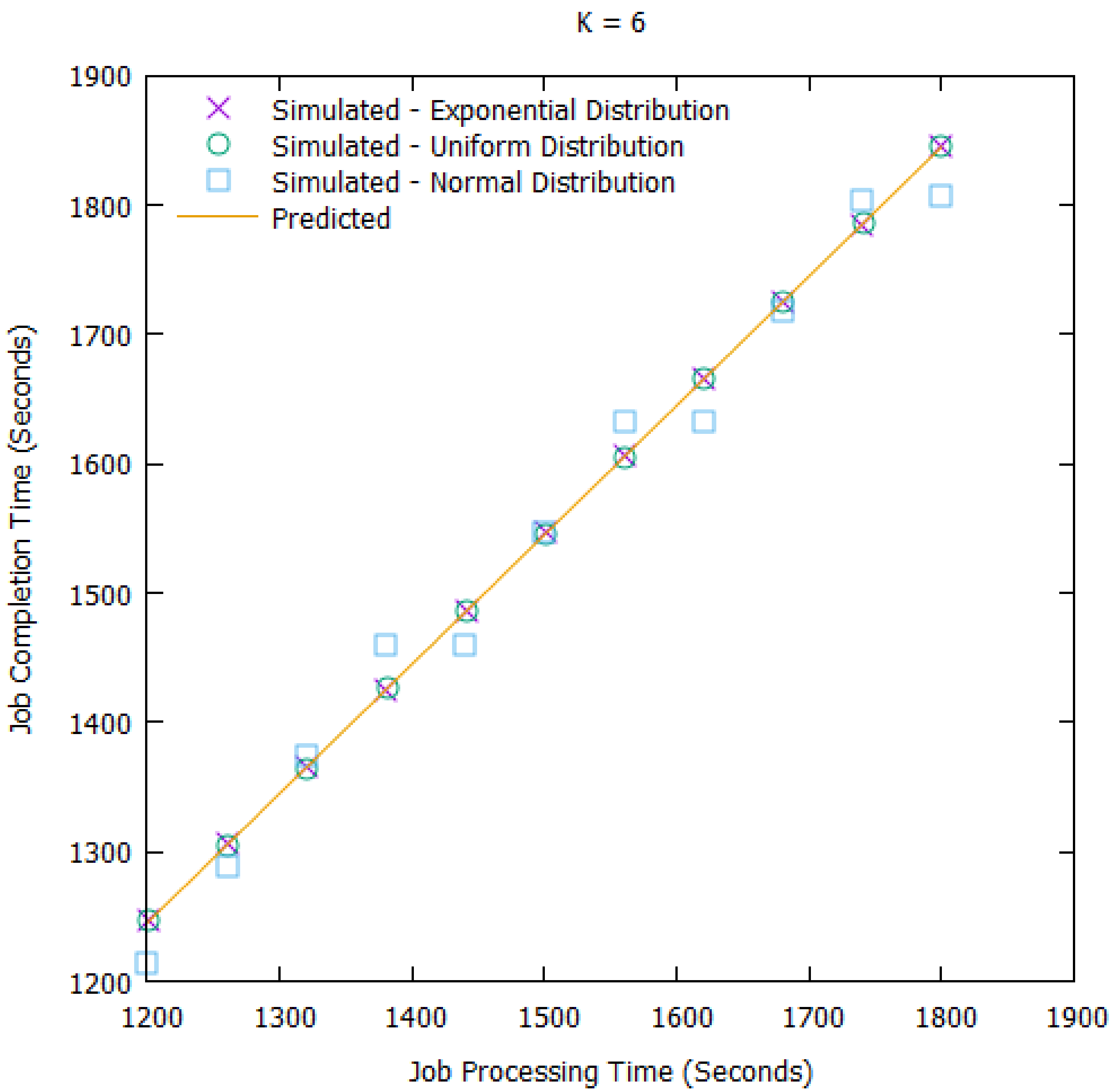


Fig. 27: Predicted and simulated job completion times for exponentially distributed, uniformly distributed and normally distributed job processing times given that six other vehicles are competing to receive a communication slot.

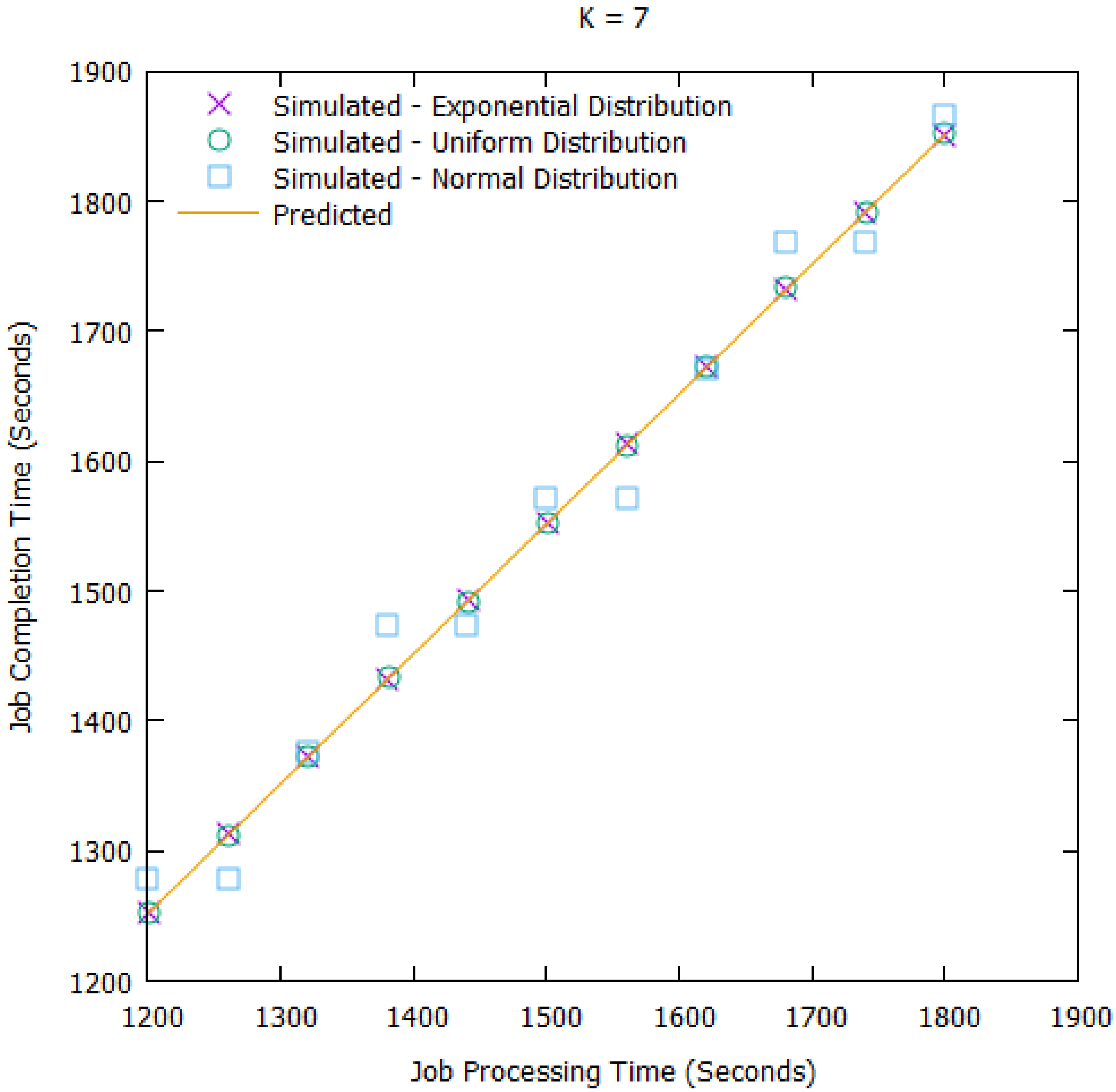


Fig. 28: Predicted and simulated job completion times for exponentially distributed, uniformly distributed and normally distributed job processing times given that seven other vehicles are competing to receive a communication slot.

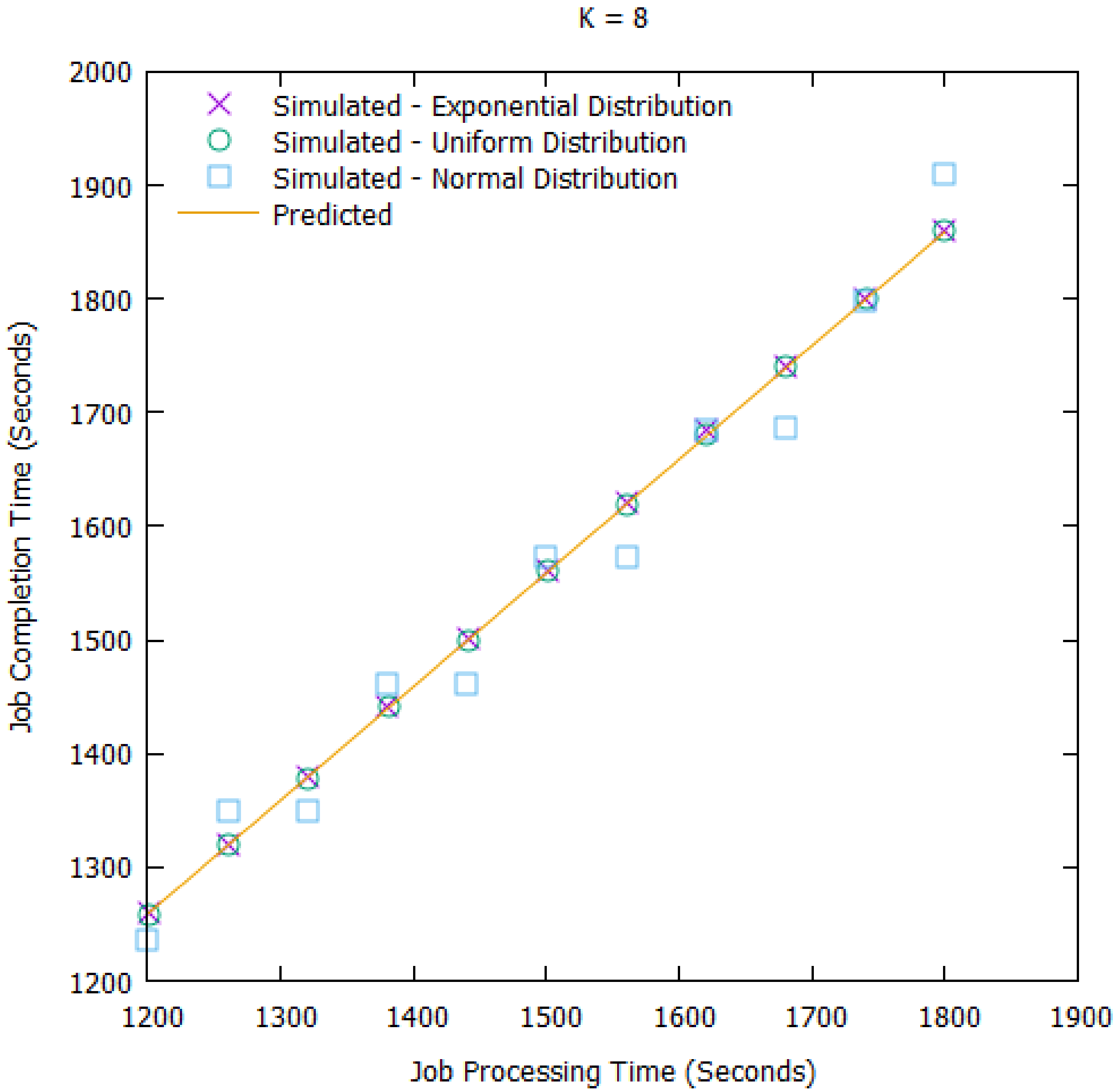


Fig. 29: Predicted and simulated job completion times for exponentially distributed, uniformly distributed and normally distributed job processing times given that eight other vehicles are competing to receive a communication slot.

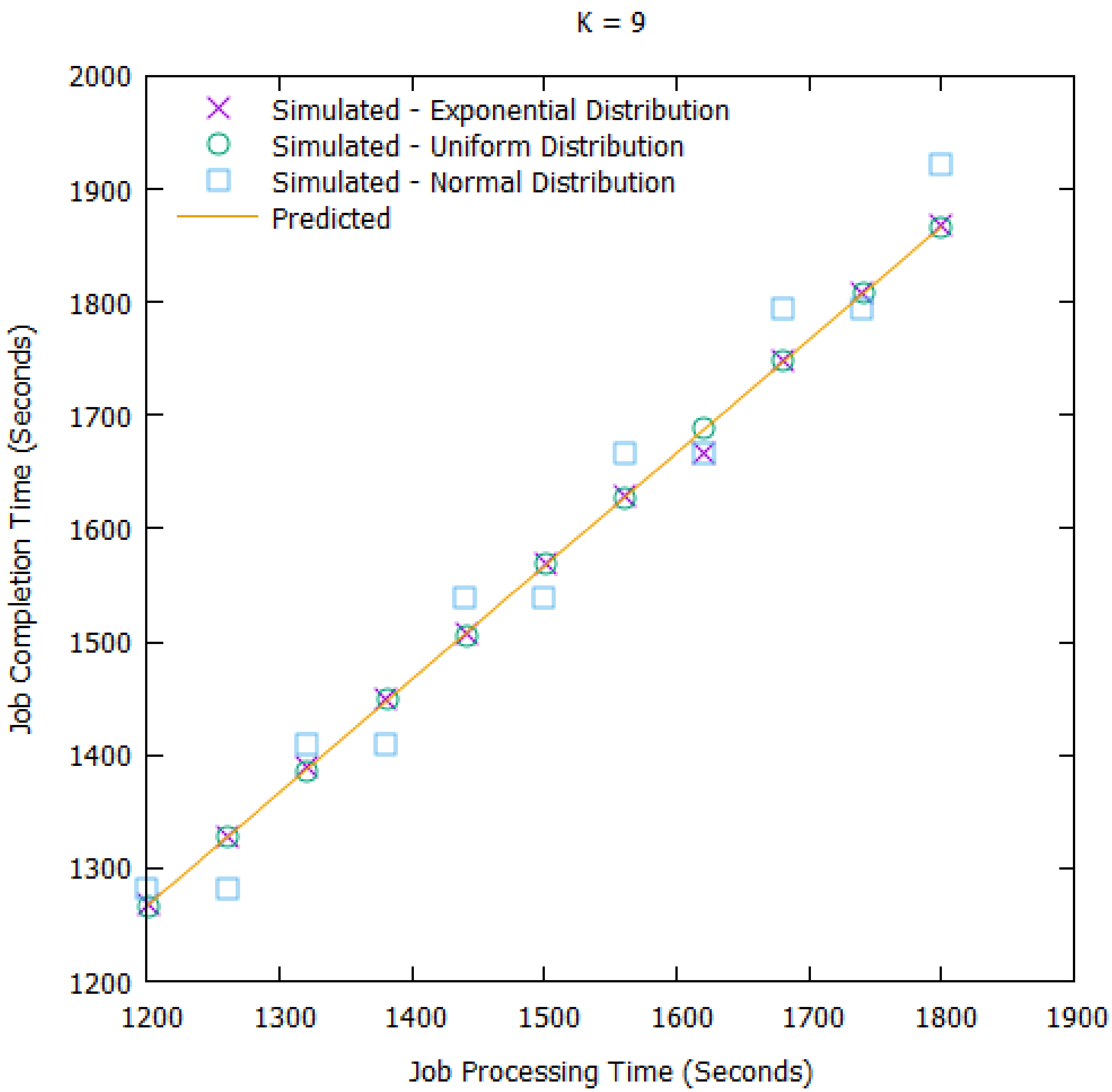


Fig. 30: Predicted and simulated job completion times for exponentially distributed, uniformly distributed and normally distributed job processing times given that nine other vehicles are competing to receive a communication slot.

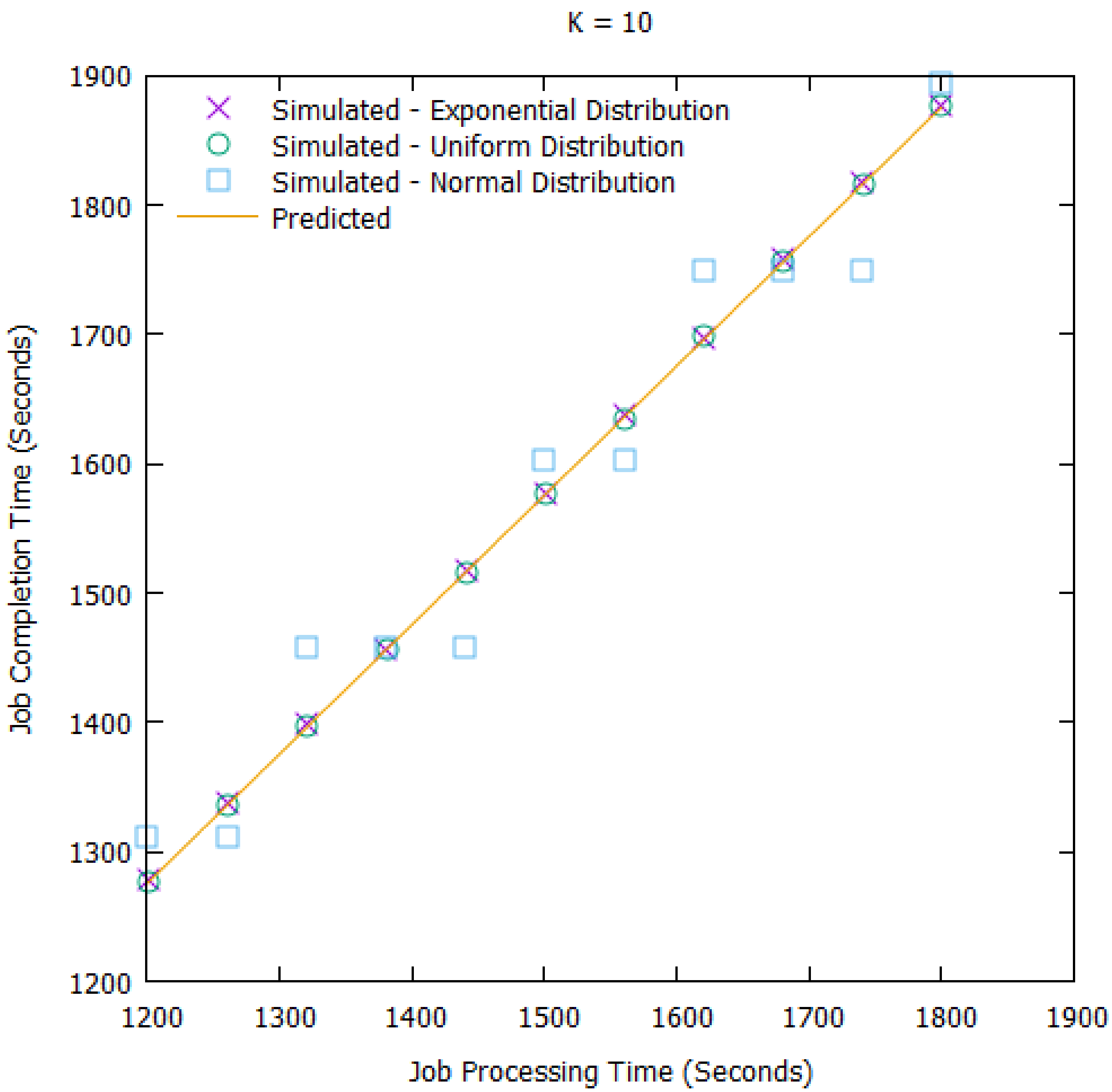


Fig. 31: Predicted and simulated job completion times for exponentially distributed, uniformly distributed and normally distributed job processing times given that ten other vehicles are competing to receive a communication slot.

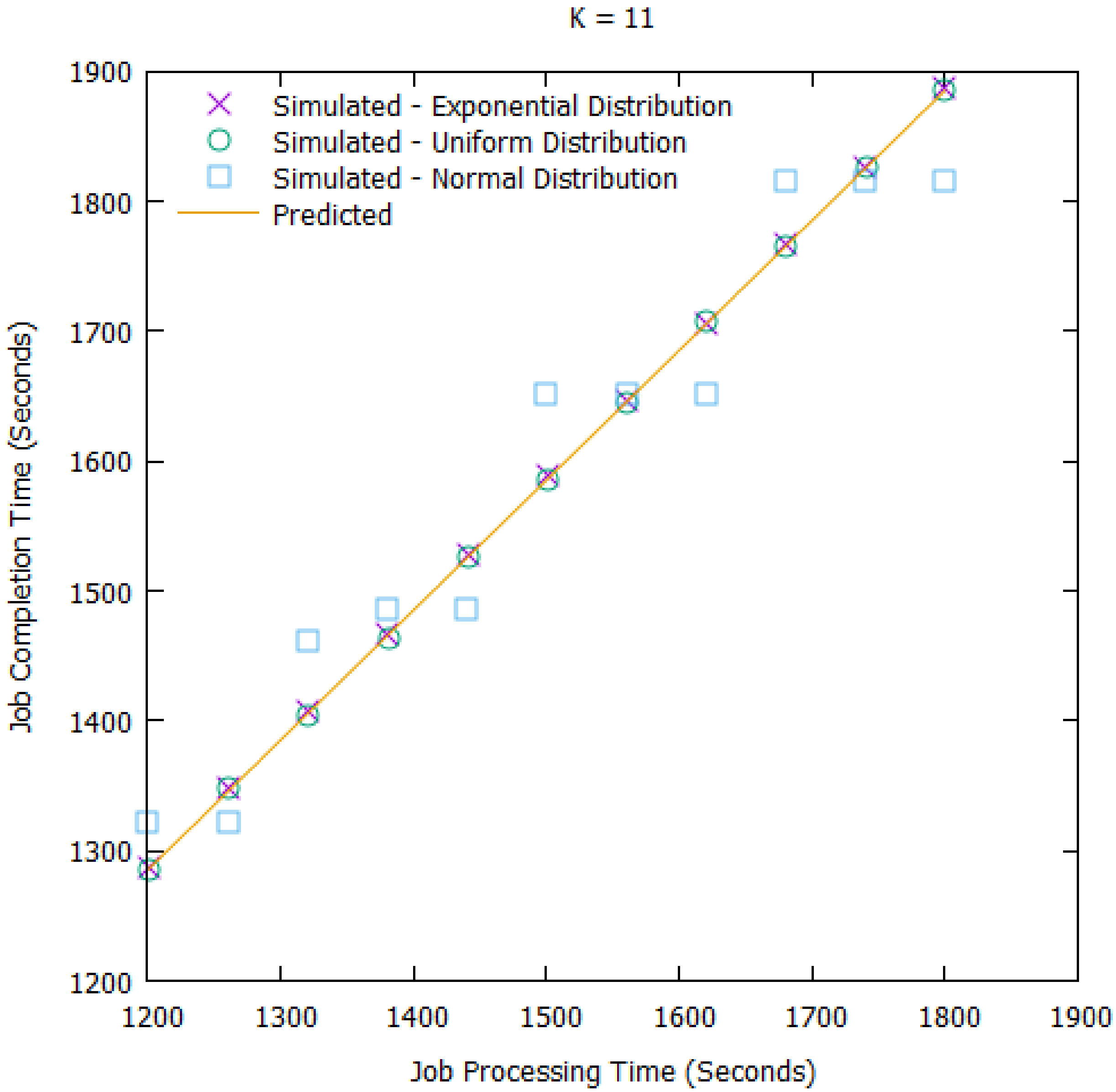


Fig. 32: Predicted and simulated job completion times for exponentially distributed, uniformly distributed and normally distributed job processing times given that 11 other vehicles are competing to receive a communication slot.

