Approximating Expected Job Completion Time in Dynamic Vehicular Clouds

Aida Ghazizadeh*
aghaziza@cs.odu.edu
American Express
Palo Alto, California, USA

Puya Ghazizadeh
ghazizap@stjohns.edu
Division of Computer Science, Mathematics and Science
St. John's University
New York, New York, USA

Ravi Mukkamala
mukka@cs.odu.edu
Department of Computer Science
Old Dominion University
Norfolk, Virginia, USA

Stephan Olariu
olariu@cs.odu.edu
Department of Computer Science
Old Dominion University
Norfolk, Virginia, USA

ABSTRACT

Motivated by the success of 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 compute resources become available while others depart, creating a volatile environment where the task of reasoning about fundamental performance metrics becomes very challenging. Just as in conventional clouds, job completion time ranks high among the fundamental quantitative performance figures of merit. With this in mind, the main contribution of this work is to offer easy-to-compute approximations of job completion time in a dynamic vehicular cloud model 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.

KEYWORDS

Cloud computing, Vehicular clouds, Job completion time, Approximations

ACM Reference Format:
Aida Ghazizadeh, Puya Ghazizadeh, Ravi Mukkamala, and Stephan Olariu. 2021. Approximating Expected Job Completion Time in Dynamic Vehicular Clouds. In *Proceedings of the 19th ACM International Symposium on Mobility Management (MobiWac '21), November 22–26, 2021, Alicante, Spain.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3479241.3486695

*While Aida Ghazizadeh currently works at American Express, she conducted this research while a student in the Department of Computer Science at Old Dominion University. No American Express data were involved in this research.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
MobiWac '21, November 22–26, 2021, Alicante, Spain
© 2021 Association for Computing Machinery.
ACM ISBN 978-1-4503-9079-8/21/11…$15.00
https://doi.org/10.1145/3479241.3486695

1 INTRODUCTION AND MOTIVATION

Cloud Computing (CC), an inspired 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 [8]. In August 2010, inspired by the success and unmistakable promise of conventional CC, Eltoweissy *et al.* [1] introduced Autonomous Vehicular Clouds. One of the defining ways in which VCs differ from CCs is in the ownership of compute resources. While in the case of CCs the compute resources have a single owner, in VCs the ownership of these resources is distributed over a large driver population. As a consequence of the distributed ownership of the compute and storage resources, the VC are highly dynamic. As vehicles enter the VC, fresh compute resources become available; when vehicles leave, often unexpectedly, their resources depart with them, creating a highly dynamic environment. In turn, the dynamically changing availability of compute resources due to vehicles joining and leaving the VC unexpectedly leads to a volatile computing environment where reasoning about system performance becomes challenging [7], [3], [2].

Recent years have seen the emergence of VCs as an active topic of research. Various VC architectures and services were outlined, in terms of desirable qualitative characteristics without any regard to, or credible study of, their feasibility and quantitative performance characteristics. All this is changing now as more and more researchers are turning their attention to quantitative aspects of VCs and of the services they contemplate especially in support of Smart Cities and Smart Communities [6], [4].

The main contribution of this work is to offer easy-to-compute approximations of job completion time in a VC model involving vehicles on a highway (see Section 2) and feasible workloads for this model. 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.

2 THE VC MODEL

In this work, 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 *access points* (APs, for short) deployed along the highway. Referring to Figure 1, 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 1, each AP has a radio coverage area of c meters [5].

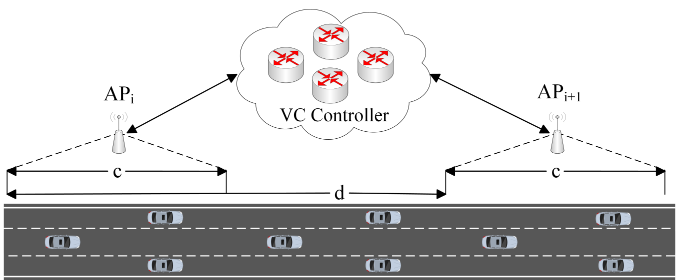

Figure 1: ***Illustrating two consecutive APs and their coverage areas.***

A vehicle can communicate with an AP only when it is in its coverage area. We assume that the APs are placed at the entry and exit ramps along the highway. This assumption implies that a vehicle entering the highway is under the coverage area of an AP. Similarly, a vehicle exits the highway under the coverage area of an AP. Upon entering the highway, 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.

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 APPROXIMATING JOB COMPLETION TIME

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

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. Instead, we only assume $E[T]$ known. 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 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.

