Vehicle-to-Vehicle Forwarding in Green Roadside Infrastructure

Morteza Azimifar[1], Terence D. Todd[1], George Karakostas[2] and Amir Khezrian[1]

[1]Department of Electrical and Computer Engineering
McMaster University
Hamilton, Ontario, CANADA
Email: todd@mcmaster.ca

[2]Department of Computing and Software
McMaster University
Hamilton, Ontario, CANADA
Email: karakos@mcmaster.ca

Abstract

Smart scheduling can be used to reduce infrastructure-to-vehicle energy costs in delay tolerant vehicular networks. In this paper we show that by combining this with vehicle-to-vehicle (V2V) forwarding, downlink traffic schedules can be generated whose energy costs are lower than that in the single hop case. This is accomplished by having the roadside units (RSUs) dynamically forward packets through vehicles which are in energy favourable locations. The paper considers both constant bit rate (CBR) and variable bit rate (VBR) air interface options. We first derive offline schedulers for the downlink RSU energy usage when V2V forwarding is added to RSU-to-vehicle communication. Both in-channel and off-channel forwarding cases are considered. The CBR and VBR cases are obtained using integer linear program and time expanded graph formulations, respectively. These schedulers provide lower bounds on energy performance and are used for comparisons with a variety of proposed online scheduling algorithms. The first algorithm is based on a greedy local optimization (GLOA). A version of this algorithm which uses a minimum cost flow graph scheduler is also introduced. A more sophisticated algorithm is then proposed which is based on a finite window group optimization (FWGO). Results from various experiments show that the proposed algorithms can generate traffic schedules with much improved downlink energy requirements compared to the case where V2V packet forwarding is not used. The performance improvements are especially strong under heavy loading conditions and when the variation in vehicle communication requirements or vehicle speed is high. Results are also presented which compare the proposed algorithms with conventional non-energy aware schedulers.

I. Introduction

Vehicular ad hoc networks (VANETs) will eventually enable a wide variety of applications ranging from road safety, to those which include context-aware advertising and Internet media streaming. This is currently being facilitated through spectral licensing in the 5.9 GHz frequency band [1] and via vehicular specific standards such as Wireless Access in Vehicular Environment (WAVE), which is based on the IEEE 802.11 wireless LAN standard [2].

VANET operation includes two communication modes, vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I). In the V2I case, fixed infrastructure will be installed at roadside locations that will enable many

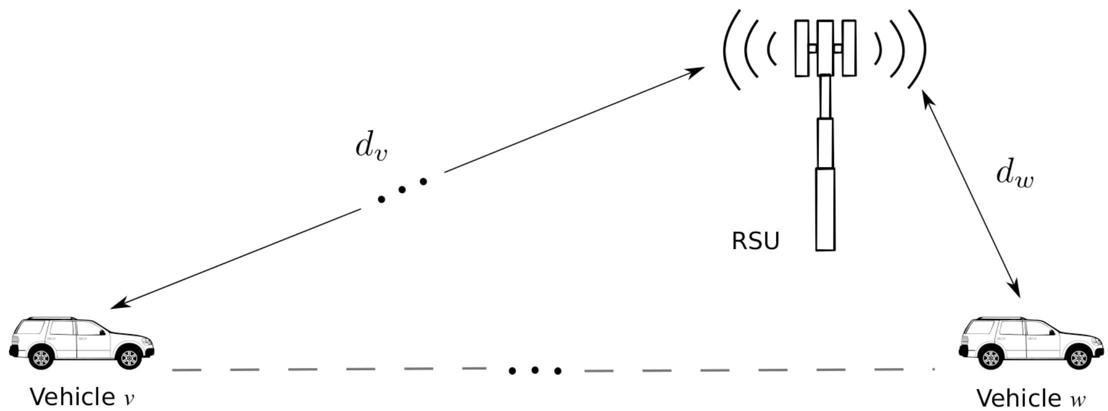

Fig. 1: Roadside Unit (RSU) Single Hop Example. Vehicles v and w are served by a single RSU as they pass through its coverage area at time t. In the single hop RSU-to-vehicle case, communication with Vehicle w at time t is preferable if $d_v \gg d_w$, since v and w are moving towards, and away from, the RSU, respectively.

new types of applications. Unlike traditional wireless networks, vehicles may only remain within roadside infrastructure radio coverage range for relatively short periods of time. This property raises the question of how vehicles should be served when passing through roadside unit coverage areas. Extensive research has already considered this problem and various schedulers have been proposed. These are discussed in Section I-A.

A serious barrier to significant roadside infrastructure deployment is the cost of providing electrical power to the deployed nodes. In some situations, electrical power is readily available. However, in many locations the associated costs are far too prohibitive to be commercially feasible. A viable alternative to power grid connections is to power the road-side units (RSUs) using green energy sources, such as solar energy. In [3] a study was presented by the US Department of Transportation's Vehicle Infrastructure Integration (VII) Initiative. This report includes cost estimates for a significant vehicular infrastructure build-out that would initially focus on vehicle monitoring and safety. It was estimated that about 40% of all rural free-way roadside infrastructure would have to be solar powered, and that roughly 63% of the roadside unit costs would be associated with solar energy provisioning, i.e., mainly panel and battery costs. The total cost estimates come to just under 1B dollars for the initial deployment round. Clearly, these costs can be reduced significantly using energy efficient roadside infrastructure designs. In vehicular infrastructure, proper traffic scheduling can lead to significant improvements in energy efficiency due to the strong dependence of power consumption on RSU-to-vehicle distance [4].

The position of vehicles passing by a given RSU can often be very accurately predicted, especially in highway settings where a given vehicle's speed remains relatively constant. In [4] it was shown that

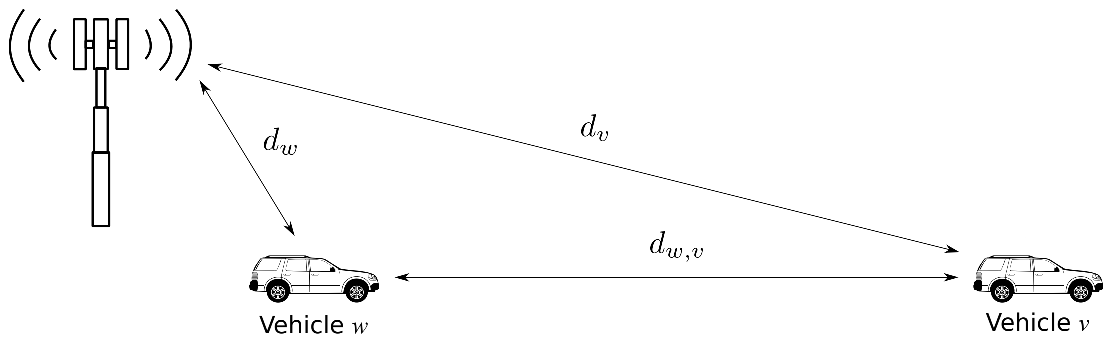

Fig. 2: Roadside Unit (RSU) V2V Multihop Forwarding Example. Instead of direct RSU-to-vehicle transmission (i.e., RSU-to-v), the RSU transmits via multihop forwarding through vehicle w (i.e., RSU-to-w followed by w-to-v forwarding). This may offer an energy savings advantage since $d_w \ll d_v$.

when there is application delay tolerance, this information can be used to reduce downlink RSU energy use. Figure 1 shows a simple example of this, where at time t, the RSU can choose to communicate with either vehicle v or w. In this example, vehicles v and w are moving towards, and away from, more energy favourable locations, respectively, i.e., d_v is decreasing and d_w is increasing. Combining this with the fact that $d_v \gg d_w$, the RSU would prefer to communicate with vehicle w from a downlink energy viewpoint.

The example in Figure 1 assumes the single-hop V2I communication case. When this is augmented with V2V forwarding however, addition energy savings may be possible. Consider the example shown in Figure 2 where the RSU wishes to communicate with vehicle v. Rather than transmitting directly over distance d_v, it may be possible to forward the transmission using multihop communication through vehicle w. So long as $d_w < d_v$, this may result in a decreased RSU energy cost compared to direct communication. Note that vehicles have virtually unlimited energy reserves, and therefore the energy cost of using the w-to-v forwarding link (shown in the figure as distance $d_{w,v}$) does not impact the RSU energy cost. It can therefore reduce energy use by forwarding packets through vehicles which are in energy favourable locations. When the w-to-v forwarding occurs on the same channel as the RSU-to-w transmission, it is referred to as *in-channel forwarding*. Conversely, when the V2V forwarding occurs on a separate channel, it is referred to as *off-channel forwarding*.

In this paper we show that by combining smart RSU scheduling with vehicle-to-vehicle packet forwarding, downlink traffic schedules can be generated whose energy costs are significantly lower than that described in [4]. The paper focuses exclusively on RSU-to-vehicle traffic flow where packets are sourced at the RSU and delivered to the passing vehicles, i.e., traffic flow is not considered which includes vehicle-to-vehicle generated traffic. Our results show that when this single hop transmission is all that is needed, and therefore when V2V would not be useful for capacity improvement, combining RSU-to-vehicle transmission with multi-hop V2V forwarding can result in decreased RSU energy use.

We first derive offline schedules for the downlink RSU energy required when V2V forwarding is enabled. These schedules provide lower bounds on downlink RSU energy requirements, and are derived for both the off-channel and in-channel forwarding cases. A formulation based on time expanded graphs is developed for the in-channel forwarding case. Both the constant bit rate (CBR) and variable bit rate (VBR) air interface options are considered. The offline scheduling bounds are used for comparisons with a variety of online schedulers.

The paper then introduces online scheduling algorithms. The first algorithm uses a greedy local optimization (GLOA) when creating schedules. A version of the algorithm is also given which uses a more sophisticated minimum cost flow graph scheduler. An algorithm is then proposed which is based on a finite window group optimization (FWGO) where vehicle groups are formed and scheduled jointly. Versions of these algorithms are proposed which use in-channel and off-channel vehicle-to-vehicle forwarding. The proposed algorithms are also adapted to the variable bit rate air interface case. Results from a variety of experiments show that the proposed scheduling algorithms can significantly improve the energy cost of the generated downlink traffic schedules compared to the case where vehicle-to-vehicle packet forwarding is not used. The performance improvements are especially strong under heavy loading conditions and when the variation in vehicle communication requirements or vehicle speed is high.

The remainder of the paper is organized as follows. In Section I-A an overview is given of related work. In Section II we give a detailed description of our system assumptions. Then in Section III we formulate offline schedules which provide bounds on the energy needed to satisfy downlink vehicular communication requirements. Both constant bit rate (CBR) and variable bit rate (VBR) air interface assumptions are considered. Section IV then presents various online scheduling algorithms for both the CBR and VBR cases. These algorithms include those based on schedulers derived using minimum cost flow graph (MCFG) and time expanded graph (TEG) constructions. Performance results are then presented in Section V which show that under certain conditions, significant reductions in roadside infrastructure energy use can be obtained using the proposed schedulers. Comparisons are also made with two non-energy aware schedulers. In Section VI, the final conclusions of our work are presented.

A. Related Work

Research in vehicular networks has been very active in the past several years. This activity has included topic areas as varied as applications [5], routing algorithms [6], security [7], and the performance of standards such IEEE 802.11p [8]. Studies have also explored the use of IEEE 802.11p in highway settings [9] [10], and in references [11], [12] and [13], protocols were proposed to improve roadside unit utilization, using vehicle-to-vehicle interaction.

Transmitter power control is a common method for adjusting between co-channel interference and network connectivity in vehicular ad hoc networks [8] [14] [15]. Energy efficiency in VANETs however, has mainly not been considered an issue, since vehicle radios are powered by the car engine, which has enormous energy reserves compared to radio energy costs. In addition, much of the previous infrastructure related work has assumed that grid connected power is available at the RSUs.

Simple road-side unit schedulers were introduced in reference [16] based on job size and deadline inputs. RSU power consumption however, was not considered in this study. Reference [17] used an optimization which maximizes the total throughput of an RSU, given vehicular location and speed inputs. In reference [13], the use of vehicle-to-vehicle forwarding was considered as a way of reducing the access load in an infrastructure based content delivery network. In these references, the energy consumption of the RSU, which is the focus of our paper, was not taken into consideration.

Energy efficient RSU scheduling has been studied in Reference [18]. Given a set of RSUs, this work formulates an optimization that minimizes RSU energy consumption by switching RSUs on and off, subject to maintaining full network connectivity. However, unlike our paper, they do not schedule vehicle-to-infrastructure communication.

II. System Model

A single Roadside Unit (RSU) with a coverage range of D_{max} is assumed, that is serving moving vehicles as shown in Figures 1 and 2. The figures show vehicles moving in one direction. However, the algorithms in this paper are also applicable to multi-way vehicular traffic. We consider the energy use of the radio interface that the RSU uses to communicate with the vehicles on the downlink (i.e., RSU-to-vehicle) direction. Since the vehicle radios are powered by the car's engine, we are not concerned about vehicular energy efficiency. The objective of the RSU is to minimize its energy use subject to satisfying the communication requests associated with vehicles passing through its coverage area. It is also assumed that vehicular communication may occur any time throughout a given vehicle's RSU transit time, that is, the communication is delay tolerant. Note that RSU-to-vehicle traffic flow is considered, i.e., packets are sourced at the RSU and delivered to the passing vehicles.

When a vehicle first enters the coverage area of the RSU, estimates are given to the scheduler which indicate the energy costs associated with downlink communication to the arriving vehicle. In [19] it was shown that in highway settings where vehicle speed is relatively constant, good energy estimates can be obtained based on current position and speed. Given these estimates, our problem is to define an RSU downlink transmission schedule such that downlink (DL) energy use is minimized. This energy consumption may be modeled differently depending on the type of air interface that is being used. Two cases are considered, as follows.