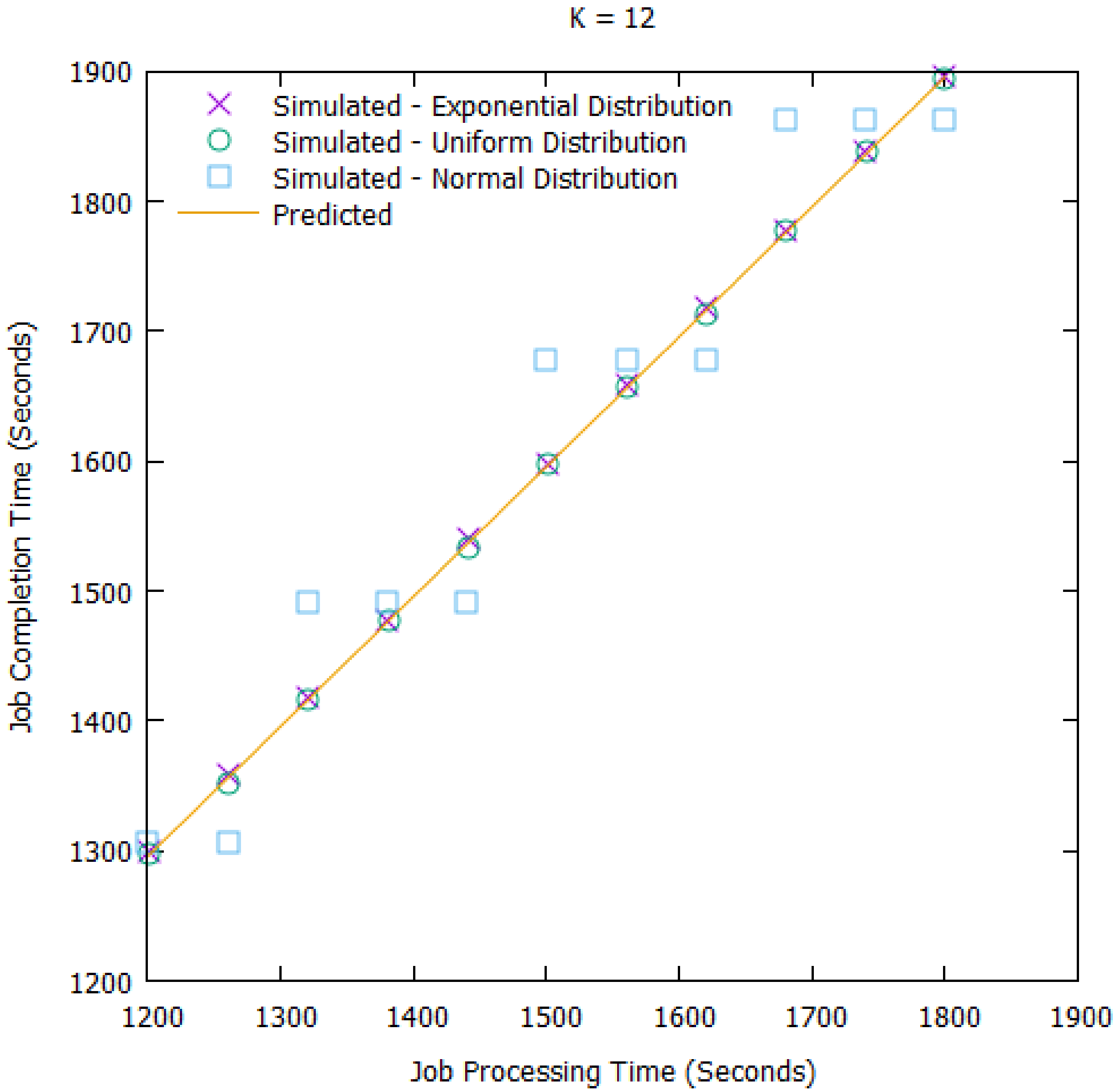


Fig. 33: Predicted and simulated job completion times for exponentially distributed, uniformly distributed and normally distributed job processing times given that 12 other vehicles are competing to receive a communication slot.

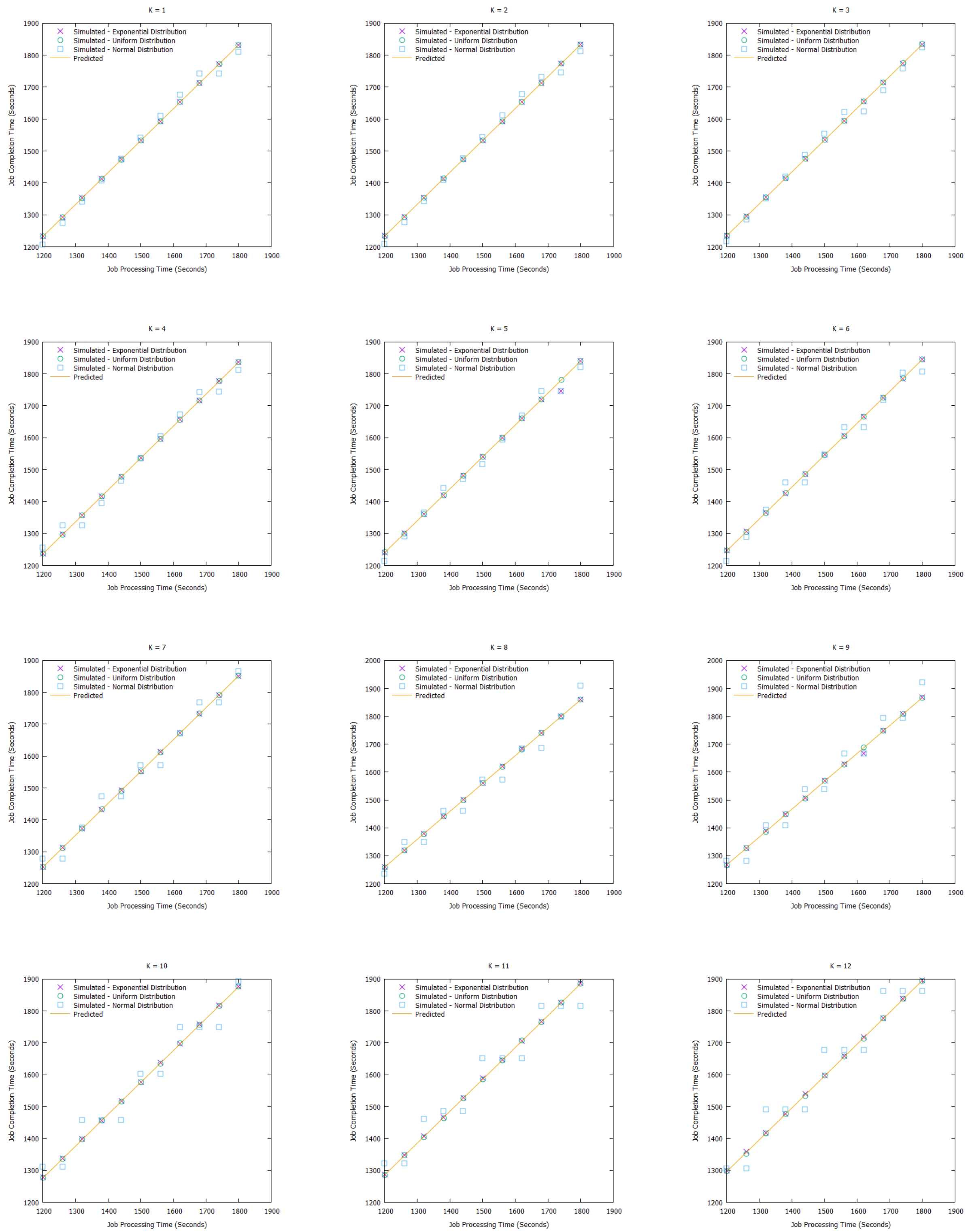


Fig. 34: *A side-by-side view of the predicted and simulated job completion times for exponentially distributed, uniformly distributed, and normally distributed job processing times given that k other vehicles are competing to receive a communication slot for k values of 1 to 12.*

TABLE 6: Simulation parameters

| **Symbol and Description** | **Value** |
|---|---|
| $l$ (number of lanes) | 3 |
| $B$ (available bandwidth) | $27 \times 10^6$ bps |
| $W$ (size of the job) | $8 \times 10^6$ bits |
| $b$ (payload in one frame) | 53792 bits |
| $F$ (frame length in bits) | 56624 bits |
| $F_s$ (frame length in seconds) | 0.002 s |
| $M$ (number of available slots for competing) | 20 |
| $c$ (access point coverage range) | 100 m |
| $d$ (distance between two consecutive APs) | 2000 m |
| $a$ (parameter used for intervals of job processing time) | 600 s |
| $v_{k+1}$ (vehicle's speed when k other vehicles are in the area) | (10, 30) m/s |
| $v_b$ (average travel speed at stop and go condition) | 9 kph $\approx$ 2.5 m/s |
| $v_f$ (free flow speed) | 107.44 kph $\approx$ 29.8 m/s |
| $k_t$ (turning point for the speed-density curve) | 17.53 |
| $\theta_1$ (scale parameter for speed-density function) | 1.8768 |
| $\theta_2$ (parameter which controls the lopsidedness of the curve) | 0.0871 |
| standard deviation for normally distributed job processing times | 1 s |

# CHAPTER 5

# JOB DOWNLOAD UNDER SEVERAL ACCESS POINTS

In this chapter, we explore the case in which the download of a job may be completed under several APs. When the download of a job takes more than one AP to complete, it means that the vehicle is not able to download the entire job under one AP and it needs to travel to the next AP to download the remainder of the job and continue the downloading from subsequent APs, until the job is completely downloaded or until the vehicle stays on the road, whichever occurs first. This scenario can happen for multiple reasons, such as having jobs that are larger in size or having limited available bandwidth to download because of the characteristics of the network or when a large number of vehicles utilize the available bandwidth simultaneously.

## 5.1 WORKLOADS OF INTEREST

The workloads contemplated for the VC in which the download of a job can be completed under several APs have a number of defining features in common:

- they are all delay-tolerant, not constrained by real-time requirements;
- the input size is relatively medium to large, say, more than 10 MBs;
- the processing time is relatively short;
- they are, as a rule, CPU-bound and do not require inter-processor communication.

In addition to the workloads presented in Chapter 4, such as text translation, scientific simulation, strategic plannings and crowdsourcing, other examples of such workloads include:

- Data mining; The recent advancements in technology have led to the generation of massive amounts of data daily, by technology such as the Internet, or the Internet of things (IoT), and with the popularity of mobile devices, more people are creating large amounts of data from different sources such as social media, or mobile devices with multiple sensors such as accelerometer, proximity sensor, GPS, light sensors, microphone, pedometer, heart rate sensor. Data mining can be used for advertisement and marketing purposes. For example, companies can predict customer behavior and target the advertisement towards potential customers using modeling techniques, profiling, and prediction models [106]. VCs can provide the computing resources required for lightweight data mining operations.

- Image processing: With the help of various computer algorithms and mathematical operations, images are processed to identify and extract valuable information and knowledge. The input can consist of an image or a series of images, and the output can consist of images or a set of parameters and attributes. Image processing can be used for classification, feature extraction, pattern and object recognition, target tracking, and template matching. VCs can provide the computing resources required for image processing, depending on the size of the input image and the estimated size of the output. Such processing tasks should not require a real-time response. An example of this is using image processing in agriculture. Images of the corp are captured at different times and processed to find any changes in the plant's growth. These findings can then help the farmers to find solutions to reduce crop failure and improve the quality of the produce [107].

## 5.2 NOTATION AND TERMINOLOGY

Since some of the notation that is used in this chapter is similar to the notation presented in the previous chapter, we refrain from repeating such notation and refer the reader to Chapter 4.

The characteristics of the workloads above motivate us to make the following assumptions about the VC.

- The size of the downloaded job is (roughly) equal to the size of the uploaded result;
- The job and the results of the job processing can be downloaded under the coverage area of several consecutive APs;
- The results of the job processing can be uploaded under the coverage area of several consecutive APs;
- A job can be completed by one vehicle and there is no need for job migration. meaning that the residency time of a vehicle is larger than job completion time.

## 5.3 APPROXIMATING JOB COMPLETION TIME

Let $G_D$ be the random variable that keeps track of the number of APs that the full coverage is used to download the job.

As previously discussed in Chapter 4, $D_{k+1}$ represents the *total number of frames* in which the vehicle has to compete, in order to complete the download of a job, given that in each frame $k$ other vehicles are also competing for a slot and the random variable $N_{k+1}$ keeps track of the number of (complete) frames a vehicle sees while in the coverage area of an arbitrary AP.

It is clear that:

$$G_D = \lfloor \frac{D_{k+1}}{N_{k+1}} \rfloor$$

For all $i$, $(i \geq 0)$, by the properties of the floor function, the probability of $G_D$ is obtained as:

$$\Pr[G_D = i] = \Pr[\lfloor \frac{D_{k+1}}{N_{k+1}} \rfloor = i] = \Pr[i \leq \frac{D_{k+1}}{N_{k+1}} < i+1] = \Pr[N_{k+1}i \leq D_{k+1} < N_{k+1}(i+1)]$$

Since $D_{k+1}$ is an integer, we then obtain:

$$\Pr[G_D = i] = \Pr[\lceil N_{k+1}i \rceil \leq D_{k+1} < \lfloor N_{k+1}(i+1) \rfloor] = \sum_{j=\lceil N_{k+1}i \rceil}^{\lfloor N_{k+1}(i+1) \rfloor} \Pr[D_{k+1} = j] \tag{27}$$

Applying the expectation operator to (27), $E[G_D]$ is then obtained as:

$$E[G_D] = \sum_{i\geq 0}^{\infty} i \sum_{j=\lceil N_{k+1}i \rceil}^{\lfloor N_{k+1}(i+1) \rfloor} \Pr[D_{k+1} = j] \tag{28}$$

As explained in Chapter 4, $D_{k+1}$ has a negative binomial distribution with parameters $r_{k+1}$ and $p_{k+1}$. It follows that for an arbitrary non-negative integer $j$,

$$\Pr[D_{k+1} = j] = \binom{j-1}{r_{k+1}-1} p_{k+1}^{r_{k+1}} (1 - p_{k+1})^{j - r_{k+1}} \tag{29}$$

Using (29) in (28), we then obtain:

$$E[G_D] = \sum_{i\geq 0}^{\infty} i \sum_{j=\lceil N_{k+1}i \rceil}^{\lfloor N_{k+1}(i+1) \rfloor} \binom{j-1}{r_{k+1}-1} p_{k+1}^{r_{k+1}} (1 - p_{k+1})^{j - r_{k+1}} \tag{30}$$

(30) shows the expectation of the number of APs that the full coverage is used to download the job.

Let $G_U$ be the random variable that keeps track of the number of APs that the full coverage is used to upload the job. Since $D_{k+1}$ and $U_{k+1}$ have the same distribution, it is clear that $E[G_U]$ can be obtained in a similar way to $E[G_D]$. Therefore we have:

$$E[G_U] = \sum_{i\geq 0}^{\infty} i \sum_{j=\lceil N_{k+1}i \rceil}^{\lfloor N_{k+1}(i+1) \rfloor} \binom{j-1}{r_{k+1}-1} p_{k+1}^{r_{k+1}} (1 - p_{k+1})^{j - r_{k+1}}$$

As discussed in Chapter 4, $T$ represents the random variable that keeps track of the execution time of the user job in the absence of any overhead attributable to the VC. We define $G_T$ as the random variable that keeps track of the number APs that the full coverage is traveled by the vehicle to execute the job. We obtain:

$$G_T = \lfloor \frac{T}{\frac{d}{v_{k+1}}} \rfloor \tag{31}$$

Applying the expectation operator to (31), $E[G_T]$ is then obtained as:

$$E[G_T] = \lfloor \frac{E[T]}{\frac{d}{v_{k+1}}} \rfloor$$

We define $\delta_D$ and $\delta_U$ respectively as the remaining of the frames needed in addition to $G_D$ and $G_U$ to download and upload the job and and $\delta_T$ is the remaining of the time (in seconds) needed to complete the job processing in addition to time spent executing the job during the $G_T \cdot \frac{d}{v_{k+1}}$ period. It is cleat that:

$$\delta_D = D_{k+1} - G_D \cdot N_{k+1} = D_{k+1} - \lfloor \frac{D_{k+1}}{N_{k+1}} \rfloor N_{k+1} \tag{32}$$

$$\delta_U = U_{k+1} - G_U \cdot N_{k+1} = U_{k+1} - \lfloor \frac{U_{k+1}}{N_{k+1}} \rfloor N_{k+1} \tag{33}$$

$$\delta_T = T - G_T \cdot \frac{d}{v_{k+1}} = T - \lfloor \frac{T}{\frac{d}{v_{k+1}}} \rfloor \frac{d}{v_{k+1}} \tag{34}$$

Applying the expectation operator to (32), (33), and (34), we then obtained:

$$E[\delta_D] = E[D_{k+1}] - E[G_D] N_{k+1}$$

$$E[\delta_U] = E[U_{k+1}] - E[G_U] N_{k+1}$$

$$E[\delta_T] = E[T] - \lfloor \frac{E[T]}{\frac{d}{v_{k+1}}} \rfloor \frac{d}{v_{k+1}}$$

To find the job completion time, we distinguish between the two cases below:

**Case 1:** *Job execution terminates under the coverage of some AP.*

In In this case, $\frac{F}{B}\delta_D + \delta_T \leq \frac{c}{v_{k+1}}$ or $\frac{F}{B}\delta_D + \delta_T \geq \frac{d}{v_{k+1}}$ and upload starts immediately under the same coverage area that the job execution terminated and therefore there is no delay.

The expected download time is obtained as the time it takes to travel to the number of APs that the full coverage is used to download the job plus the time it takes to download the remainder of the frames needed to download the job ($\frac{F}{B}E[D_{k+1}] + \frac{(d-c)}{v_{k+1}}E[G_D]$). We obtain the expected job completion time as

$$E[J_1] = E[T] + \frac{F}{B}E[D_{k+1}] + \frac{(d-c)}{v_{k+1}}E[G_D] + \frac{F}{B}E[U_{k+1}] + \frac{(d-c)}{v_{k+1}}E[G_U].$$

By algebraic manipulations we then obtain

$$E[J_1] = E[T] + \frac{F}{B}(E[D_{k+1}] + E[U_{k+1}]) + \frac{(d-c)}{v_{k+1}}(E[G_D] + E[G_U]).$$

By recalling (12) we then obtain

$$E[J_1] = E[T] + \frac{2F}{B}E[D_{k+1}] + \frac{2(d-c)}{v_{k+1}}E[G_D].$$

**Case 2:** *Job execution terminates between the coverage areas of two adjacent APs.* In this case, $\frac{c}{v_{k+1}} < \frac{F}{B}\delta_D + \delta_T < \frac{d}{v_{k+1}}$, and upload starts after some delay at the next available AP.

The job execution terminates after leaving the coverage area of $\mathrm{AP_{N\text{-}1}}$ but before entering the coverage area of $\mathrm{AP_N}$. The vehicle starts uploading the results upon entering the coverage area of $\mathrm{AP_N}$. In this case, it is natural to define the job completion time $J_2$ as the summation of the time it takes to travel to $\mathrm{AP_N}$ and the time it takes to upload the job:

$$J_2 = \frac{dN}{v_{k+1}} + \frac{d}{v_{k+1}}G_U + \frac{F}{B}\delta_U. \tag{35}$$

To justify (35) notice that the vehicle must first physically reach the coverage area of $\mathrm{AP_N}$ before it can start uploading the results. The former time is $\frac{dN}{v_{k+1}}$, while the latter is $\frac{d}{v_{k+1}}G_U + \frac{F}{B}\delta_U$. Here, $N$ is the *unique* natural number for which job download and execution terminates strictly between the coverage areas of $\mathrm{AP_{N\text{-}1}}$ and that of $\mathrm{AP_N}$. In other words, $N$ satisfies

$$\frac{(N-1)d + c}{v_{k+1}} < \frac{d}{v_{k+1}}G_D + \frac{F}{B}\delta_D + T \leq \frac{Nd}{v_{k+1}}. \tag{36}$$

By applying the expectation operator to (35), we obtain

$$E[J_2] = \frac{dE[N]}{v_{k+1}} + \frac{d}{v_{k+1}}E[G_U] + \frac{F}{B}E[\delta_U].$$

In order to obtain an expression for $E[J_2]$ we proceed as follows: From (36) by simple algebra we obtain, in stages,

$$\frac{d}{v_{k+1}}G_D + \frac{F}{B}\delta_D + T < \frac{dN}{v_{k+1}} \leq \frac{d}{v_{k+1}}G_D + \frac{F}{B}\delta_D + T + \frac{d-c}{v_{k+1}}. \tag{37}$$

Applying the expectation operator to (37) yields

$$\frac{d}{v_{k+1}}E[G_D] + \frac{F}{B}E[\delta_D] + E[T] < \frac{dE[N]}{v_{k+1}} \leq \frac{d}{v_{k+1}}E[G_D] + \frac{F}{B}E[\delta_D] + E[T] + \frac{d-c}{v_{k+1}}. \tag{38}$$

After adding $\frac{d}{v_{k+1}}E[G_U] + \frac{F}{B}E[\delta_U]$ throughout in (38) and recalling (35), we write

$$\begin{aligned} &\tfrac{d}{v_{k+1}}E[G_U] + \tfrac{F}{B}E[\delta_U] + \tfrac{d}{v_{k+1}}E[G_D] + \tfrac{F}{B}E[\delta_D] + E[T] < E[J_2] \\ &\leq \frac{d}{v_{k+1}}E[G_U] + \frac{F}{B}E[\delta_U] + \frac{d}{v_{k+1}}E[G_D] + \frac{F}{B}E[\delta_D] + E[T] + \frac{d-c}{v_{k+1}}. \end{aligned} \tag{39}$$

By (39) and recalling (12), we write

$$\frac{2d}{v_{k+1}}E[G_D] + \frac{2F}{B}E[\delta_D] + E[T] < E[J_2] \leq \frac{2d}{v_{k+1}}E[G_D] + \frac{2F}{B}E[\delta_D] + E[T] + \frac{d-c}{v_{k+1}},$$

which yields the following approximation for $E[J_2]$ that turns out to be quite accurate and the accuracy of which will be assessed in detail.

$$E[J_2] \approx E[T] + \frac{2d}{v_{k+1}}E[G_D] + \frac{2F}{B}E[\delta_D] + \frac{d-c}{2v_{k+1}}. \tag{40}$$

### 5.3.1 COMPLETING THE APPROXIMATION

The goal of this subsection is to combine the two cases discussed above into a coherent approximation of the job completion time.

Let $\pi_1$ and $\pi_2$ be, respectively, the *limiting* probabilities of Case 1 and Case 2 occurring. Using the Law of Total Expectation Theorem (Theorem 4), the expectation $E[J]$ of job completion time can be computed as

$$E[J] = \pi_1 E[J_1] + \pi_2 E[J_2]. \tag{41}$$

To evaluate the limiting probabilities $\pi_1, \pi_2$, consider the time interval $I$ of length $\frac{d}{v_{k+1}}$.

$$I = \left[\frac{d(N-1)+c}{v_{k+1}}, \frac{dN+c}{v_{k+1}}\right]$$

Since the probability distribution of $T$ is not known, to a first approximation, we assume that job execution terminates, uniformly at random in the time interval $I$. In turn, this assumption implies that $\pi_1$ and $\pi_2$ are given by the expressions

$$\pi_1 = \frac{c}{d}; \pi_2 = \frac{d-c}{d}. \tag{42}$$

Upon replacing the expressions of $\pi_1$ and $\pi_2$ obtained in (42) back into (41), and recalling (35) and (40) we obtain

$$E[J] = \frac{c}{d}E[J_1] + \frac{d-c}{d}E[J_2].$$

## 5.4 EVALUATION OF JOB COMPLETION TIME

This section presents our simulation results performed in order to validate the analytical results obtained in Chapter 5.3. The simulation model that we use to validate the theoretical results is similar to the model that we described in Chapter 4.5. Table 7 shows the simulation parameters. We have performed several sets of experiments that we now describe.

In the first set of experiments, the model has been set up in such a way that job processing times are exponentially distributed with a mean of $\frac{1}{\lambda}$, where $\lambda$ is 1200 seconds. The simulation and prediction results of job completion times for k values of 1 to 12 are shown in Table 8. The maximum relative error is less than 0.26%, with an average of 0.15% for the exponential distribution.

In the second set of experiments we considered jobs with processing times that are uniformly distributed on the interval from $\lambda - a$ to $\lambda + a$, where $a$ is 5 seconds and $\lambda$ is 1200 seconds. Table 9 shows the simulation and prediction results for k values of 1 to 12. The maximum relative error is less than 0.31%, with an average of 0.16% for the described uniform distribution.

In the third set of experiments we considered jobs with processing times that are uniformly distributed on the interval from $\lambda - a$ to $\lambda + a$, where $a$ is 600 seconds and $\lambda$ is 1200 seconds. Table 10 shows the simulation and prediction results for k values of 1 to 12. The maximum relative error is less than 0.26%, with an average of 0.17% for the described uniform distribution.

Table 11 shows the simulation results for $G_D$, the number of APs that the full coverage is used to download the job and also the predicted values using (30). The maximum relative error is less than 0.03%, with an average of 0.002%.

The comparison of simulation results and analytical results confirms the accuracy of the theoretical predictions.

TABLE 7: Simulation parameters

| **Symbol and Description** | **Value** |
|---|---|
| $l$ (number of lanes) | 3 |
| $B$ (available bandwidth) | $27 \times 10^6$ bps |
| $W$ (size of the job) | $8 \times 10^7$ bits |
| $b$ (payload in one frame) | 53792 bits |
| $F$ (frame length in bits) | 56624 bits |
| $F_s$ (frame length in seconds) | 0.002 s |
| $M$ (number of available slots for competing) | 20 |
| $c$ (access point coverage range) | 100 m |
| $d$ (distance between two consecutive APs) | 2000 m |
| $v_{k+1}$ (vehicle's speed when k other vehicles are in the area) | (10, 30) m/s |
| average job processing time | 1200 s |
| number of runs | $10^4$ |

## 5.5 JOB MIGRATION

In conventional data centers, cloud providers use VM migration techniques for many benefits that they provide, such as zero-downtime hardware maintenance, load balancing, server consolidation, improving data and network locality, reducing hosting, and energy costs, and facilitating maintenance [108] [109].

In dynamic VCs (unlike the static nature of the computational resources in conventional data centers), computational resources are dynamic. The vehicles, which are the VC's computational resources, arrive and depart at random times. The residency time of the vehicles or, in other words, the time that each computational resource can be utilized in the VC is random, limited and not infinite, and as a result, this characteristic may cause interruptions in the ongoing services.

