2018 IEEE 11th International Conference on Cloud Computing

Bandwidth Optimal Data/Service Delivery for Connected Vehicles via Edges

Deepak Gangadharan, Oleg Sokolsky and Insup Lee
University of Pennsylvania, USA
Email:{deepakg@seas,sokolsky@cis,lee@cis}.upenn.edu

BaekGyu Kim, Chung-Wei Lin and Shinichi Shiraishi
Toyota InfoTechnology Center, USA
{bkim,cwlin,sshiraishi}@us.toyota-itc.com

***Abstract*—The paradigm of connected vehicles is fast gaining lot of attraction in the automotive industry. Recently, a lot of technological innovation has been pushed through to realize this paradigm using vehicle to cloud (V2C), infrastructure (V2I) and vehicle (V2V) communications. This has also opened the doors for efficient delivery of data/service to the vehicles via edge devices that are closer to the vehicles. In this work, we propose an optimization framework that can be used to deliver data/service to the connected vehicles such that a bandwidth cost objective is optimized. For the first time, we also integrate a vehicle flow model in the optimization framework to model the traffic flow in the coverage area of the edges. Using the optimization framework, we study the variation of the optimal bandwidth cost for varying problem sizes and vehicle flow model parameter values for both data and service delivery.**

***Index Terms*—connected vehicles; edge computing; V2C; V2I; data delivery; service delivery; bandwidth cost**

I. Introduction

In the past few years, there has been a focus to develop technologies that enable connectivity of vehicles in order to improve the driving safety and experience. In particular, this has been realized by connecting vehicles to cloud (V2C), infrastructure (V2I) and other vehicles (V2V). The advent of these modes of connectivity has also opened the doors to efficiently deliver data and services to the vehicles. For example, a driver travelling to a new destination may request to download the map of a certain area or may want an update to an already installed application in the vehicle. The above mentioned data can be delivered to the vehicle either directly from the cloud or through a device that is located in close proximity to the vehicle (called the *edge device* or the edge). Similarly, computation offloading is one service in which a computation intensive function belonging to an application residing in a vehicle, such as Lane Keep Assist (LKA), may be offloaded to an edge device proximally closer to the vehicle while on the move. The offloaded computation is then executed on the edge and the results are sent back to the vehicle.

The hierarchical architecture that we consider in this work is shown in Fig. 1. There are three distinct levels in the architecture. The topmost level in the architecture is the cloud. The cloud provides various kinds of services and communicates with both the vehicle and the edge devices. The communication between the cloud and the edge device, denoted as *Type 1* communication in Fig. 1, typically involves transfer of a large amount of data. The middle level includes the edge devices, which are intermediate devices between the cloud and the vehicle. An example of an edge device would be a traffic light control node. The edges may communicate with each other (denoted as *Type 2* communication in Fig. 1) with lesser latency in comparison to *Type 1* communication. The bottom level in the architecture consists of the vehicles. As a vehicle passes by an edge device, it may communicate with the edge device (denoted as *Type 3* communication). The communication latency between the vehicle and the edge is the smallest. In addition, the vehicles also communicate with the cloud, which initiates data or service delivery. In this work, we do not consider V2V communication as we do not consider vehicles as relays to deliver data or service.

In order to deliver services, offloading to the edge is more beneficial as it incurs much smaller delay compared to offloading to the cloud. In the case of data also, delivery via the edge is efficient for the following reason — if multiple vehicles request for the same large sized data, multiple transmissions of the same data from the cloud to the different vehicles can be avoided if two or more vehicles pass through the same edges. In the above case, it is enough to send the data once to a group of edges so that all the vehicles passing through them can receive the data. Therefore, delivery of data or services via the edge is more efficient than delivering directly from the cloud.

Problem Motivation In this work, we propose an optimization framework to deliver data (such as update data) or service (such as computation offloading) from the cloud to the vehicles via the edges. The vehicle first interacts with the cloud letting

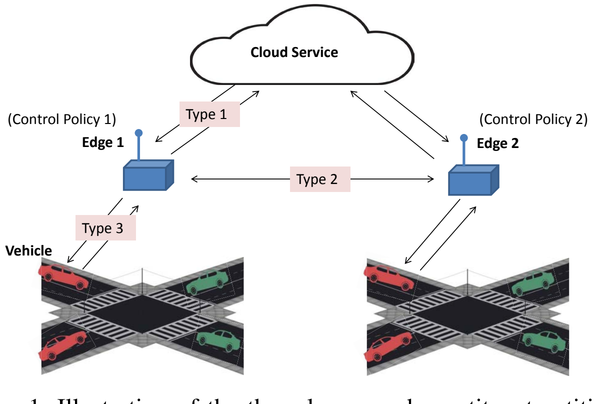


Fig. 1: Illustration of the three layers and constituent entities in vehicle connectivity architecture

2159-6190/18/$31.00 ©2018 IEEE
DOI 10.1109/CLOUD.2018.00021

the cloud know what data/service it wants and the route it plans to take towards its destination. We do not consider the scenario of route changes in this work. Let us assume that the route of the vehicle consists of a set of edges. Further, let us also consider that multiple vehicles send similar requests to the cloud. The problem that needs to be solved in the cloud is how to map the data/service requested by the vehicles to the edges such that they receive the requested data/service during their trip.

