International Journal of Pure and Applied Mathematics
Volume 118 No. 24 2018
ISSN: 1314-3395 (on-line version)
url: http://www.acadpubl.eu/hub/
Special Issue

http://www.acadpubl.eu/hub/

MOBILE EDGE COMPUTING BY VEHICULAR EDGE NETWORKS FOR EFFICIENT AND SCALABLE TASK OFFLOADING

Mohammad Pasha[1], Khaleel Ur Rahman Khan[2],
[1]Information Technology Department,
[2]Computer Science Engineering Department,
[1]MJCET, ACE[2]
Hyderabad, India.
muhammed.pasha@gmail.com,
khaleelrkhan@gmail.com

May 27, 2018

Abstract

Vehicular Edge Networks proposes the use of computation facilities i.e. On-board units, available at the vehicles to support services offered by remote cloud. Being termed as Opportunistic Vehicular Edge Computing lately, it has the potential to supplement the services provided by the Edge devices. Vehicular Edge Computing architectures have been proposed recently which support opportunistic task offloading. One of the foremost challenges in these networks is efficient utilization of the resources available at the vehicular nodes. The present work proposes DATOVC, a distributed and adaptive protocol for efficient task offloading in these networks to address the scalability of vehicular clouds. Results obtained by extensive simulations are presented to evaluate and compare its performance with existing protocols.

***Key Words*:** Vehicular Cloud Computing; Mobile Edge Computing; Vehicular Ad-Hoc Networks; Computation Offloading.

1 INTRODUCTION

A Mobile Computing platform called as Vehicular Cloud is envisioned due to the increase in processing, storage and communication capabilities of Vehicular Nodes. It is lately being referred to as Vehicular Edge Computing [1]. To improve the on-road safety and other services various infrastructural services are required. The under-utilized resources in vehicles can be exploited to fulfil this need. The forefront challenges in these dynamic environments are efficient resource utilization and task offloading. The concept of executing tasks remotely is predominantly adapted from mobile cloud computing (MCC) due to the inherent limitations of mobile devices in processing-heavy tasks and to an extent avoid battery drain [2]. But vehicular networks do not face this challenge. These rather can help overcome the need of depending on the remote servers for processing and storage by utilizing services provided by potential nodes and road-side equipment. Further, these vehicles can assist in distributed and coordinated execution of tasks that are related to effectively managing various transport activities on roads. Planned evacuation during road traffic congestion and accidents can be addressed using vehicular networks [3]. Various recently proposed works address the concept of task offloading in vehicular networks [7], Successful task completion [8], time constrained execution [9], distributed task scheduling [10], fault tolerant task scheduling [11], [12], surrogate selection algorithms [13], [18].

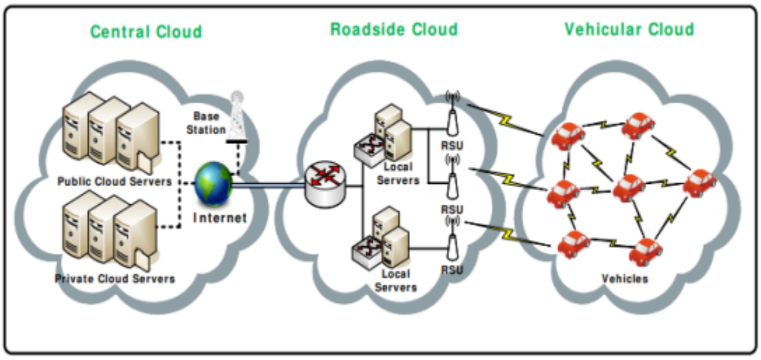

Figure 1 Vehicular Cloud Computing Architecture [4] is composed of (a) Central Cloud, (b) Roadside Cloud and (c) Vehicular Cloud.

2 PROBLEM DEFINITION AND FORMULATION

TABLE I Notations and their meanings used in problem formulation and related derivations.