TABLE 8: Simulated and predicted job completion times for jobs of size 10 MB for exponentially distributed job processing times with parameter $\frac{1}{\lambda} = 1200s$ (20 minutes).

| k | Simulated | Predicted | %Error |
|---|---|---|---|
| 1 | 1372.971177 | 1370.085158 | 0.210202446 |
| 2 | 1506.602746 | 1504.023341 | 0.17120671 |
| 3 | 1642.814903 | 1640.536782 | 0.1386718 |
| 4 | 1793.948394 | 1792.201082 | 0.097400349 |
| 5 | 1851.216304 | 1847.766624 | 0.186346673 |
| 6 | 2098.006361 | 2097.589206 | 0.0198834 |
| 7 | 2227.232571 | 2226.370874 | 0.038689134 |
| 8 | 2299.213903 | 2297.220068 | 0.086718117 |
| 9 | 2298.473231 | 2292.467841 | 0.261277352 |
| 10 | 2445.785379 | 2439.69165 | 0.249152238 |
| 11 | 2607.28175 | 2601.495095 | 0.221942067 |
| 12 | 2783.440482 | 2778.096422 | 0.191994765 |

In order to handle these interruptions, we propose job migration strategies that help reduce the overhead of the job completion time, minimize the total bandwidth usage, improve the Quality of service (QoS), and improve the performance and reliability of the system.

In the previous section, we explored the case in which the download of a job may be completed under several APs, and a job can be completed by one vehicle, and there is no need for job migration, meaning that the residency time of a vehicle is larger than job completion time.

In this section, we explore the case in which the download of a job may be completed under several APs, and a job can be completed by one or several vehicles, and there may be a need for job migration, meaning that the residency time of a vehicle is not necessarily larger than the job completion time.

We propose and investigate job migration strategies for such types of jobs in dynamic VCs.

### 5.5.1 JOB MIGRATION STRATEGY 1

TABLE 9: Simulated and predicted job completion times for jobs of size 10 MB with job processing times uniformly distributed on the interval from $\lambda - a$ to $\lambda + a$, where $a = 5s$ and $\lambda = 1200s$.

| k | Simulated | Predicted | %Error |
|---|---|---|---|
| 1 | 1373.45613 | 1370.085158 | 0.2454372 |
| 2 | 1507.096864 | 1504.023341 | 0.2039367 |
| 3 | 1643.080926 | 1640.536782 | 0.1548398 |
| 4 | 1794.929018 | 1792.201082 | 0.1519802 |
| 5 | 1851.925975 | 1847.766624 | 0.224596 |
| 6 | 2098.557275 | 2097.589206 | 0.0461302 |
| 7 | 2227.832187 | 2226.370874 | 0.0655935 |
| 8 | 2301.024405 | 2297.220068 | 0.1653323 |
| 9 | 2299.763016 | 2292.467841 | 0.3172142 |
| 10 | 2446.058988 | 2439.69165 | 0.2603101 |
| 11 | 2602.666214 | 2601.495095 | 0.0449969 |
| 12 | 2779.290042 | 2778.096422 | 0.0429469 |

When a vehicle enters the VC, it receives a job and immediately attempts to download the job. The vehicle competes with other vehicles in the coverage areas at several consecutive APs to complete the job download.

After successfully downloading the job, the vehicle starts the processing of the job. The processing time of the job may be larger than the residency time of the vehicle, or in other words, the vehicle might not have enough time to complete the job.

By having the estimated residency time of the vehicle and the estimated upload time of the job, the vehicle can identify at which AP number it should start uploading the intermediate results.

When the vehicle enters the designated AP coverage area, it starts the migration of the job and uploads the intermediate results.

In this migration strategy, other vehicles cannot start the download of the job until the intermediate results are entirely and successfully uploaded. This adds an overhead to the total job completion time of the job.

TABLE 10: Simulated and predicted job completion times for jobs of size 10 MB with job processing times uniformly distributed on the interval from $\lambda - a$ to $\lambda + a$, where $a = 600s$ and $\lambda = 1200s$.

| **k** | **Simulated** | **Predicted** | **%Error** |
|---|---|---|---|
| 1 | 1373.785119 | 1370.085158 | 0.269326019 |
| 2 | 1507.875563 | 1504.023341 | 0.255473457 |
| 3 | 1644.49119 | 1640.536782 | 0.240463914 |
| 4 | 1795.881202 | 1792.201082 | 0.204920036 |
| 5 | 1851.490241 | 1847.766624 | 0.201114574 |
| 6 | 2102.673828 | 2097.589206 | 0.241816972 |
| 7 | 2227.065675 | 2226.370874 | 0.031198049 |
| 8 | 2297.955536 | 2297.220068 | 0.032005332 |
| 9 | 2293.827335 | 2292.467841 | 0.059267513 |
| 10 | 2443.808426 | 2439.69165 | 0.168457409 |
| 11 | 2605.877605 | 2601.495095 | 0.168177899 |
| 12 | 2782.792566 | 2778.096422 | 0.168756522 |

Consider a scenario that vehicle A is assigned a job, and after downloading the job, the vehicle starts the processing of the job. Vehicle A does not remain in the VC long enough to complete the processing of the job and therefore needs to migrate the job before departing the VC. Vehicle A starts the upload of the intermediate results at the $AP_i$ (Figure 36). The upload of the job cannot finish under one AP; therefore, vehicle A uploads a chunk of the job at the $AP_i$ and continues to upload the rest of the job at the $AP_{i+1}$. While the vehicle A is uploading the remaining of the job at the $AP_{i+1}$, vehicle B enters the VC, but it is not assigned the job until vehicle A finishes the upload of the job. Once vehicle A completes the upload of the job, vehicle B starts the download of the job at the next available AP.

### Job Migration Failure

The migration of a job may be successful, but there is also a chance of failed migration. As the vehicle competes with other vehicles in the coverage area of the AP, to receive a communication slot, it may not be able to acquire the necessary bandwidth to complete the

TABLE 11: Simulated and predicted number of APs that the full coverage is used to download the job for jobs of size 10 MB.

| k | Simulated | Predicted | %Error |
|---|---|---|---|
| 1 | 1.000000 | 0.999999 | 0.000100 |
| 2 | 2.000000 | 2.000000 | 0.000000 |
| 3 | 3.000000 | 2.999999 | 0.000033 |
| 4 | 4.000000 | 3.999999 | 0.000025 |
| 5 | 4.000000 | 4.000000 | 0.000000 |
| 6 | 5.000000 | 4.999999 | 0.000020 |
| 7 | 5.000000 | 4.999999 | 0.000020 |
| 8 | 4.645316 | 4.646726 | 0.030353 |
| 9 | 4.000000 | 4.000000 | 0.000000 |
| 10 | 4.000000 | 4.000000 | 0.000000 |
| 11 | 4.000000 | 4.000000 | 0.000000 |
| 12 | 4.000000 | 4.000000 | 0.000000 |

migration and leaves the VC before uploading the intermediate results.

In such cases, a job migration failure occurs, and the work that is competed by the vehicle is lost.

Additionally, this causes an unwanted overhead to the total job completion time of the job, and the next available vehicle must start the job from a previously saved state.

Consider a scenario that vehicle A enters the VC and receives a job with a total processing time of $x$ seconds (Figure 35).

- Vehicle A completes $x_1$ seconds of the processing time and successfully migrates the job to vehicle B.
- Vehicle B enters the VC and receives the job and starts processing the job from the point that vehicle A left and completes $x_2$ seconds of the processing time and fails to migrate the job before departing the VC.
- Vehicle C enters the VC and receives the job and starts processing the job from the point that vehicle A left and completes the remaining $x_3$ seconds of the processing time and successfully uploads the results to the next available AP.

In this scenario, when vehicle B fails to migrate the job, the work that is competed by vehicle B is lost, and therefore vehicle C must start the processing from a previously saved state. This situation adds an overhead from the failed migration to the total job completion time.

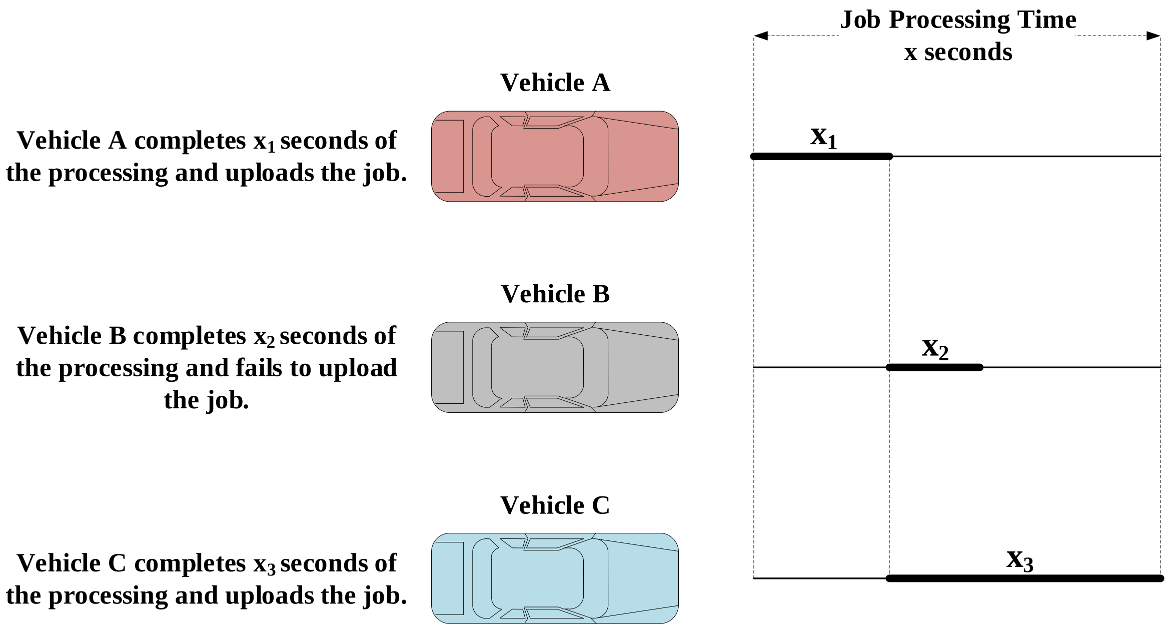


Fig. 35: Illustrating a migration scenario in which vehicle A completes $x_1$ seconds of the processing time of the job and successfully uploads the intermediate results. Vehicle B starts processing the job from the point that vehicle A left and completes $x_2$ seconds of the processing time and fails to migrate the job before departure. Vehicle C starts processing the job from the point that vehicle A left and completes the remaining $x_3$ seconds of the processing time and successfully uploads the results.

### 5.5.2 JOB MIGRATION STRATEGY 2

This job migration strategy is an improvement of the job migration strategy 1 (discussed in Chapter 5.5.1), and the main distinction of this migration strategy is that in this migration strategy, vehicles entering the VC can start to download chunks of the job and the intermediate results uploaded by the previous vehicles and to not wait for the other vehicle to entirely upload the job. Depending on the size of the job, vehicles may need to travel to multiple APs to successfully upload a job, and therefore by reducing the wait time and exploiting the resources of the available vehicles as soon as possible, we can significantly reduce the overhead to the job completion time.

Consider a scenario that vehicle A, is assigned a job, and after downloading the job, vehicle A starts the processing of the job. Vehicle A does not remain in the VC long enough to complete the processing of the job and therefore needs to migrate the job before departing the VC. Vehicle A starts the upload of the intermediate results at the $AP_i$ (Figure 37). The upload of the job cannot finish under one AP; therefore, vehicle A uploads a chunk of the job at the $AP_i$ and continues to upload the rest of the job at the $AP_{i+1}$. While the vehicle A is uploading the remaining of the job at the $AP_{i+1}$, vehicle B enters the VC, and is assigned the same job at the $AP_i$. Vehicle B does not wait until vehicle A finishes the upload of the job, and starts the download of the chucks of the job that was previously uploaded by vehicle A at the $AP_i$. Once vehicle A completes the upload of the job at the $AP_{i+1}$, it departs the VC. Vehicle B completes the download of the job at the $AP_{i+1}$.

### 5.5.3 SIMULATION MODEL

We have simulated a three-lane highway with APs placed every 2000 meters. Each AP has a coverage area of 100 meters in which the vehicles driving along the highway can transmit or receive messages. The APs continuously send out frames of length 56624 bits with a payload of 53792 bits.

The speed of each vehicle is determined by the traffic density. In our simulation, we have used the five-parameter logistic speed-density function described in Chapter 3.2 to determine the vehicle's speed based on the number of the vehicles in the coverage area, the values of which are available in Table 2.

Vehicles enter the coverage area of an AP at random times, in which the inter-arrival time, or the average waiting time for a vehicle to arrive at the AP is exponentially distributed, and the mean inter-arrival time is 5 seconds.

Each vehicle that enters the VC has a residency time. The residency time of a vehicles is in terms of the number of APs, meaning the number of APs that a vehicle will be under the coverage of while remaining in the VC. The residency times of the vehicles are normally distributed with different means of 15, 20, 25, 30, 35, 40, 45, 50, 55 and 60 APs and a standard deviation of 1 AP.

When a vehicle enters the coverage area of an AP and receives the beginning of the frame, it competes with the other vehicles in the same coverage area. For this purpose, it chooses at random one of the 20 slots in the first contention period and the same procedure is repeated in the second contention period. Vehicles that select a unique slot in either contention periods are *successful*. The available payload is then divided equally among the successful vehicles.

A vehicle that contacts the AP for the first time is assigned a job of size 10 MB with a processing time normally distributed with a mean of 1200 seconds (or 20 minutes) and a standard deviation of 60 seconds. The vehicle then starts the download of the job and continues to compete in the next frames until the job is fully downloaded. The job execution starts immediately after the download of the job. If a vehicle must leave before the the job execution is completed, it needs to migrate the job. By having the estimated upload time of the job and the residency time of the vehicle, the AP number at which the vehicle must start the migration is determined.

- Job Migration Strategy 1

  In this migration strategy, when the vehicle approaches the coverage area of the designated AP at which it should start the migration, it starts the migration by uploading the partially completed job. Similar to download, the vehicle competes with other vehicles in the coverage area, for acquiring transmission slots. The upload of the partially completed job may take several APs. Once the upload of the intermediate results is completed, the job is then assigned to another available vehicle in the VC.
- Job Migration Strategy 2

  In this migration strategy, when the vehicle approaches the coverage area of the designated AP at which it should start the migration, it starts the migration by uploading the partially completed job. Similar to download, the vehicle competes with other vehicles in the coverage area, for acquiring transmission slots. The upload of the partially completed job may take several APs. If another vehicle is available in the VC to accept the job and the intermediate results are partially uploaded then we do not wait for the upload of the intermediate results to be completed, and assign the job to the available vehicle in the VC to start the download of the job immediately.

If a vehicle is unsuccessful at migration and uploading the intermediate results, we do not take into account the processing time that was completed by the vehicle. If the job execution is completed, the vehicle immediately attempts to upload the results. The process of uploading of the results is similar to download, in terms of competing for transmission slots. If a vehicle is not under the coverage area at the time that the job execution is completed, then it attempts to upload the results at the next available AP. We record the job completion time from the moment that the job is assigned to a vehicle until the final results are uploaded.

Our simulations were developed in house, using the Java programming language, with each experiment repeated $10^4$ times. Table 12 shows the values of each parameter.

### 5.5.4 SIMULATION RESULTS

In this section we present our simulation results. The simulation results for job completion times for k values of 1 to 12 are shown in Figures 38 to 49.

The average job completion time values from simulations for different job migration strategies are plotted. As the vehicle residency time increases, the average job completion time for both migration strategies decreases and eventually remains relatively at the same value. The reason for this is as the average vehicle residency time increases, the average number of migrations decreases and eventually remains at the value 0.

The simulation results for the average number of successful job migration for k values of 1 to 12 are shown in Figure 50 to 61.

The average total job completion time using the migration strategy 2 decreases compared to when using the migration strategy 1, for cases in which job migration occurs. For instance the average decrease in total job completion time for average vehicle residency time of 15 APs for k values of 1 to 12 is 800 seconds or 13.33 minutes.

TABLE 12: Simulation parameters

| **Symbol and Description** | **Value** |
|---|---|
| $l$ (number of lanes) | 3 |
| $B$ (available bandwidth) | $27 \times 10^6$ bps |
| $W$ (size of the job) | $8 \times 10^7$ bits |
| $b$ (payload in one frame) | 53792 bits |
| $F$ (frame length in bits) | 56624 bits |
| $F_s$ (frame length in seconds) | 0.002 s |
| $M$ (number of available slots for competing) | 20 |
| $c$ (access point coverage range) | 100 m |
| $d$ (distance between two consecutive APs) | 2000 m |
| $v_{k+1}$ (vehicle's speed when k other vehicles are in the area) | (10, 30) m/s |
| average job processing time | 1200 s |
| standard deviation of job processing time | 60 s |
| average inter-arrival time of vehicles | 5 s |
| average vehicle residency time | (15, 60) APs |
| number of runs | $10^4$ |

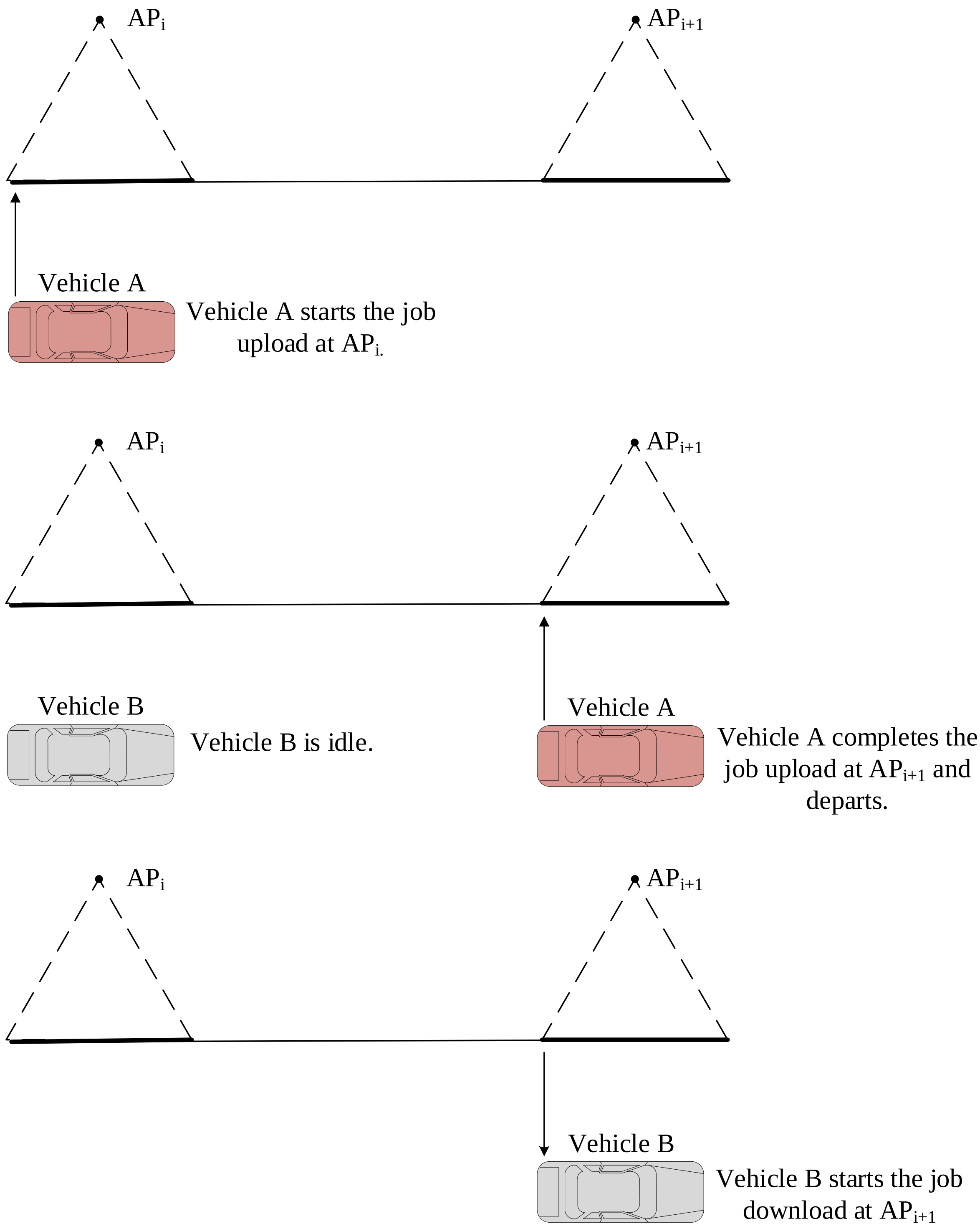


Fig. 36: Illustrating a scenario for job migration strategy 1.

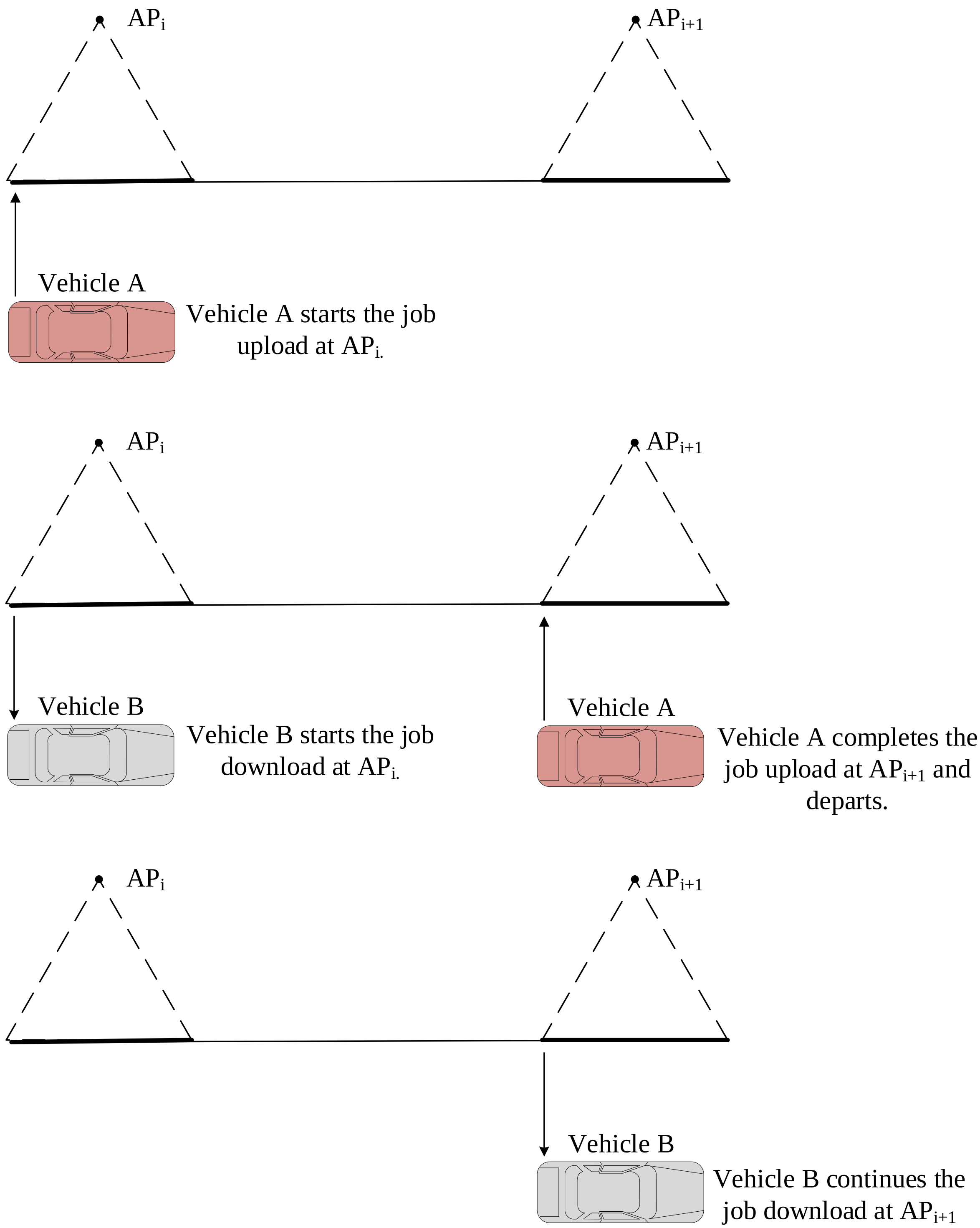


Fig. 37: Illustrating a scenario for job migration strategy 2.