1) Constant Bit Rate (CBR): In the CBR case, power control is used to maintain a fixed bit rate over the RSU-to-vehicle links as the channel path loss changes. In this case the objective is to schedule downlink transmissions such that the required average downlink *power consumption* is minimized. In this model, channel time is assumed to be a contiguous stream of time slots, where each time slot can carry B bits, regardless of vehicle position within the coverage area of the RSU. The objective of a scheduler is to satisfy incoming downlink vehicular requests with the minimum possible energy.
2) Variable Bit Rate (VBR): In the VBR case, the transmission power used by the RSU is assumed to be fixed, and changes in channel path loss are compensated for by varying the bit rate used over the RSU-to-vehicle links. A given packet transmission may therefore occupy a different amount of time based on the bit rate used when it is transmitted. For this reason, it is convenient to consider the time-line as a contiguous stream of frame epochs, where frame f is of duration τ_f[1]. Note that the frames are defined on the downlink but are only used by the RSU for scheduling convenience, i.e., they do not have to exist on the radio air interface itself. The only requirement is that the frames it uses for scheduling are short enough that the bit rates available for a given station do not change significantly in the frame. When scheduling is performed, the scheduler must ensure that the total duration of packet transmissions occurring in frame f must not exceed τ_f. In this case the objective of a scheduler is to schedule downlink transmissions such that the *total time* that the RSU spends transmitting is minimized.

In both cases it is usually advantageous to serve vehicles when they are as close as possible to the RSU. This results in lower transmit power requirements in the CBR case, and correspondingly higher data rates in the VBR case. This will tend to reduce downlink power consumption or downlink transmission time, respectively. In both cases the transmit power or bit rate needed can be set in a variety of ways such as

[1]For generality we assume that frames can have different durations. In practical systems such as IEEE 802.11, frame times of this type are typically fixed in duration.

using a short two-way handshake prior to actual vehicle data packet transmission.

In Figure 2 we showed the case where downlink transmission may also occur via vehicular multihop forwarding. For simplicity, at first we will assume the off-channel forwarding case where V2V forwarding is done using a separate radio. However, offline schedules and online schedulers will also be considered for the in-channel forwarding case. Given the appropriate energy cost inputs, our problem is to generate energy efficient downlink traffic schedules. The inputs and outputs of our scheduling problem are summarized more formally as follows.

INPUT: We consider a finite sequence of V arriving vehicles indexed by the set $\mathcal{V} = \{1, 2, \ldots, V\}$, where each vehicle v has a given RSU-to-vehicle communication requirement, R_v bits. The entire time considered in the CBR case consists of T time slots given by the set $\mathcal{T} = \{1, 2, \ldots, T\}$ over which the scheduling is to occur. In the VBR case the time considered is the set of frames $\mathcal{F} = \{1, 2, \ldots, F\}$ and their duration, τ_f for all $f \in \mathcal{F}$. We are also given estimates of the energy costs (for CBR) or achievable bit rates (for VBR) for downlink communication from the RSU to vehicle v in time slot t, i.e., $\epsilon_{v,t}$ (for CBR) or the available bit rate in frame f (for VBR), $B_{v,f}$[2]. The scheduler is also given estimates of the time slots (CBR) or frame periods (VBR) during which vehicles are within multihop forwarding range.

OUTPUT: A scheduler output gives an RSU-to-vehicle transmission schedule. Given the inputs discussed above, the objective of the scheduler is to find a downlink transmission schedule such that all vehicle communication requirements are satisfied, and the downlink RSU energy cost is minimized.

In the CBR case this is accomplished by assigning values to the following binary variables

$$x_{w,v,t} = \begin{cases} 1 & \text{if the RSU sends indirectly to vehicle } v \text{ via direct transmission to vehicle } w \text{ in time slot } t, \\ 0 & \text{otherwise.} \end{cases} \quad (1)$$

Note that when $x_{v,v,t} = 1$, the RSU transmits directly to vehicle v in time slot t, i.e., without any multihop communication.

In the VBR case, the system is modeled using a sequence of frames in time, denoted by the set $\mathcal{F} = \{1, 2, \ldots, F\}$. In this case a scheduler must specify values for the variables, $z_{v,f}$, which are defined to be the number of bits sent from the RSU to vehicle v in frame f. The value of $z_{v,f}$ must

[2] In this paper we focus exclusively on the scheduling problem itself, i.e., given sets of energy cost estimates, what is the best schedule that can be found?

TABLE I: Variable Definitions

$\mathcal{V}$	Set of Vehicles
$\mathcal{G}$	Vehicle group
R_v	Vehicle job sizes
H_v	Backlog at vehicle v
S_v	Speed of vehicle v
CBR Air Interface	
$\mathcal{T}$	Set of time slots
$\epsilon_{v,t}$	Energy cost estimates for RSU-to-vehicle v in time slot t
$\mathcal{T}_{v,w}$	Set of time slots for which vehicle v and w are in communication range
VBR Air Interface	
$\mathcal{F}$	Set of frames
$z_{v,f}$	Bits sent from RSU to vehicle v in frame f
$y_{v,w,f}$	Bits sent from vehicle v to vehicle w in frame f
$B_{v,w,f}$	V2V bit rate between v and w in frame f
$B_{v,f}$	Bit rate for vehicle v in frame f (RSU-to-vehicle communication)
τ_f	Frame duration

be such that when bits are forwarded to other vehicles, the number arriving at each final destination must satisfy its communication requirement.

The schedulers to be considered can operate either *offline* or *online*. An *offline scheduler* is provided with the complete input sample function, which it then uses to generate an offline schedule. This is used to derive lower bounds on energy performance in Section III. In contrast, an *online scheduler* is provided the inputs in real time and it must make causal downlink transmission decisions. In Section IV, online scheduling algorithms are proposed. When defining the offline schedulers in Section III, T and F are set to include the entire input traffic sample function, so that lower bounds on performance can be obtained. In the online algorithms introduced in Section IV, T and F are set to include the current scheduling window of known vehicles. In the following section, we formalize these definitions and derive offline bounds on the energy needed for RSU-to-vehicle communication, for both the CBR and VBR cases. As an aid to the reader, the commonly used variable definitions are listed in Table I.

III. Offline Energy Scheduling Bounds

A. Constant Bit Rate (CBR) Energy Bound

In this section we present an integer linear programming (ILP) formulation which can be used to find a lower bound on the energy needed for the time-slotted CBR case. It is clear from the example shown in Figure 2 that if $d_v \gg d_w$, the downlink RSU energy cost associated with multihop communication may be much less than that for direct RSU-to-vehicle communication. This will be so if the energy cost of the RSU-to-vehicle w link is lower than that for vehicle v since the vehicle-to-vehicle hop does not affect the RSU energy consumption. For this reason, the RSU prefers to communicate with nearby vehicles rather than more distant ones. However, this scheduling must be done in a way which guarantees that the packet reception requirements of the vehicles are satisfied.

Given the RSU energy cost inputs discussed previously, the objective is to define an algorithm which will schedule energy efficient downlink transmissions. For a given input sample function, a lower bound is formulated for the energy needed to satisfy downlink vehicular requests. We assume that there are V vehicles indexed by the set $\mathcal{V} = \{1, 2, \ldots, V\}$ and there are T time slots given by the set $\mathcal{T} = \{1, 2, \ldots, T\}$ over which the scheduling is to take place. R_v is defined to be the communication requirement for vehicle v in bits. We also define $\mathcal{T}_{v,w}$ to be the set of time slots for which vehicles v and w are within V2V forwarding range and $\epsilon_{v,t}$ is the single time slot energy cost of downlink communication to vehicle v at time t. The inputs to the problem are therefore given by the set of n-tuples

$$\mathcal{I} = \{(R_v, \epsilon_{v,t}, \mathcal{T}_{v,w})\} \quad \forall v, w \in \mathcal{V},\ \forall t \in \mathcal{T} \tag{2}$$

and a lower bound on total energy use can be computed using the following integer linear program (ILP).

$$\underset{x_{w,v,t}}{\text{minimize}} \quad \sum_{t\in\mathcal{T}} \sum_{v\in\mathcal{V}} \sum_{w\in\mathcal{V}} \epsilon_{w,t}\ x_{w,v,t} \tag{CBR-BOUND}$$

$$\text{subject to} \quad \sum_{w\in\mathcal{V}} \sum_{t\in\mathcal{T}_{w,v}} x_{w,v,t} = \lceil R_v/B \rceil, \quad \forall v \in \mathcal{V} \tag{3}$$

$$\sum_{v\in\mathcal{V}} \sum_{w\in\mathcal{V}} \sum_{t\notin\mathcal{T}_{w,v}} x_{w,v,t} = 0, \tag{4}$$

$$\sum_{v\in\mathcal{V}} \sum_{w\in\mathcal{V}} x_{w,v,t} \leq 1, \qquad \forall t \in \mathcal{T} \tag{5}$$

$$x_{w,v,t} \in \{0, 1\}, \qquad \forall v, w \in \mathcal{V}, \forall t \in \mathcal{T} \tag{6}$$

In CBR-BOUND, the objective function is simply the downlink energy used by the RSU, i.e., via first-hop transmission from the RSU to vehicle w. Constraint 3 ensures that vehicle communication requirements are fulfilled by summing the appropriate values of $x_{w,v,t}$ but only over the time intervals for which vehicles v and w are within forwarding range. The values of $x_{w,v,t}$ when $t \notin \mathcal{T}_{v,w}$ are set to zero as shown in Constraint 4. Constraint 5 ensures that a given time slot can contain at most one transmission. Note that in CBR-BOUND we have assumed that V2V forwarding occurs on a separate communication channel from that which is used for RSU-to-vehicle transmission since we do not include a constraint to prohibit V2I communication at the same time as V2V, and vice versa. This result, however, is clearly also a lower bound on energy use when V2V and V2I communication occurs over the same channel.

CBR-BOUND can be solved directly using branch and bound techniques, and CPLEX has been used with data generated from MATLAB. These results are used for comparisons with on-line algorithms to be introduced in Section IV.

B. Time Expanded Graph (TEG) Energy Bounds

In this section we formulate a more complex energy bound that can be used in both the CBR and VBR cases that includes in-channel V2V forwarding. The formulation uses *time expanded graphs* (TEG) [20]. Unlike static network flow problems, time expanded graphs model the temporal evolution of the static case and can account for flow storage. This is done by modeling the system as a sequence of frames in time. In the general case, the frames can be variable in length, where τ_f is the duration of frame f, as discussed in Section II. $\mathcal{F} = \{1, 2, \ldots, F\}$ is defined to be the set of all frames to be scheduled. A simple illustration of how TEGs are used for our problem is shown in Figure 3. The figure shows three frames and three vehicles, numbered 1, 2 and 3. The TEG models the evolution of RSU and vehicular connectivity from one frame to the next. The vertical edges connect different time instances of the same vehicle and are used to model the carrying of flow forward in time. In the figure, vehicles 1 and 3 are communicated to directly from the RSU, whereas vehicle 2 receives its flow from V2V communication through vehicles 1 and 3. The graph shown is directed, with the direction of the vertical edges being from top to bottom and the horizontal edges between vehicles being bi-directional. Node D is the sink. RSU$_f$ is defined to be the RSU node in frame f and $V_{i,f}$ is defined to be the vehicle i node in frame f. The edges emanating from the RSU nodes have a cost $c_{i,f}$ which is equal to the *reciprocal* of the bit rate that the RSU can use to communicate with vehicle i in frame f. All other edge costs are zero. In this example, Vehicles 1, 2 and 3 require 5, 10 and 10 units of flow, in bits, respectively. The arrows show those graph edges which are carrying non-zero units of flow. In frame 1, 25 units of flow arrive at the RSU which transmits 15 to vehicle 3. Ten of these are destined for vehicle 3 and the other 5 are destined for vehicle 2. Similarly, in frame 3, the RSU transmits the remaining 5 units of flow directly to vehicle 1. Note that the use of vertical graph edges correspond to the act of *storing flow* at the associated node, and carrying it forward in time. In frame 1, 5 units of flow for vehicle 2 are sent via vehicle 1, which is

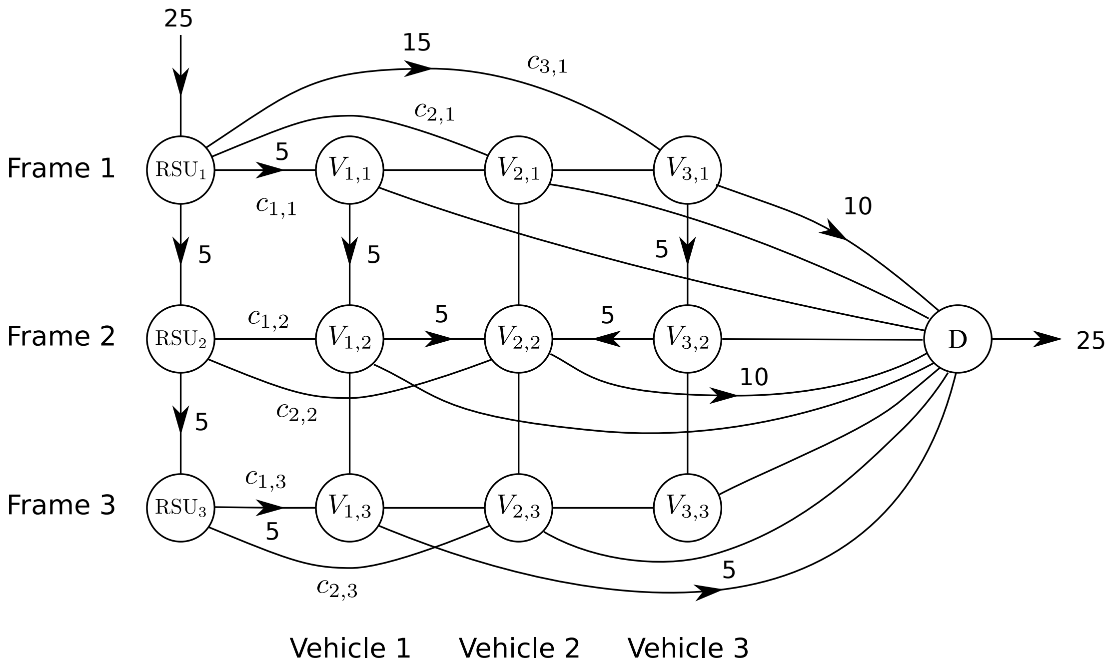

Fig. 3: Time Expanded Graph (TEG) Example. The TEG models the temporal evolution of the network. Only three frames and three vehicles are shown.

forwarded by V2V communication to vehicle 2 in frame 2. Similarly, in frame 2, vehicle 3 forwards 5 units of flow to vehicle 2. In this example, vehicle 2 is provided for, solely by V2V forwarding.