Given the resource availabilities at the various edges in the edge network and the routes that the vehicles take, there are multiple possibilities of how data can be partitioned and mapped to the edges that belong to its route. It may be possible to map all the requested data to one of the edges or the data can be broken down into chunks and the data chunks can be mapped to the set of edges. Similarly, it might be possible to retrieve a service from any of the edges in the route. Communication bandwidth is a scarce resource and the available bandwidth at the edges depends on the number of vehicles passing the coverage area of the edge and the velocity of the vehicles in the coverage area. We utilize the notion of bandwidth cost based on a well known bandwidth pricing mechanism. Based on this notion, we present an optimization framework that will be used in the cloud to map the data chunks/service for all vehicles to the edges in order to minimize the total bandwidth cost over all the edges.

The idea behind using the bandwidth pricing mechanism is to increase the cost of bandwidth usage of an edge with increase in its bandwidth utilization. Traditionally, bandwidth pricing mechanism has been used in allocating bandwidth to users on links in order to prevent link over congestion in traditional networks. Therefore, we use this mechanism in our optimization framework so that the requested data is mapped to the edges such that all the vehicles can retrieve their data from the edges without over congestion as it enables serving higher number of vehicles. The likelihood of over congestion of vehicles at an edge is very much expected in a realistic scenario as evident in pilot projects started in New York City, Tampa and Wyoming [2], which consider thousands or tens of thousands of vehicles and few hundreds of edges.

Challenges and Contributions: To the best of our knowledge, none of the earlier works on delivering data/service to mobile nodes used any vehicle flow model to perform analysis. The vehicle flow model is essential to capture the traffic flow in the vicinity of an edge. The density of the traffic influences the delivery of data/services to the vehicle requiring them. We utilize a linear vehicle flow model in this work, but the framework can easily handle other types of flow models.

To the best of our knowledge, this is the first work which proposes a centralized (to be executed in the cloud) optimization framework for bandwidth optimal data/service delivery to connected vehicles via edges considering a vehicle flow model and edge resource constraints. More specifically, our main contributions in this work are

1) Proposed optimization constraints (which integrate a linear vehicle flow model) for delivery of data and service to mobile vehicles.
2) Demonstrated the usefulness of a bandwidth pricing mechanism to minimize the bandwidth cost over all the edges.
3) Performed several experiments for synthetic vehicle traffic and showed the optimization results obtained.

The rest of the paper is organized as follows. In the next section, we present the related work on mobile edge computing and data/service delivery to mobile nodes. Subsequently, we present the problem formulation in Section III. The optimization constraints and objective functions for data and service delivery are then presented in Section IV. The experimental results are discussed in Section V. Finally, we conclude the paper with our remarks in Section VI.

II. Related Work

In this section, we first present a brief evolution of edge computing and mobile edge computing (MEC) in particular and their advantages and limitations. Then, we look at prior research done in two areas namely resource allocation in edge computing and data/service delivery to mobile nodes via edge nodes, which are closely related to the problem we address in this paper. The primary advantage of MEC is the proximity of the edge nodes to the mobile devices, which brings a limited amount of memory, compute and communication resources close to the mobile devices thereby enabling low latency turn around time of the results [9], [4], [11]. This is in sharp contrast to its predecessor mobile cloud computing (MCC), where the computation was offloaded to a cloud server situated far away from the mobile user [3]. Although MCC provided powerful compute and large amount of storage resources, the communication latencies between the cloud server and the mobile device are very long.

The first idea of edge computing was proposed in [13] and was called *cloudlet* in which powerful computers were placed in appropriate locations to give the mobile users the required compute and storage resources near them. This presented the challenge of switching between mobile network and WiFi when using the resources on the cloudlet. Another alternative that was proposed was to build adhoc clouds [14] from the mobile devices themselves, but this presented its own set of challenges such as finding the appropriate mobile devices for computation and delivery of results, incentivizing the mobile devices to provide their compute resources, security and privacy, etc. Fog computing [5] generalizes the concept of edge computing better than cloudlets and adhoc clouds, where processing is performed on many connected devices at the edge of the network. The problem with all the above mentioned edge computing paradigms is that the computing is not integrated into the architecture of the mobile network. In order to solve this, European Telecommunications Standards Institute (ETSI) came with a solution known as MEC [9], which seamlessly integrates the computing into the mobile network.

One of the key services provided in MEC is computation offloading as it greatly reduces the energy and speeds up the computation process. This service was earlier proposed in the domain of MCC [18] where a code partitioning algorithm was proposed for mobile code offloading. Once a decision is taken to offload a task, adequate resources need to be allocated on the edge. There are several works which present techniques to allocate resources for computation offloading on a single node such that the execution delay is minimized [19], [8]. In [8], power consumption is also optimized along with delay. Similarly, there are few works that propose techniques for resource allocation on multiple edge nodes while minimizing execution delay [16] and a combination of delay with power consumption [12]. While all these works present interesting techniques to allocate resources on the edge for computation offloading, they consider task offloading from a mobile device and therefore do not consider any vehicle flow model, which is very essential to incorporate in the scenario of data/service delivery to connected vehicles via edges. Moreover, our objective function in this work is optimization of bandwidth cost under delay constraints.