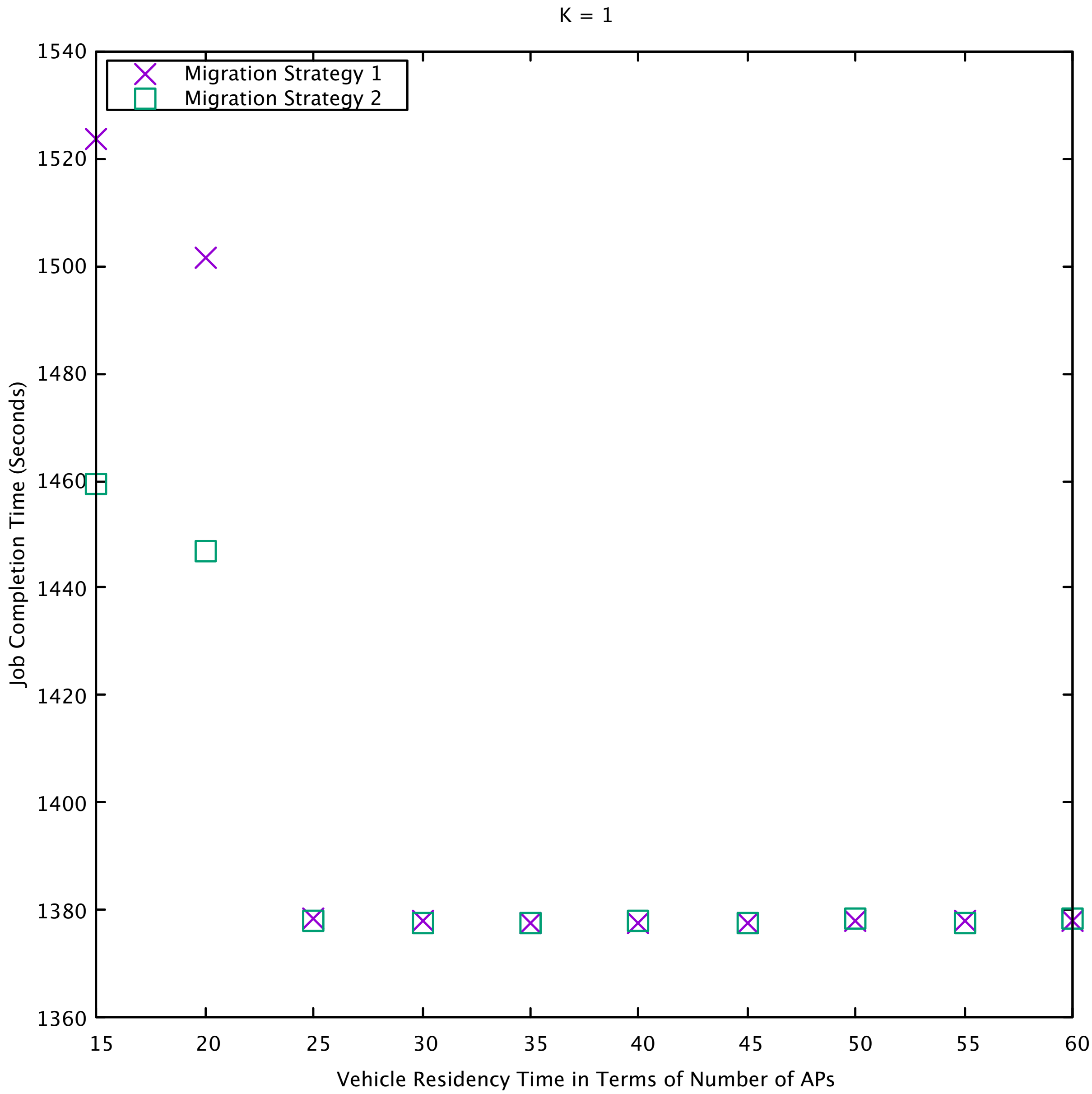


Fig. 38: Job completion times for different migration strategies given that one other vehicle is competing to receive a communication slot.

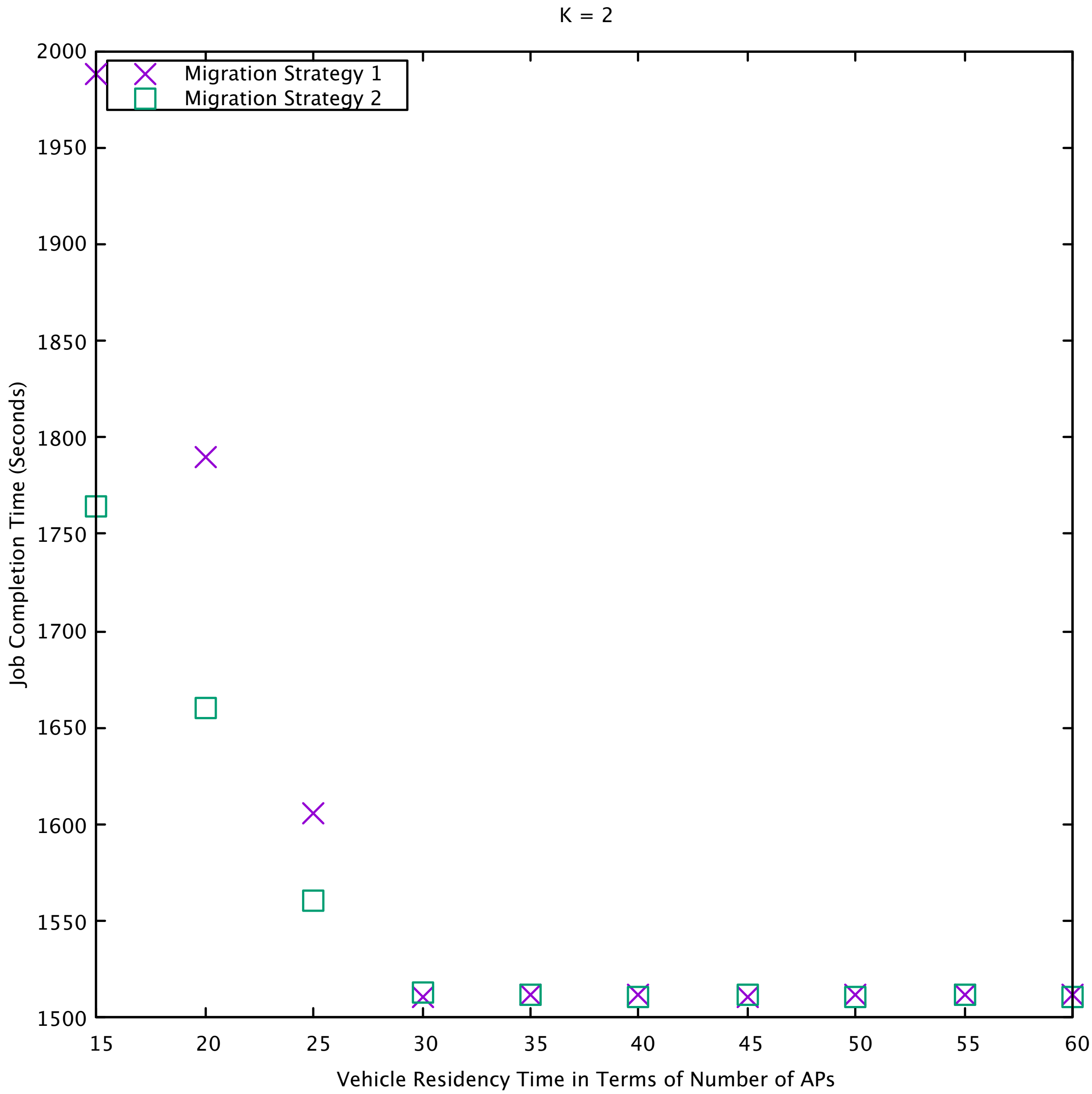


Fig. 39: Job completion times for different migration strategies given that two other vehicles are competing to receive a communication slot.

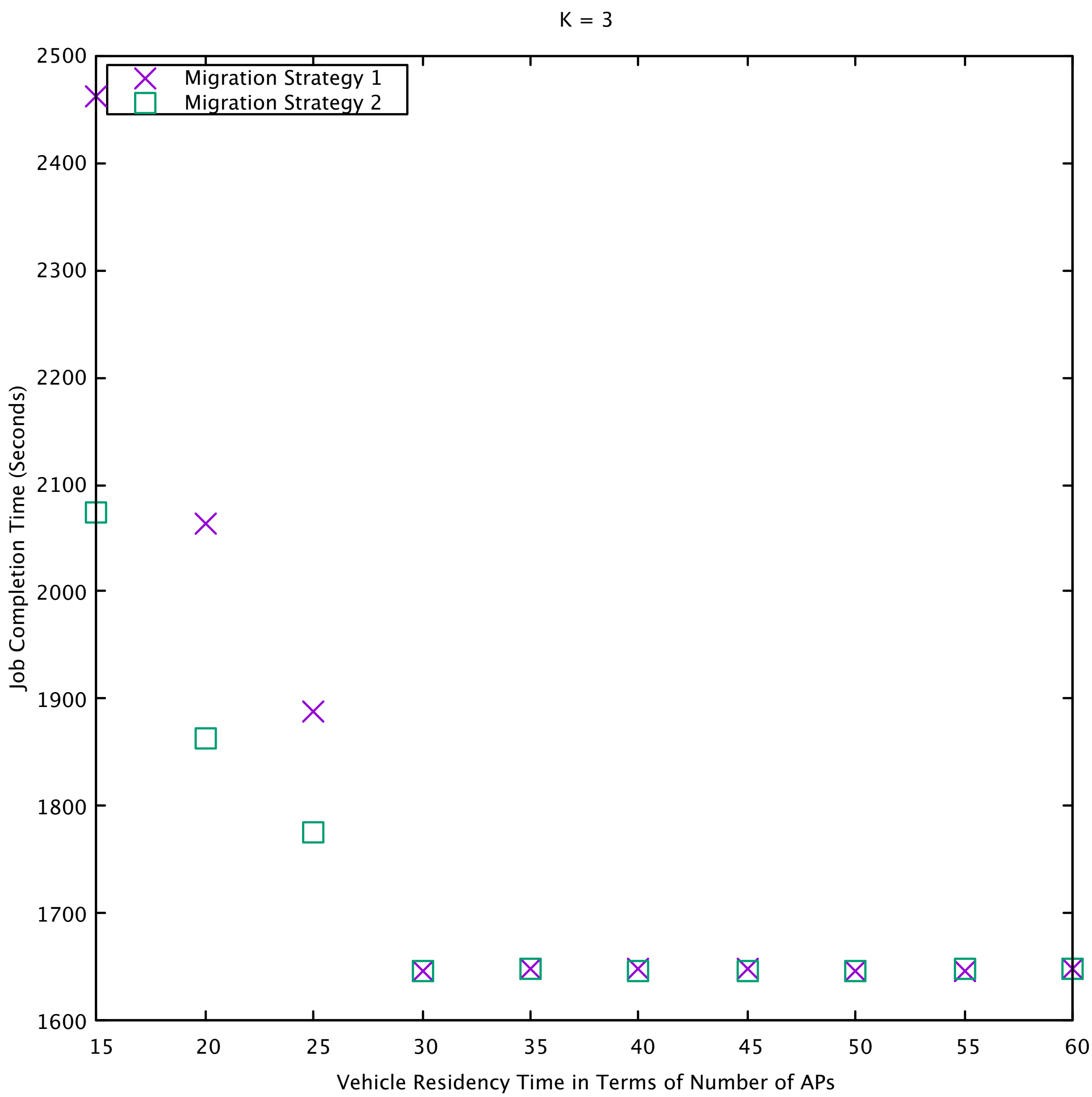


Fig. 40: Job completion times for different migration strategies given that three other vehicles are competing to receive a communication slot.

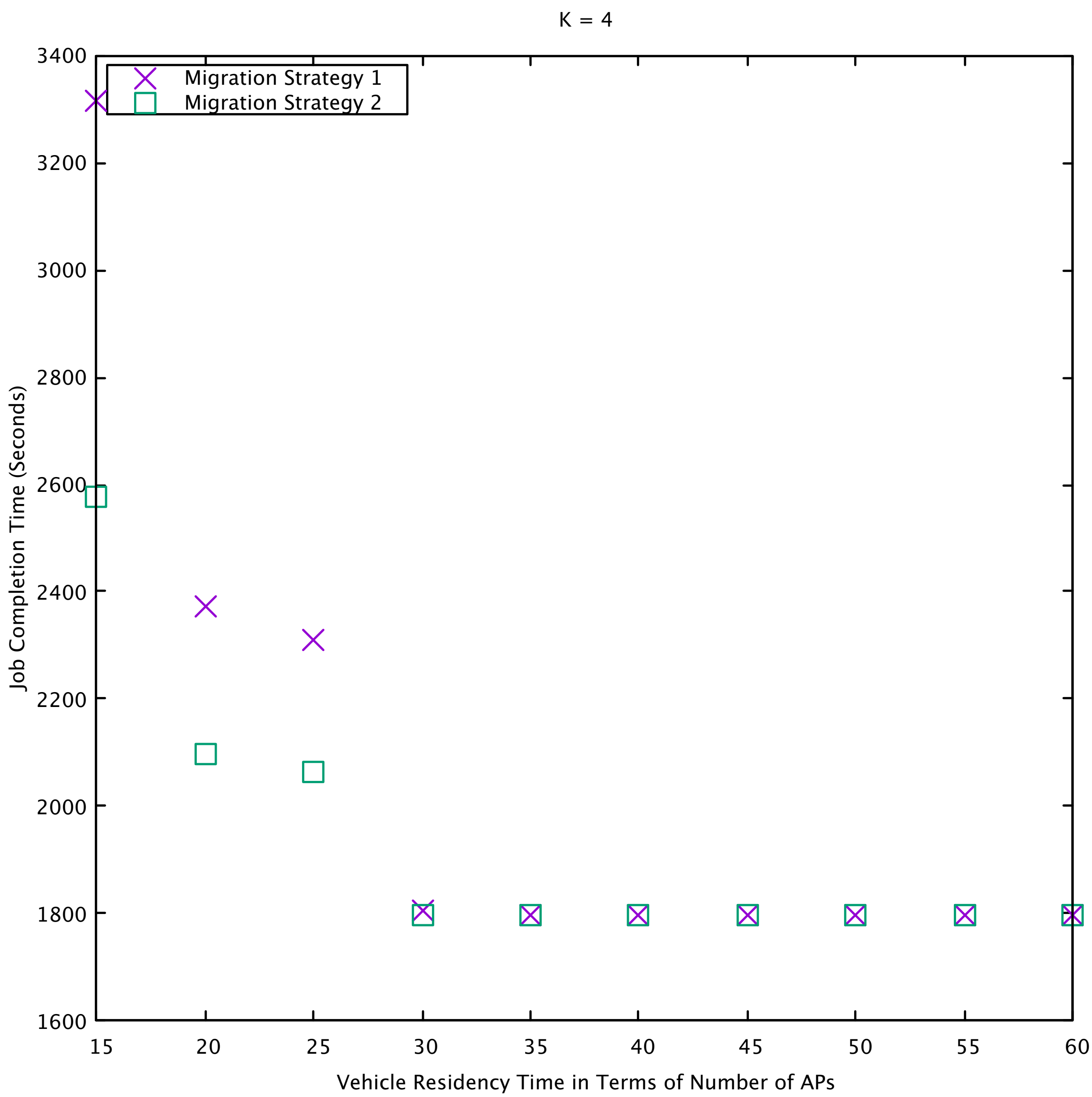


Fig. 41: Job completion times for different migration strategies given that four other vehicles are competing to receive a communication slot.

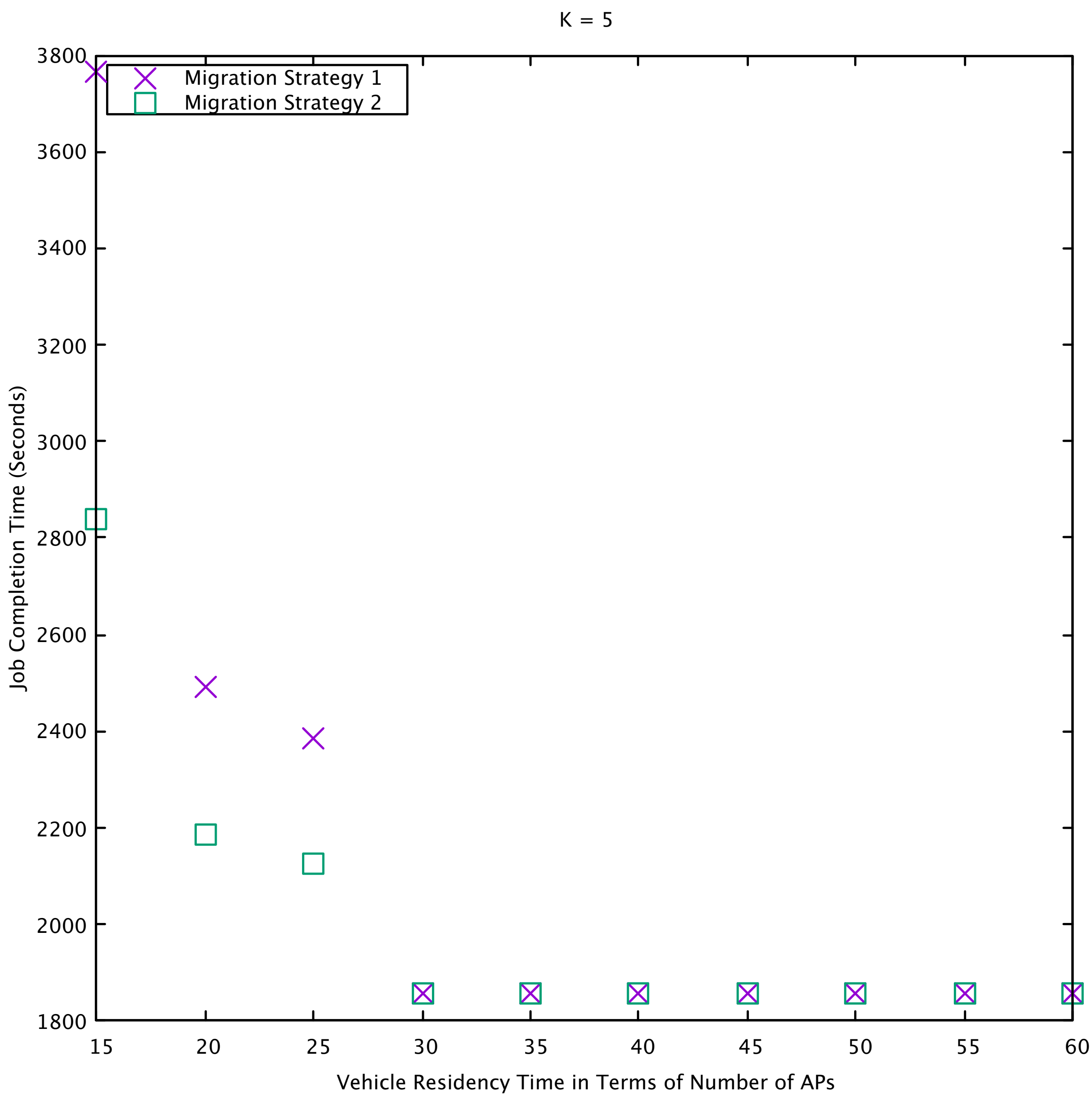


Fig. 42: Job completion times for different migration strategies given that five other vehicles are competing to receive a communication slot.

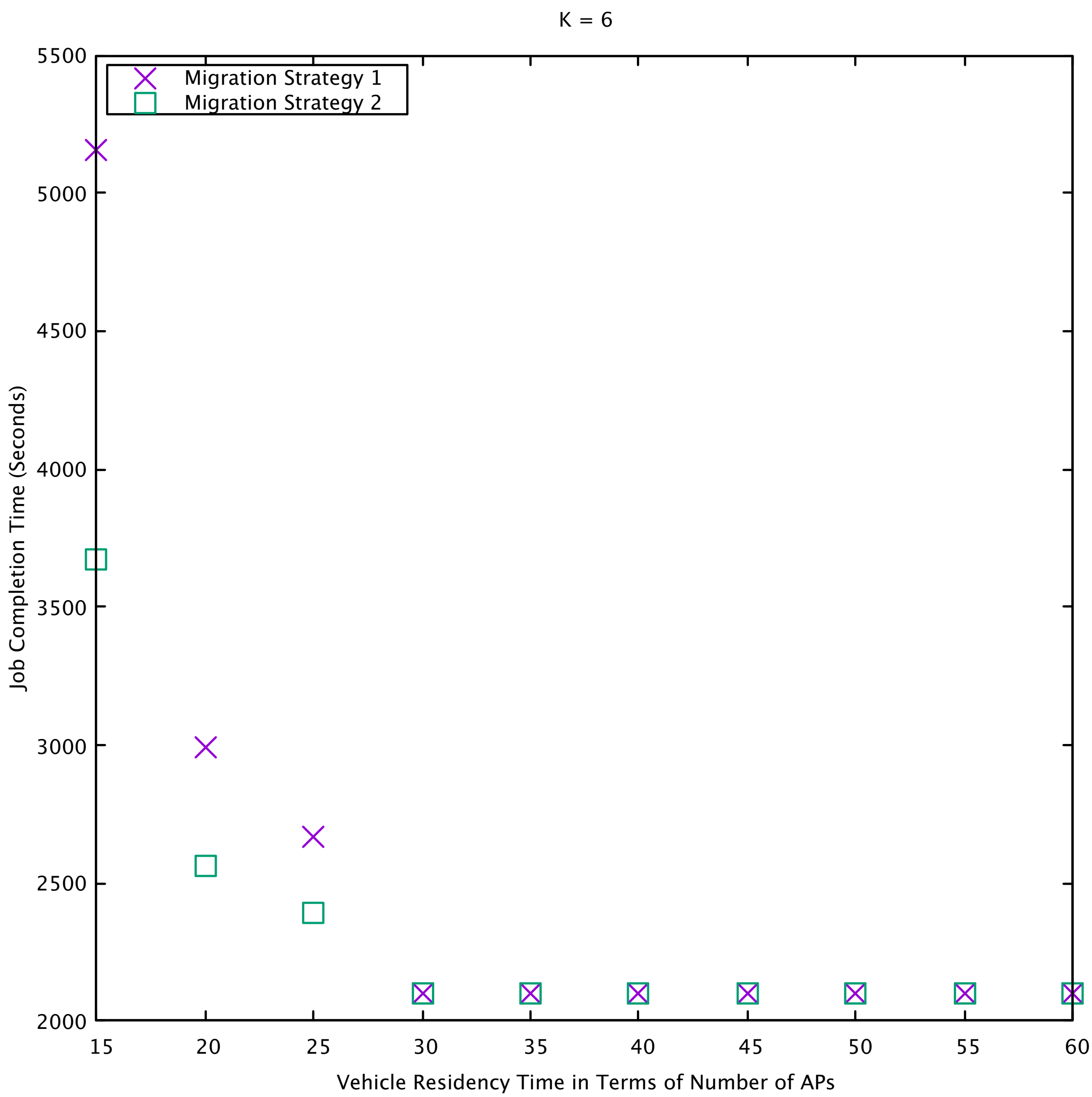


Fig. 43: Job completion times for different migration strategies given that six other vehicles are competing to receive a communication slot.

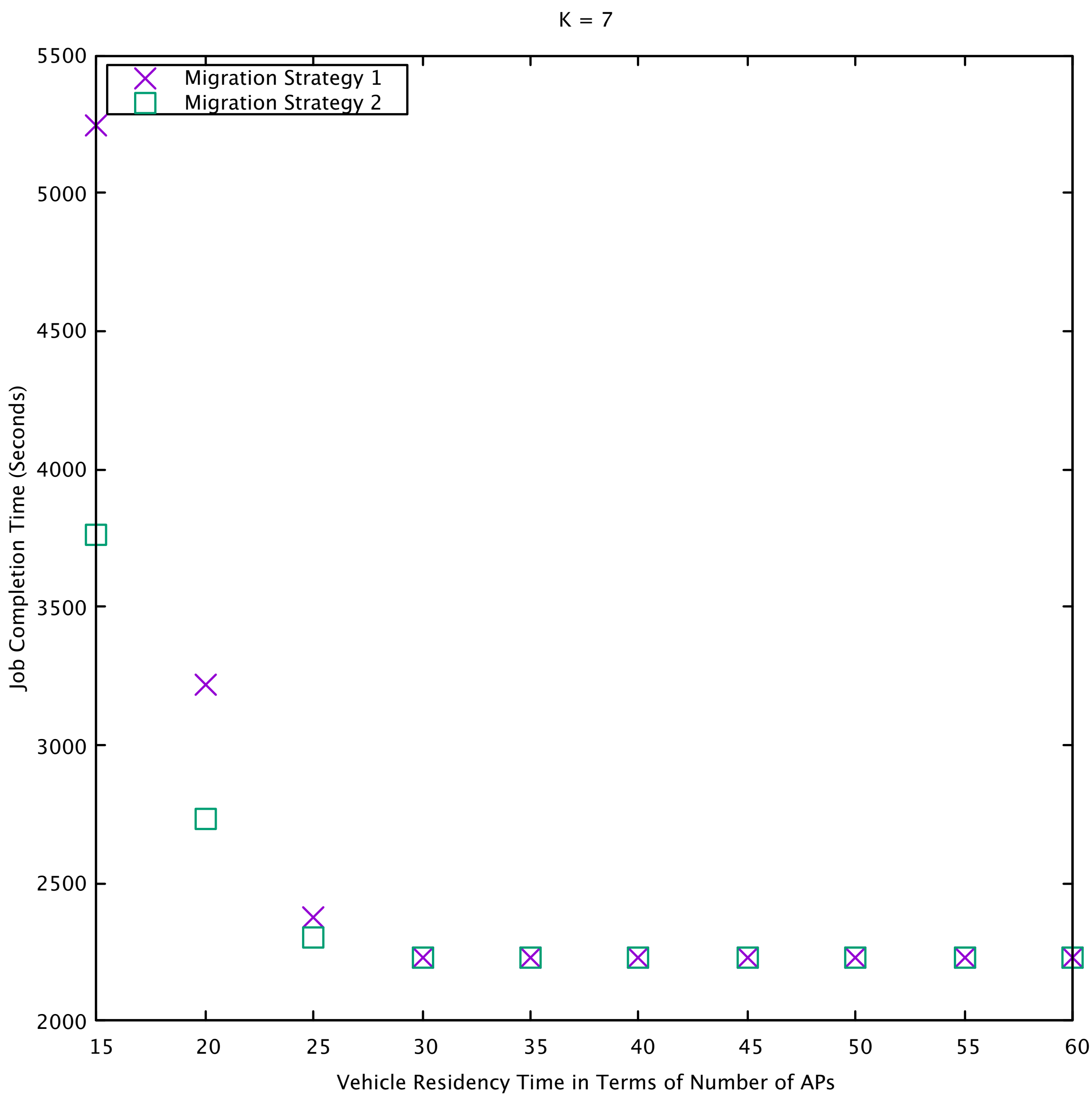


Fig. 44: Job completion times for different migration strategies given that seven other vehicles are competing to receive a communication slot.