Table 1: ***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 that a frame is successful for a vehicle, given that k other vehicles are also competing for slots
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

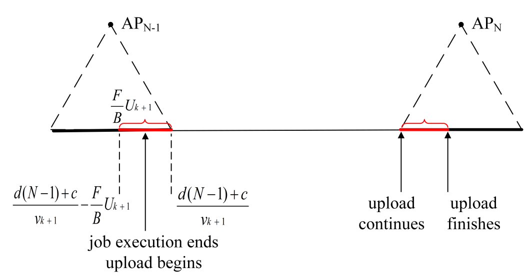

Figure 2: ***Illustrating Case 1***

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 2, assume that job execution terminates under the coverage of AP_{N-1} and that the upload of the results begin in the same the coverage area but finishes under 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:

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

To justify (1), observe that $\frac{F}{B}D_{k+1} + T$ is the time it takes the vehicle to download the container image 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 (1), and using the linearity of expectation yields

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

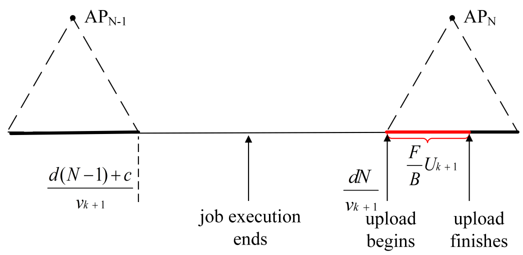

Figure 3: *Illustrating Case 2*

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

Referring to Figure 3, 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 results upon entering the coverage area of AP_N. In this case, it is natural to define the job completion time J_2 as:

$$J_2 = \frac{dN}{v_{k+1}} + \frac{F}{B}U_{k+1}. \tag{3}$$

To justify (3) notice that the vehicle must first physically reach the coverage area of 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 AP_{N-1} and that of AP_N. In other words, N satisfies

$$\frac{(N-1)d + c}{v_{k+1}} < \frac{F}{B}D_{k+1} + T \leq \frac{Nd}{v_{k+1}}. \tag{4}$$

By applying the expectation operator to (3), we obtain

$$E[J_2] = \frac{dE[N]}{v_{k+1}} + \frac{F}{B}E[U_{k+1}]. \tag{5}$$

In order to obtain and expression for $E[J_2]$ we proceed as follows: From (4) by simple algebra we obtain, in stages,

$$\frac{F}{B}D_{k+1} + T < \frac{dN}{v_{k+1}} \leq \frac{F}{B}D_{k+1} + T + \frac{d-c}{v_{k+1}}. \tag{6}$$

Applying the expectation operator to (6) yields

$$\frac{F}{B}E[D_{k+1}] + E[T] \leq \frac{dE[N]}{v_{k+1}} \leq \frac{F}{B}E[D_{k+1}] + E[T] + \frac{d-c}{v_{k+1}}. \tag{7}$$

After adding $\frac{F}{B}E[U_{k+1}]$ throughout in (7) and recalling (3), 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.

$$E[J_2] \approx E[T] + \frac{2F}{B}E[D_{k+1}] + \frac{d-c}{2v_{k+1}}. \tag{8}$$

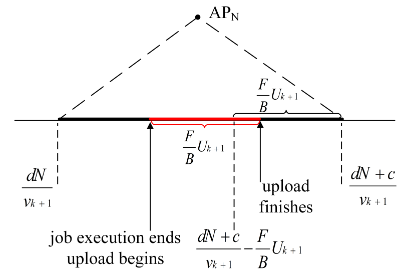

Figure 4: *Illustrating Case 3*

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 4, job execution finishes under the coverage area of AP_N and the results are uploaded under the coverage of AP_N. In this case, it is natural to define the job completion time J_3 as

$$J_3 = \frac{F}{B}[D_{k+1} + U_{k+1}] + T. \tag{9}$$

Upon applying the expectation operator to (9) and using the linearity of expectation we obtain:

$$E[J_3] = E[T] + \frac{2F}{B}E[D_{k+1}]. \tag{10}$$

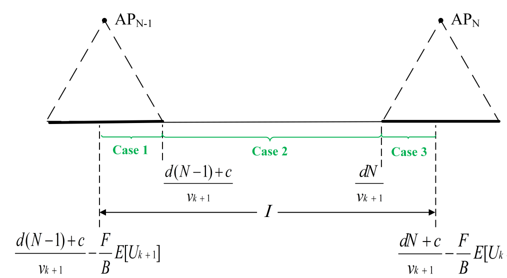

Figure 5: *Illustrating the evaluation of* $E[J]$.