In the TEG formulation, we model flow in units of bits. As before, we define $\mathcal{V} = \{1, 2, \dots, V\}$ to be the set of vehicles and R_v is the demand for vehicle v. $z_{v,f}$ is the number of bits sent from the RSU to vehicle v in frame f. $y_{i,j,f}$ is defined to be the number of bits sent from vehicle i to vehicle j in frame f. We will first show the variable bit rate case, and therefore we define $B_{v,f}$ to be the bit rate available between the RSU and vehicle v in frame f. $B_{i,j,f}$ is the bit rate available between vehicles i and j in frame f. Given these definitions, the TEG optimization is written as follows.

$$\underset{z_{v,f}, y_{i,j,f}}{\text{minimize}} \quad \sum_{f\in\mathcal{F}} \sum_{v\in\mathcal{V}} z_{v,f}/B_{v,f} \tag{TEG-BOUND}$$

$$\text{subject to} \quad \sum_{v\in\mathcal{V}} z_{v,f}/B_{v,f} + \sum_{i\in\mathcal{V}} \sum_{j\in\mathcal{V}} y_{i,j,f}/B_{i,j,f} \le \tau_f, \qquad \forall f \in \mathcal{F} \tag{7}$$

$$\sum_{f=1}^{\theta-1} (z_{v,f} + \sum_{i\in\mathcal{V}} y_{i,v,f}) - \sum_{f=1}^{\theta} \sum_{j\in\mathcal{V}} y_{v,j,f} \ge 0, \qquad \forall v \in \mathcal{V}, \forall \theta \in \{1, 2, \dots, F\} \tag{8}$$

$$\sum_{f\in\mathcal{F}} (z_{v,f} + \sum_{i\in\mathcal{V}} y_{i,v,f}) - \sum_{f\in\mathcal{F}} \sum_{j\in\mathcal{V}} y_{v,j,f} = R_v, \qquad \forall v \in \mathcal{V} \tag{9}$$

$$z_{v,f} \ge 0, \qquad \forall v \in \mathcal{V}, \forall f \in \mathcal{F} \tag{10}$$

$$y_{i,j,f} \geq 0, \qquad \forall i, j \in \mathcal{V}, \forall f \in \mathcal{F} \qquad (11)$$

The objective minimizes the total time needed by the RSU for downlink transmission. Constraint 7 ensures that the number of bits transmitted in each frame, f, does not exceed the frame duration, τ_f. The next two are flow conservation constraints. Constraint 8 ensures that for each vehicle v, the total number of bits vehicle v forwards to other vehicles by the end of frame θ must not exceed the number of bits that were received by v by the end of frame $\theta - 1$. The inequality permits a given vehicle to hold bits and carry them forward into future frames. Constraint 9 makes sure that at the end of the total time period, the difference between the flow into each vehicle node v and the flow out is the number of bits in its demand, R_v.

Note that only trivial changes to the above LP are needed to model the in-channel constant bit rate, variable packet length, case. Since the bit rates are fixed we define $B_{v,f} \triangleq B$ and $B_{i,j,f} \triangleq B$, for all v, i, j and f. The objective will become

$$\underset{z_{v,f}}{\text{minimize}} \sum_{f \in \mathcal{F}} \sum_{v \in \mathcal{V}} z_{v,f} \; \epsilon_{v,f} \qquad (12)$$

where $\epsilon_{v,f}$ is redefined as the per bit energy cost of transmitting to vehicle v in frame f. Constraint 7 will become

$$\sum_{v \in \mathcal{V}} z_{v,f} + \sum_{i \in \mathcal{V}} \sum_{j \in \mathcal{V}} y_{i,j,f} \leq B\tau_f, \quad \forall f \in \mathcal{F} \qquad (13)$$

Constraints 8 to 11 remain unchanged.

The above formulations permit the computation of offline schedules which provide bounds on the energy needed for the air interface assumptions that have been considered. They can also be used as a starting point for online algorithms which are discussed in the next section.

IV. Online Scheduling Algorithms

Section III defined offline schedulers which give lower bounds on the RSU energy needed to fulfill vehicular packet requirements. In this section we propose online algorithms which generate transmission schedules when the input data is given in real time. In the online case, the scheduling decisions can only be made using data consisting of past and current inputs. Two basic algorithms are defined, however, these are applied to both the CBR and VBR air interface cases, and can use either in-channel and off-channel forwarding. Before introducing the details of each algorithm, the basic ideas are given as follows.

- Greedy Local Optimization Scheduler (GLOA): When a vehicle arrives, it is assigned to a group of vehicles to which multihop forwarding is possible. Channel time is then greedily assigned to the group, using vehicle speed prioritization. The time assignment is based on minimum energy downlink RSU communication, i.e., the RSU forwards packets through the vehicle in the group for which energy cost is minimum. The ensuing transmissions are forwarded to the appropriate destination vehicle.
- Finite Window Group Optimization (FWGO): FWGO uses the same group assignment as in GLOA. However, instead of greedy scheduling, time is assigned using an optimization that includes all known vehicles and outstanding transmission backlogs.

The details of these algorithms are given below, first for the constant bit rate case, then for the variable bit rate option.

A. *Constant Bit Rate (CBR)*

1) Greedy Local Optimization Scheduler (GLOA): In the basic GLOA scheduler, vehicle-to-vehicle forwarding is assumed to occur on a separate channel than the one used for RSU-to-vehicle transmission. To accomplish this, arriving vehicles are formed into groups for combined RSU-to-vehicle and vehicle-to-vehicle forwarding. We consider the case where at some time instant a group of vehicles has been identified for scheduling. The scheduling is based on a greedy local optimization defined below. We denote $\mathcal{G}$ as the group of vehicles that will be scheduled and will engage in hybrid RSU-to-vehicle and vehicle-to-vehicle forwarding, $\mathcal{K}$ as the set of time slots *currently available* to the vehicles in set $\mathcal{G}$, i.e., the set of unscheduled time slots, H_i as the transmission requirement for vehicle i in units of time slots, and $\epsilon_{i,k}$ as the energy cost for RSU transmission to vehicle i in time slot k. We define $x_{i,k}$ as a binary variable equal to one if the RSU communicates to vehicle i in time slot k, and zero otherwise. We would like to minimize the energy needed to process this group of vehicles, i.e.,

$$\underset{x_{i,k}}{\text{minimize}} \quad \sum_{i\in\mathcal{G}}\sum_{k\in\mathcal{K}} x_{i,k}\,\epsilon_{i,k} \tag{14}$$

$$\text{subject to} \quad \sum_{k\in\mathcal{K}} x_{i,k} \geq H_i \qquad \forall i \in \mathcal{G} \tag{15}$$

$$\sum_{i\in\mathcal{G}} x_{i,k} \leq 1 \qquad \forall k \in \mathcal{K} \tag{16}$$

$$x_{i,k} \in \{0,1\} \qquad \forall i \in \mathcal{G}, \forall k \in \mathcal{K} \tag{17}$$

Note that

$$\min_{x_{i,k}} \sum_{k\in\mathcal{K}} \sum_{i\in\mathcal{G}} x_{i,k}\ \epsilon_{i,k} \tag{18}$$

$$\geq \min_{x_{i,k}} \sum_{k\in\mathcal{K}} \min_{i\in\mathcal{G}}\{\epsilon_{i,k}\} \sum_{i\in\mathcal{G}} x_{i,k} \tag{19}$$

$$= \min_{x_{i,k}} \sum_{k\in\mathcal{K}} \mathcal{E}_{\mathcal{G},k}\ X_{\mathcal{G},k} \tag{20}$$

where we have defined $X_{\mathcal{G},k} = \sum_{i\in\mathcal{G}} x_{i,k}$ and $\mathcal{E}_{\mathcal{G},k} = \min_{i\in\mathcal{G}}\{\epsilon_{i,k}\}$. Also note that $X_{\mathcal{G},k} \leq 1\ \forall k \in \mathcal{K}$. Since we allow vehicle-to-vehicle forwarding for group $\mathcal{G}$, we can aggregate the time slots needed by this group, and interpret $X_{\mathcal{G},k}$ as

$$X_{\mathcal{G},k} = \begin{cases} 1 & \text{if the RSU transmits to vehicle group } \mathcal{G} \\ & \text{in time slot } k, \\ 0 & \text{otherwise,} \end{cases} \tag{21}$$

and the optimization can then be written as

$$\begin{aligned} \underset{X_{\mathcal{G},i}}{\text{minimize}} \quad & \sum_{k\in\mathcal{K}} X_{\mathcal{G},k}\ \mathcal{E}_{\mathcal{G},k} && \text{(GROUP-OPT)} \\ \text{subject to} \quad & \sum_{k\in\mathcal{K}} X_{\mathcal{G},k} \geq \sum_{i\in\mathcal{G}} H_i && (22) \\ & X_{\mathcal{G},k} \in \{0,1\} \qquad \forall k \in \mathcal{K} && (23) \end{aligned}$$

The optimization GROUP-OPT reflects the fact that due to vehicle-to-vehicle sharing, a time slot available for use by a vehicle at time k incurs an RSU energy cost which is the minimum of the costs for all vehicles in the group. This is where the RSU energy advantage is obtained. For this reason the optimization need only allocate $\sum_{i\in\mathcal{G}} H_i$ of these time slots. This optimization can be easily solved by sorting the set

$$\{\min_{i\in\mathcal{G}}\{\epsilon_{i,k}\}\} \tag{24}$$

for all k in increasing order of energy cost and then by allocating those time slots corresponding to the first $\sum_{i\in\mathcal{G}} H_i$ entries.

If the group $\mathcal{G}$ consists of vehicles which are platooning together at the same speed, then there is very little energy gain to be made even if their job requirements are different. i.e., for a given k the minimum in Equation 24 is the same for all vehicles. The groups that are formed must instead give the

Algorithm 1 Greedy Local Optimization Scheduler (GLOA)

1: **for all** $t \in \{0, 1, \ldots\}$ **do**
2: **if** a vehicle v arrives to the system **then**
3: Find the largest vehicle group currently within RSU coverage range, $\mathcal{G}$, such that for all vehicles $w \in \mathcal{G}$, $\{t, t+1, \ldots, T_c\} \subset \mathcal{T}_{v,w}$, where $\mathcal{T}_{v,w}$ is the set of time slots for which vehicles v and w are within multihop communication range and T_c is a target contact duration, i.e., all vehicles in $\mathcal{G}$ remain within V2V communication range with v for a time duration T_c.
4: Define $\widetilde{H}_T$ to be the total residual backlog for the vehicles in $\mathcal{G}$, i.e., $\widetilde{H}_T = \sum_{i \in \mathcal{G}} \widetilde{H}_i$ where $\widetilde{H}_i$ is the residual backlog for vehicle i.
5: Assign the $\widetilde{H}_T$ backlog across the vehicles in $\mathcal{G}$ in proportion to each vehicle's relative speed, i.e., vehicle i is assigned a backlog fraction of $1/S_i \sum_{j \in \mathcal{G}} (1/S_j)$, where S_j is the speed of vehicle i.
6: Time slots are allocated using an earliest-RSU-arrival (ERA) prioritization.
7: **end if**
8: **end for**

system "diversity in energy". In the special case where vehicles travel at the same speed but are not platooning and have different job sizes, energy improvements are possible. For example, consider the case where $\mathcal{G}$ consists of 2 vehicles, V_1 and V_2. Also, assume that the system is lightly loaded so that all time slots are available. If the vehicles are sufficiently separated, then the set of $H_1 + H_2$ time slots where $\sum_{k \in \mathcal{K}} \min\limits_{i \in \mathcal{G}} \{\epsilon_{i,k}\}$ is minimized will be those time slots where V_1 and V_2 are closest to the RSU i.e., we can view the solution as the same as the case for two vehicles travelling in a disjoint fashion, each having a job size of $(H_1 + H_2)/2$.

The intuition above motivates the proposed scheduling algorithm, GLOA, as shown in Algorithm 1. The idea is that the energy costs for a vehicle group can be the minimum energy cost of all the group's vehicles, but these costs are time varying as the group moves through the RSU coverage area. However, note that the time that a fast moving vehicle spends in energy favourable locations is proportionately less than that of a slow moving vehicle in the same group. For this reason when the total remaining vehicular backlog is distributed across the group, it is done in decreasing proportion of vehicle speed.

Referring to Algorithm 1, GLOA operates as follows. When a vehicle v arrives at time t, a vehicle group is found such that vehicle-to-vehicle communication is possible for a contact duration time T_c. This is shown in Steps 1 to 3. If a group cannot be found then the vehicle initiates a new vehicle group of its own. In Step 4 the total residual backlog for the group is determined. This consists of the total number of remaining time slots needed for the group, denoted by $\widetilde{H}_T$. This is "assigned" across the vehicles in the group based on vehicle speed as shown in Step 5. Vehicles with lower speed are assigned a proportionately larger fraction of the total residual backlog, and the computations shown are truncated to the appropriate integer slot times. Note that this is a "virtual assignment" in that a time slot needed for vehicle i which is "assigned" to vehicle j means that the RSU communicates that packet to vehicle i through multi-hop forwarding via vehicle j. In the case where all vehicles in $\mathcal{G}$ are travelling at the same speed, the remaining backlog will be distributed uniformly across all $|\mathcal{G}|$ vehicles in the group. Finally, in Step 6 the vehicle demands are allocated across the desired time slots in order of earliest-RSU-arrival (ERA) priority, which is an energy efficient schedule; it is precisely the grouping of vehicles that V2V permits which allows this scheduling.

GLOA is the simplest and least computationally complex of the algorithms. The running time needed for each arriving vehicle is given by $\mathcal{O}(N \min\{N, T\})$ where N is the number of vehicles within range of the RSU and T is the number of time slots over which scheduling occurs. The min in this expression is because we must look ahead for at most N vehicles in a range of T time slots.