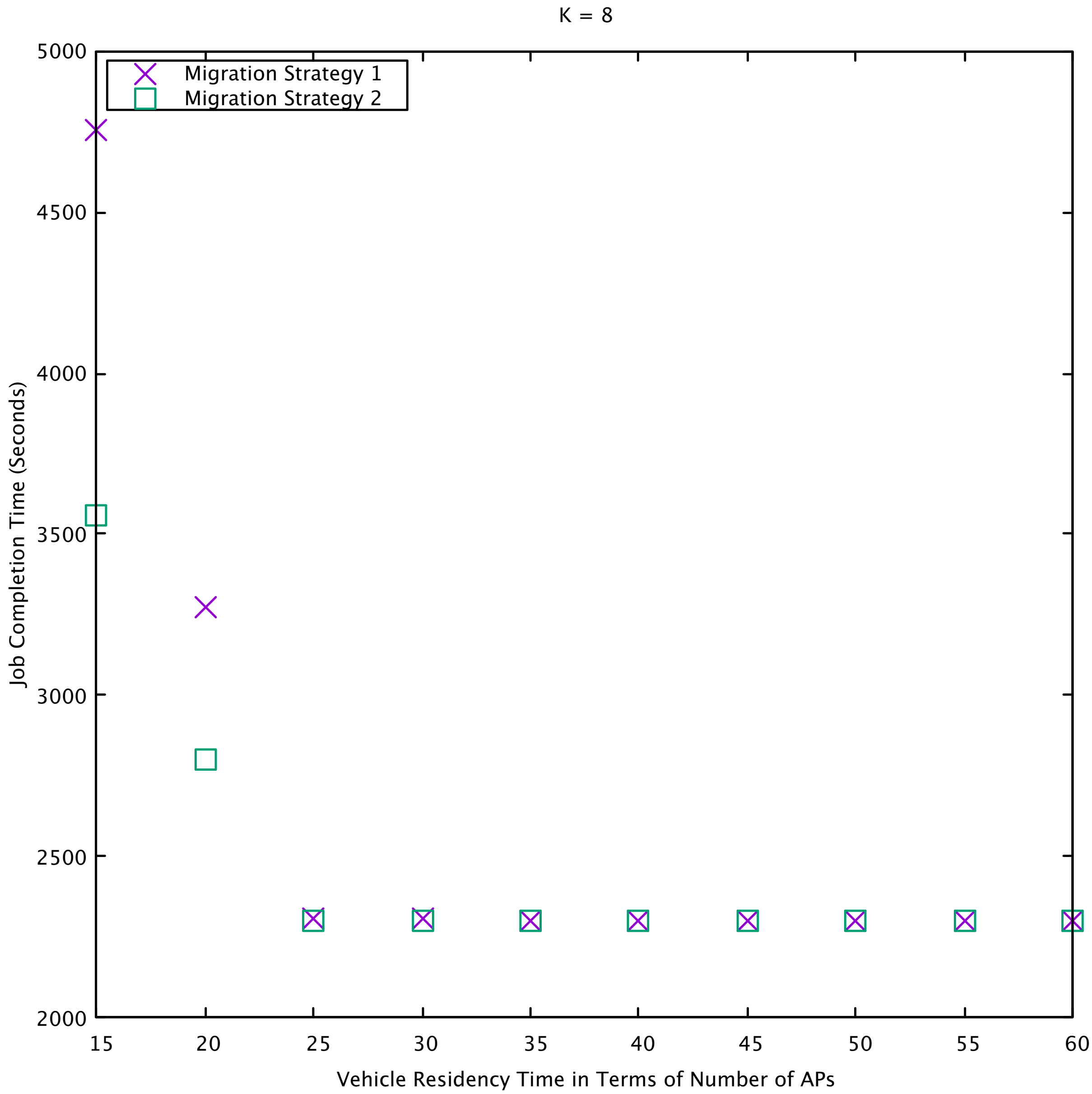


Fig. 45: Job completion times for different migration strategies given that eight other vehicles are competing to receive a communication slot.

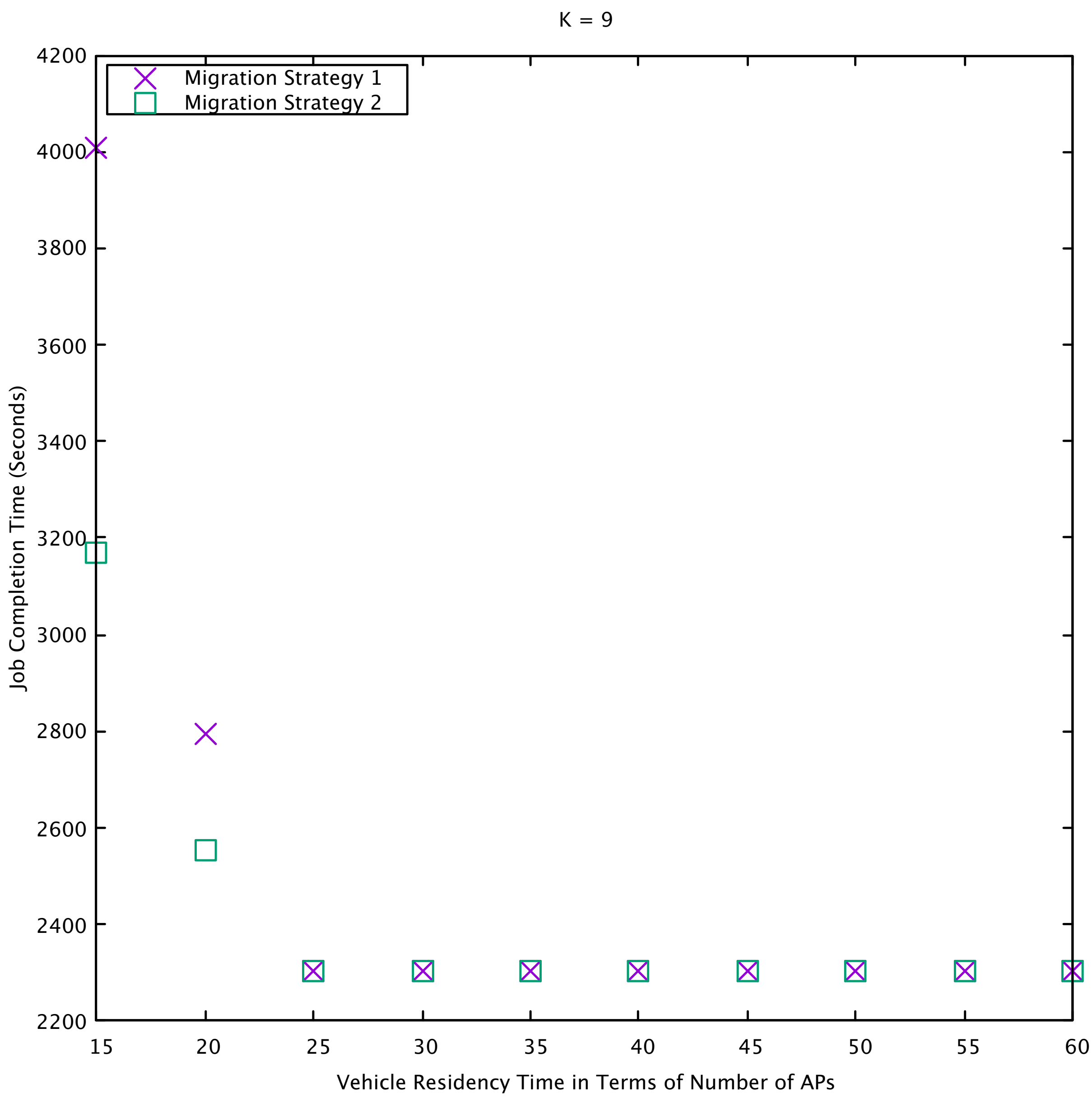


Fig. 46: Job completion times for different migration strategies given that nine other vehicles are competing to receive a communication slot.

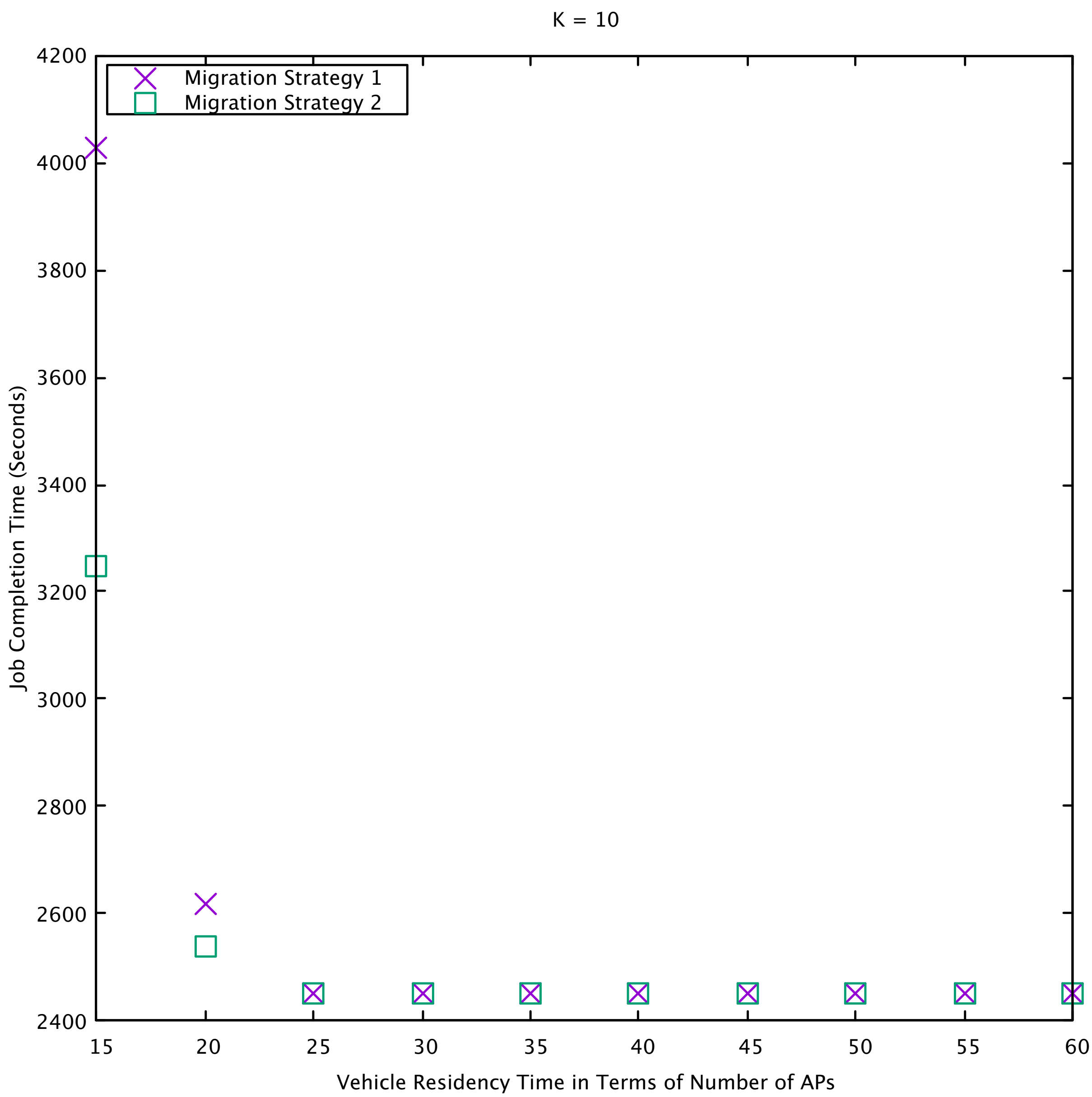


Fig. 47: Job completion times for different migration strategies given that ten other vehicles are competing to receive a communication slot.

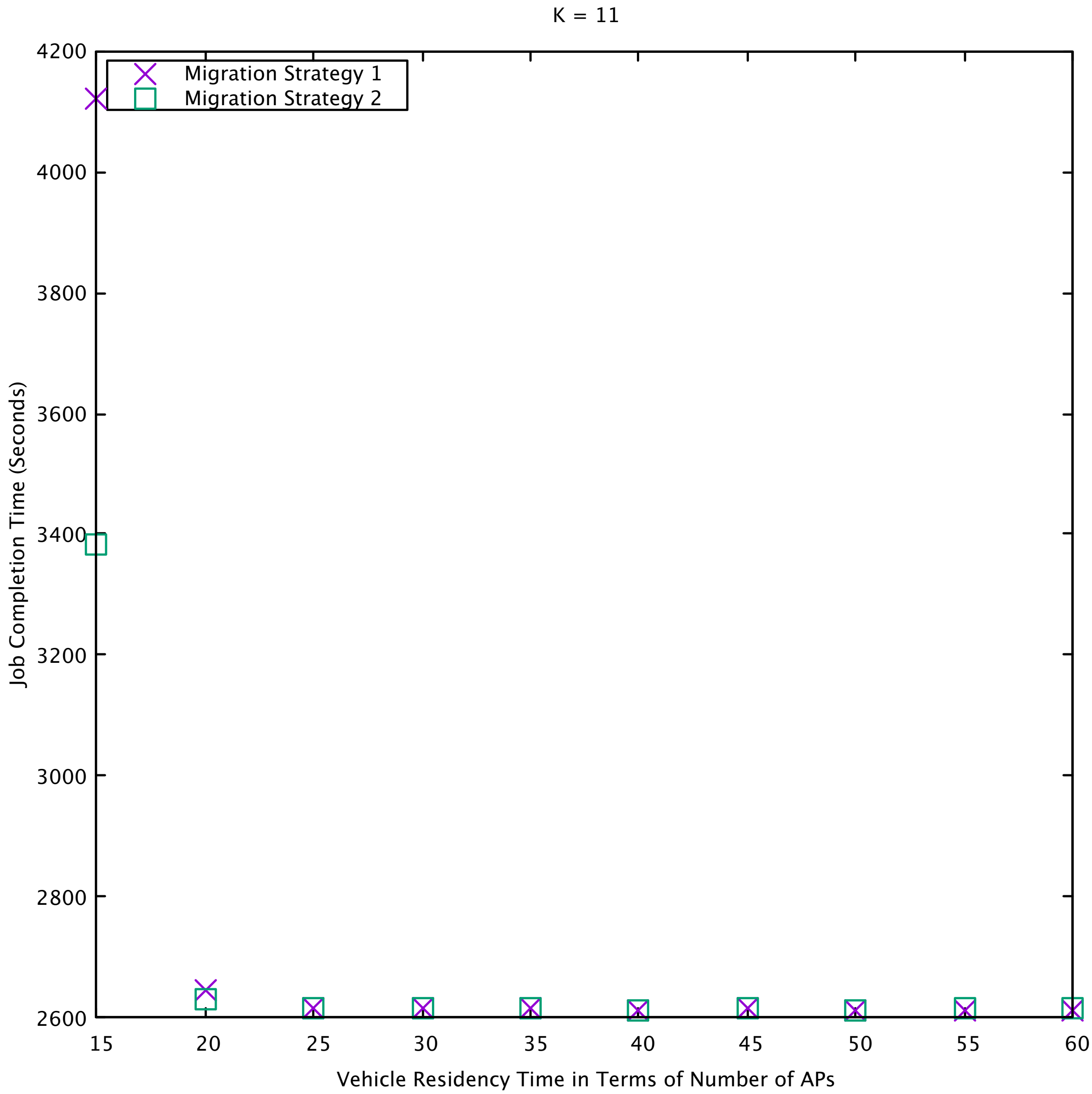


Fig. 48: Job completion times for different migration strategies given that 11 other vehicles are competing to receive a communication slot.

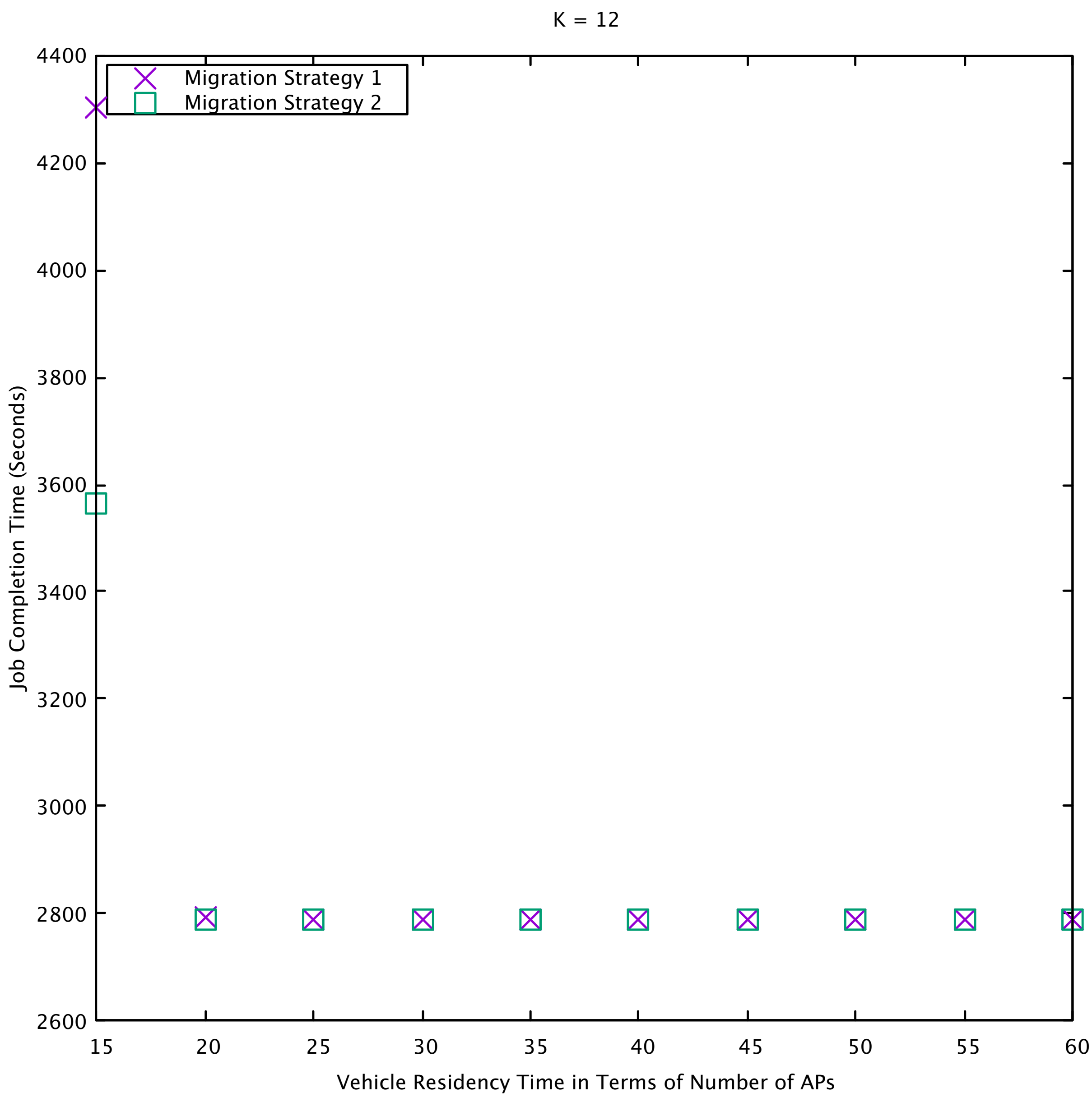


Fig. 49: Job completion times for different migration strategies given that 12 other vehicles are competing to receive a communication slot.

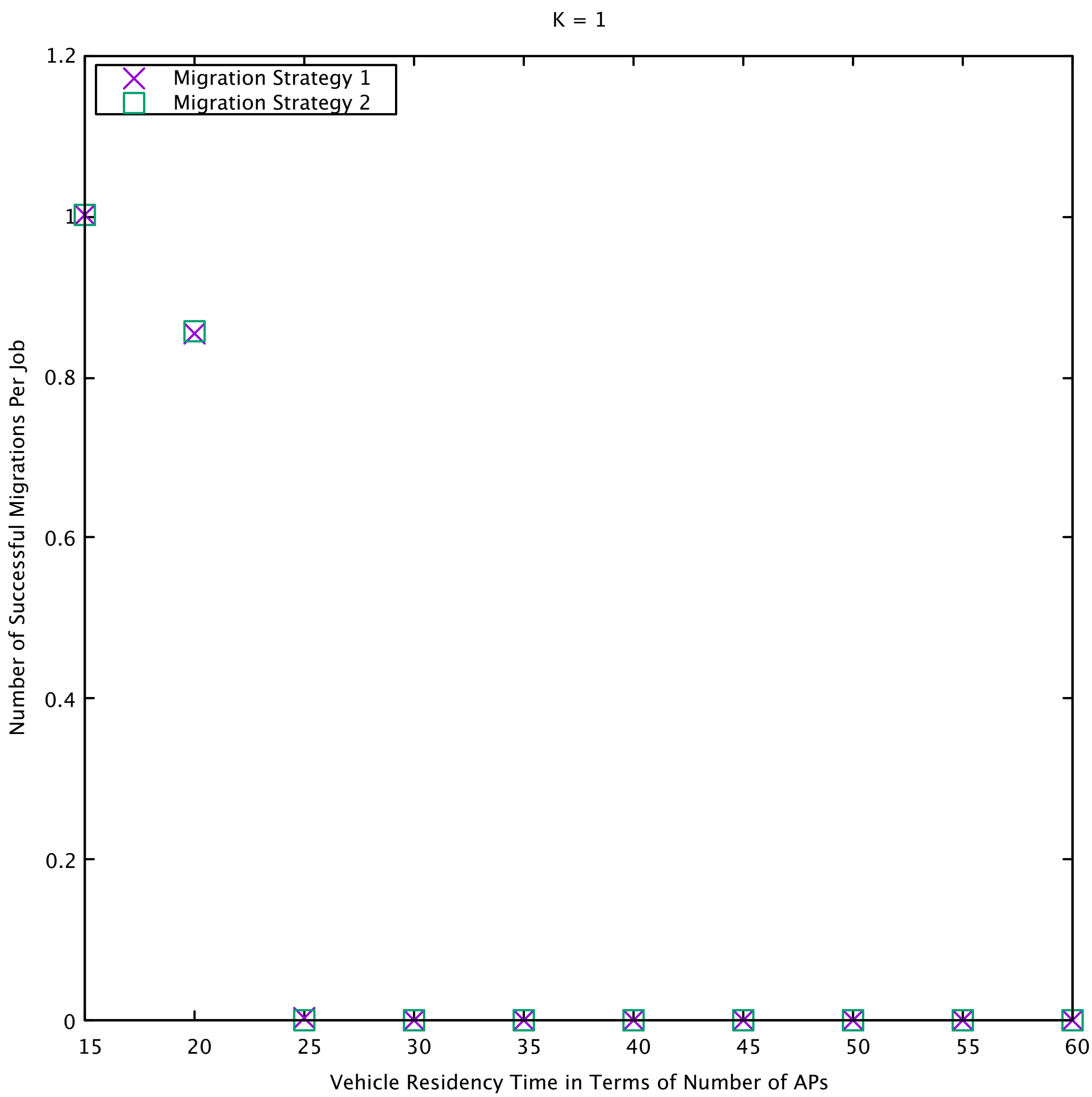


Fig. 50: Average number of successful migrations per job, for different migration strategies given that one other vehicle is competing to receive a communication slot.