The research that is closest to our work are the papers on data delivery from infrastructure to vehicles [10] and online resource allocation to deliver services like computation offloading [17] for mobile nodes. In [10], the authors propose a trajectory-based forwarding scheme to deliver data from infrastructure nodes to moving vehicles in vehicular adhoc networks. A target node is determined based on the route of the destination vehicle and data packets are forwarded to the selected target node which the vehicle is expected to pass by. The authors utilize packet's delivery delay distribution and destination vehicle's travel delay distribution to select the optimal target node which minimizes delivery delay and satisfies a packet delivery probability. An online edge cloud resource allocation algorithm is proposed in [17], which considers arbitrary user movement and variation in resource prices. Due to arbitrary user movement, the cloud does not a priori know the route that the vehicle will take. In contrast, it is a very likely scenario that a driver knows beforehand the route to the destination. In such circumstances, the cloud can exploit the route information to deploy data/service, which has been used in [10]. However, both these techniques do not consider any vehicle flow model, which characterizes the movement of traffic near the edge or between edges. In this paper, our proposed optimization framework addresses this problem in the context of optimal bandwidth allocation.

III. Problem Formulation

In this section, we present an optimization framework to deliver data/service to connected vehicles such that bandwidth cost incurred at the edges is minimized. The optimization flow starts in the cloud when it receives requests from the vehicles in order to either deliver some data or service. Along with the request, the vehicles also send information regarding the route that they intend to take. We assume that the cloud maintains a route map from which it can infer the edges that the vehicle will pass through. Let us assume that there are N vehicles requesting for either data or service and there are M edges in the network available for data/service delivery.

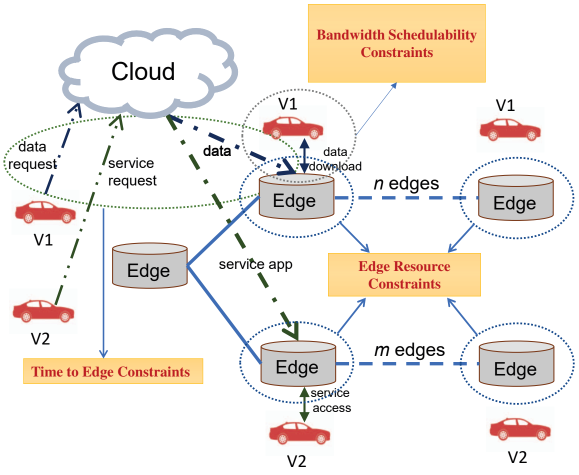


Fig. 2: Flow in the connected vehicle scenario and associated constraints

Before describing the constraints necessary for the optimization, we introduce some terms that will be used in the formulation of the constraints and the objective function.

Decision variables $m_{i,j}$ and $serv_{i,j}$: For data delivery, the decision variable that we use is the amount of data $m_{i,j}$ belonging to a vehicle V_i that the cloud partitions to an edge E_j. It is an integer variable. For service delivery, we use the binary decision variable $serv_{i,j}$, which denotes that vehicle V_i receives its service from edge E_j if $serv_{i,j} = 1$ and the contrary if $serv_{i,j} = 0$. For edges E_j that do not fall within the route of vehicle V_i, $m_{i,j} = 0$.

Route parameter: We use a binary variable $x_{i,j}$ to denote whether an edge falls within the route of a vehicle. If $x_{i,j} = 0$, then edge E_j does not fall in the route of vehicle V_i and therefore $m_{i,j} = 0$. On the other hand, if $x_{i,j} = 1$, then edge E_j falls in the route of vehicle V_i and $m_{i,j}$ may assume a non zero value.

Data: The data that a vehicle requests is characterized using the amount of memory it requires on the edge memory resources. Let us assume that the memory requirement for some data that vehicle V_i requests for is M_i. The data transmitted between the edge and the vehicle for service delivery is described in Section IV.

Edge parameters: The edge is characterized by the memory, processing resources and the bandwidth that it offers. Let the memory capacity of an edge E_j be M_j. We also denote the memory that is already occupied on edge E_j as M_j^{occ}. Therefore, the available memory on the edge is given by $M_j - M_j^{occ}$. The processing capacity at an edge is given by P_j resource units (i.e., it could be the number of processing units of time or the number of VMs available) and the number of resources already in use is given by P_j^{occ} units. The bandwidth that the edge E_j offers is denoted by B_j.

Timing parameters: We denote the time that the vehicle V_i takes to travel to edge E_j after initiating the download by

$t\_trv_{i,j}$. Also, the time taken to send a data chunk $m_{i,j}$ to edge E_j after the vehicle V_i initiates the download is given by $t\_comm_{i,j}$.

The goal of optimization step in the cloud is to find the values of $m_{i,j}$ (or chunks of the total data) and $serv_{i,j}$ for relevant edges of a vehicle (edges that fall in the route) such that the total bandwidth cost is minimized while satisfying the constraints.

IV. Data/Service Delivery Optimization

In this section, we present the optimization constraints for data delivery and service delivery. The first part of this section includes the description of some basic constraints that will be used for both data and service delivery. Subsequently, we present the optimization constraints necessary specifically for feasible service delivery to vehicles. Finally, the optimization objective functions are formulated and discussed.