2) GLOA with Minimum Cost Flow Graph Scheduling (GLOA-MCFG): This algorithm is a more sophisticated version of GLOA, where step 6 is replaced with a minimum cost flow graph (MCFG) scheduler discussed in reference [4]. Although this scheduler is more complex, the schedule can be generated in time which is polynomial in the number of time slots to be scheduled. This algorithm is referred to as GLOA-MCFG.

The running time of GLOA-MCFG is given by $\mathcal{O}(N \min\{N, T\}) + F(N, T)$ where the first term is from the complexity expression for GLOA, and $F(N, T)$ is the time needed to solve the minimum cost flow graph for a window size of W time slots. As stated in the description above, $F(N, T)$ is polynomial in N and T.

3) Finite Window Group Optimization (FWGO): More sophisticated schedulers than the GLOA versions can be obtained by solving an ILP at each vehicular arrival over a finite window period consisting of the union of time slots for which the current set of vehicles is within RSU coverage. When a new vehicle v arrives, the algorithm proceeds as in Algorithm 1 up to and including Step 3. However, the time slot assignment is done using a minimum cost flow graph scheduler [4][21] using a simplification of CBR-BOUND, which is possible when the vehicle groups are constant throughout the time window considered. First we define $\mathcal{G}_v$ to be the vehicle group for vehicle v as discussed in Section IV-A1. Taking the objective function from CBR-BOUND and noting that $x_{w,v,t} = 0$ when $w \notin \mathcal{G}_v$, the objective of CBR-BOUND can be written as

$$\sum_{t \in \mathcal{T}} \sum_{v \in \mathcal{V}} \sum_{w \in \mathcal{G} v} \epsilon_{w,t} x_{w,v,t} \tag{25}$$

Algorithm 2 Finite Window Group Optimization (FWGO)

1: **for all** $t \in \{1, 2, \dots\}$ **do**
2:   **if** a vehicle v arrives to the system **then**
3:     Find the largest vehicle group currently within RSU coverage range, $\mathcal{G}_v$, such that for all vehicles $w \in \mathcal{G}$, $\{t, t+1, \dots, T_c\} \subset \mathcal{T}_{v,w}$ where T_c is a target contact duration, i.e., all vehicles in $\mathcal{G}_v$ remain within V2V communication range with v for a time duration T_c. (This is the same as step 3 in Algorithm 1.)
4:     Define $\mathcal{V}_t$ to be the current set of vehicles within RSU coverage.
5:     Solve optimization CBR-BOUND using the residual (unserved) vehicular traffic of vehicles $\mathcal{V}_t$ to obtain the minimum cost time slot assignments, $x_{\mathcal{G}_v,t}$. Note that when vehicles v and w are within V2V communication range for the entire time window, this optimization can be found in time which is polynomial in the number of time slots using a minimum cost flow graph (MCFG) using the construction shown in Figure 4.
6:     Execute the schedule obtained in step 5.
7:   **end if**
8: **end for**

and therefore

$$\sum_{t\in\mathcal{T}}\sum_{v\in\mathcal{V}}\sum_{w\in\mathcal{G}_v}\epsilon_{w,t}x_{w,v,t} \geq \sum_{t\in\mathcal{T}}\sum_{v\in\mathcal{V}}\{\min_{w\in\mathcal{G}_v}\epsilon_{w,t}\}\sum_{w\in\mathcal{G}_v}x_{w,v,t} \tag{26}$$

$$\triangleq \sum_{t\in\mathcal{T}}\sum_{v\in\mathcal{V}}\mathcal{E}_{\mathcal{G}_v,t}x_{\mathcal{G}_v,t} \tag{27}$$

where $x_{\mathcal{G}_v,t} \triangleq \sum_{w\in\mathcal{G}_v} x_{w,v,t}$ and $\mathcal{E}_{\mathcal{G}_v,t} \triangleq \min_{w\in\mathcal{G}_v}\epsilon_{w,t}$. The scheduling is now done in a group-wise fashion where $x_{\mathcal{G}_v,t}$ is equal to one if the RSU transmits to vehicle group $\mathcal{G}_v$ during time slot t, and zero otherwise. Similarly, $\mathcal{E}_{\mathcal{G}_v,t}$ is defined to be the minimum energy cost available between the RSU and the vehicles in group $\mathcal{G}_v$, at time t. The resulting ILP is as follows.

$$\underset{x_{\mathcal{G}_v,t}}{\text{minimize}} \quad \sum_{t\in\mathcal{T}}\sum_{v\in\mathcal{V}}\mathcal{E}_{\mathcal{G}_v,t}x_{\mathcal{G}_v,t} \tag{FWGO-ILP}$$

$$\text{subject to} \quad \sum_{t\in\mathcal{T}} x_{\mathcal{G}_v,t} = \widetilde{H}_v, \quad \forall v \in \mathcal{V} \tag{28}$$

$$\sum_{v\in\mathcal{V}} x_{\mathcal{G}_i,t} \leq 1, \quad \forall t \in \mathcal{T} \tag{29}$$

$$x_{\mathcal{G}_i,t} \in \{0,1\}, \quad \forall i \in \mathcal{V}, t \in \mathcal{T} \tag{30}$$

This ILP generates a minimum energy schedule for the inputs used at time t. A major advantage of this approach is that FWGO-ILP can be solved in time which is polynomial in the number of time slots, using a minimum cost flow graph construction [21]. An example of this scheduler is shown in Figure 4, where

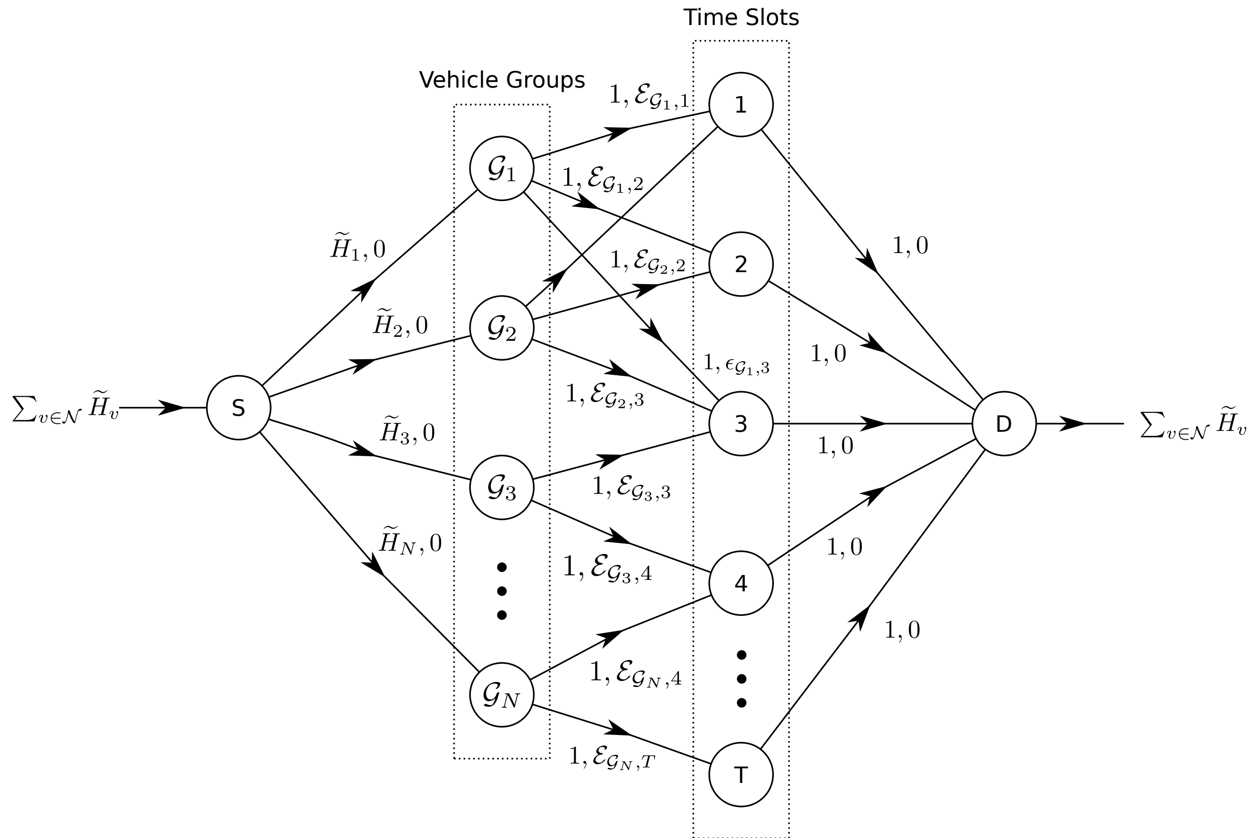

Fig. 4: Minimum Energy Vehicle Group Flow Graph Scheduler. Each edge is labeled with an ordered pair, $(u_{i,t}, c_{i,t})$, where $u_{i,t}$ and $c_{i,t}$ are the capacity and cost of using edge (i, t). The input and output nodes, I to O, carry a flow of $\sum_{v\in\mathcal{N}}\widetilde{H}_v$. They are connected to the rest of the graph by zero cost edges.

graph $G = (V, E)$ has a set of vertices, V, and a set of edges E [4]. Each edge $(i, j) \in E$ has a capacity $u_{i,j}$, and a cost, $c_{i,t}$, that denotes the energy penalty paid per unit of flow on that edge. These are written as ordered pairs $(u_{i,t}, c_{i,t})$ on each graph edge.

The flow enters and exits the graph at source and collection nodes S and D. The first column of nodes represents all N currently active vehicle groups. The second column represents all time slots $\mathcal{T}_t$, where $T = |\mathcal{T}_t|$. Each vehicle group node has edges connected to the time slot nodes during which vehicles in that group are inside the RSU coverage area with a cost, $\mathcal{E}_{\mathcal{G}_v,t}$, which is the minimum of the RSU to vehicle costs for group v as defined above. The capacity for an edge from the source S to a vehicle node is the residual communication requirement for vehicle v in time slots, denoted by $\widetilde{H}_v$.

The capacity for an edge from any time slot node to the destination D is 1, which prevents time slots from being used more than once. The edges between a vehicle i and the time slots also have a capacity of 1, which ensures that only one unit of transmission requirement can be assigned to a given time slot. The cost for the edges originating from S or terminating at D is zero. Finding the minimum cost flow for graph

G provides the minimum energy the RSU must consume to schedule vehicle transmission requirements for the given set of inputs. The Integrality Property [21] ensures that when the demand and capacities are integer, there is an integer minimum cost flow. Since our vehicle groups to time slot edge capacities are 1, the resulting flow paths are binary and give the values $x_{\mathcal{G}_v,t}$ which achieve minimum energy.

4) Off-Channel and In-Channel V2V Scheduling (GLOA-IC, GLOA-MCFC-IC and FWGO-IC): The algorithms introduced in sections IV-A1, IV-A2 and IV-A3 assume that the V2V relaying occurs on a separate channel than that of V2I communication, i.e., the off-channel forwarding case. To assess its potential advantage, we assume that this is done using a separate vehicle radio interface. Since V2V transmission will happen on a separate channel, there will be no contention between V2V and RSU-to-vehicle communication. For this reason, packet collisions which occur during forwarding do not affect the energy performance of the RSU. Assume vehicle v receives a packet from the RSU in time slot t (or in frame f, in the VBR case) and intends to send the packet to vehicle w via V2V forwarding. In our results we assume that the packet is forwarded off-channel immediately after it is received. Hence it is enough for the schedulers to ensure that no collision happens in RSU-to-vehicle transmissions.

We also consider in-channel V2V forwarding where vehicle-to-vehicle relaying occurs on the same channel. In this case the algorithms are identified by appending "-IC" (in-channel) to the algorithm name. Unlike the off-channel case, when in-channel V2V is used, the time slots needed are explicitly scheduled by the RSU using the online algorithms. In GLOA-IC and GLOA-MCFG-IC, at each vehicle arrival time we execute the algorithms as discussed in Sections IV-A1 and IV-A2. Once time slots have been assigned for RSU-to-vehicle transmission, they are excluded from consideration and V2V time slots are assigned using a simple higher-speed-first prioritization. In a similar way, FWGO-IC executes the FWGO Algorithm as given in Section IV-A3. Following this, the assigned V2I time slots are excluded and V2V time slots are assigned as in the GLOA-IC algorithm.

B. Variable Bit Rate (VBR)

In this section we adapt the algorithms introduced in Section IV-A for use in the variable bit rate air interface case. This involves minor changes to how the algorithms operate, which are discussed below.

1) GLOA, VBR Version (GLOA-VBR): This algorithm is a variation of GLOA applied to the VBR case. The algorithm is the same as Algorithm 1 except that time is allocated for each vehicle using higher bit rates associated with that vehicle and time whenever possible.

2) FWGO, VBR Version (FWGO-VBR): The algorithm given in Section IV-A3 is followed exactly except that time is allocated for each vehicle in accordance with the bit rate associated with that vehicle and time. The assignment to frames is done based on the time expanded graph formulation discussed in Section III-B.

An issue which may arise in both the GLOA-VBR and FWGO-VBR algorithms has to do with the nature of the output obtained from the time expanded graph formulation. A difference between the CBR and VBR models presented in Sections III-A and III-B respectively, is that the VBR flow formulation gives an output which is fractional. This is fine, since our units of flow in the VBR case are in bits; so, for the online algorithms, the fractional output can be simply rounded to the nearest integer. An issue which may arise is that the number of bits assigned to a given frame may be below a desired minimum packet length. We have found that for parameters of interest, this rarely happens, but when it does, the VBR algorithms round the assigned bits up to the minimum packet length and transmit them in the first available frame where they can be accommodated. This frame may, of course, be in a less energy favourable location than that where the bits were originally assigned.

The expressions for running time for these algorithms are the same as their CBR counterparts, but now instead of N and T in the previous expressions for complexity, we have NT, i.e., for GLOA-VBR the result is $\mathcal{O}((NT)^2)$ and for FWGO-VBR we have $\mathcal{O}((NT)^2) + F(NW)$, where W is the window size.