Notation or Symbol	*Meaning*
Task related parameters	
T_{id}	Task Identifier
T_d	Time constraint of task or deadline
C	Amount of computations/ instructions to be processed
Surrogate related parameter	
μ_s	Task processing capability of a node in million instructions per second (MIPS)
T_Defferal	Intentional Delay in Response time calculated by surrogate nodes
Task completion time related parameters	
TCT_i	Task completion time for a given task i
T_{tx}	Task transmission time
T_{rx}	Task receiving time
T_{exec}	Task execution time
T_{queue}	Task queuing time
D_i	Total size of the task parameters sent in bytes
$B_{c,s}$	Shared Bandwidth between client and surrogate vehicle in Mbits/sec.
Link lifetime related parameters	
T_{link}	Link lifetime between vehicles
v_a, v_b	Instantaneous velocity of vehicle a, Instantaneous velocity of vehicle b
$d_{(a,b)}$	Communication distance between vehicle a and vehicle b
Mf	Mobility factor
Mf_{pre}	Previous value of Mobility factor
N	Node density
w	Parameter used to regulate mobility factor value
r	Relative velocity
Parameters for calculating worst case communication time in 802.11p based VANETs	
$T_{preamble}$	Time taken for identification of frame type
T_{sym}	Symbol duration time for 802.11p
T_{signal}	Signal duration time for 802.11p
B_c	Bandwidth in Mbits/sec.
l	Payload value in bits
N_{DBPS}	Number of data bits per OFDM symbol
CWmin	Minimum value of contention window in slots, i.e., 15
CWmax	Maximum value of contention window in slots, i.e., 1023
T_{ack}	Time taken for receiving acknowledgement
T_{slot}	Slot time for 802.11p, i.e., 13 microseconds

3 PROBLEM DEFINITION

Using the network and task models we present the calculations for the task completion time. The total task completion time for executing a task is composed of communication time, the computation time and the queuing delay of the task in the surrogate queue.

$$TCT_i = T_{tx} + T_{rx} + T_{exec} + T_{queue} \tag{1}$$

Where

- $T_t x = T_r x = D_i / B_{c,s}$ Where D_i is the total size of task parameters and $B_{c,s}$ is the available shared bandwidth between the client and the surrogate;
- $T_{exec} = C_i / \mu_i$Where C_i is the computations requested for a task and μ_s is the computation capacity of the surrogate;
- $T_q ueue$ is the queuing delay of the task;

In vehicular networks the maximum time that can be allocated for completion of task is dependent on the link duration time between the client and surrogate. The link duration is derived from the relative velocity of the vehicles. Thus we have

$$T_{link(a,b)} = (V_a - V_b)/d_{(a,b)} \tag{2}$$

Where

- $T_{link(a,b)}$ is the link duration time between vehicle a and vehicle b;
- V_a and V_b are instantaneous velocities of vehicle a and vehicle b;
- $d_{(a,b)}$ is the instantaneous communication distance between vehicle a and vehicle b;

Thus to successfully offload tasks to surrogates the following inequality needs to be realized.

$$2 * (D_i / Bc, s) + (C_i / \mu_s) + T_{queue}(V_a - V_b)/d_{(a,b)} \tag{3}$$

With the following constraints

- $TCT_i \Leftarrow \mathrm{T}_d$. (4)
- $\mu_s >> C_i$ per unit of time.... (5)
- $D_i <$Wave Short Message Packet payload, i.e.,4096 bytes.........(6)

4 DISTRIBUTED AND ADAPTIVE TASK OFFLOADING IN VEHICULAR COLOUDS (DATOVC)

The proposed algorithm is composed of three phases namely resource discovery phase, fitness evaluation phase and surrogate selection phase.

A. Vehicular Resource Discovery Protocol

The proposed resource discovery protocol is broadcast protocol to exchange both positional as well as task related information from the local neighborhood of a node. All nodes participating in task offloading process compute a minimum threshold fitness score which is a function of its computation capacity (measured in million instructions per second i.e., represented by MIPS) and task queue length. The client nodes include it in the sent task offloading request and broadcasts it.