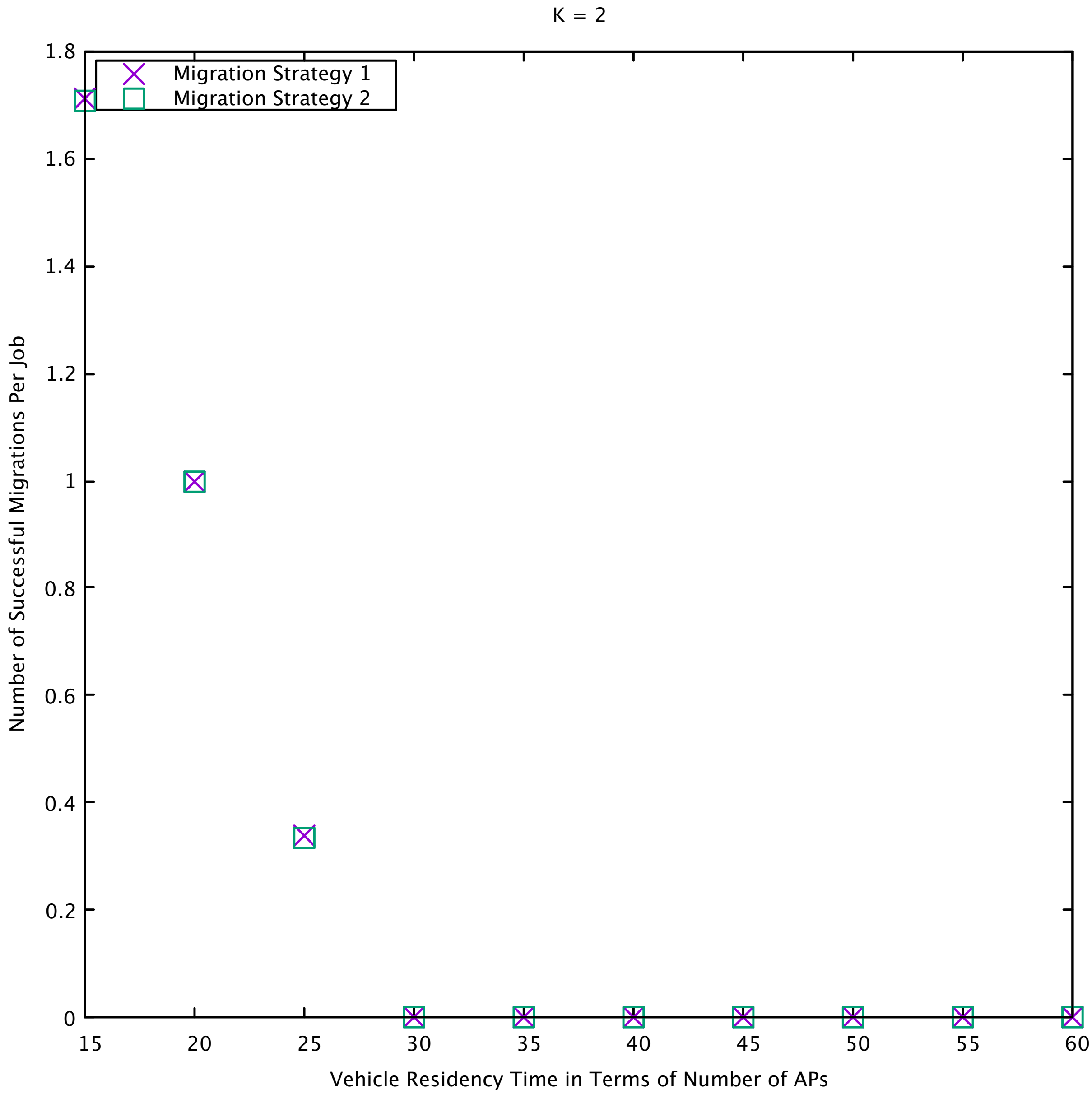


Fig. 51: Average number of successful migrations per job, for different migration strategies given that two other vehicles are competing to receive a communication slot.

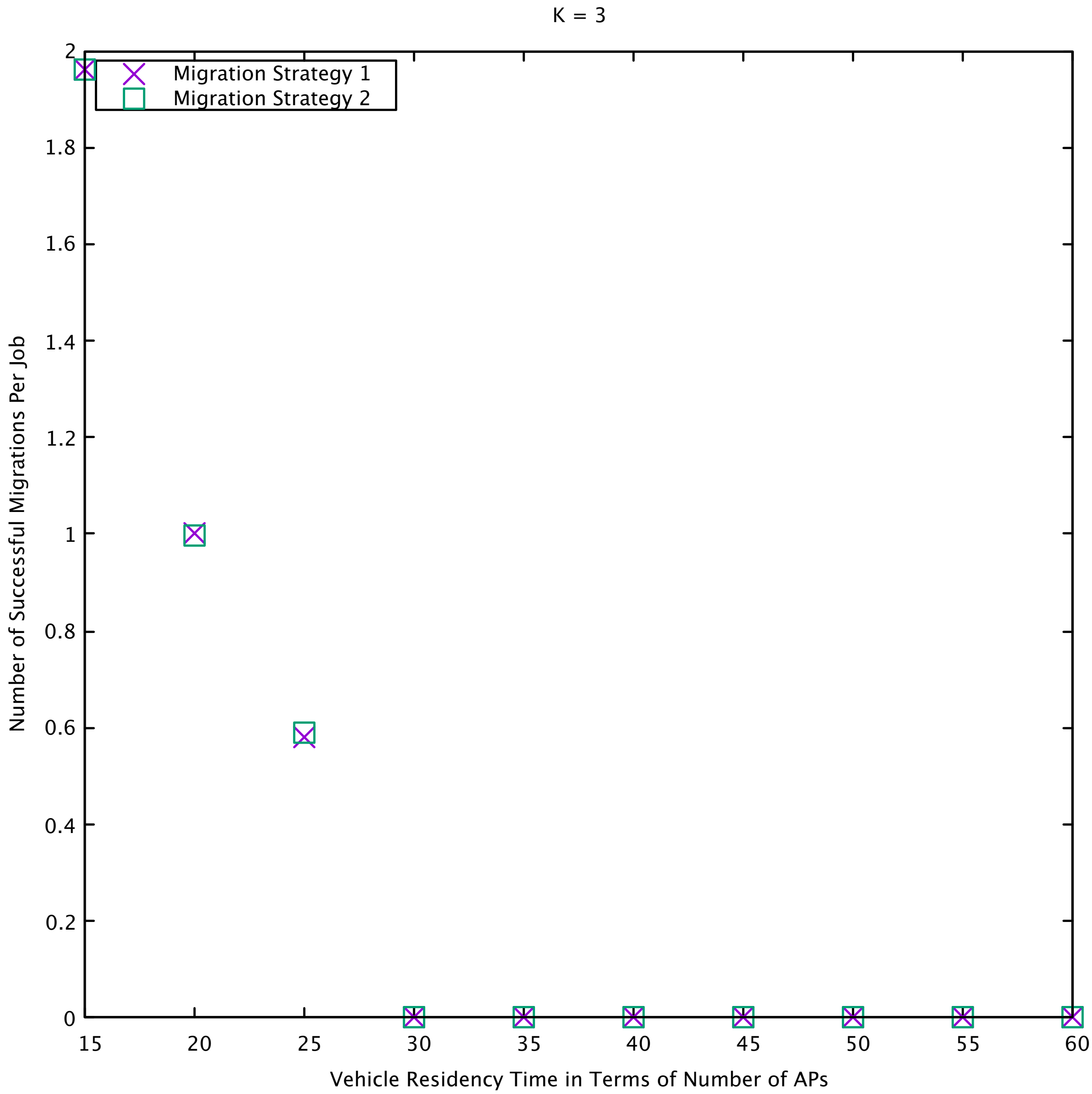


Fig. 52: Average number of successful migrations per job, for different migration strategies given that three other vehicles are competing to receive a communication slot.

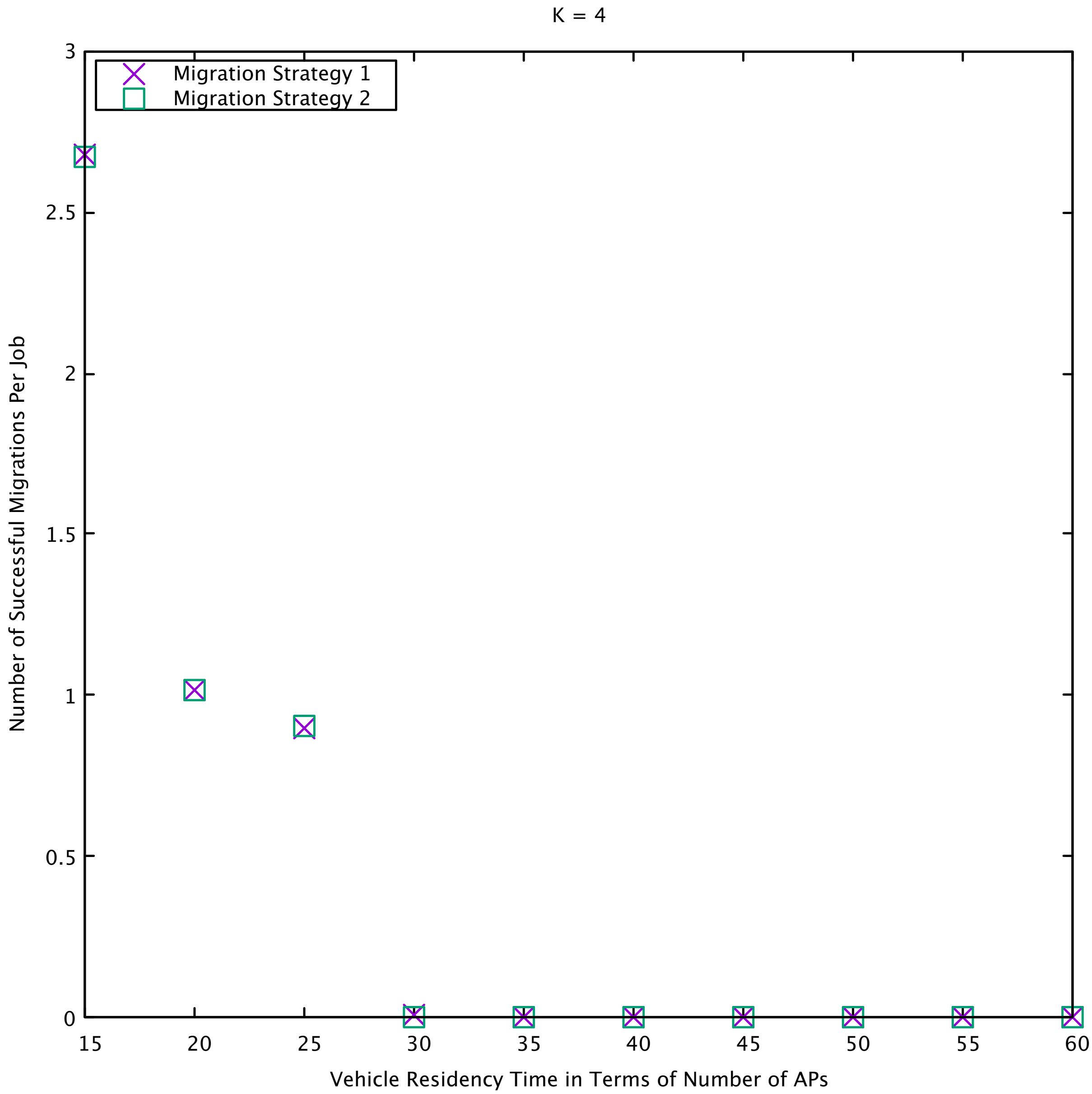


Fig. 53: Average number of successful migrations per job, for different migration strategies given that four other vehicles are competing to receive a communication slot.

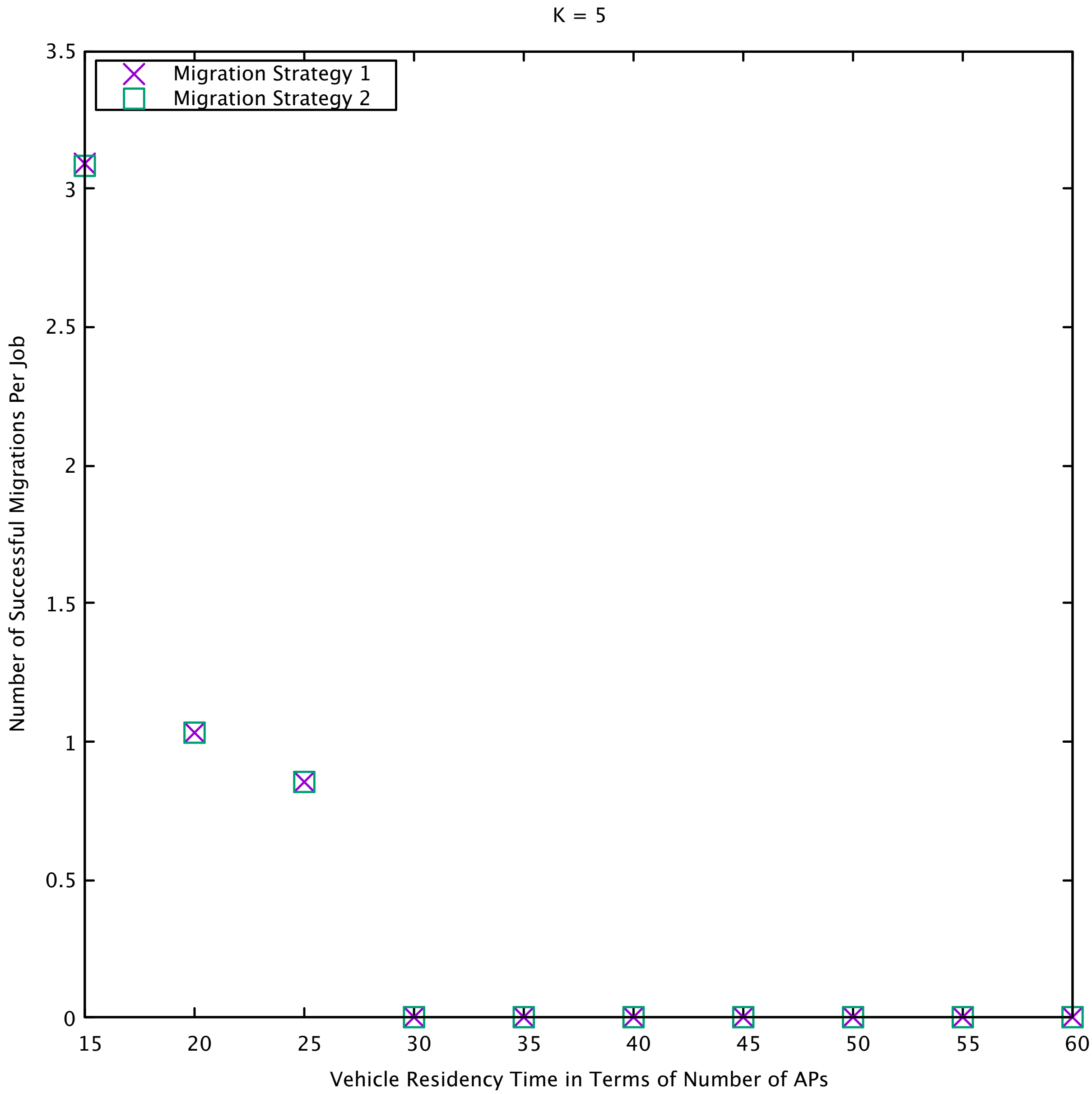


Fig. 54: Average number of successful migrations per job, for different migration strategies given that five other vehicles are competing to receive a communication slot.

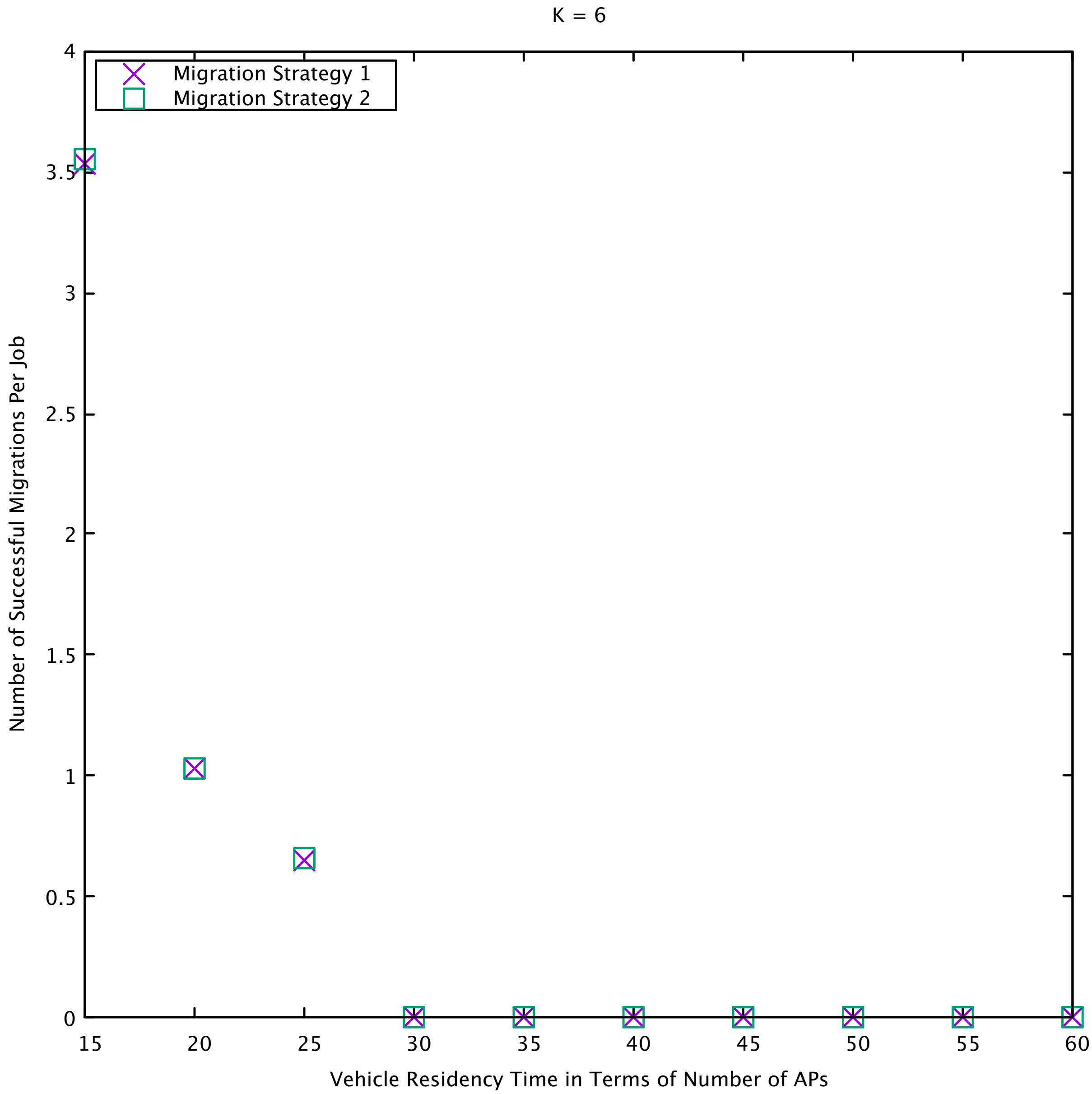


Fig. 55: Average number of successful migrations per job, for different migration strategies given that six other vehicles are competing to receive a communication slot.

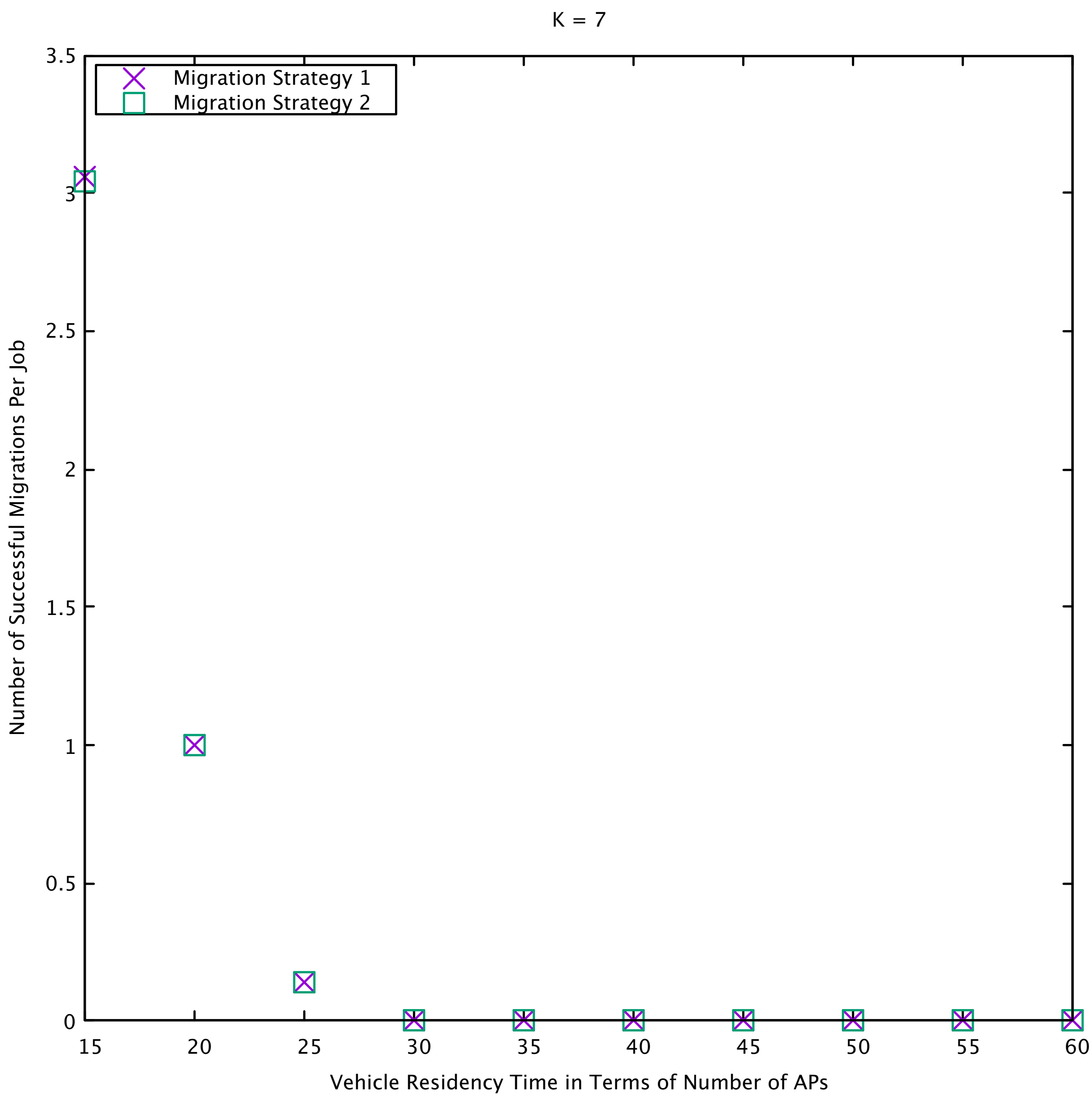


Fig. 56: Average number of successful migrations per job, for different migration strategies given that seven other vehicles are competing to receive a communication slot.

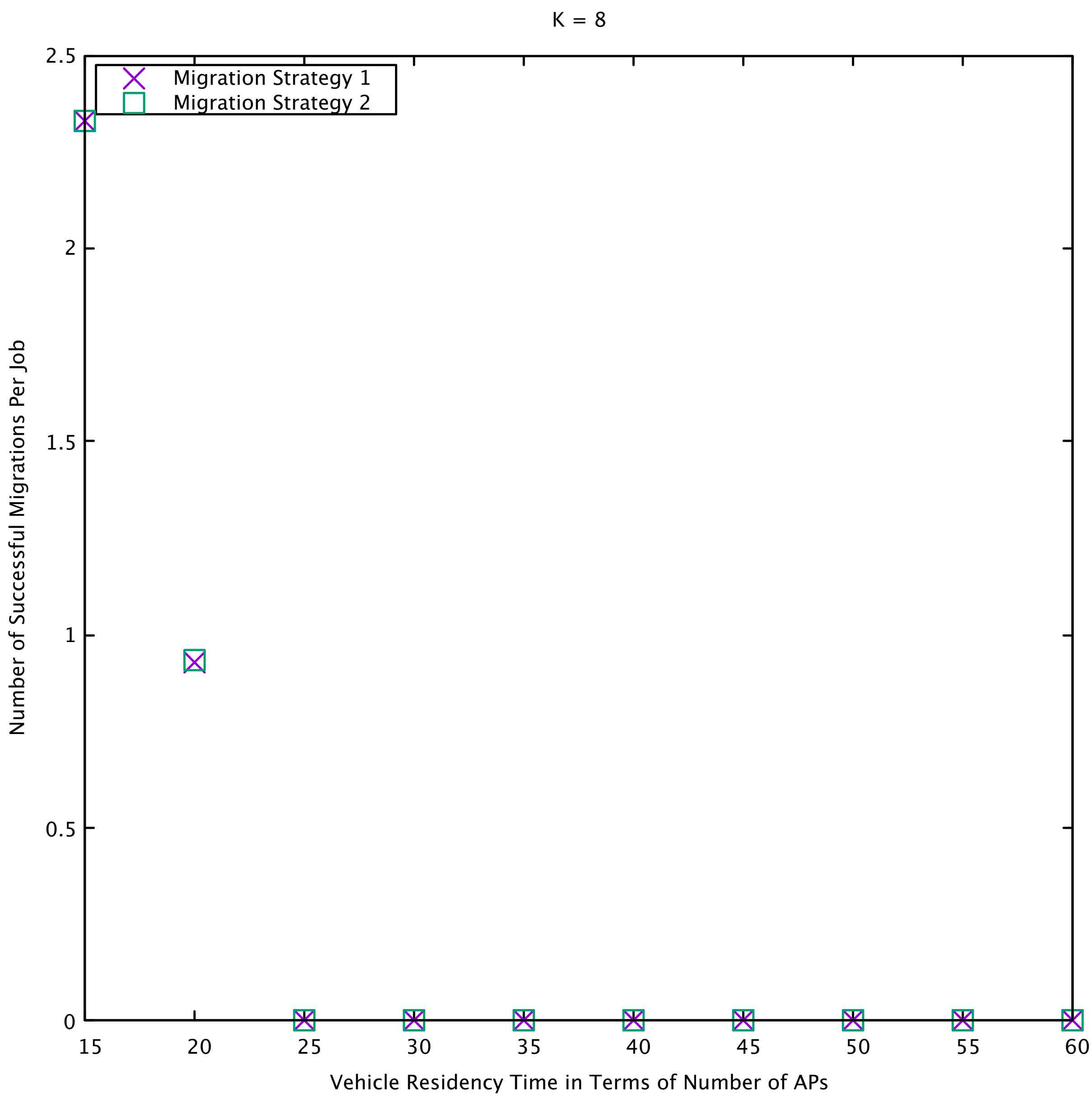


Fig. 57: Average number of successful migrations per job, for different migration strategies given that eight other vehicles are competing to receive a communication slot.