A. General Constraints for Data/Service Delivery

Firstly, we present two preliminary constraints and then we present the timing and resource constraints that are relevant to the connected vehicle scenario considered in this work. These constraints are illustrated in Fig. 2 and described in detail below.

1) **Range Constraint**: This constraint sets the upper and lower bound for the decision variable $m_{i,j}$ and is given as follows.

$$\begin{aligned} m_{i,j} &= 0, i = 1..N, j = 1..M : x_{i,j} = 0 \\ m_{i,j} &\geq 0, i = 1..N, j = 1..M : x_{i,j} = 1 \\ m_{i,j} &\leq M_i, i = 1..N, j = 1..M : x_{i,j} = 1 \end{aligned} \tag{1}$$

The upper bound for $m_{i,j}$ is the entire size of the data that the vehicle requests for, while the lower bound is 0.

2) **Accumulation Constraint**: For data delivery, as data may be broken into chunks and mapped to more than one edge, the summation of the sizes of the data chunks over all the edges should not exceed the entire data size M_i. This constraint is formulated as shown below.

$$\sum_{j=1}^{M} m_{i,j} \times x_{i,j} = M_i, i = 1..N \tag{2}$$

In the case of service delivery, the above constraint can be used as follows

$$\sum_{j=1}^{M} serv_{i,j} \times x_{i,j} = 1, i = 1...N \tag{3}$$

which signifies that a vehicle V_i receives its service from only one of the edges on the route.

3) **Time to Edge Constraint**: In the case of data delivery, this timing constraint ensures that the data chunk has been transferred by the cloud to the edge before a vehicle reaches an edge. We formulate this constraint as follows.

$$m_{i,j} \times t\_comm_{i,j} \leq m_{i,j} \times t\_trv_{i,j}, i = 1..N, j = 1..M \tag{4}$$

We use the decision variable $m_{i,j}$ on both sides of the inequality in order not to map any data chunk to the edges where the timing relationship $t\_comm_{i,j} \leq t\_trv_{i,j}$ is not satisfied. In the case of service delivery also, Eqn. 4 can be used by replacing $m_{i,j}$ with $serv_{i,j}$.

4) **Edge Resource Constraint**: The resource constraints considered within this class of constraints are the memory constraint and the processing resource constraint. Under the memory constraint, we ensure that the amount of memory required for all the chunks belonging to all the vehicles passing through an edge does not exceed the memory capacity of the edge while considering the memory that is already occupied by other applications or data residing on the edge. This constraint is formulated as below.

$$\sum_{i=1}^{N} m_{i,j} + M_j^{occ} \leq M_j, j = 1..M \tag{5}$$

For service delivery to vehicle V_i, we assume that d_i is the input data for processing and r_i is the result data that is sent back to V_i. Hence, the memory constraint that needs to be satisfied for service delivery is

$$\sum_{i=1}^{N} serv_{i,j} * (d_i + r_i) + M_j^{occ} \leq M_j, j = 1..M \tag{6}$$

The processor resource constraint that needs to be checked for service delivery is that processor resource usage for all the vehicles $\{V_i\}$ (given by p_i) that retrieve service from E_j does not exceed the maximum processing capacity of the edge. This is given by

$$\sum_{i=1}^{N} serv_{i,j} * p_i + P_j^{occ} \leq P_j, j = 1..M \tag{7}$$

5) **Bandwidth Schedulability Constraint**: Given the bandwidth B_j offered by the edge E_j, the amount of data available to a vehicle V_i can be computed based on a macroscopic vehicle flow model in the coverage area of an edge, which is described in [15]. The macroscopic flow model depends on three quantities — vehicle density k_j, vehicle flow q_j and vehicle velocity $v_{i,j}$. Vehicle density k_j is defined as the number of vehicles per unit distance. Vehicle flow q_j is defined as the number of vehicles passing a fixed point per unit time. The three quantities in the flow model are related as follows

$$q_j = k_j \times v_{i,j}$$

Greenshields [7] proposed a linear relationship between speed and density based on field experiments given by

$$v_{i,j} = v_j^f \times (1 - \frac{k_j}{k_j^{jam}})$$

where v_j^f is the free flow velocity near edge E_j, which is the velocity of the vehicle when it has no obstructions from other vehicles (usually assumed to be the speed limit) and k_j^{jam} is the vehicle density during a jam. The vehicular traffic in the coverage area of the edge can be modelled using a M/D/C/C queuing model for a steady-state traffic flow model described above as shown in [15], whereby the traffic arrival at the edge follows a Poisson process, servicing the traffic is deterministic and there are C servers. If the coverage distance of an edge is L_j, then $C = k_j^{jam} \times L_j$ is the maximum number of vehicles that can be accommodated in the coverage range of the edge. As per M/D/C/C queue model, the minimum number of bytes received by a vehicle is given by (can be easily proved from Eqn 7 in [15] by considering that all vehicles receive the same service weight)

$$D_{i,j}^{min} = \frac{B_j}{k_j^{jam} \times v_{i,j}} \tag{8}$$

where B_j is the bandwidth of the edge device.

The bandwidth schedulability constraint for vehicle requiring data is given by

$$m_{i,j} \leq D_{i,j}^{min}, i = 1..N, j = 1..M \tag{9}$$

which simply depicts that the data chunk mapped to edge E_j for vehicle V_i must be less than or equal to the minimum number of bytes that V_i will receive from E_j during its journey across the coverage distance.