B. Fitness Score Evaluation Process

Each target surrogate which receives a task offloading request also compares the client fitness score to its own to determine its suitability for accepting the task for remote processing. Nodes that already have higher utilization of resources (as measured by the task queue length) avoid this calculation and simply drop the task offloading requests. This directly influences the overall power consumption of the network as a whole as it reduces the processor cycles at each node. A node which provides faster processing units and with a least bounded task queue computes a delay period, called $T\_Defferal$. It is inversely proportional to its fitness score. This favors the less utilized surrogates over highly utilized ones. to clients. Hence the fittest surrogate gets a chance to reply to the client node first. This move reduces the response storm created by blind sending of replies by all surrogates. Moreover, taking advantage of the broadcast nature of the wireless medium the other surrogates when overhear a response from a surrogate compares the offered fitness score to its own fitness score. All surrogate nodes wait for a period equal to $T\_Defferal$ calculated individually before replying Now overhearing surrogates may decide to cancel its delay timer of reporting the fitness score and avoid the reply to the client depending on the number of nodes in its neighborhood. Because retransmissions are used to enhance the reliability of the transmitted messages it is important to adjust the retransmissions according to neighborhood density.

C. Surrogate Selection Process

The client node waits for a predefined period of time to accept responses from surrogates. When it receives the fitness score from potential surrogates it selects the surrogate with best fitness score. Due to the fact that vehicular networks are prone to packet losses due to various reasons the surrogate may offload tasks to multiple surrogates also. Table 2. highlights the important phases of the protocol.

5 PERFORMANCE EVALUATION

The algorithms are implemented using vehicles in network simulation framework [21] which couples OMNET++ [22] and SUMo [23] for realistic mobility modelling. To evaluate the performance, we consider that the client nodes generate 2 to 8 jobs per minute. These jobs are partitioned into 10 to 20 tasks each [8]. Thus we vary the number of tasks from 20 to 160 tasks for different experiments. Each task has a computation requirement of 1000 MIPS to 2000MIPS.

6 SIMULATION SETUP

We consider a one-dimensional vehicular network formed on a 2-Lane highway which follows the Level of Service concept of transport management as shown in Table 3 is considered. To evaluate the effect of the vehicular movement on the offloading decision we make use of the Gaussian exponential mixture of mobility model proposed in [26].We compare DATOVC against ACATOVC and plot its performance in terms of being scalable.

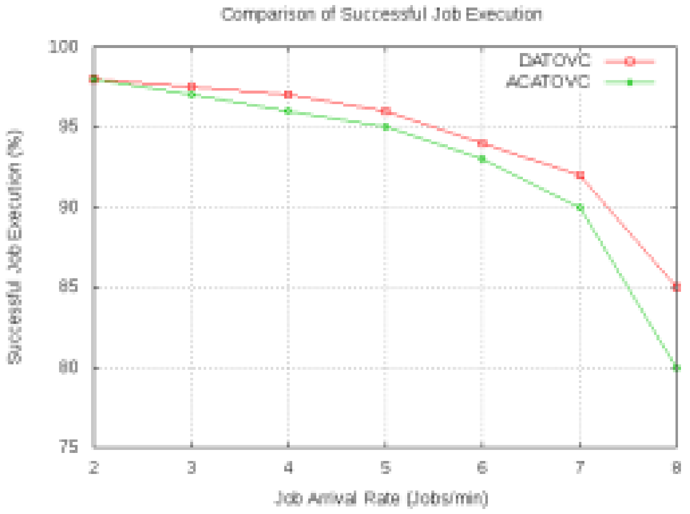

Figure 2 Comparison of Successful Job Execution (%) for varying Job Arrival Rate

Figure 2 shows the performance of the proposed algorithms for varying job arrival rate. The vehicular density is fixed at 40 veh/ km. As the number of job requests increases the Successful Job Execution (%) decreases. This is owing to the fact that with more number of job requests being flooded into the network it competes with the task response being sent by the surrogates. The DATOVC protocol performs better than ACATOVC due to lessening of broadcast storm. Hence offers 85% successful job execution performance.