3) GLOA-IC and FWGO-IC, VBR: Just as we consider in-channel forwarding for the CBR case, the same is true for the VBR case. GLOA-IC-VBR is the same algorithm proposed in Section IV-B1. In GLOA-IC, however, after scheduling V2I transmissions, the scheduler will put aside frame time allocated for RSU-to-Vehicle communication and then assign V2V communication using the remaining capacity. The same goes for the FWGO-IC-VBR algorithm.

V. Performance Evaluation Examples

In this section, some representative examples of the performance of the proposed algorithms are presented. The simulation model is first described. Results are then presented in Section V-A for the CBR case where there is distance dependent exponential channel path loss. Then in Section V-B, results are given which include random components consisting of log-normal channel shadowing. In Section V-C, results are then given for the VBR air interface case. Finally, in Section V-D comparisons are made between two non-energy aware schedulers and our algorithms.

RSU coverage range, D_{max}	1000 m
V2V forwarding range	1000 m
Vehicle arrivals	Poisson process
Path loss exponent, α (CBR)	3
Vehicle bit rates (VBR)	IEEE 802.11p

TABLE II: Default Simulation Parameter Values

The input data to the schedulers is taken from a highway environment where vehicles are assumed to maintain constant speed throughout the RSU coverage areas [19][22]. It has been shown that in this type of scenario, good estimates of energy costs can be readily made [23][24]. The energy cost inputs are based on vehicle position and associated estimates of downlink transmission energy costs. We assume that the energy costs come from a distance dependent exponential path loss model with a path loss exponent of $\alpha = 3$. However, in many practical systems there will be dominant deterministic propagation with random components due to effects such as shadowing. Therefore, we also include results which incorporate errors due to strong shadowing components. The models used assume Poisson process vehicle arrivals as in [22], [24] and [25]. We also assume that T_c is set to the transit time of the given vehicle through the RSU coverage area, which is taken to be 1 km on either side of the RSU. In the VBR experiments, the IEEE 802.11p standard has been used for selecting the channel bit rates. The default parameters are summarized in Table II. Additional parameters used for the various experiments are given in the text or in the figure captions. The output of the schedulers consist of downlink traffic schedules whose total energy costs are plotted in the graphs. In our experiments, the vehicle forwarding range is chosen to be constant, and the same as that assumed for the RSU (i.e., 1 km). For comparison purposes, in the CBR case, the energy values plotted on each graph are normalized to the minimum energy cost achieved. In the VBR case, the total transmission time is normalized to the simulation run length. As discussed in Sections I and II, we assume that traffic flow is from the RSU to the passing vehicles, i.e., there is no vehicle-to-vehicle generated traffic.

A. CBR Results (Exponential Path Loss)

In Figure 5 the total downlink RSU energy use is plotted versus vehicular density for the constant bit rate case. The graph includes results for the offline bounds derived in Section III and we have included these bounds for the cases with and without V2V forwarding. It can be seen from the graph that vehicle-to-vehicle forwarding can significantly decrease the RSU energy, especially as vehicle density becomes

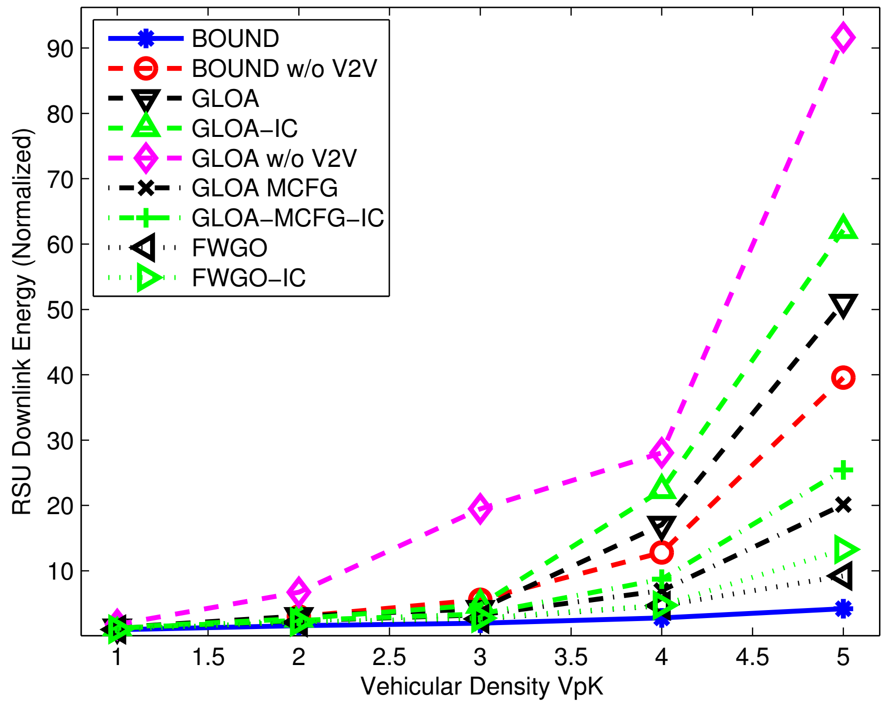

Fig. 5: Total RSU Downlink Energy vs. Vehicular Density (Vehicle per km). Speed is constant and equal to 20 m/sec for all the vehicles. The job sizes were taken from a Gaussian distribution with a mean of 5 and a standard deviation of 2.

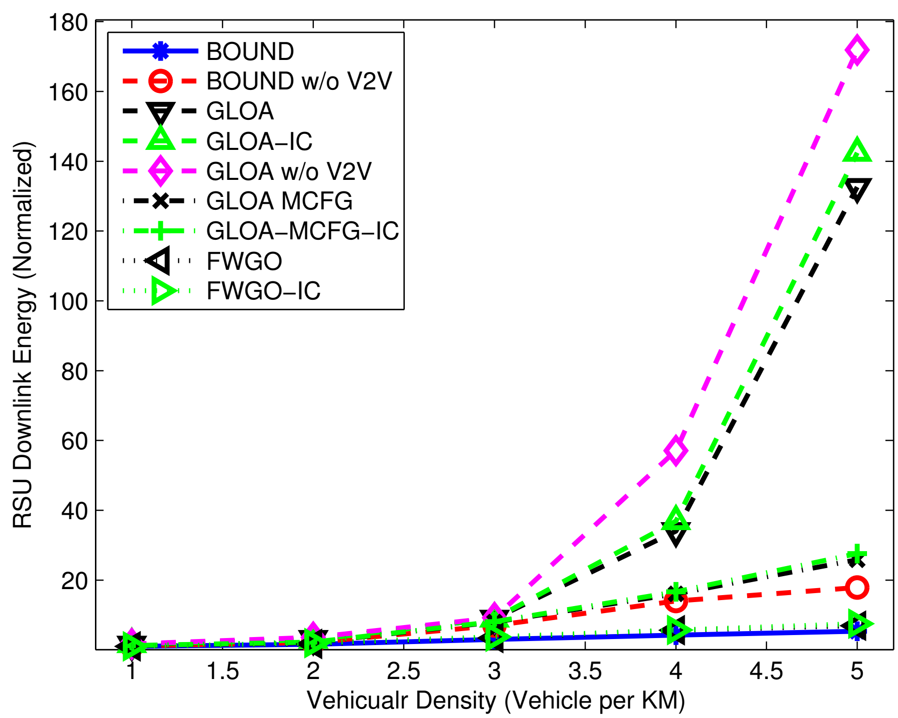

Fig. 6: Total RSU Downlink Energy vs. Vehicular Density (Vehicle per km). The job sizes of all the vehicles are constant and equal to 5 (time slots), and the speed is taken from a Gaussian distribution with a mean 20 m/sec and the standard deviation of 2.

higher. For example, when comparing the simple GLOA algorithm with and without V2V forwarding (GLOA vs. GLOA w/o V2V) we find that the energy use without V2V is about 100% higher at the largest vehicle density shown. When doing the forwarding in-channel (GLOA-IC), there is typically a slight loss in energy efficiency due to the contention for downlink time slots. At the value of density quoted above, the downlink energy for GLOA-IC is about 20% higher than the off-channel case. The graph also shows that by combining V2V forwarding with a more sophisticated time slot assignment algorithm (i.e., GLOA-MCFG and GLOA-MCFG-IC), the energy efficiency can be improved significantly. For example, the GLOA-MCFG algorithm requires less than 50% lower energy than basic GLOA at the higher vehicular density. As before, when requiring the algorithm to schedule the V2V transmissions in-channel, a small energy penalty is paid. The best performance was obtained by the FWGO algorithms (FWGO and FWGO-IC) and we can see that their energy requirements are about 100% above the offline bound. This is quite good, considering how poorly the other algorithms perform compared with the offline bound. Clearly the more sophisticated algorithms are closer to the bounds, as expected.

It is possible to operate some of the non-V2V schedulers using the more complex MCFG time slot assignment. In the interests of clarity we have not included results for these cases in the figures. However, we have included the offline bound for the non-V2V case from Section III (shown as “BOUND w/o V2V”). In practice, the best online algorithms perform somewhat above this bound, and it can be seen that the best algorithms with V2V forwarding perform significantly below it. At the highest vehicular density shown in Figure 5, for example, the bound gives an energy value which is over 400% higher than the online FWGO algorithm. Clearly, V2V forwarding can significantly decrease RSU energy use even if an online algorithm without V2V forwarding could achieve the offline bound.

We now consider the advantages of V2V forwarding when there are differences in vehicle speed. First consider the case where two vehicles are moving in the same direction with the same speed, and will therefore remain at a constant distance relative to each other. If there is no difference in the vehicular job sizes, then it is easy to see that there is no energy advantage to vehicle-to-vehicle forwarding. Now consider the case where one of the vehicles is travelling at a much higher speed than the other. The fast moving vehicle will be at the most energy favourable locations, e.g., closer to RSU, for less time than the slower moving vehicle. Assuming the vehicles maintain contact, vehicle-to-vehicle forwarding can play a significant role in improving energy efficiency. Figure 6 uses the same parameters as Figure 5, except that we have drawn the speed of each vehicle from a Gaussian distribution, as discussed in the figure

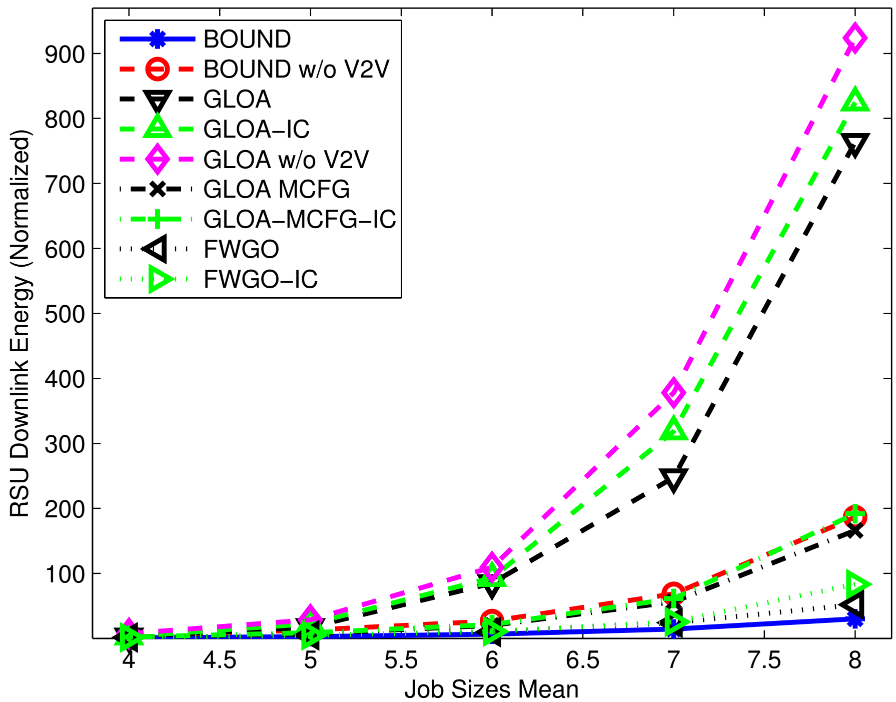

Fig. 7: Total RSU Downlink Energy vs. Job Sizes Mean. Vehicular Density is constant and equal to 4 vehicles per km, the job sizes are from a Gaussian distribution with Standard Deviation of 2

caption. It is clear that the vehicle speed differences will increase the advantage of vehicle-to-vehicle forwarding. However, the benefit of V2V forwarding is not comparable with the previous case when there were differences in job sizes. For example, GLOA with and without V2V forwarding shows about a 30% advantage at the highest vehicular density compared to about 100% in the case of Figure 5. Contrary to Figure 5, where GLOA-MCFG was doing better than the Bound w/o V2V, in this figure it is worse by about 20%. This is due to the decrease in advantage that vehicle speed differences make. It should be noted that the GLOA-MCFG w/o V2V algorithm would be somewhere above the Bound w/o V2V. Also, the FWGO algorithm is doing the best compared to the other online algorithms, as before. Both the FWGO and FWGO-IC algorithms are below the Bound w/o V2V by about a 60% advantage, and they are both quite close to Bound with V2V, i.e., off by about 25% which is very good compared to the other online algorithms. This plot shows the pure performance of GLOA and GLOA-MCFG compared to FWGO, which is due to the nature of GLOA and GLOA-MCFG, which assign time slots based on relative speed in a greedy fashion.