B. Service Delivery Specific Constraints

We now present the constraint that is necessary to be checked specifically for feasible service delivery to the vehicles. For any service such as computation offloading to the edges or some other application execution on the edge, computation resources are used on the edge. The computation resources on the edge are normally considered to be virtualized. Therefore virtual machines (VMs) are the computation resources allocated to the service required by a vehicle. In this paper, we consider that the computation resource required by vehicle V_i (denoted earlier as p_i) is the number of VMs for execution of the service and the VMs on all edges are homogeneous with similar computing capability. We also assume that the cloud is privy to information of how much time it would take to execute a particular service if p_i number of VMs are allocated for the service. Let us denote this time as t_{p_i}. One another assumption in our framework for service delivery is that each service is allocated p_i resources entirely on one edge.

Considering all the above assumptions, the schedulability constraint for processing the service on an edge resource is that the time required to process the service and the time taken to transfer the data and result between the vehicle and the edge is lesser than the time the vehicle remains in the coverage distance of the edge. The minimum number of bytes that a vehicle may transfer during its transit in the coverage area is given by

$$D_{i,j}^{min,serv} = \frac{B_j \times (\frac{L_j}{v_{i,j}} - t_{p_i})}{k_j^{jam} \times L_j} \tag{10}$$

which signifies that the transfer of data must take place within the time remaining after processing the service and before the vehicle moves out of the coverage area. The time remaining for a vehicle to transfer the data to process and get the result back for a service required by vehicle V_i is $t_{rem,i}^{serv} = \frac{L_j}{v_{i,j}} - t_{p_i}$. From M/D/C/C queue modelling [15], it can be easily derived that the minimum number of bytes that can be transferred with equal bandwidth share to all vehicles in a coverage area is given by $\frac{B_j \times t_{rem,i}^{serv}}{k_{jam} \times L_j}$. This results in Eqn. 10 by substituting for $t_{rem,i}^{serv}$.

The bandwidth schedulability constraint is then given by

$$serv_{i,j} * (d_i + r_i) \leq D_{i,j}^{min,serv}, i = 1..N, j = 1..M \tag{11}$$

C. Objective Function

In this work, we use the following two objective functions and demonstrate the results of optimization.

1) Objective 1: Minimization of maximum bandwidth utilization over all edges
In the case of data delivery, the bandwidth utilization at an edge is a summation of the bandwidth utilizations of the vehicles which retrieve some data of size $m_{i,j}$ and vehicles which demand a total bandwidth B^{res}. The bandwidth utilization for vehicles requiring data can be obtained as follows.

$$bw_j^{util,data} = \sum_{i=1}^{N} \frac{\frac{m_{i,j} \times v_{i,j}}{L_j}}{B_j} \tag{12}$$

where $\frac{m_{i,j} \times v_{i,j}}{L_j}$ is the bandwidth requirement of the vehicle V_i retrieving data. For service delivery, the bandwidth utilization at an edge of a vehicle is given by

$$bw_j^{util,serv} = \sum_{i=1}^{N} \frac{d_i + r_i}{\left(\frac{L_j}{v_{i,j}} - t_{p_i}\right) \times B_j} \tag{13}$$

Let B_j^{res} be the bandwidth reserved for other vehicles near edge E_j. Then the bandwidth utilization for other vehicles with reserved bandwidth of B_j^{res} is computed as follows.

$$bw_j^{util,oth} = \frac{B_j^{res}}{B_j} \tag{14}$$

The total bandwidth utilization at edge E_j considering vehicles requiring data and service both is then computed as follows.

$$bw_j^{util} = bw_j^{util,data} + bw_j^{util,serv} + bw_j^{util,oth} \tag{15}$$

Hence, the first objective function considered is given by

$$minimize \max_{\forall j=1..M} bw_j^{util} \tag{16}$$

which denotes the minimization of maximum bandwidth utilization considering all the M edges.

2) Objective 2: Minimization of total bandwidth cost over all edges
The bandwidth cost is computed based on a pricing mechanism used in [6], which balances the bandwidth load across all the switches in cloud networks thereby controlling congestion. We apply the same principle in this work, where we use the pricing mechanism to balance the bandwidth load considering the vehicles which request data from the cloud. The bandwidth cost at an edge E_j is computed using a non linear pricing policy as follows.

$$bw_j^{cost} = \beta \times (1 + bw_j^{util})^2 \tag{17}$$

where β is the cost factor. The second objective function is obtained by

$$minimize \sum_{j=1}^{M} bw_j^{cost} \tag{18}$$

V. Experimental Results

In this section, we demonstrate the utility of our optimization framework by obtaining solutions for different problem sizes by varying the number of edges and the number of vehicles requiring data/service. In the first two sets of experiments, we show the results for optimal data delivery to vehicles via the edges. The two sets of experiments correspond to the two objective functions discussed in Section IV — minimization of max bandwidth utilization and minimization of total bandwidth cost. The final set of experiments shows the results for optimal service delivery to vehicles via the edges.