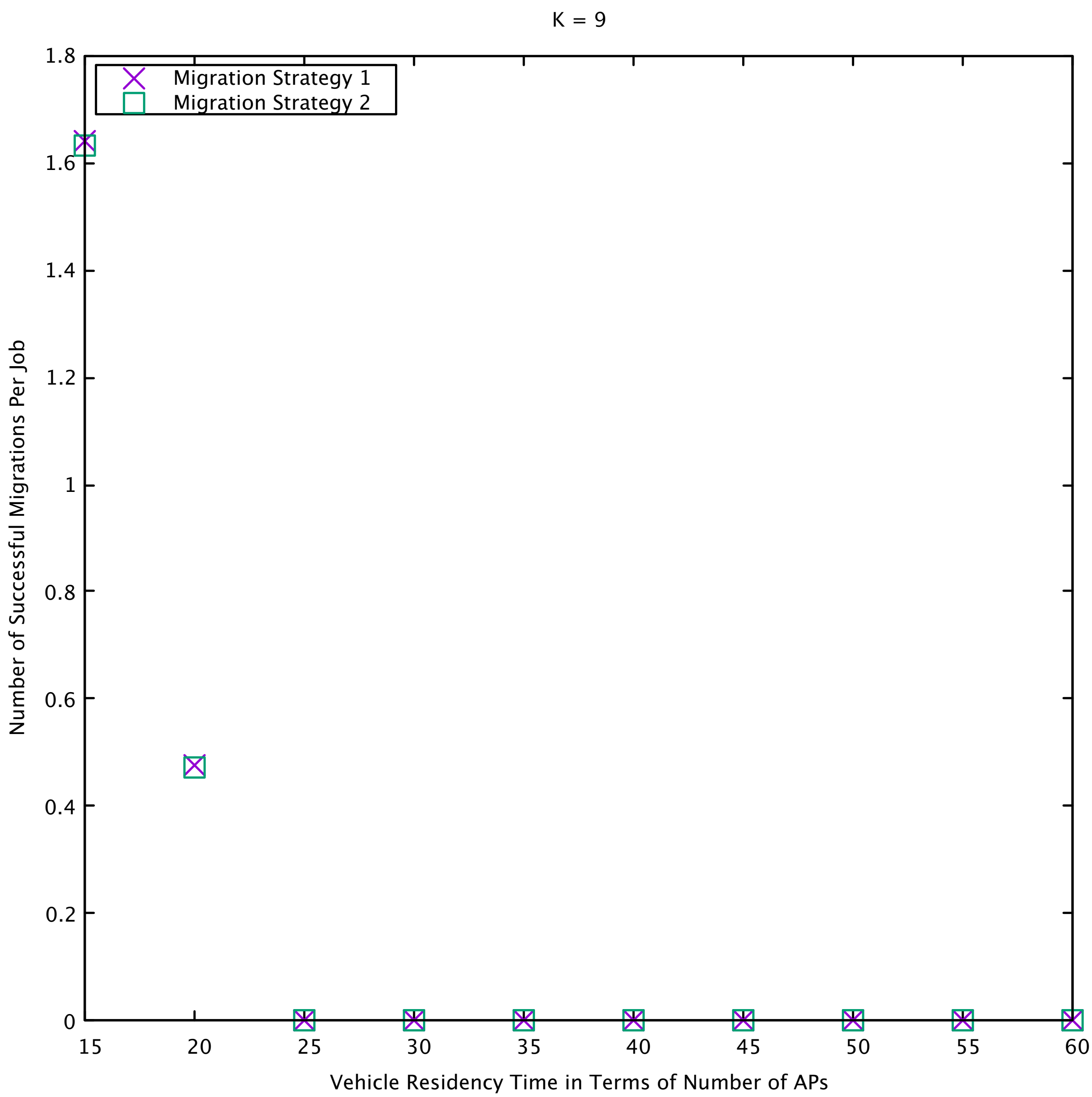


Fig. 58: Average number of successful migrations per job, for different migration strategies given that nine other vehicles are competing to receive a communication slot.

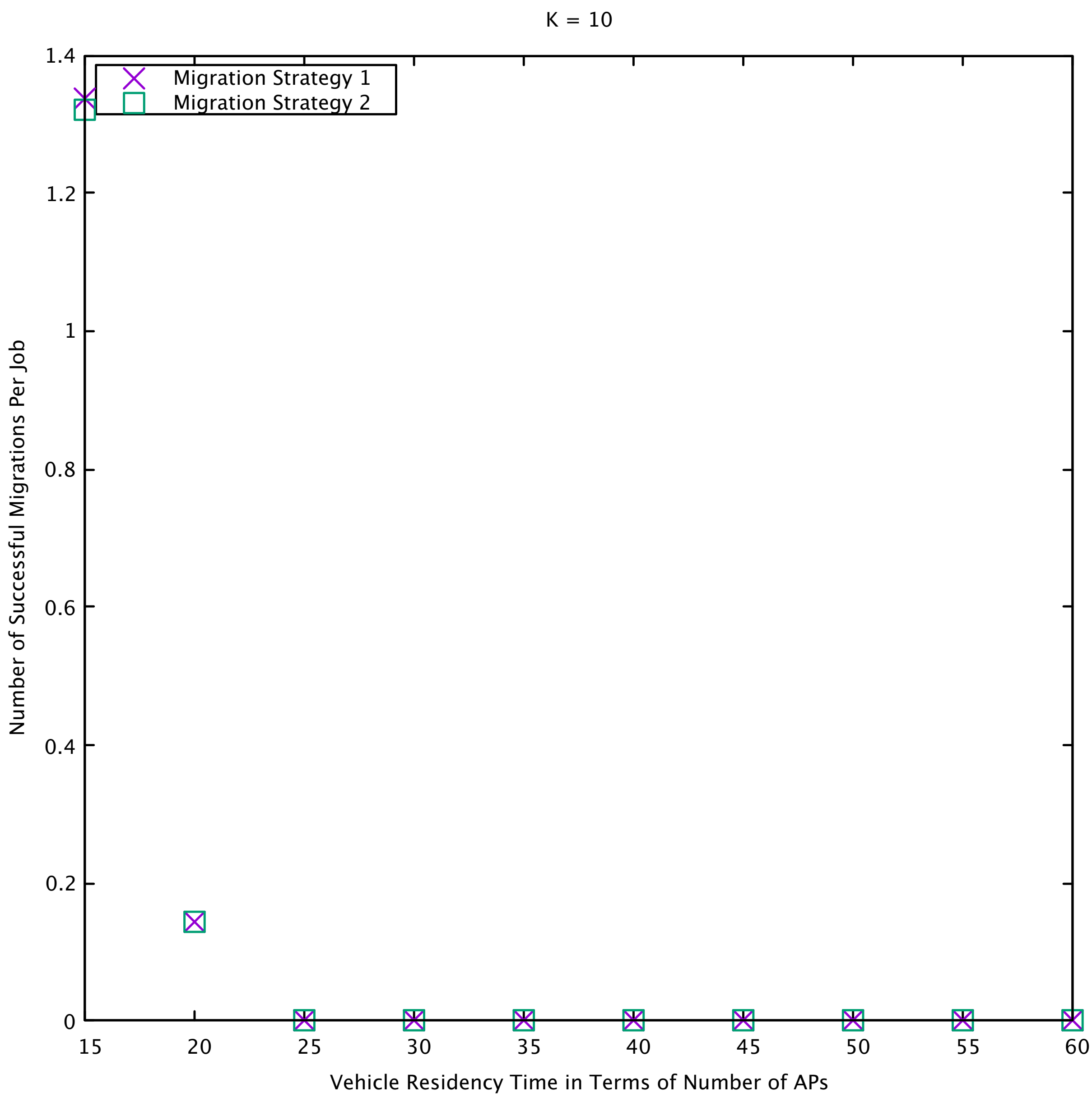


Fig. 59: Average number of successful migrations per job, for different migration strategies given that ten other vehicles are competing to receive a communication slot.

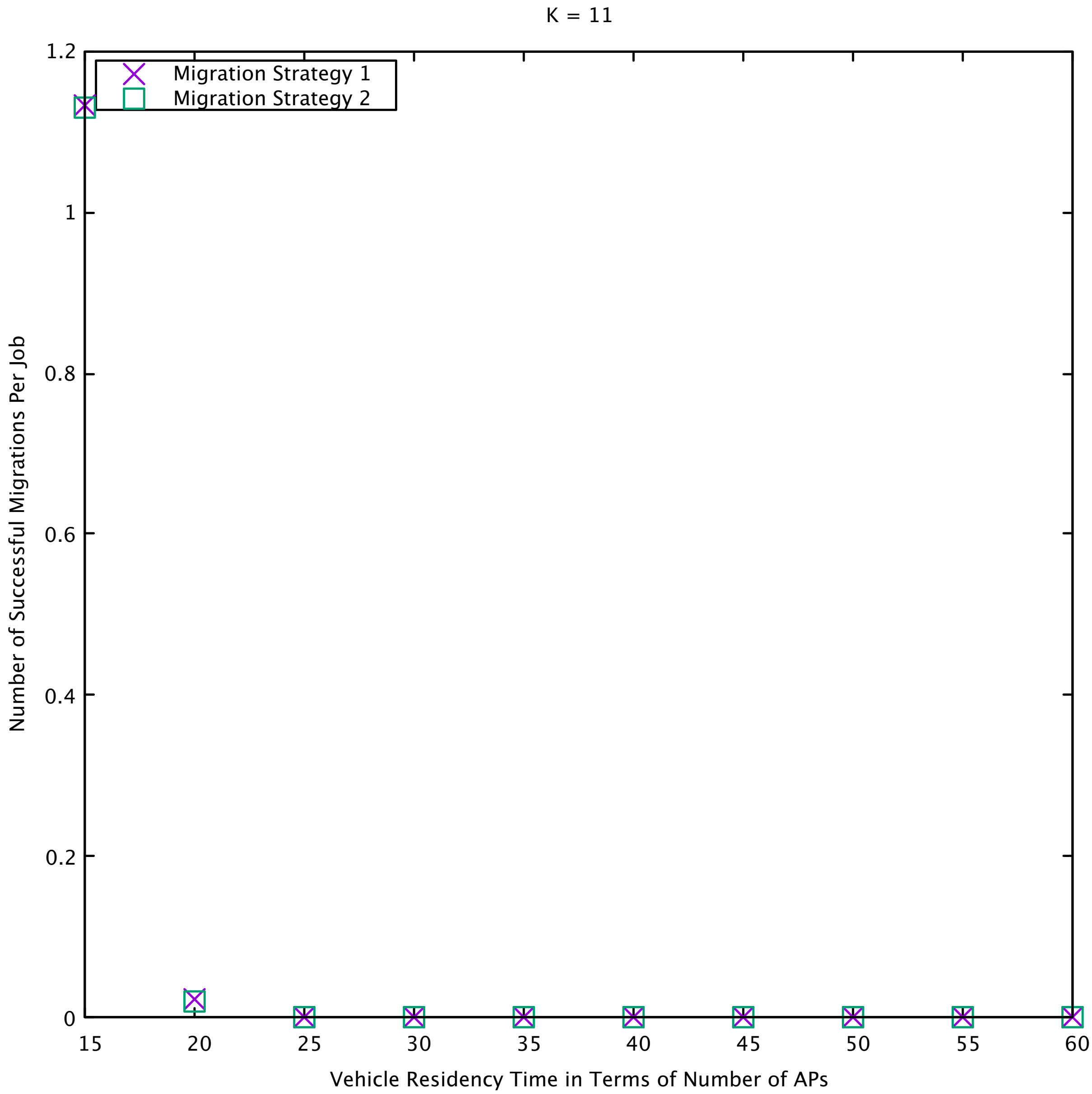


Fig. 60: Average number of successful migrations per job, for different migration strategies given that 11 other vehicles are competing to receive a communication slot.

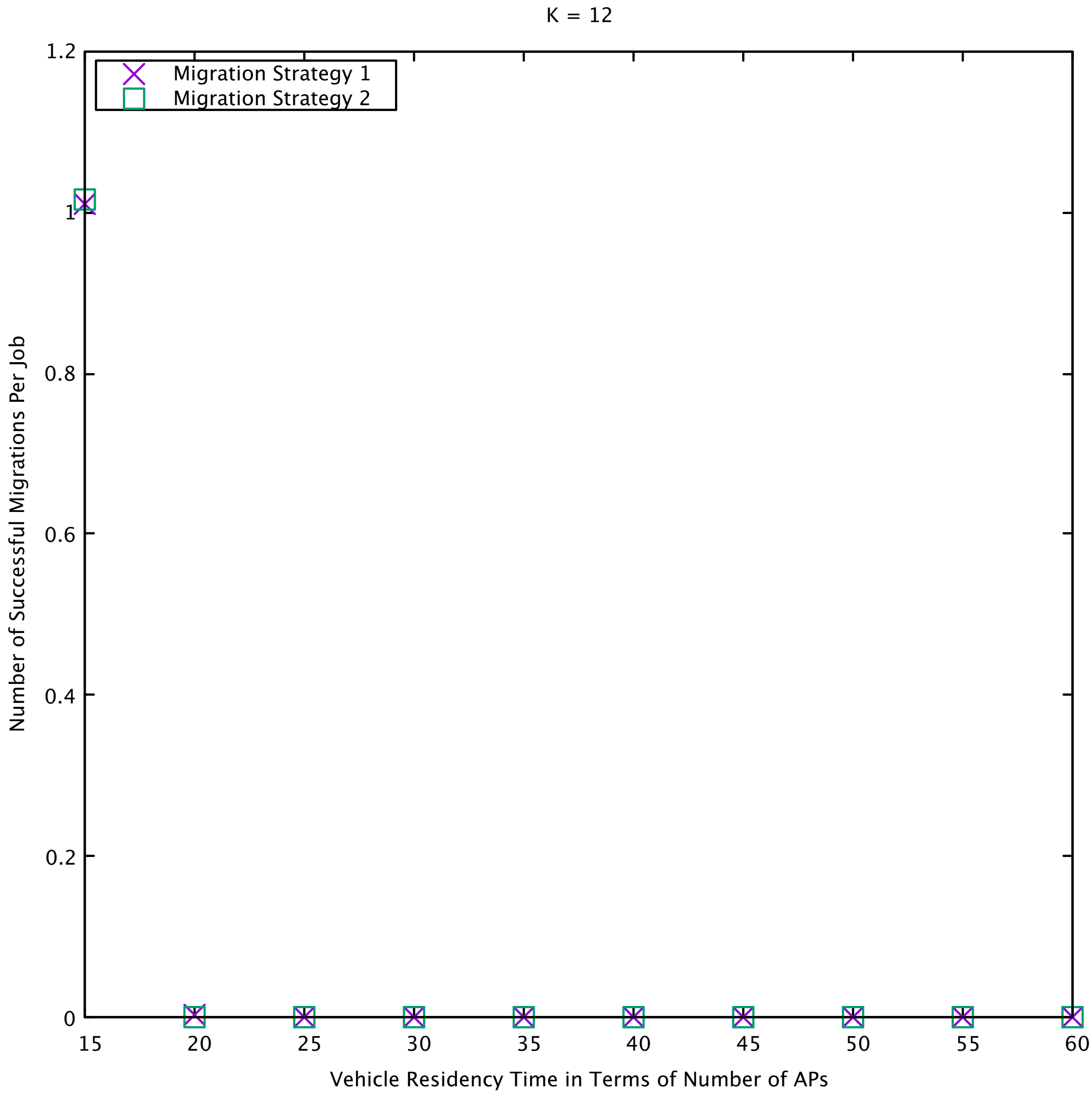


Fig. 61: Average number of successful migrations per job, for different migration strategies given that 12 other vehicles are competing to receive a communication slot.

# CHAPTER 6

# PUTTING THE WORK IN PERSPECTIVE

In this work, we have presented pioneering contributions to the field of dynamic Vehicular Clouds, particularly in the area of the job completion time.

When envisioning a dynamic VC, there are many questions that come to mind, such as, “What are the limitations of a dynamic VC?” or “What is a dynamic VC suitable for?”. In this chapter, we aim at answering these questions and putting our work in perspective.

## 6.1 WHAT ARE THE LIMITATIONS OF THE DYNAMIC VC?

In this section, we discuss some of the limitations of the proposed dynamic VC.

As previously discussed in Chapter 4, the job completion time depends on the available bandwidth in the coverage area of each AP. This indicates that bandwidth plays an imperative role in the performance and employing the full potential of the dynamic VC.

While the DSRC technology is a potential solution to support the communications between the vehicles and the roadside APs, and can provide low-latency, secure transmissions, fast network acquisition, and reliability in adverse weather conditions, however, we should not ignore the limitations of it.

One of the drawbacks of the DSRC, is that compared to the recent technologies such as 5G, it can support less data transmission rate and throughput.

Another limitations originates in the “short-range” characteristics of the DSRC technology. For instance, in the proposed dynamic VC, when a vehicle moves with a high speed (e.g., free-flow speed) on the highway and under the coverage area of an AP, it stays in the coverage area for only a couple of seconds, which appropriately limits the applicable time to download or upload a job.

Such limitations trigger the consideration of investing in new communication technologies to enhance the performance of the dynamic VC.

## 6.2 WHAT ARE SOME INTERESTING FINDINGS?

In this section, we present some thought-provoking findings that are worth noting.

When investigating the job completion time in a dynamic VC, the impact of a vehicle’s speed is an interesting parameter to take a closer look at.

As discussed in Chapter 3, it is clear that the speed of a vehicle is a function of the traffic density. As the the traffic density under the coverage of an AP increases, the vehicle's speed decreases, which hints that the vehicle spends more time under the coverage of the AP and has more time to download or upload a job.

As the traffic density under the coverage area of an AP increases, the number of vehicles that compete with each other to receive a communication slot in a frame increases and the probability of the slot allocation decreases, but interestingly, when the number of the successful vehicles decreases, the available bandwidth is divided between fewer successful vehicles.

As discussed in Chapter 5, the download time of a job affects the total job completion time. The number of APs that the full coverage is used to download or upload the job is a function of many parameters, not only the speed of the vehicle; therefore, the job's download time does not necessarily increase in all cases as the traffic density increases.

It is also interesting to note that as discussed in Chapter 4, we found that the total number of frames in which the vehicle has to compete, in order to complete the download or upload of a job, given that in each frame other vehicles are also competing for a slot is independent of the success probability. In other words, this criterion does not depend on the bidding policy that the vehicle chooses, however it does depend on the size of the job, the available payload, and the number of other vehicles in the coverage area that are competing for a communication slot.

## 6.3 WHAT IS A DYNAMIC VC SUITABLE FOR?

In this section we discuss the type of workloads most suited for running on the dynamic VC.

We can not say what jobs are exactly fit for the dynamic VC in the future. The reason for this is, as the technology advances, the non-addressed challenges in the current technologies may be addressed in the future. However, our analysis and claims are based on our best knowledge of the current everyday used technologies.

The random nature of vehicles' arrival and departure creates stochastic environment in terms of resource availability. Considering this characteristic, jobs that require non-preemptive scheduling are less preferable candidates for VCs. These types of jobs requiring non-preemptive scheduling can be assigned to the dynamic VC in the events that the residency time of the vehicles is sufficiently larger than job duration. On the other hand, jobs that are preemptive in terms of scheduling are good candidates for running on the dynamic

VC.

In terms of dependencies, jobs can be classified to job with dependencies and without dependencies. In jobs with dependencies the execution starts after all job's dependents have been completed. Because of the VC's stochastic nature, jobs without dependencies can be a good fit for the dynamic VC.

In another type of classification, flexible jobs in terms of deadline or tardiness are appropriate candidates to be executed in the dynamic VC. With the current technologies presented and the proposed architecture, jobs with strict deadlines are less preferable to be considered. The dynamic VC is not suitable for the workload that requires real-time computing. The real-time computing should not be misunderstood with the high-performance computing or compute-intensive applications. Jobs that are compute bound are good candidates for the dynamic VC.

# CHAPTER 7

# CONCLUDING REMARKS AND DIRECTIONS FOR FUTURE WORK

The main goal of this chapter is to discuss our fundamental contributions, the problems that we have addressed, and the solutions that we have proposed in this dissertation. We then will point out some of the features that will be implemented as part of future work.

## 7.1 CONCLUDING REMARKS

Recently a new concept of Vehicular Cloud has been introduced and motivated by the compelling success and popularity of cloud computing, and the understanding that the current and future vehicles are equipped with powerful computers, transceivers, and sensing devices and employing these resources in a meaningful way has compelling economic and environmental impacts.

Vehicular Cloud provides an opportunity so that the owners of vehicles can rent their excess computational and storage resources to customers in a similar approach to companies and corporations that rent out their excess resources in exchange for benefits.

Nowadays, millions of vehicles are moving on roads or are parked in parking lots every day. While the purpose of static Vehicular Cloud is to employ the under-utilized resources of the stationary (e.g., parked) vehicles, dynamic Vehicular Cloud harnesses the resources of moving vehicles.

In this thesis, we have focused on the computational capabilities of the vehicles that are moving on highways.

We have proposed an architecture for dynamic Vehicular Cloud, consisting of vehicles moving on the highways and multiple communication stations installed along the highway, and investigated the feasibility of such systems. The dynamic Vehicular Cloud system is based on two-way communications between vehicles and the stations.

We have proposed and investigated a communication model and frame structure for V2I communications in dynamic Vehicular Clouds.

As in conventional clouds, job completion time is one of the crucial quantitative performance figures of merit.

Vehicular Cloud, is characterized by resource volatility. As vehicles enter, new computing resources become available, and when vehicles depart, often unexpectedly, they take their resources with them, creating a volatile environment where the task of reasoning about job completion time, becomes very challenging.

In this thesis, we have discussed the job completion time in dynamic Vehicular Clouds, and have offered easy-to-compute approximations of job completion time for jobs of different sizes that can be downloaded by vehicles under one or several stations.

We have also investigated the feasibility of job migration in dynamic Vehicular Clouds. As discussed, the vehicles, which are the VC's computational resources, arrive and depart at random times. In other words, the time that each computational resource can be employed in the VC is random, limited, and not permanent, and as a result, this characteristic may cause failure in executing jobs and interruptions in the ongoing services.

To handle these interruptions, once a vehicle is ready to leave the Vehicular Cloud if the vehicle is running a job, the job and all intermediate data stored by the departing vehicle must be migrated to an available vehicle in the VC. In this thesis, we have investigated job migration in dynamic Vehicular Cloud and proposed different job migration strategies to reduce the job completion time's overhead and improve the performance and reliability of the system.

## 7.2 DIRECTION FOR FUTURE WORK

As cloud providers work on improving the quality of the services that they provide, the customers of cloud, expect more accelerated responses, scalability, flexibility, and lower latency from the cloud services.

As part of our future work, we plan to continue improving the current Vehicular Cloud model by developing new solutions to improve the system's performance and reliability.

In our future work, we look more into additional improved job migration strategies to reduce the overhead of the job completion time in dynamic VCs.

We will investigate Vehicular Cloud models in which a user may execute several parallel jobs that are dependent on each other, and the speed of the execution for each job may be different. In such situations, the slowest executing job creates a bottleneck and barrier to timely completion of the job. In our future work, we consider investigating such scenarios and dependencies in the dynamic Vehicular Cloud.

Another solution to reduce the latency in VCs is by employing redundancy solutions. A job can be executed simultaneously by multiple vehicles that have different computational capabilities. The VC then waits for and selects the earliest copy of the job that is completed, which then may reduce the overall job completion time and the latency. In our future work, we will investigate such scenarios and solutions comprehensively and thoroughly.

# REFERENCES


[1] S. Olariu and M. C. Weigle, eds., *Vehicular Networks: From Theory to Practice*. CRC Press/Taylor & Francis, March 2009.

[2] S. Olariu, T. Hristov, and G. Yan, "The next paradigm shift: from vehicular networks to vehicular clouds," in *Basagni, S.* et al.*, (Eds), Mobile Ad Hoc Networking Cutting Edge Directions*, pp. 645–700, Wiley and Sons, New York, 2013.

[3] G. Yan, S. Olariu, and D. Popescu, "NOTICE: an architecture for the notification of traffic incidents," *IEEE Intelligent Transportation Systems Magazine*, vol. 4, no. 4, pp. 6–16, 2012.

[4] X. Wang, S. El-Tawab, A. Alhafdhi, M. Almalag, and S. Olariu, "Toward probabilistic data collection in the NOTICE architecture," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 12, pp. 3354–3363, 2016.