The benefits of vehicle-to-vehicle forwarding come from differences in vehicle speed, job sizes and system loading. Figure 7 shows that as the vehicle job sizes increase, the benefit of V2V forwarding becomes more pronounced. The reason for this advantage is clearly because, in the non-V2V case, traffic must be transmitted in less energy favourable situations. If the system load remains constant, as the variation in job sizes increases, it is expected that the energy use with smart V2V forwarding will remain constant. In a system with no V2V communication, increases in energy would be expected. This phenomenon is the focus of Figure 8. In Figure 7 the effect of the average job size is shown and the advantage of V2V forwarding with the increase in loading is obvious. In Figure 7 all the points have the same standard deviation, unlike Figure 8, in which the mean is the same. It is clear from this example that the increase in variation makes V2V forwarding more advantageous. For example, at the lowest standard deviation, the Bounds with and without V2V forwarding differ by about 70% whereas at a standard deviation of 3, the Bound with V2V forwarding is almost 10 times better than the Bound without V2V case. This is a large difference in energy usage which can also be seen in the online algorithms. For example, GLOA with V2V is 2 times better than without V2V. Also, FWGO and FWGO-IC, GLOA-MCFG and GLOA-MCFG-IC, are all are doing better than the Bound without V2V (and any non-V2V scheduler). FWGO is about 4 times better than Bound w/o V2V. Also, it is expected that increases in variation will not affect energy usage when there is V2V forwarding. This can be seen in Figure 8 where Bound with V2V only increases by about 10% while Bound w/o V2V increases by almost 275%. The better the online algorithm, the less increase in energy use is expected as the variation in job sizes increases. As an example, FWGO only increases by 75%, GLOA-MCFG increases by about 80%, and simple GLOA experiences a 75% increase. This result, with only a 10% increase compared to the Bound, shows better time slot usage.

B. CBR Results (Exponential Path Loss with Shadowing)

Figure 9 uses the same assumptions as in Figure 5 except that the energy cost data that is fed to the schedulers includes random components. This is done to ensure that when the scheduler input data are not ideal, the algorithms do not produce results which would be biased in some way, due to this randomness. In the results presented, this was done by adding propagation shadowing effects to the extracted data, which result in unpredictable randomness in these estimates. The scheduling is therefore based on this input, but the actual costs incurred include the energy perturbations due to the random components. It can be seen in Figure 9 that the same relative performance comparisons are true for this case except that the energy values obtained are higher. For example, the RSU energy cost for the basic GLOA algorithm is about double what it was before. The same may be said for the other algorithms and offline bounds. As was illustrated earlier, the input data fed to the on-line algorithms use estimates which include errors due to randomness of the wireless channel. In the presence of such errors, as shown in Figure 9, the difference between the on-line algorithms and Bound becomes wider, as the Bound has complete knowledge, whereas the online algorithms schedule vehicles based on the provided data inputs. In experiments that were performed for higher levels of shadowing, i.e., for σ of 12 dB, we found that the general trends are the same. The overall energy consumption increases as does the spread between the different algorithms. However, we found that their relative performance remains the same. This observation was also made in reference [19] when shadowing components were used with direct RSU-to-vehicle transmission.

C. VBR Results (Exponential Path Loss)

Figures 10 to 12 are samples of experiments conducted for the VBR case. In Figure 10 the total RSU downlink transmission time is plotted versus vehicular density. As the vehicular density increases, the benefit of vehicle-to-vehicle forwarding increases. Clearly all of the online algorithms are doing better than the Bound w/o V2V, and the FWGO-VBR off channel case at the highest bit rate is using almost 3 times less transmission time. The advantage of the FWGO-VBR algorithm over GLOA is apparent as the density increases. It can be seen that FWGO is doing better than FWGO-IC by about 15%, about 25% better than GLOA-DFG and over 70% better than GLOA-DFG-IC. The worst online algorithm for the V2V case, i.e., GLOA-DFG-IC, is doing about 80% better than the Bound w/o V2V. It should be noted that in the case simulated here, vehicles have small job sizes, but there is a small fraction of vehicles, i.e., 15%, which have large job sizes.

Figure 12 uses the same parameters as in the previous experiment except that the speed is also variable. As the results from the CBR case have shown, speed variation is one of the reasons vehicle-to-vehicle forwarding is advantageous. Obviously, adding that to the experiment of Figure 10 makes the vehicle-to-vehicle forwarding more useful and the differences between the online V2V algorithms and the Bound w/o V2V are very obvious. Figure 11 shows the effect of job size: as in the CBR case, a higher system load results in increased performance with vehicle-to-vehicle forwarding. At the highest load, there is about a 130% advantage comparing FWGO and Bound w/o V2V. Although other online algorithms are performing poorly compared to FWGO, they are still far better than Bound w/o V2V. For example, FWGO-IC is doing about 100% better, GLOA-DFG about 110% and GLOA-DFG-IC about 80%.

In summary, as the vehicular density increases, the number of vehicles in the coverage area also increases, and the benefits from V2V forwarding become more apparent, since there are more vehicles

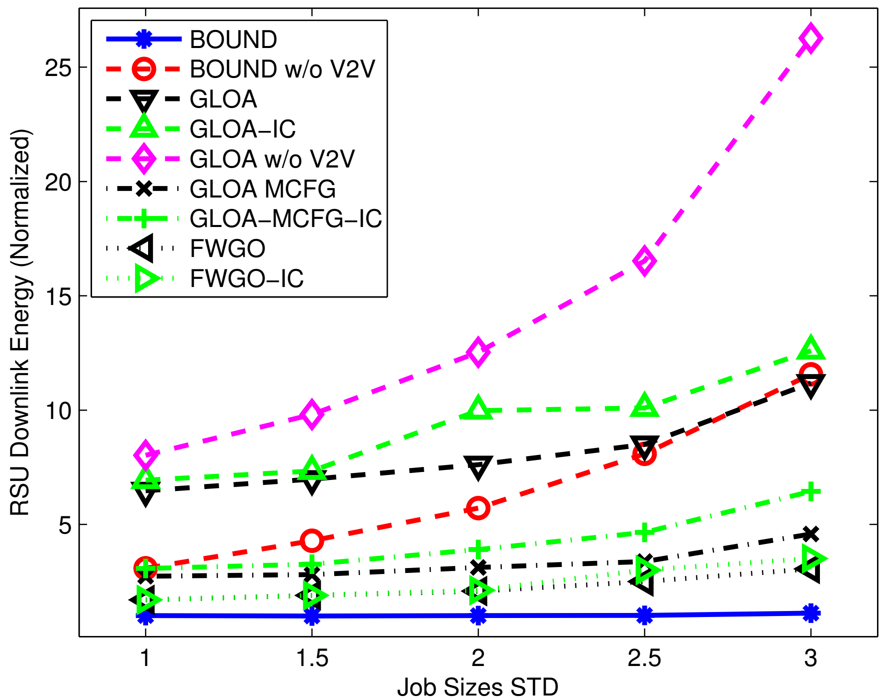

Fig. 8: Total RSU Downlink Energy vs. Job Size Standard Deviation. The mean job size is constant and equal to 5. Vehicular density is constant and equal to 4 vehicles per km and vehicle speed is constant and equal to 20 m/sec.

within mutual communication range. In the case where there is no vehicle-to-vehicle communication, it is obvious that by increasing the number of vehicles in the system, the total downlink energy, or equivalently, the transmission time in the VBR case, will increase. At the same time, when intelligent vehicle-to-vehicle forwarding is available, large improvements in the energy efficiency of the downlink traffic schedules are possible. Examples of such improvements have been shown above.

D. Non-Energy Aware Schedulers

The results presented above involve comparisons with schedulers whose focus is on energy efficient performance. In this section, results are presented which consider both energy and mean delay performance when the proposed algorithms are compared with schedulers that do not consider energy efficiency and do not assume delay tolerance. The additional algorithms which are used for these comparisons are as follows [16] [26].

- First Come First Served (FCFS): FCFS assigns (CBR) time slots or (VBR) transmission time to each vehicle immediately when it arrives to the RSU coverage area. The RSU will actively service vehicle requests whenever there are vehicles within its coverage range, which have unfulfilled job requests.

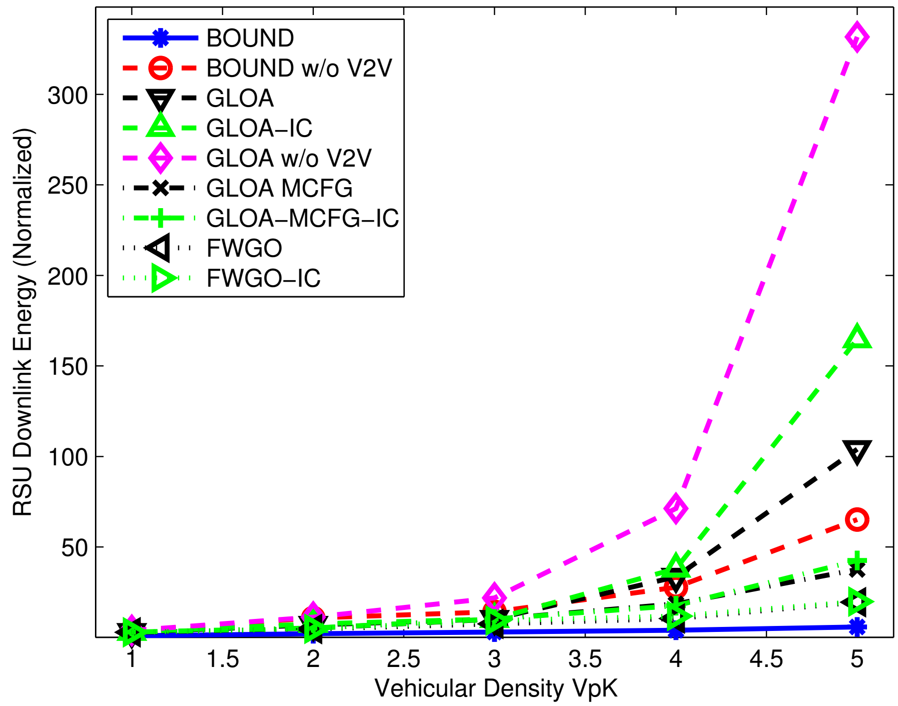

Fig. 9: Total RSU Downlink Energy vs. Vehicular Density (Vehicle per km). This plot has the same assumptions as Figure 5 but including log-normal shadowing with a zero mean and a standard deviation of $\sigma = 4$ dB.

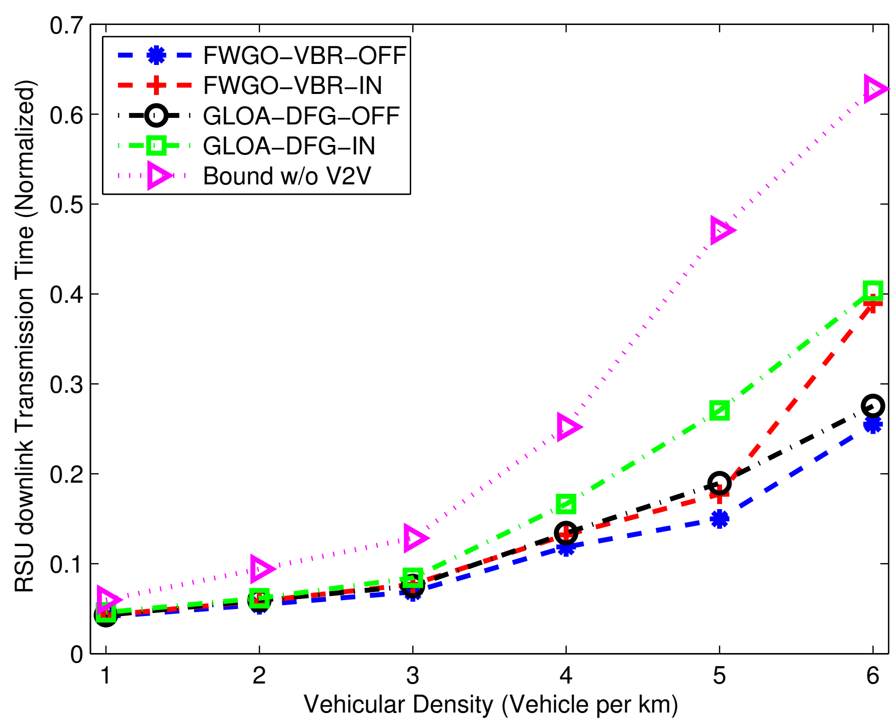

Fig. 10: RSU Downlink Transmission Time vs. Vehicular Density. Vehicle demand is taken such that 85% of vehicles have no job, but the rest have a demand of 15 time slots. The vehicle speeds are all 20 m/s.

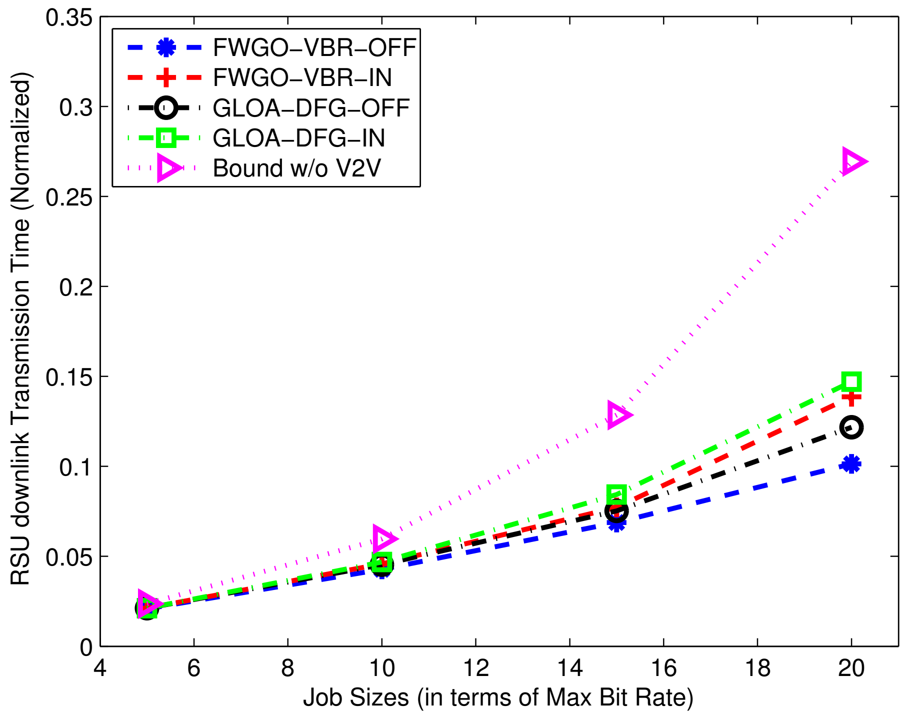

Fig. 11: RSU Downlink Transmission Time vs. Vehicle Job Demand. This graph uses the same parameters as in Figure 10 except that the high vehicle demands are varied. Vehicular density is 4/km.