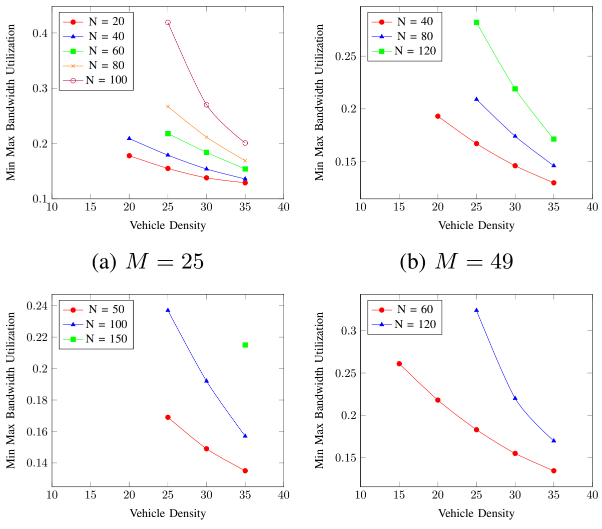
We will first explain the experimental settings and then present our inferences from the results obtained. The data for the experiments were generated using Matlab scripting and the optimization was carried out using the IBM ILOG CPLEX Solver.

A. Data Delivery: Variation of objective 1 with varying density values for different N

The intention of this experiment is to study the effects of variation of vehicle density values on the minimal maximum bandwidth utilization for different values of N and M. Firstly, we draw inferences regarding the general trend considering different problem sizes and then observe more detailed variations in objective functions.

The parameters used in this experiment are as follows.

1) We used 4 values for the number of edges (M) from the set $\{25, 49, 81, 100\}$. Although, these edges were generated as a square grid of sizes 5×5, 7×7, 9×9 and 10×10 respectively, the topology of the edge network does not change our proposed optimization framework including the formulated constraints.
2) Number of vehicles (N) requiring update was varied in steps between 20 and 200.
3) Vehicle jam density (k_j^{jam}) was fixed to 50 at all edges.



(a) $M = 25$

(b) $M = 49$

(c) $M = 81$

(d) $M = 100$

Fig. 3: Variation of minimal maximum bandwidth utilization with vehicle density for varying N

4) The actual vehicle density (k_j) in the coverage area was varied between 35 and 10 in steps of 5.
5) Coverage distance (L_j) of each edge was assigned arbitrary values between 0.6 miles and 1.6 miles.
6) Memory requirement of data requested by each vehicle (M_i) was randomly generated using uniform distribution between 60 Mbits and 80 Mbits. These values are very much in the range of realistic software update sizes as given in [1].
7) Memory capacity of the edge (M_j) was randomly generated using uniform distribution between 400 Mbits and 500 Mbits.
8) Maximum bandwidth capacity of the edges (B_j) was randomly generated using uniform distribution between 8 Mbps and 15 Mbps.
9) Free flowing velocity (v_j^f) of the vehicles in the coverage area of edges was randomly generated between 50 and 70 mph.
10) The route of the vehicles was randomly generated by picking connected edges randomly from a square grid.
11) The earliest travel times of the vehicles to the edges on its route was generated based on the distance of the edges from each other and free flowing velocity of the vehicles between the edges, both being generated randomly.

The results obtained from the above experimental setting are shown in Fig. 3. The graphs demonstrate the variation in minimal max bandwidth utilization across all the edges for varying vehicle density values and for varying N. The four sub-plots correspond to the four edge network sizes we have used for this experiment. There are three important characteristics we observe from the results in Fig. 3a–Fig. 3d. Firstly, for any value of N, the minimum of the max bandwidth utilization decreases as the vehicle density (k_j) at the edges increases. This clearly is because of the decrease in the velocities of the vehicle in the coverage area of the edge as the density increases resulting in more time spent in the coverage area thereby requiring a lesser bandwidth.

The second result we observe is that at lower vehicle densities, the minimum of the max bandwidth utilization across all the edges increases rapidly with the increase in the value of N and the increase in the objective function is marginal at higher vehicle densities. This behaviour occurs because of the combination of the earlier result where bandwidth requirement increases at lower densities due to increased velocity and the increase in bandwidth utilization at the edges due to increase in the number of vehicles requiring data delivery (N). For higher vehicle densities, the increase in minimal max bandwidth utilization is marginal because the increase in bandwidth requirement for the additional number of vehicles is marginal due to lower velocities.

Finally, for some lower vehicle density values, there is no feasible solution, because the velocity of vehicles is high and therefore does not receive enough bandwidth. This infeasible density point shifts upwards to higher densities with increase in the value of N. This is clear from all the sub-plots in Fig. 3 and specifically highlighted in the case of $N = 150$ and $M = 81$, where there is only feasibility for $k_j = 35$. However, the infeasible density point can be lowered with increase in the number of edges, which is explained below. There is no feasible solution for $k_j = 15$ for any plot from Fig. 3a–Fig. 3c. But, feasibility for $k_j = 15$ is found with increase in the value of M to 100 and $N = 60$ as shown in Fig. 3d. This result shows that there is a range of values of M for each N, which will lead to feasibility for lower values of k_j as the vehicle bandwidth requirement can be shared across larger number of edges resulting in lower requirement of bandwidth at each edge.