[5] M. Abuelela and S. Olariu, "Taking VANET to the clouds," in *Proceedings of the ACM MoMM*, (Paris, France), November 2010.

[6] D. C. Marinescu, *Cloud computing: theory and practice*. Morgan Kaufmann, 2017.

[7] A. Boukerche and R. E. De Grande, "Vehicular cloud computing: Architectures, applications, and mobility," *Computer Networks*, vol. 135, pp. 171–189, 2018.

[8] M. Whaiduzzaman, M. Sookhak, A. Gani, and R. Buyya, "A survey of vehicular cloud computing," *Journal of Network and Computer Applications*, vol. 40, pp. 325–344, 2014.

[9] M. Eltoweissy, S. Olariu, and M. Younis, "Towards autonomous vehicular clouds," in *Proceedings of AdHocNets'2010*, (Victoria, BC, Canada), August 2010.

[10] L. A. Barroso, J. Clidaras, and U. Hölzle, "The datacenter as a computer: An introduction to the design of warehouse-scale machines," *Synthesis lectures on computer architecture*, vol. 8, no. 3, pp. 1–154, 2013.

[11] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. H. Katz, A. Konwinski, G. Lee, D. A. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, "Above the clouds: A berkeley view of cloud computing," in *Technical Report No. UCB/EECS-2009-28*, (U.C. Berkeley), February 2009.

[12] D. C. Marinescu, A. Paya, J. P. Morrison, and S. Olariu, "An approach for scaling cloud resource management," *Cluster Computing*, vol. 20, no. 1, pp. 909–924, 2017.

[13] National Institute of Standards and Technology (NIST), "NIST definition of cloud computing." `http://csrc.nist.gov/groups/SNS/cloud-computing/cloud-def-v15.doc`, October 2009.

[14] A. Bento and R. Bento, "Cloud computing: A new phase in information technology management," *Journal of Information Technology Management*, vol. 22, no. 1, pp. 39–46, 2011.

[15] R. Buyya, C. Vecchiola, and S. T. Selvi, *Mastering cloud computing: foundations and applications programming.* Newnes, 2013.

[16] Q. Zhang, L. Cheng, and R. Boutaba, "Cloud computing: state-of-the-art and research challenges," *Journal of Internet Services and Applications*, vol. 1, no. 1, pp. 7–18, 2010.

[17] National Institute of Standards and Technology (NIST), "DRAFT cloud computing synopsis and recommendationsdefinition of cloud computing." `http://csrc.nist.gov/publications/drafts/800-146/Draft-NIST-SP800-146.pdf`, May 2011.

[18] G. DeCandia, D. Hastorun, M. Jampani, G. Kakulapati, A. Lakshman, A. Pilchin, S. Sivasubramanian, P. Vosshall, and W. Vogels, "Dynamo: Amazons highly available key-value store," in *Proc. 23-th ACM Symposium on Operating Systems Principles, (SOSP07)*, (Stevenson, Washington), 14-17 October 2007.

[19] J. L. Hennessy and D. A. Patterson, *Computer architecture: a quantitative approach.* Elsevier, 2011.

[20] P. Ghazizadeh, R. Mukkamala, and S. Olariu, "Data integrity evaluation in cloud database-as-a-service," in *2013 IEEE Ninth World Congress on Services*, pp. 280–285, 2013.

[21] H. Krcmar, R. Reussner, and B. Rumpe, *Trusted cloud computing.* Springer, 2014.

[22] Amazon Inc., "Amazon web services." `http://aws.amazon.com`, 2018.

[23] C. Baun, M. Kunze, J. Nimis, and S. Tai, *Cloud Computing: Web-Based Dynamic IT Services.* Springer Science & Business Media, 2011.

[24] Google, Inc., "Google app engine." `http://code.google.com/appengine/`, 2010.

[25] Microsoft Corporation, "Windows azure." `http://www.microsoft.com/windowazure/`, 2018.

[26] IBM Inc., "IBM smart cloud." `http://www.ibm.com/cloud-computing`, 2017.

[27] S. Olariu, I. Khalil, and M. Abuelela, "Taking VANET to the clouds," *International Journal of Pervasive Computing and Communications*, vol. 7, pp. 7–21, February 2011.

[28] S. Olariu, M. Eltoweissy, and M. Younis, "Towards autonomous vehicular clouds," *ICST Transactions on Mobile Communications and Applications*, vol. 11, no. 7-9, pp. 1–11, 2011.

[29] P. Ghazizadeh, "Resource allocation in vehicular cloud computing." PhD Thesis, Old Dominion University, July 2014.

[30] R. Florin, P. Ghazizadeh, A. Ghazi Zadeh, and S. Olariu, "Enhancing dependability through redundancy in military vehicular clouds," in *Proceedings of IEEE MILCOM'2015*, (Tampa, Florida), October, 26-28 2015.

[31] S. Arif, S. Olariu, J. Wang, G. Yan, W. Yang, and I. Khalil, "Datacenter at the airport: Reasoning about time-dependent parking lot occupancy," *IEEE Transactions on Parallel and Distributed Systems*, vol. 23, no. 11, pp. 2067–2080, 2012.

[32] R. Hussain, F. Abbas, J. Son, and H. Oh, "TIaaS: Secure cloud-assisted traffic information dissemination in vehicular ad hoc networks," in *Cluster, Cloud and Grid Computing (CCGrid), 2013 13th IEEE/ACM International Symposium on*, pp. 178–179, May 2013.

[33] A. Ghazizadeh and S. Olariu, "Vehicular clouds: A survey and future directions," in *Cloud Computing for Optimization: Foundations, Applications, and Challenges*, pp. 435–463, Springer, 2018.

[34] D. Lu, Z. Li, D. Huang, X. Lu, Y. Deng, A. Chowdhary, and B. Li, "Vc-bots: a vehicular cloud computing testbed with mobile robots," in *Proc. First International Workshop on Internet of Vehicles and Vehicles of Internet*, pp. 31–36, ACM, 2016.

[35] R. Florin, S. Abolghasemi, A. Ghazi Zadeh, and S. Olariu, “Big Data in the parking lot,” in *Big Data Management and Procesing* (K.-C. Li, H. Jiang, and A. Zomaya, eds.), ch. 21, pp. 425–449, Boca Raton, Florida: Taylor and Francis, 2017.

[36] L. Gu, D. Zeng, and S. Guo, “Vehicular cloud computing: A survey,” in *Proc. IEEE Globecom Workshops*, pp. 403–407, Dec 2013.

[37] P. B. Goes, H. Kawashima, and U. Sumita, “Analysis of a new thruway communication system with discrete minimal zones,” *IEEE Transactions on Communications*, vol. 40, no. 4, pp. 754,764, 1992.

[38] J. Ott and D. Kutscher, “Drive-thru internet: Ieee 802.11 b for automobile users,” in *IEEE INFOCOM 2004*, vol. 1, IEEE, 2004.

[39] J. Ott and D. Kutscher, “A disconnection-tolerant transport for drive-thru internet environments,” in *Proceedings IEEE 24th Annual Joint Conference of the IEEE Computer and Communications Societies.*, vol. 3, pp. 1849–1862, IEEE, 2005.

[40] W. L. Tan, W. C. Lau, and O. Yue, “Modeling resource sharing for a road-side access point supporting drive-thru internet,” in *Proceedings of the sixth ACM international workshop on VehiculAr InterNETworking*, pp. 33–42, 2009.

[41] D. Hadaller, S. Keshav, T. Brecht, and S. Agarwal, “Vehicular opportunistic communication under the microscope,” in *Proceedings of the 5th international conference on Mobile systems, applications and services*, pp. 206–219, 2007.

[42] R. Yu, Y. Zhang, S. Gjessing, W. Xia, and K. Yang, “Toward cloud-based vehicular networks with efficient resource management,” *IEEE Network*, vol. 27, no. 5, pp. 48–55, 2013.

[43] V. Bychkovsky, B. Hull, A. Miu, H. Balakrishnan, and S. Madden, “A measurement study of vehicular internet access using in situ wi-fi networks,” in *Proceedings of the 12th annual international conference on Mobile computing and networking*, pp. 50–61, 2006.

[44] N. Banerjee, M. D. Corner, D. Towsley, and B. N. Levine, “Relays, base stations, and meshes: enhancing mobile networks with infrastructure,” in *Proceedings of the 14th ACM international conference on Mobile computing and networking*, pp. 81–91, 2008.

[45] I. Jang, S. Choo, M. Kim, S. Pack, and G. Dan, "The software-defined vehicular cloud: A new level of sharing the road," *ieee vehicular technology magazine*, vol. 12, no. 2, pp. 78–88, 2017.

[46] N. Kumar, R. Iqbal, S. Misra, and J. J. Rodrigues, "Bayesian coalition game for contention-aware reliable data forwarding in vehicular mobile cloud," *Future Generation Computer Systems*, vol. 48, pp. 60–72, 2015.

[47] K. Mershad and H. Artail, "Finding a star in a vehicular cloud," *IEEE Intelligent transportation systems magazine*, vol. 5, no. 2, pp. 55–68, 2013.

[48] Z. Jiang, S. Zhou, X. Guo, and Z. Niu, "Task replication for deadline-constrained vehicular cloud computing: Optimal policy, performance analysis, and implications on road traffic," *IEEE Internet of Things Journal*, vol. 5, no. 1, pp. 93–107, 2017.

[49] C. Pahl, A. Brogi, J. Soldani, and P. Jamshidi, "Cloud container technologies: a state-of-the-art review," *IEEE Transactions on Cloud Computing*, vol. DOI 10.1109/TCC.2017.2702586, pp. 1–14, 05 2017.

[50] "Docker containers." `https://www.docker.com/resources/what-container/`, 2019.

[51] B. Baron, M. Campista, P. Spathis, L. H. Costa, M. Dias de Amonim, O. C. Duarte, G. Pujolle, and Y. Viniotis, "Virtualizing vehicular node resources: Feasibility study of virtual machine migration," *Vehicular Communications*, vol. 4, pp. 39–46, April 2016.

[52] D. Kapil, E. S. Pilli, and R. Joshi, "Live virtual machine migration techniques: Survey and research challenges," in *Proc. 3rd International IEEE Advance Computing Conference (IACC)*, (Ghaziabad, India), February 2013.

[53] P. Kaur and R. A., "Virtual machine migration in cloud computing," *International Journal of Grid Distribution Computing*, vol. 8, no. 5, pp. 337–342, 2015.

[54] T. K. Refaat, B. Kantarci, and H. T. Mouftah, "Virtual machine migration and management for vehicular clouds," *Vehicular Communications*, vol. 4, pp. 47–56, 2016.

[55] P. Ghazizadeh, R. Mukkamala, and S. El-Tawab, "Scheduling in vehicular cloud using mixed integer linear programming," in *Proceedings of the first international workshop on Mobile sensing, computing and communication*, pp. 7–12, 2014.

[56] B. Baron, M. Campista, P. Spathis, L. H. M. Costa, M. D. de Amorim, O. C. M. Duarte, G. Pujolle, and Y. Viniotis, “Virtualizing vehicular node resources: Feasibility study of virtual machine migration,” *Vehicular Communications*, vol. 4, pp. 39–46, 2016.

[57] C. Harrison and I. A. Donnelly, “The theory of smart cities,” in *Proceedings of the 55th Annual Meeting of the International Society for the Systems Sciences, (ISSS'2011), Hull, U.K.*, 2011.

[58] P. Olson, “Why Google's Waze is trading user data with local governments.” `http://www.forbes.com/sites/parmyolson/2014/07/07/why-google-waze-helps-local-governments-track-its-users/`, July 7, 2014.

[59] USDOT, “The smart/connected city and its implications for connected transportation,” 2014.

[60] Y. A. Zuev, “On the estimation of efficiency of voting procedures,” *Theory of Probability & Its Applications*, vol. 42, no. 1, pp. 73–81, 1998.

[61] Y. Zuev and S. Ivanov, “The voting as a way to increase the decision reliability,” *Journal of the Franklin Institute*, vol. 336, no. 2, pp. 361 – 378, 1999.

[62] L. F. Sarmenta, “Sabotage-tolerance mechanisms for volunteer computing systems,” *Future Generation Computer Systems*, vol. 18, no. 4, pp. 561 – 572, 2002. Best papers from Symp. on Cluster Computing and the Grid (CCGrid2001).

[63] K. Watanabe, M. Fukushi, and S. Horiguchi, “Expected-credibility-based job scheduling for reliable volunteer computing,” *IEICE transactions on information and systems*, vol. 93, no. 2, pp. 306–314, 2010.

[64] K. Nakano, S. Olariu, and J. L. Schwing, “Broadcast-efficient protocols for mobile radio networks,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 10, no. 12, pp. 1276–1289, 1999.

[65] K. Nakano and S. Olariu, “Randomized leader election protocols in radio networks with no collision detection,” *International Symposium on Algorithms and Computation, (ISAAC'2000)*, pp. 362–373, August 2000.

[66] R. Elmasri and S. B. Navathe, *Fundamentls of Database Systems*. Addison-Wesley, 6-th edition, 2011.

[67] S. Toueg and O. Babaoglu, "On the optimum checkpoint selection problem," *SIAM Journal of Computing*, vol. 13, pp. 630–649, August 1984.

[68] P. Ghazizadeh, S. Olariu, A. Ghazi Zadeh, and S. El-Tawab, "Towards fault-tolerant job assignment in vehicular cloud," *Proceecings IEEE SCC*, pp. 17–24, June 2015.

[69] P. Ghazizadeh, R. Florin, A. Ghazi Zadeh, and S. Olariu, "Reasoning about the Mean Time to Failure in vehicular clouds," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, pp. 751–761, March 2016.

[70] R. Florin, A. Ghazizadeh, P. Ghazizadeh, S. Olariu, and D. C. Marinescu, "Enhancing reliability and availability through redundancy in vehicular clouds," *IEEE Transactions on Cloud Computing*, pp. 1–1, 2019.

[71] P. Ghazizadeh, R. Mukkamala, and R. Fathi, "Modeling and predicting fault tolerance in vehicular cloud computing," in *2015 International Conference on Computing and Network Communications (CoCoNet)*, pp. 395–400, 2015.

[72] R. Florin, A. Ghazi Zadeh, P. Ghazizadeh, and S. Olariu, "Towards approximating the mean time to failure in vehicular clouds," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 7, pp. 2045–2054, 2018.

[73] R. Florin, P. Ghazizadeh, A. Ghazi Zadeh, S. El-Tawab, and S. Olariu, "Reasoning about job completion time in vehicular clouds," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, pp. 1762–1771, 2017.

[74] R. Florin, P. Ghazizadeh, A. Ghazi Zadeh, R. Mukkamala, and S. Olariu, "A tight estimate of job completion time in vehicular clouds," *IEEE Transactions on Cloud Computing*, pp. 1–1, 2018.

[75] P. Ghazizadeh and R. Fathi, "A propagation based model for communication in vehicular cloud," in *2018 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, pp. 295–299, 2018.

[76] K. Bachman, "Study: Teens still big mall shoppers." `http://www.adweek.com/news/advertising-branding/study-teens-still-big-mallshoppers-112487`, accessed January 18 2017.

[77] "Teen marketplace." `http://www.jcdecauxna.com/sites/default/files/assets/mall/documents/studies/TeensFact.pdf`, accessed January 18 2017.

[78] "The mall phenomenon." `http://www.jcdecauxna.com/sites/default/files/assets/mall/documents/studies/mallphenomenon.pdf`, accessed January 18 2017.

[79] H. Hu and C. R. Jasper, "Men and women: A comparison of shopping mall behavor," *Journal of Shopping Center Research*, vol. 11, no. 1/2, pp. 113–123, 2004.

[80] H. Moussanif, I. Khalil, and S. Olariu, "Cooperation as a Service in VANET: Implementation and simulation results," *Mobile Information Systems*, vol. 8, no. 2, pp. 153–172, 2012.

[81] US Department of Transportation, "Standard Specification for Telecommunications and Information Exchange Between Roadside and Vehicle Systems - 5 GHz Band Dedicated Short Range Communications (DSRC) Medium Access Control (MAC) and Physical Layer (PHY) Specifications." ASTM E2213-03, August 2003.

[82] R. S. Raw, L. Loveleen, A. Kumar, A. Kadam, and N. Singh, "Analysis of message propagation for intelligent disaster management through vehicular cloud network," in *Proc. 2-nd ACM International Conference on Information and Communication Technology for Competitive Strategies, (ICTCS'2016)*, March 2016.

[83] M. Chaqfeh, M. Nader, I. Jawhar, and J. Wu, "Vehicular cloud data collection for intelligent transportation systems," in *Proc. IEEE International Conference on Smart Cloud Networks & Systems, (SCNS'2016)*, December 2016.

[84] S. K. Sood *et al.*, "Smart vehicular traffic management: An edge cloud centric iot based framework," *Internet of Things*, p. 100140, 2019.

[85] D. Kwak, R. Liu, D. Kim, B. Nath, and L. Iftode, "Seeing is believing: Sharing real-time visual traffic information via vehicular clouds," *IEEE Access*, vol. 4, pp. 3617–3631, 2016.

[86] J. Saleem, N. Jamil, M. Naseem, A. Jahan, and S. Javed, "Vehicular cloud computing (vcc),"

[87] G. Yan, D. Wen, S. Olariu, and M. C. Weigle, "Security challenges in vehicular cloud computing," *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 1, pp. 284–294, 2013.

[88] G. Yan, S. Olariu, J. Wang, and S. Arif, “Towards providing scalable and robust privacy in vehicular networks,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 7, pp. 1896–1906, 2014.

[89] C. Huang, R. Lu, H. Zhu, H. Hu, and X. Lin, “PTVC: Achieving privacy-preserving trust-based verifiable vehicular cloud computing,” in *Proc. of IEEE Global Communications Conference (GLOBECOM'2016)*, pp. 1–6, IEEE, 2016.

[90] A. Ghazizadeh, P. Ghazizadeh, R. Mukkamala, and S. Olariu, “Towards approximating expected job completion time in dynamic vehicular clouds,” in *2019 IEEE 12th International Conference on Cloud Computing (CLOUD)*, pp. 481–483, July 2019.

[91] H. Wang, Q.-Y. Chen, and D. Ni, “Logistic modeling of the equilibrium speed-density relationship,” *Transportation Research Part A: Policy and Practice*, vol. 45, no. 6, pp. 554–566, 2011.

[92] A. Ghazizadeh, P. Ghazizadeh, and S. Olariu, “Reasoning about a communication protocol for vehicular cloud computing systems,” in *2019 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, pp. 214–219, 2019.

[93] A. Ghazizadeh and P. Ghazizadeh, “Design and analysis of a communication protocol for dynamic vehicular clouds in smart cities,” in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pp. 1–6, 2018.

[94] J. Carbonell, S. Klein, D. Miller, M. Steinbaum, T. Grassiany, and J. Frey, “Context-based machine translation,” in *Proc. 7-th Conference of the Association for Machine Traslation in the Americas*, (Cambridge, August), pp. 19–28, 2006.

[95] P. B. Callahan and R. Kosaraju, “A decomposition of multidimensional point sets with applications to $k$-nearest neighbors and $n$-body potential fields,” *Journal of the ACM*, vol. 42, pp. 67–90, January 1995.

[96] T. Schlick, *Molecular modeling and simulation: an interdisciplinary guide: an interdisciplinary guide*, vol. 21. Springer Science & Business Media, 2010.

[97] A. Leach and L. B. Kier, “Molecular modeling: principles and applications,” *Journal of Medicinal Chemistry*, vol. 40, no. 18, p. 2969, 1997.

[98] G. S. Fishman, *A First Course in Monte Carlo.* Thomson Brooks/Cole, Belmont, CA, 2006.

[99] M. K. Lindell and C. S. Prater, "Critical behavioral assumptions in evacuation time estimate analysis for private vehicles: Examples from hurricane research and planning," *Journal of Urban Planning and Development*, vol. 133, pp. 18–29, March 2007.

[100] P. Murray-Tuite and H. Mahmassani, "Transportation network evacuation planning with household activity interactions," *Transportation Research Record: Journal of Transportation Research Board*, vol. 1894, pp. 150–159, 2004.

[101] M. Satyanarayanan, P. Bahl, R. Caceres, and N. Davies, "The case for VM-based cloudlets in mobile computing," *Pervasive Computing, IEEE*, vol. 8, no. 4, pp. 14–23, 2009.

[102] C. Forbes, M. Evans, N. Hastings, and B. Peacock, *Statistical distributions.* John Wiley & Sons, 2011.

[103] G. S. Grimmett *et al.*, *Probability and random processes.* Oxford university press, 2020.

[104] D. Romik, "Stirling's approximation for n!: the ultimate short proof?," *The American Mathematical Monthly*, vol. 107, no. 6, pp. 556–557, 2000.

[105] J. Blitzstein and J. Hwang, *Introduction to Probability, Second Edition.* Chapman & Hall/CRC Texts in Statistical Science, CRC Press, 2019.

[106] G. S. Linoff and M. J. Berry, *Data mining techniques: for marketing, sales, and customer relationship management.* John Wiley & Sons, 2011.

[107] A. Kapoor, S. I. Bhat, S. Shidnal, and A. Mehra, "Implementation of iot (internet of things) and image processing in smart agriculture," in *2016 International Conference on Computation System and Information Technology for Sustainable Solutions (CSITSS)*, pp. 21–26, IEEE, 2016.

[108] F. Zhang, G. Liu, X. Fu, and R. Yahyapour, "A survey on virtual machine migration: Challenges, techniques, and open issues," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 2, pp. 1206–1243, 2018.

[109] R. Boutaba, Q. Zhang, and M. F. Zhani, "Virtual machine migration in cloud computing environments: Benefits, challenges, and approaches," in *Communication Infrastructures for Cloud Computing*, pp. 383–408, IGI Global, 2014.

# APPENDIX

Consider the semi-closed interval $[0, \infty)$ ruled into finite segments of length $x > 0$ and a sliding window (window, for short) of length $Z > 0$ with its left endpoint at $t \geq 0$. We say that a segment defined above is *completely covered* (covered, for short) by the window if both its endpoints lies inside the window. For example, when $t = 0$, $\lfloor \frac{Z}{x} \rfloor$ of the segments are completely covered by it.

The sliding window is moving continuously from left to right, covering various segments as it moves. For $t$ arbitrary, we are interested in the expected number of segments covered by the sliding window.

It is clear that when $Z < x$ the answer is 0. When $Z = x$ the answer is 0 with probability 1. This follows because the set of points in $[0, \infty)$ where the window overlaps a segment of size $x$ is countable and thus has measure 0. With this in mind, in the remainder of this Appendinx we will assume that $Z > x$.

**Lemma 1.** *If $Z > x$ then the expected number of segments covered is $\frac{Z}{x} - 1$.*

*Proof:* Observe that the stochastic process that keeps track of the number $C$ of segments covered by the window is a *renewal process* and, as a consequence, for the purpose of studying the expected number of segments covered, we can restrict ourselves to $t \in [0, x)$. As before, we notice that when $t = 0$, the window covers $\lfloor \frac{Z}{x} \rfloor$ segments. However, for an arbitrarily small $\epsilon > 0$, when $t = \epsilon$, the number of such segments drops to $\lfloor \frac{Z}{x} \rfloor - 1$ and it stays at that level until $t = x\lfloor \frac{Z}{x} \rfloor + x - Z$.

Next, as $t$ moves over the interval $(x\lfloor \frac{Z}{x} \rfloor + x - Z, x)$, the number of covered segments increases to $\lfloor \frac{Z}{x} \rfloor$. Assuming that $t$ is *uniformly* distributed in $[0, x)$, the expected number, $E[C]$, of covered segments is

$$
\begin{aligned}
E[C] &= \int_{u=0}^{x\lfloor \frac{Z}{x} \rfloor + x - Z} \left( \left\lfloor \frac{Z}{x} \right\rfloor - 1 \right) \frac{du}{x} + \int_{u=x\lfloor \frac{Z}{x} \rfloor + x - Z}^{x} \left\lfloor \frac{Z}{x} \right\rfloor \frac{du}{x} \\
&= \frac{1}{x} \left[ \left( x \left\lfloor \frac{Z}{x} \right\rfloor + x - Z \right) \left( \left\lfloor \frac{Z}{x} \right\rfloor - 1 \right) + \left( Z - x \left\lfloor \frac{Z}{x} \right\rfloor \right) \left\lfloor \frac{Z}{x} \right\rfloor \right] \\
&= \frac{1}{x} \left[ x \left\lfloor \frac{Z}{x} \right\rfloor^2 - x \left\lfloor \frac{Z}{x} \right\rfloor + x \left\lfloor \frac{Z}{x} \right\rfloor - x - Z \left\lfloor \frac{Z}{x} \right\rfloor + Z + Z \left\lfloor \frac{Z}{x} \right\rfloor - x \left\lfloor \frac{Z}{x} \right\rfloor^2 \right] \\
&= \frac{1}{x} (Z - x) \\
&= \frac{Z}{x} - 1.
\end{aligned}
\tag{43}
$$

■

# VITA

Aida Ghazizadeh
Department of Computer Science
Old Dominion University
Norfolk, VA 23529

## EDUCATION

Ph.D. Computer Science, Old Dominion University, 2020
M.S. Computer Science, Old Dominion University, 2014

## PUBLICATIONS

A complete list of publications is available at `https://scholar.google.com/citations?user=f29yAywAAAAJ&hl=en`

Typeset using LaTeX.