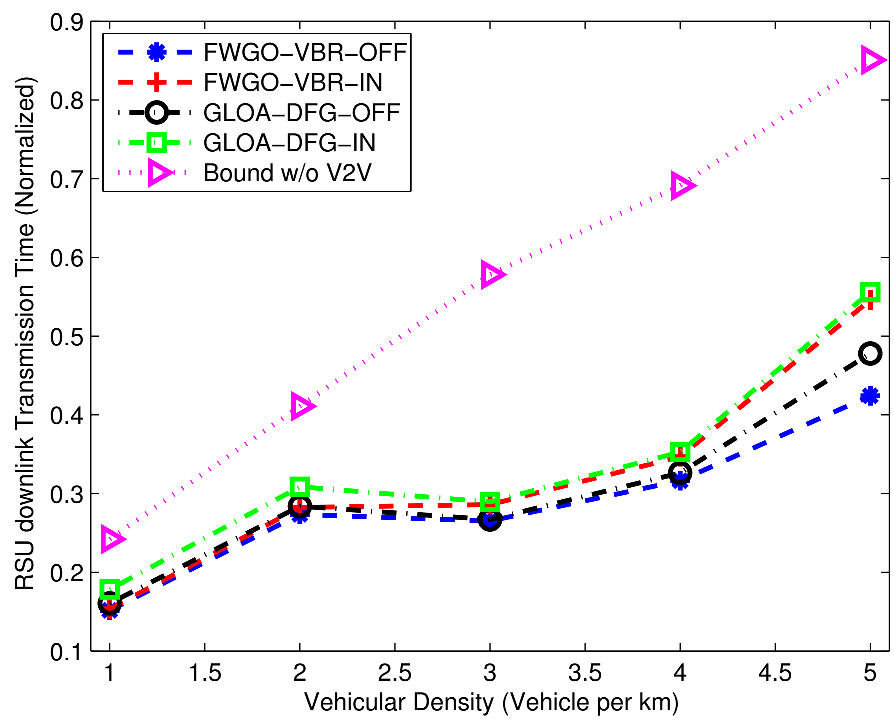

Fig. 12: RSU Downlink Transmission time vs. Vehicular Density. This plot has the same parameters as Figure 10 except that vehicle speed has been taken from a Gaussian distribution with a mean 20 m/s and a standard deviation of 2.

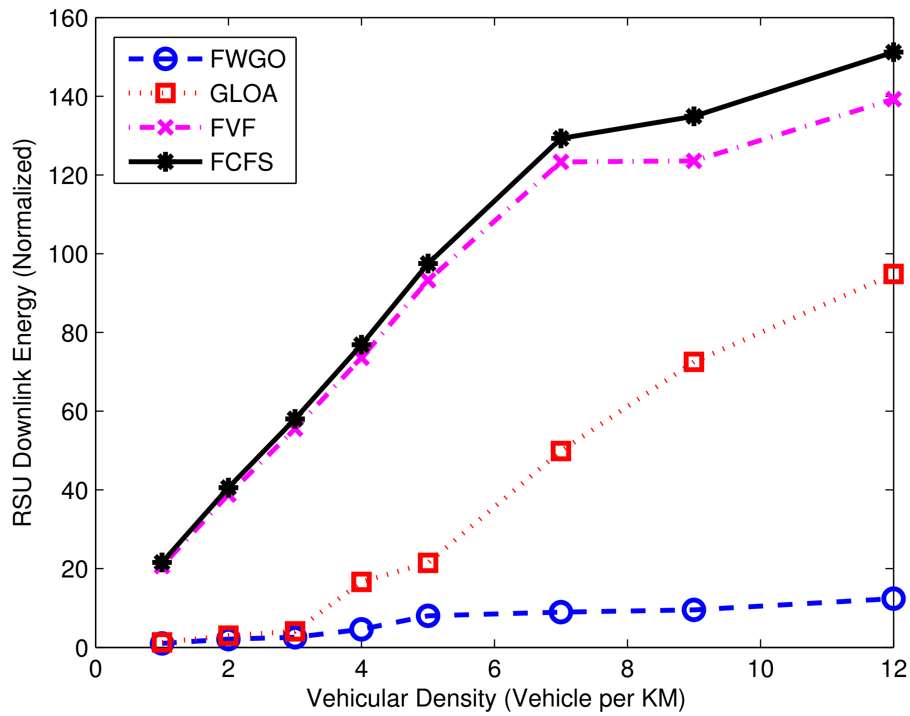

Fig. 13: RSU Downlink Energy (Normalized) vs. Vehicular Density. Vehicle job size is taken from a Normal distribution with a mean of 5 and a standard deviation of 2. Vehicle speeds follow a Normal distribution with a mean of 20 m/s and a standard deviation of 2.

- Fastest Vehicle First (FVF): As in FCFS, FVF actively services vehicles whenever there is any outstanding vehicle backlog. However, priority is based on vehicle speed, with faster moving vehicles given higher priority, i.e., when the transmission of a packet is completed, the vehicle with the highest speed is the next one served.

It is assumed that there is no vehicle-to-vehicle traffic generation, and since FCFS and FVF are designed for the CBR case, we do not consider V2V forwarding for these schedulers. For this reason the FCFS and FVF algorithms use single hop RSU-to-vehicle transmission as would be the case in practice.

In the first set of results we assume Poisson process vehicle arrivals where vehicle job sizes are selected from a a Normal distribution with a mean of 5 time slots and standard deviation of 2. Vehicle speeds follow a Normal distribution with a mean 20 m/s and standard deviation of 2. Figures 13 and 14 show comparisons of downlink RSU energy and mean packet delay, respectively, as vehicle density is varied for the off-channel forwarding case. As one would expect, the performance of FCFS and FVF are far worse that that of the energy-aware schedulers. These large differences are to be expected since the FCFS and FVF algorithms transmit to vehicles as soon as possible, thus ignoring the energy required. As vehicular

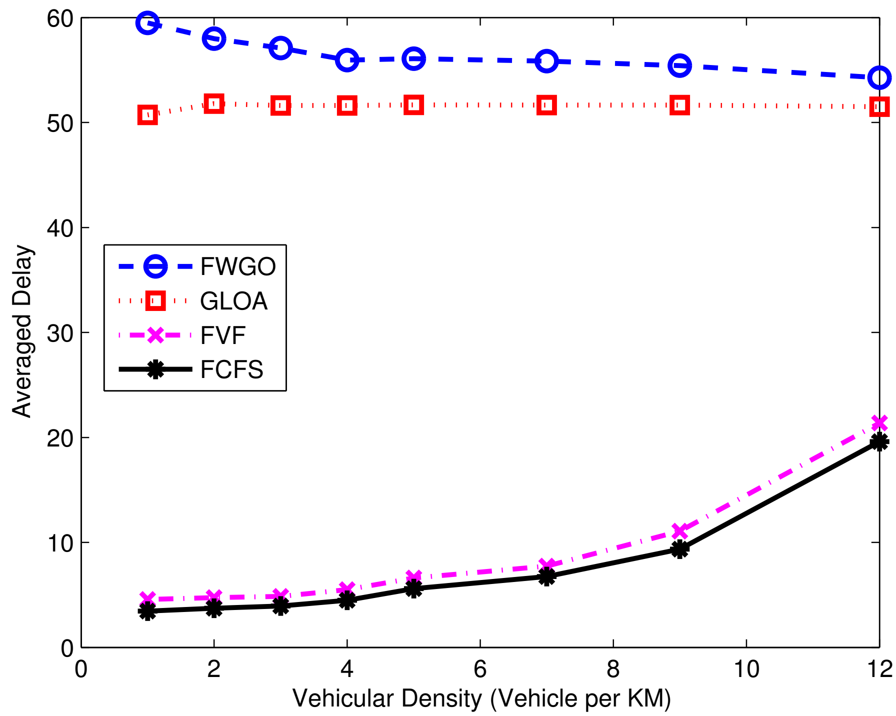

Fig. 14: Average Delay vs. Vehicular Density. Vehicle job size is taken from a Normal distribution with a mean of 5 slots and a standard deviation of 2. Vehicle speeds are from a Normal distribution with a mean of 20 m/s and a standard deviation of 2.

density increases, these differences increase, as do the differences between the FCFS and FVF algorithms. As seen in Figure 13, FVF eventually performs slightly better than FCFS since prioritizing vehicles based on speed will schedule faster moving vehicles before they move into energy unfavourable locations. Figure 13 at the highest vehicular densities, FCFS and FVF perform with almost ten times more energy cost than FWGO, and almost twice that of GLOA.

Figure 14 shows the corresponding comparisons of mean packet delay. Again, as one would expect, the price that is paid for the high energy efficiency of the GLOA and FWGO schedulers is a far higher average delay performance. Since FCFS and FVF cause the RSU to transmit whenever there is vehicle backlog, they achieve very low delays compared to that of both GLOA and FWGO. We can see that comparing these two algorithms there is a slight increase in delay associated with the more energy efficient of the two. Also, the slight decrease in delay for FWGO as vehicular density increases is caused by the fact that as contention increases, the algorithms serve vehicles at locations which are further from the RSU. These graphs provide a good illustration of the trade-off between delay and energy efficient operation of the RSU.

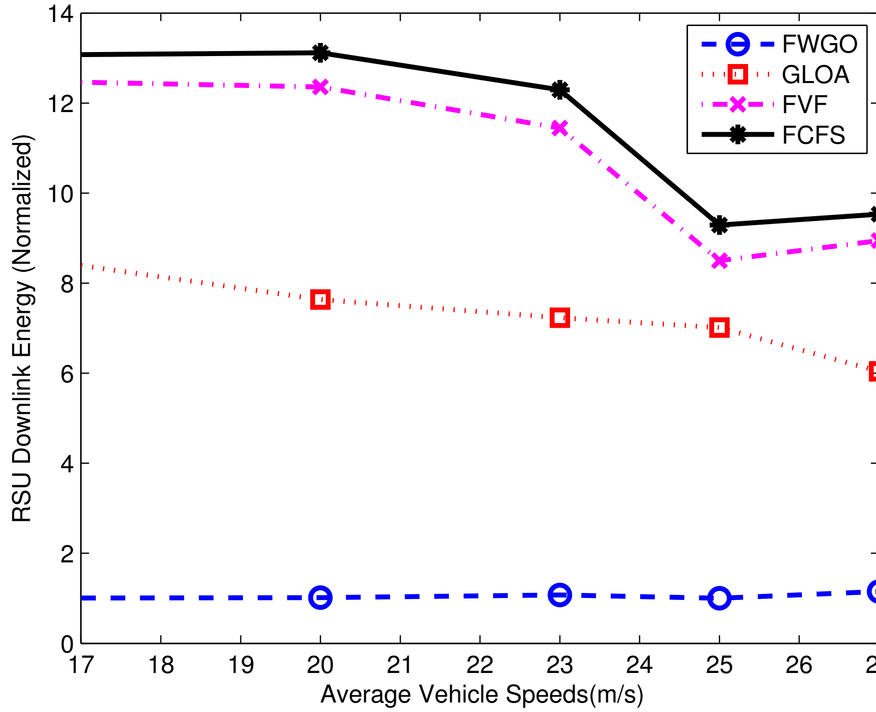

Fig. 15: RSU Downlink Energy (Normalized) vs. Average Vehicle Speed (m/s). Vehicular density of 5 (v/km). Vehicle job size is taken from a Normal distribution with a mean of 7 and a standard deviation of 2. Vehicle speeds follow a Normal distribution with a mean shown on the x-axis and a standard deviation of 2.

The effect of vehicle speed under the same assumptions is shown in Figures 15 and 16. As expected, FWGO performs far better energy-wise than the other three algorithms. Interestingly, the RSU energy use for FCFS and FVF drops as the average speed increases. This is caused by the fact that vehicles arrive more quickly to increasingly favourable energy positions as their speed goes up. This effect is less so for FWGO and GLOA since the schedulers wait until vehicles are in energy favourable locations before communication occurs. This is unlike FCFS and FVF which have the RSU communicate whenever there is vehicular backlog. The corresponding mean delay results are shown in Figure 16. As before, the delay performance of the GLOA and FWGO algorithms is far worse than FCFS and FVF, for the reasons discussed previously. We see that for the former algorithms, the mean delay decreases with speed. As in the previous set of results, this is caused by the fact that as speed increases, vehicles arrive in energy favourable locations more quickly, which decreases their mean delay.