3.1 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. Let π_1, π_2 and π_3 be, respectively, the *limiting* probabilities of Case 1, Case 2 and Case 3 occurring. Using the Law of Total Expectation, the expectation $E[J]$ of job completion time can be computed as

$$E[J] = \pi_1 E[J_1] + \pi_2 E[J_2] + \pi_3 E[J_3]. \tag{11}$$

To evaluate the limiting probabilities π_1, π_2, π_3, consider the time interval $I = \left[\frac{d(N-1)+c}{v_{k+1}} - \frac{F}{B}E[U_{k+1}], \frac{dN+c}{v_{k+1}} - \frac{F}{B}E[U_{k+1}]\right]$ of length $\frac{d}{v_{k+1}}$ and refer to Figure 5. 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 π_1, π_2, π_3 are given by the expressions

$$\pi_1 = \frac{FE[U_{k+1}]v_{k+1}}{Bd}; \ \pi_2 = \frac{d-c}{d}; \ \pi_3 = \frac{c}{d} - \pi_1. \tag{12}$$

Upon replacing the expressions of π_1, π_2, π_3 obtained in (12) back into (11), we obtain our approximation of the job completion time:

$$E[J] = E[T] + \frac{(3d-c)FE[D_{k+1}]}{Bd} + \frac{(d-c)^2}{2dv_{k+1}}. \tag{13}$$

4 SIMULATION MODEL AND RESULTS

In order to validate the analytical results in Section 3, we have run extensive simulations. Specifically, we have compared our simulation results for the job completion times against our predicted values of $E[J]$ derived in 13 in our model. In this section we describe the simulation model and offer the details of our simulation results.

4.1 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. 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 detailed discussion of which is out of the scope of this paper. 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 [9] 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 λ. We vary λ 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. In this paper 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 λ and uniformly distributed on the interval from $\lambda - a$ to $\lambda + a$. Table 2 shows the value of each parameter. Our simulations were developed in house 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.

4.2 Simulation Results

The simulation results for job completion times for k values of 1 to 12 are shown in Figure 6. 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 π_1, π_2, π_3 of (12) are respectively 0.0093, 0.95, and 0.0407. These values match closely the predicted probability values explained in Section 3.

Table 2: Simulation Parameters

Symbol and Description	Value
l (number of lanes)	3
B (available bandwidth)	27×10^6 bps
W (size of the job)	8×10^6 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
v_f (free flow speed)	107.44 kph
k_t (turning point for the speed-density curve)	17.53
θ_1 (scale parameter for speed-density function)	1.8768
θ_2 (parameter which controls the lopsidedness of the curve)	0.0871

5 CONCLUDING REMARKS AND DIRECTIONS FOR FUTURE WORK

It is well known that job completion time is one of the basic performance figures of merit both in both CCs and VCs. In general, predicting job completion time requires full knowledge of the probability distributions of the intervening random variables. In practice, the datacenter manager does not know these distribution functions. Instead, she may have an estimate of the first moment of job execution time, in the absence of any overhead attributable to the VC. The main contribution of this work was to offer easy-to-compute approximations of job completion time in a VC model involving vehicles on a highway. Our extensive simulations have shown that our approximations are very accurate. In future work, we will look at other workloads where the underlying VM cannot be downloaded under the coverage area of a single AP and needs several APs to complete this operation. Also, of interest are scenarios involving short vehicular residency times, where VMs (or containers) need to be migrated to other vehicles.

REFERENCES

[1] M. Eltoweissy, S. Olariu, and M. Younis. 2010. Towards Autonomous Vehicular Clouds. In *Proceedings of AdHocNets'2010*. Victoria, BC, Canada.

[2] R. Florin, A. Ghazizadeh, P. Ghazizadeh, S. Olariu, and D. C. Marinescu. 2019. Enhancing Reliability and Availability Through Redundancy in Vehicular Clouds. *IEEE Transactions on Cloud Computing* (2019), 1–1. https://doi.org/10.1109/TCC.2019.2905590

[3] R. Florin, P. Ghazizadeh, A. Ghazi Zadeh, S. El-Tawab, and S. Olariu. 2018. Towards approximating the mean time to failure in vehicular clouds. *IEEE Transactions on Intelligent Transportation Systems* 18, 7 (2018).

[4] Aida Ghazizadeh. 2020. Towards Dynamic Vehicular Clouds. (2020).

[5] A. Ghazizadeh, P. Ghazizadeh, R. Mukkamala, and S. Olariu. 2019. Towards Approximating Expected Job Completion Time in Dynamic Vehicular Clouds. In *2019 IEEE 12th International Conference on Cloud Computing (CLOUD)*. 481–483. https://doi.org/10.1109/CLOUD.2019.00084

[6] Aida Ghazizadeh and Stephan Olariu. 2018. Vehicular clouds: A survey and future directions. In *Cloud Computing for Optimization: Foundations, Applications, and Challenges*. Springer, 435–463.