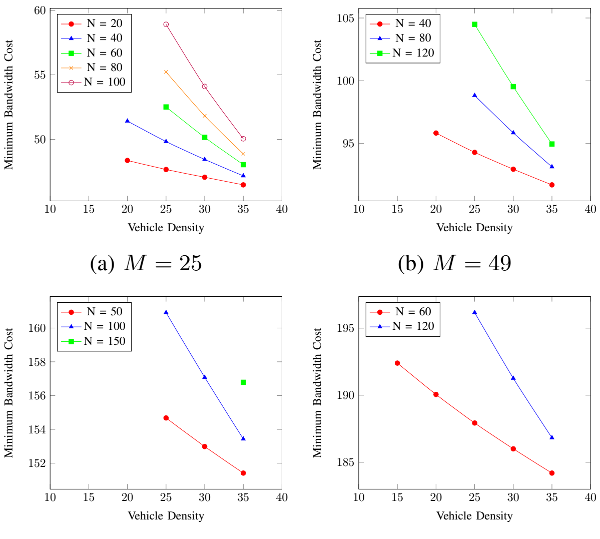
B. Data Delivery: Variation of objective 2 with varying density values for different N

In this experiment, we vary some parameters of the vehicle flow model along with N and M. The intention of this experiment is to study the effects of variation of vehicle density on the minimum bandwidth cost considering all the edges in the network. Here, we draw inferences regarding the general trend for different problem sizes and observe if there are any important changes from the results in Fig. 3.

The parameters used in this experiment are as follows.

1) The bandwidth cost factor β was fixed to 1.5.
2) Rest of the parameters were used from the previous experiment.

The results obtained from the above experimental setting are shown in Fig. 4. The graphs demonstrate the variation in minimum bandwidth cost for varying density values at the edges and for different number of vehicles. The four sub-plots correspond to the four edge network sizes we have used for this experiment. The trends that we observed in the earlier experiments for minimum of max bandwidth utilization are valid in this set of experiments also, i.e., (a) the bandwidth cost reduces with increase in vehicle density, (b) the rise in the bandwidth cost with increasing values of N is higher for lower densities than higher densities, and (c) there are no feasible solutions for specific lower density values for each N in an edge network with M edges.



(a) $M = 25$

(b) $M = 49$

(c) $M = 81$

(d) $M = 100$

Fig. 4: Variation of minimum bandwidth cost with vehicle density for varying N

Although the general trend in the results are the same in this experiment also, the solutions obtained are more inclined towards reducing the cost of bandwidth utilization at all the edges. In the earlier experiment, we observed from the results obtained that the minimum of max bandwidth utilization objective led to solutions where the data was more evenly distributed across the edges on the route of individual vehicles all the time while this was not necessarily the best solution in terms of the bandwidth cost considering all the edges. For example, in the case of $M = 25$ and $N = 40$, we saw that the bandwidth cost was minimal with non even data distribution to the edges. The reason for this is that evenly distributing data results in data being deployed on some edges, which incurs more cost, which is something not considered when using the objective function that minimizes max bandwidth utilization objective. Minimizing bandwidth cost is a good objective when pricing is attached to allocation of bandwidth, while minimizing max bandwidth utilization is a good objective when it is necessary to balance the bandwidth load over all the edges.

In Fig. 3a, the minimum max bandwidth utilization for $k_j = 35$ is very close for $N = 20$ and $N = 40$. Additionally, the increase in minimum max bandwidth utilization is very less with increase in the values of N for $k_j = 35$. This characteristic does not hold in the case of minimum bandwidth cost objective, where there is higher cost difference between $N = 20$ and $N = 40$ for $k_j = 35$. This is because the first objective does not take into consideration the extra cost incurred by adding the bandwidth load for new vehicles due to deployment of data on edges with lower utilization. Therefore, the minimum max bandwidth utilization value is not affected significantly whereas there is significant increase in the minimum bandwidth cost.

C. Data+Service Delivery: Variation of objective 2 with varying percentage of vehicles requiring service

In this section, we consider a practical scenario where there are vehicles requiring both data and service. We use the experiment in Section V-B as the starting point where all the vehicles require data and increase the percentage of vehicles that require delivery of some service. The parameters used in this experiment are as follows

1) The parameters for vehicles requesting data remains the same from previous experiment.
2) The computation capacity of the edges (P_j in number of VMs) was randomly generated using uniform distribution between 24 and 40.
3) P_j^{occ} was randomly generated between 1 and 3, p_i was randomly generated between 1 and 10.
4) t_{p_i} was randomly generated between 1 and 10 sec.
5) d_i uses the same values generated for M_i in Section V-A, while r_i is randomly generated between 1 and 25 Mbits.

Due to space constraints we do not plot the graphs for this result and give a brief overview of the results obtained only for one problem size. The percentage of vehicles requiring service was varied between 10% and 40% for $M = 49$ and $N = 120$. The density values used were $k_j = 35$ and $k_j = 30$. The two results observed in this experiment are described below. The bandwidth cost increased only slightly with increase in percentage of vehicles requiring service delivery for $k_j = 35$, i.e, total bandwidth cost increased from 94.96 (only vehicles requiring data delivery) to 95.57 (40% vehicles requiring service delivery). The amount of increase in bandwidth cost depends on the values of d_i and r_i. The larger these values, higher will be the increase in bandwidth cost. For $k_j = 30$, the bandwidth cost increase was small, but feasible solution ended after 20% increase in vehicles requiring service delivery.