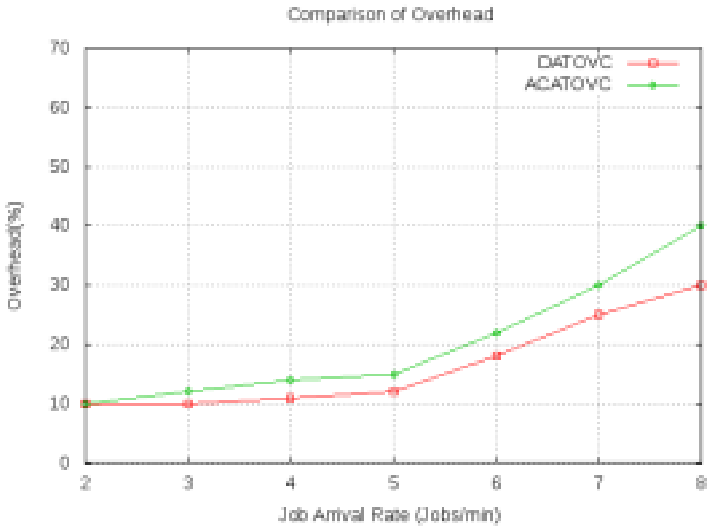

Figure 3 Comparison of Overhead (%) for varying Job Arrival Rate

Figure 3 shows the performance of the proposed algorithms for varying job arrival rate. The vehicular density is fixed at 40 veh/ km. As the number of job requests increases the Overhead (%) increases. This is owing to the fact that with more number of job requests being flooded into the network it competes with the task response being sent by the surrogates. The DATOVC protocol performs better than ACATOVC due to lessening of broadcast storm. Hence offers 85% successful job execution performance.

7 CONCLUSION

An efficient task offloading scheme was presented for vehicular networks to support Edge Computing. The proposed scheme considers task offloading from multiple client nodes to nearby surrogate nodes, satisfying the different tasks constraints while considering the wireless channel dynamics. One significant point to note that it is just not enough that a particular surrogate is selected purely based fulfilling the tasks deadline constraint. The link lifetime between the client and the surrogate should be of enough length for the client to send the task workload to the surrogate. The effect of the delay that occurs due to contention at the MAC layer should be taken into account when offloading a task. The proposed algorithm outperforms existing protocols in terms of Scalability, Average Job Execution time and percentage of Successfully Completed jobs.

References

[1] Hu, Yun Chao, et al. Mobile edge computingA key technology towards 5G. ETSI white paper, 2015.

[2] Fernando, Niroshinie, Seng W. Loke, and Wenny Rahayu. Mobile cloud computing: A survey. Future generation computer systems, 2013.

[3] Sardellitti, Stefania, Gesualdo Scutari, and Sergio Barbarossa. Joint optimization of radio and computational resources for multicell mobile-edge computing. IEEE Transactions on Signal and Information Processing over Networks, 2015.

[4] [4]Uzcategui, Roberto A. Universidad Nacional Experimental Politecnica Antonio Jose de Sucre Guillermo Acosta-Marum. Georgia Institute of Technology WAVE: A Tutorial TOPICSIN Automotive Networking, 2009.

[5] Abdelhamid, Sherin, Hossam S. Hassanein, and Glen Takahara. Vehicle as a mobile sensor. Procedia Computer Science, 2014.

[6] Chen, Lei, and Cristofer Englund. Cooperative intersection management: a survey. IEEE Transactions on Intelligent Transportation Systems, 2016.

[7] Shojafar, Mohammad, Nicola Cordeschi, and Enzo Baccarelli. Energy-efficient adaptive resource management for real-time vehicular cloud services. IEEE Transactions on Cloud computing, 2016.