[7] P. Ghazizadeh, R. Florin, A. Ghazi Zadeh, and S. Olariu. 2016. Reasoning about the Mean Time to Failure in vehicular clouds. *IEEE Transactions on Intelligent Transportation Systems* 17, 3 (March 2016), 751–761.

[8] D. C. Marinescu. 2017. *Cloud Computing, Theory and Applications* (2-nd ed.). Morgan Kaufman, Elsevier.

[9] H. Wang, Q.-Y. Chen, and D. Ni. 2011. Logistic modeling of the equilibrium speed-density relationship. *Transportation Research Part A: Policy and Practice* 45, 6 (2011), 554–566.

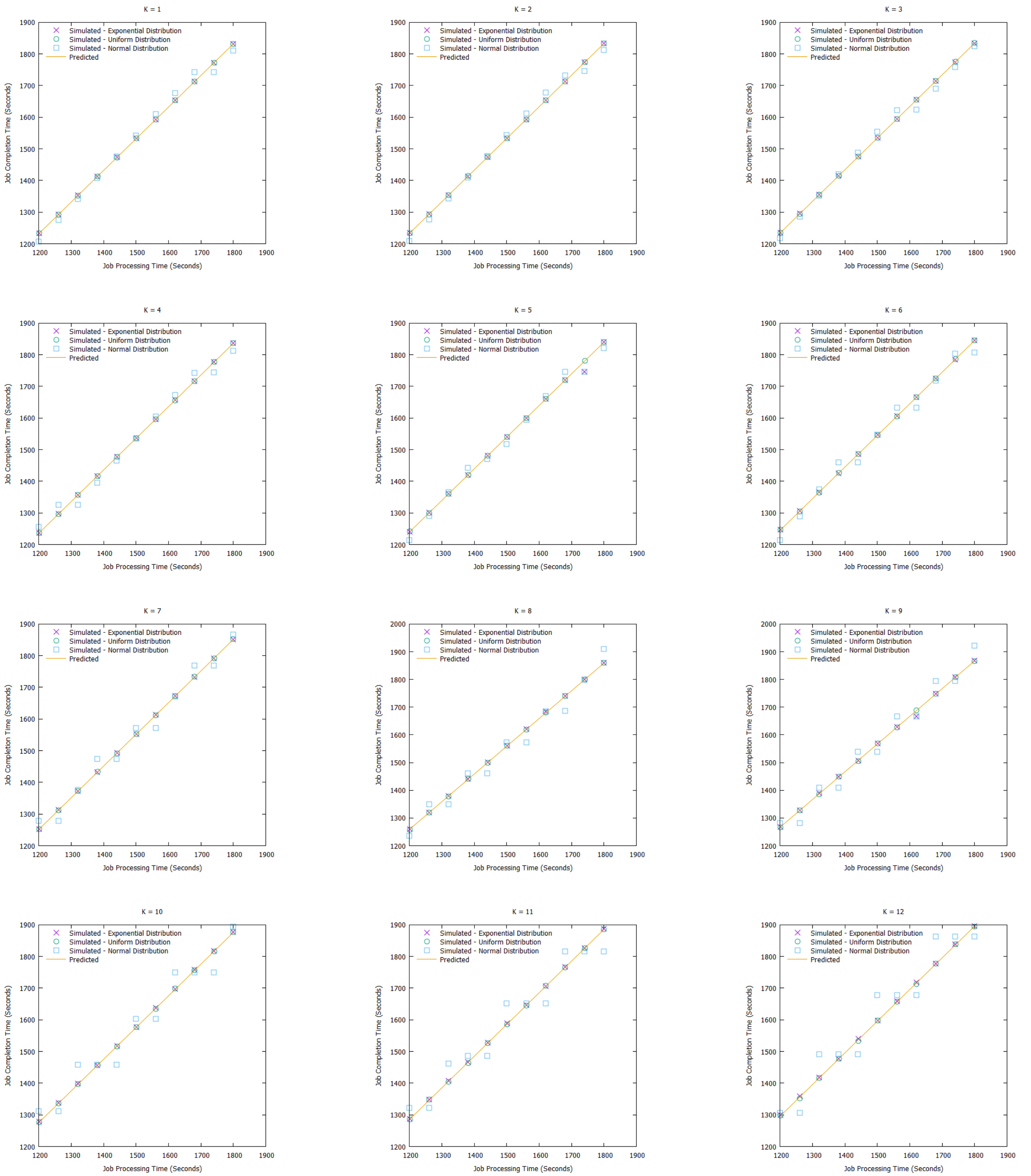

Figure 6: ***Predicted and simulated job completion times for exponentially distributed, uniformly distributed and normally distributed job processing times given that k other vehicles are competing to receive a communication slot.***