In the VBR air interface case, examples are shown for RSU energy performance (i.e., RSU transmission time) as a function of vehicular density in Figure 17. In this case the results compare our best algorithm, FWGO, with FCFS and FVF. The modelling assumptions used are the same as those in the previous sets of

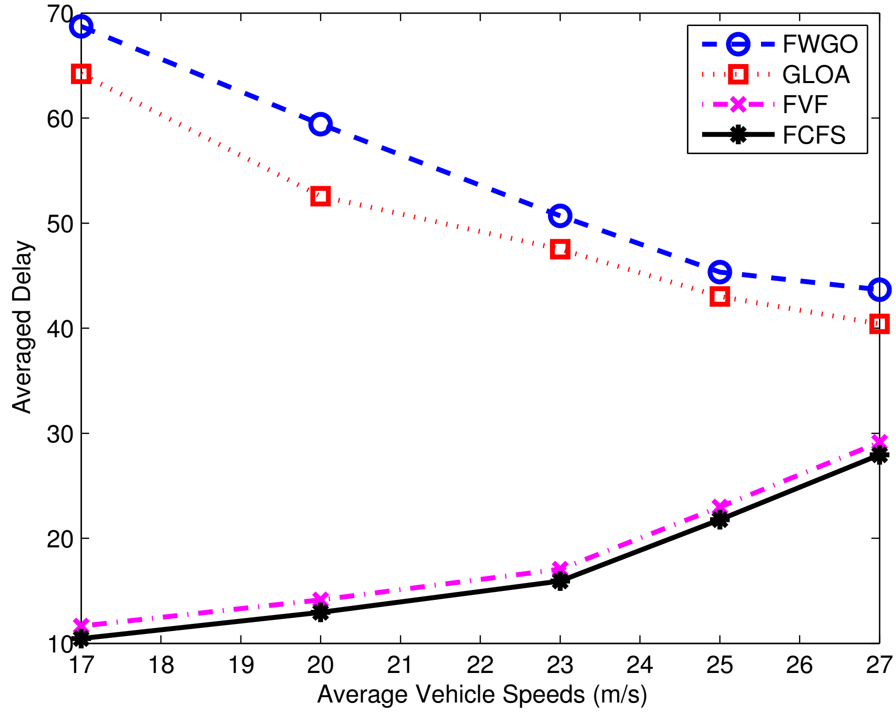

Fig. 16: Average Delay vs. Average Vehicle Speed (m/s). Vehicular density is 5 (v/km). Vehicle job size is taken from a Normal distribution with a mean of 7 and a standard deviation of 2. Vehicle speeds follow a Normal distribution with a mean shown on the x-axis and a standard deviation of 2.

results. The results obtained are qualitatively similar to the comparisons done for the CBR case. In Figure 17, we find that FWGO performs significantly better than FCFS and FVF from an energy viewpoint. For this set of parameters the energy performance of FCFS and FVF is over 500% compared to FWGO. Also, as expected, FVF performs better (by about 25%) than FCFS. This is due to the fact that faster moving vehicles will be in closer range of the RSU for a smaller portion of time, and therefore giving them priority will increase energy efficiency. In Figure 17, the average delay performance is significantly lower for the FCFS and FVF algorithms, for the reasons discussed previously.

VI. Conclusions

In this paper we have presented results which show that by combining smart scheduling with vehicle-to-vehicle (V2V) packet forwarding, downlink vehicular traffic schedules can be generated with reduced energy cost compared to the non-V2V case. This happens because vehicles have virtually unlimited energy reserves, and as a result, vehicular forwarding does not incur infrastructure energy costs. Roadside infrastructure can therefore reduce energy use by forwarding packets through vehicles which are in energy favourable locations. Offline bounds were derived for the downlink energy usage when V2V forwarding

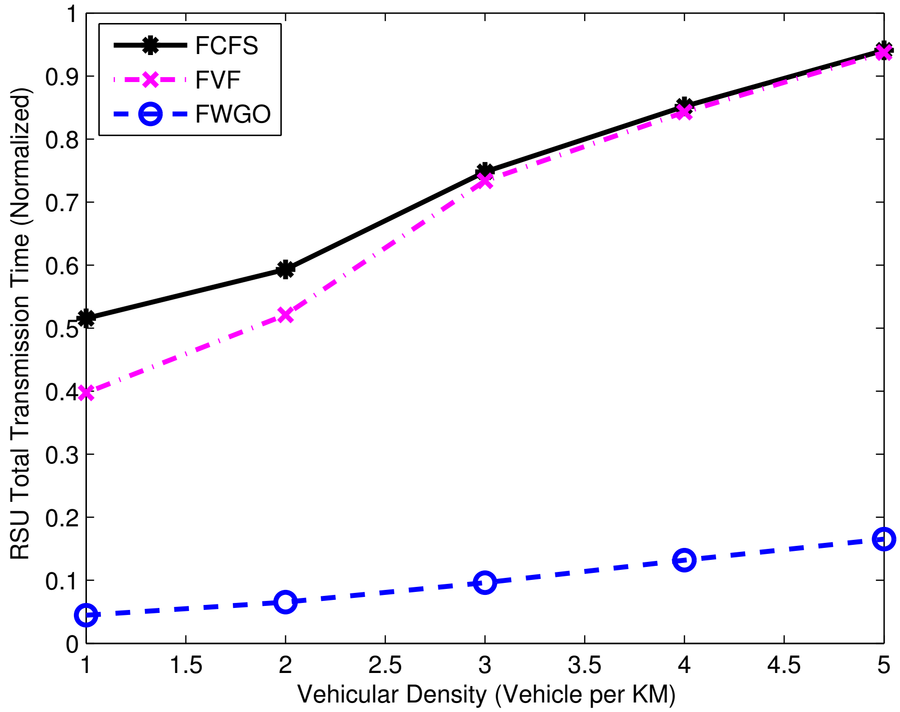

Fig. 17: RSU Downlink Transmission Time vs. Vehicular Density. All vehicles have the same job size of 3 (in terms of maximum bit rate). Vehicle speeds follow a Normal distribution with a mean of 20 m/s and a standard deviation of 3.

is used. This includes both the off-channel and in-channel V2V forwarding cases. The in-channel bound uses a formulation based on a time expanded graph model. These bounds were used for comparisons with various online scheduling algorithms where both fixed and variable bit rate air interface options were considered.

The paper then introduced online algorithms. The first algorithm used a greedy local optimization, referred to as GLOA. A version of this algorithm was also considered which uses a minimum cost flow graph scheduler to perform the time slot assignment. A more complex algorithm was also introduced which is based on a proposed finite window group optimization (FWGO). Versions of these algorithms were introduced which use in-channel vehicle-to-vehicle scheduling, and were also adapted to the variable bit rate air interface case. Results from a variety of experiments show that under certain conditions, the proposed scheduling algorithms can significantly improve the downlink energy requirements of the road-side unit compared to the case where vehicle-to-vehicle packet forwarding is not used. The performance improvements are especially strong under heavy loading and when the variation in vehicle communication requirements or vehicle speed is high. Results were also presented which compare the proposed algorithms

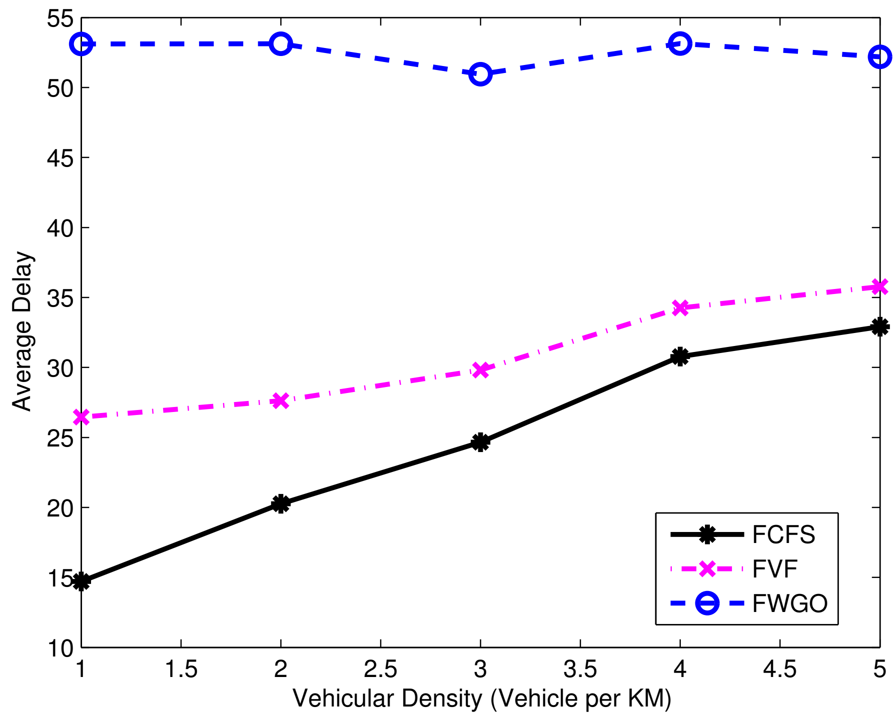

Fig. 18: RSU Downlink Transmission Time vs. Vehicular Density. All vehicles have the same job size of 3 (in terms of maximum bit rate). Vehicle speeds follow a Normal distribution with a mean of 20 m/s and a standard deviation of 3.

with conventional non-energy aware schedulers.

References

[1] Federal Communications Commission, "FCC 03-024. FCC Report and Order," February 2004.

[2] R. Uzcategui and G. Acosta-Marum, "Wave: A Tutorial," *Communications Magazine, IEEE*, vol. 47, no. 5, pp. 126–133, 2009.

[3] S. Peirce and R. Mauri, "Vehicle-Infrastructure Integration (VII) Initiative: Benefit-Cost Analysis: Pre-Testing Estimates," *Intelligent Transportation Systems Joint Program Office, United States Department of Transportation, Washington, DC.*, March 2007.

[4] A. A. Hammad, G. H. Badawy, T. D. Todd, A. A. Sayegh, and D. Zhao, "Traffic Scheduling For Energy Sustainable Vehicular Infrastructure," in *IEEE Global Communications Conference (IEEE GLOBECOM'2010), Miami, Fla.*, December 2010.

[5] Y. Khaled, M. Tsukada, J. Santa Lozano, and T. Ernst, "On The Design of Efficient Vehicular Applications," *Proc. of IEEE VTC09, Barcelona, Spain, April*, 2009.

[6] F. Li and Y. Wang, "Routing in Vehicular Ad Hoc Networks: A Survey," *IEEE Vehicular Technology Magazine*, vol. 2, no. 2, pp. 12–22, 2007.

[7] L. Zhang, Q. Wu, A. Solanas, and J. Domingo-Ferrer, "A Scalable Robust Authentication Protocol For Secure Vehicular Communications," *IEEE Transactions on Vehicular Technology*, pp. 1606–1617, 2009.

[8] J. Mittag, F. Schmidt-Eisenlohr, M. Killat, J. Harri, and H. Hartenstein, "Analysis and Design of Effective and Low-Overhead Transmission Power Control for VANETs," in *Proceedings of the Fifth ACM International Workshop on Vehicular Inter-Networking*. ACM, 2008.

[9] 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*. ACM, 2006.

[10] J. Ott and D. Kutscher, "Drive-thru Internet: IEEE 802.11b for Automobile Users," *INFOCOM 2004*, vol. 1, March 2004.

[11] M. F. Jhang and W. Liao, "On Cooperative and Opportunistic Channel Access for Vehicle to Roadside (V2R) Communications," in *IEEE Global Communications Conference (GLOBECOM'2008)*, Dec. 2008, pp. 1–5.

[12] J. Zhao, T. Arnold, Y. Zhang, and G. Cao, "Extending Drive-Thru Data Access by Vehicle-to-Vehicle Relay," *VANET '08: Proceedings of the Fifth ACM International Workshop on Vehicular Inter-Networking*, 2008.

[13] A. Nandan, S. Das, G. Pau, M. Gerla, and M. Y. Sanadidi, "Co-operative Downloading in Vehicular Ad-hoc Wireless Networks," in *Second Annual Conference on Wireless On-demand Network Systems and Services, WONS'2005*, January 2005, pp. 32–41.

[14] A. Festag, R. Baldessari, and H. Wang, "On Power-aware Greedy Forwarding in Highway Scenarios," in *5th International Workshop on Intelligent Transportation (WIT), Hamburg, Germany*, 2007.

[15] D. Rawat, G. Yan, D. Popescu, M. Weigle, and S. Olariu, “Dynamic Adaptation of Joint Transmission Power and Contention Window in VANET,” *VTC*, Sept. 2009.

[16] Y. Zhang, J. Zhao, and G. Cao, “On Scheduling Vehicle-Roadside Data Access,” in *Proceedings of the Fourth ACM International Workshop on Vehicular Ad Hoc Networks*. ACM, 2007, pp. 9–18.

[17] J. Alcaraz, J. Vales-Alonso, J. García-Haro *et al.*, “Control-Based Scheduling With QoS Support for Vehicle to Infrastructure Communications,” *IEEE Communications*, pp. 32–39, 2009.

[18] F. Zou, J. Zhong, W. Wu, D.-Z. Du, and J. Lee, “Energy-efficient Roadside Unit Scheduling for Maintaining Connectivity in Vehicle Ad-hoc Network,” in *Proceedings of the 5th International Conference on Ubiquitous Information Management and Communication*. ACM, 2011, p. 64.

[19] A. A. Hammad, T. D. Todd, G. Karakostas, and D. Zhao, “Downlink Traffic Scheduling in Green Vehicular Roadside Infrastructure,” *IEEE Transactions on Vehicular Technology*, vol. 62, no. 3, pp. 1289–1302, March 2013.

[20] B. Kotnyek, “An Annotated Overview of Dynamic Network Flows,” *Rapport de Recherche. Institut National de Recherche en Informatique et en Automatique*, no. 4936, pp. 1–28, 2003.

[21] R. Ahuja, T. Magnanti, and J. Orlin, “Network Flows: Theory, Algorithms, and Applications,” *Journal of the Operational Research Society*, vol. 45, no. 11, pp. 1340–1340, 1994.

[22] M. Khabazian and M. Ali, “A Performance Modeling of Connectivity in Vehicular Ad Hoc Networks,” *IEEE Transactions on Vehicular Technology*, vol. 57, no. 4, pp. 2440–2450, 2008.

[23] R. G. C. Sommer, D. Eckhoff and F. Dressler, “A Computationally Inexpensive Empirical Model of IEEE 802.11p Radio Shadowing in Urban Environments,” in *Eighth International Conference on Wireless On-Demand Network Systems and Services*, 2011, pp. 84–90.

[24] S. Wang, “The Effects of Wireless Transmission Range on Path Lifetime in Vehicle-formed Mobile Ad Hoc Networks on Highways,” in *IEEE International Conference on Communications, 2005. ICC’2005.*, vol. 5. IEEE, 2005, pp. 3177–3181.

[25] K. Bilstrup, E. Uhlemann, E. Strom, and U. Bilstrup, “Evaluation of the IEEE 802.11p MAC Method for Vehicle-to-Vehicle Communication,” in *IEEE 68th Vehicular Technology Conference, 2008. VTC 2008-Fall.*, September 2008.

[26] V. Kumar and N. Chand, “Data Scheduling in VANETs: A Review,” *International Journal of Computer Science and Communication*, vol. 1, no. 2, pp. 399–403, 2010.