VI. Conclusion

In this work, we introduced a centralized optimization framework for data/service delivery to connected vehicles via edges. Our optimization framework introduced constraints necessary for both data and service delivery. In addition, we integrated a well known vehicle flow model into our optimization framework. This allowed us to optimize the delivery of data/services for a bandwidth cost function considering the traffic flow. The experiments demonstrated the effect of considering the vehicle flow parameters in bandwidth optimal data/service delivery.

Acknowledgment

This research was supported in part by NSF/Intel grant CNS-1505799, a gift from Toyota ITC, ONR N00014-16-1-2195, and Global Research Laboratory Program (2013K1A1A2A02078326) through NRF, and the DGIST Research and Development Program (CPS Global Center) funded by the Ministry of Science, ICT & Future Planning.

References

[1] Updating car software: Why delta technology is better than compression. http://www.embedded-computing.com/embedded-computing-design/updating-car-software-why-delta-technology-is-better-than-compression.

[2] U.S. department of transportation, connected vehicle pilot deployment program. https://www.its.dot.gov/pilots/.

[3] S. Barbarossa, S. Sardellitti, and P. Di Lorenzo. Communicating while computing: Distributed mobile cloud computing over 5G heterogeneous networks. *IEEE Signal Processing Magazine*, 31(6):45–55, 2014.

[4] M. T. Beck, M. Werner, S. Feld, and S. Schimper. Mobile edge computing: A taxonomy. In *Proc. of the Sixth International Conference on Advances in Future Internet*, pages 48–55, 2014.

[5] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli. Fog computing and its role in the internet of things. In *Proceedings of the first edition of the MCC workshop on Mobile cloud computing*, pages 13–16, 2012.

[6] A. Ghavami, Z. Li, and H. Shen. On-demand bandwidth pricing for congestion control in core switches in cloud networks. In *Cloud Computing (CLOUD), 2016 IEEE 9th International Conference on*, pages 867–870, 2016.

[7] B. Greenshields, W. Channing, H. Miller, et al. A study of traffic capacity. In *Highway research board proceedings*, volume 1935. National Research Council (USA), Highway Research Board, 1935.

[8] X. Guo, R. Singh, T. Zhao, and Z. Niu. An index based task assignment policy for achieving optimal power-delay tradeoff in edge cloud systems. In *Communications (ICC), 2016 IEEE International Conference on*, pages 1–7, 2016.

[9] Y. C. Hu, M. Patel, D. Sabella, N. Sprecher, and V. Young. Mobile edge computing a key technology towards 5G. *ETSI white paper*, 11(11):1–16, 2015.

[10] J. Jeong, S. Guo, Y. Gu, T. He, and D. H. Du. Trajectory-based statistical forwarding for multihop infrastructure-to-vehicle data delivery. *IEEE Transactions on Mobile Computing*, 11(10):1523–1537, 2012.

[11] P. Mach and Z. Becvar. Mobile edge computing: A survey on architecture and computation offloading. *IEEE Communications Surveys & Tutorials*, 19(3):1628–1656, 2017.

[12] J. Oueis, E. C. Strinati, and S. Barbarossa. Small cell clustering for efficient distributed cloud computing. In *Personal, Indoor, and Mobile Radio Communication (PIMRC), 2014 IEEE 25th Annual International Symposium on*, pages 1474–1479, 2014.

[13] M. Satyanarayanan, P. Bahl, R. Caceres, and N. Davies. The case for VM-based cloudlets in mobile computing. *IEEE pervasive Computing*, 8(4), 2009.

[14] C. Shi, V. Lakafosis, M. H. Ammar, and E. W. Zegura. Serendipity: enabling remote computing among intermittently connected mobile devices. In *Proceedings of the thirteenth ACM international symposium on Mobile Ad Hoc Networking and Computing*, pages 145–154, 2012.

[15] W. L. Tan, W. C. Lau, O. Yue, and T. H. Hui. Analytical models and performance evaluation of drive-thru internet systems. *IEEE Journal on selected areas in Communications*, 29(1):207–222, 2011.

[16] S. S. Tanzil, O. N. Gharehshiran, and V. Krishnamurthy. Femtocloud formation: A coalitional game-theoretic approach. In *Global Communications Conference (GLOBECOM), 2015 IEEE*, pages 1–6, 2015.

[17] L. Wang, L. Jiao, J. Li, and M. Mühlhäuser. Online resource allocation for arbitrary user mobility in distributed edge clouds. In *Distributed Computing Systems (ICDCS), 2017 IEEE 37th International Conference on*, pages 1281–1290, 2017.

[18] Y. Zhang, H. Liu, L. Jiao, and X. Fu. To offload or not to offload: an efficient code partition algorithm for mobile cloud computing. In *Cloud Networking (CLOUDNET), 2012 IEEE 1st International Conference on*, pages 80–86, 2012.

[19] T. Zhao, S. Zhou, X. Guo, Y. Zhao, and Z. Niu. A cooperative scheduling scheme of local cloud and internet cloud for delay-aware mobile cloud computing. In *Globecom Workshops (GC Wkshps), 2015 IEEE*, pages 1–6, 2015.