Downlink Traffic Scheduling in Green Vehicular Roadside Infrastructure

Abdulla A. Hammad, Terence D. Todd, George Karakostas and Dongmei Zhao
Department of Electrical and Computer Engineering
McMaster University
Hamilton, Ontario, CANADA
Email: todd@mcmaster.ca

Abstract

In this paper we consider the problem of scheduling for energy efficient road-side infrastructure. In certain scenarios, vehicle locations can be predicted with a high degree of accuracy, and this information can be used to reduce downlink infrastructure-to-vehicle energy communication costs. The paper first presents off-line scheduling results which provide lower bounds on the energy needed to satisfy arriving vehicular communication requirements. We show that the *packet-based scheduling* case can be formulated as a generalization of the classical single-machine job scheduling problem with a tardiness penalty, referred to as α-Earliness-Tardiness. A proof is given which shows that even under a simple distance-dependent exponential radio path loss assumption, the problem is NP-complete. The remainder of the paper then focuses on *timeslot-based scheduling*. We formulate this problem as a Mixed Integer Linear Program (MILP) which is shown to be solvable in polynomial time using a proposed minimum cost flow graph construction. The paper then introduces three energy efficient online traffic scheduling algorithms for common vehicular scenarios where vehicle position is strongly deterministic. The first, Greedy Minimum Cost Flow (GMCF), is motivated by our minimum cost flow graph formulation. The other two algorithms have reduced complexity compared with GMCF. The Nearest Fastest Set (NFS) scheduler uses vehicle location and velocity inputs to dynamically schedule communication activity. The Static Scheduler (SS) performs the same task using a simple position-based weighting function. Results from a variety of experiments show that the proposed scheduling algorithms perform well when compared to the energy lower bounds in vehicular situations where path loss has a dominant deterministic component so that energy costs can be estimated. Our results also show that near-optimal results are possible but come with increased computation times compared to our heuristic algorithms.

I. Introduction

Vehicular ad hoc networks (VANETs) will become a major commercial force in the near future. These systems will enable applications ranging from road safety, to those involving context-aware advertising and in-vehicle Internet media streaming. Recognizing the importance of VANETs, the FCC has licensed the operation of dedicated short range communication (DSRC) in the 5.9 GHz frequency band [1]. A new standard for vehicular networks, known as Wireless Access in Vehicular Environment (WAVE), has been developed based on the IEEE 802.11 wireless LAN standard [2].

VANETs define two modes of communication, Vehicle-to-Vehicle (V2V) and Vehicle-to-Infrastructure (V2I). In the latter case, fixed infrastructure can be used to broadcast safety messages or can be used as a gateway to the Internet. Vehicular network infrastructure will eventually evolve into a platform which will permit an even larger variety of mobile applications. Unlike many traditional wireless networks, vehicles are often moving very quickly, and may only remain in an RSU radio coverage area for a relatively short period of time. In addition, since multiple vehicles may be present in the RSU coverage area, the question arises as to the order with which vehicles should be served. Extensive research has already considered this problem and various schedulers have been proposed as will be discussed in Section I-A.

In many highway locations, deploying roadside infrastructure is difficult due to the unavailability or prohibitive expense of wired electrical power. In these situations, an alternative to wired power connections is to operate some of the RSUs using an energy sustainable source such as solar power. In these types of node designs, it is well-known that the energy provisioning costs are a strong function of average power consumption and that they can be a significant fraction of the total node cost [3][4]. This motivates the need for energy efficient vehicular roadside infrastructure. 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 [5].

In this paper, energy efficient road-side unit scheduling is considered. In certain vehicular installations, the location of vehicles passing through the RSU radio coverage area can be predicted with a high degree of accuracy. This information can then be used to reduce downlink infrastructure-to-vehicle energy communication costs. The paper starts by presenting off-line scheduling bounds which provide lower limits on the energy needed to satisfy vehicular communication requests. The paper considers both *packet* and *timeslot* based scheduling. In the former case, the problem can be formulated as a generalization of the classical single-machine job scheduling problem with earliness and tardiness penalties, referred to as α-Earliness-Tardiness. Even under a simple distance-dependent exponential radio path loss assumption, the problem is shown to be NP-complete. The paper also considers packet-based scheduling. This version of the problem can be formulated as a Mixed Integer Linear Program (MILP), which is shown to be solvable in polynomial time by modeling it as a minimum cost flow problem and using the integrality property of its solution. The paper then introduces online traffic scheduling algorithms for the common vehicular scenario where there is a strong deterministic radio path loss component. The first algorithm (Greedy Minimum Cost Flow (GMCF)) is based on a local optimization using our minimum cost flow model. Two other algorithms are proposed with reduced complexity compared with the GMCF Algorithm. The first algorithm, the Static Scheduler (SS), assigns time slots according to a simple position-based weighting function. The second is a Nearest Fastest Set (NFS) scheduler that uses vehicle location and velocity inputs to dynamically assign communication slots. Results from a variety of experiments show that the proposed scheduling algorithms perform well when compared to the energy lower bounds in vehicular situations where path loss has strong deterministic components. Our results also show that near-optimal results are possible but come with increased computation times compared to our heuristic algorithms.

The remainder of the paper is organized as follows. In Section I-A a brief overview is given of related work. In Section II we give a detailed description of our system assumptions. Section III formulates lower bounds on the energy performance of the offline RSU scheduling problem. In Section III-A2 the packet-based scheduling problem is shown to be a generalization of the classic single machine job scheduling problem with earliness and tardiness penalties. The complexity of this problem is considered in this section and a proof of NP-completeness is given in the Appendix. Following this, in SectionIII-B1 the timeslot-based scheduling problem is formulated as a mixed integer linear program and in Section III-B2 a minimum cost flow formulation is used for offline scheduling. These provide lower bounds on the energy consumption needed to satisfy arriving vehicular requirements. The paper then introduces online scheduling algorithms in Section IV. In Sections IV-B, IV-C and IV-D, the Greedy Minimum Cost Flow (GMCF), Static, and Nearest Fastest Set schedulers are introduced. Performance comparisons are then presented in Section V Finally, in Section VI, the paper presents the conclusions of our work.

A. Related Work

VANET research has spanned a wide variety of topics in recent years. This includes applications [6], routing protocols [7], authentication [8], and the performance analysis of the IEEE 802.11p standard [9]. Several studies (e.g., [10] and [11]), have illustrated the suitability of IEEE 802.11p for highway applications and in [12],[13] and [14], proxy vehicles are used to improve roadside unit utilization, and to decrease vehicle contention.

Vehicle transmitter power control has been used as a mechanism for trading off network connectivity and reduced interference levels between vehicles (e.g., [9], [15] and [16]). The energy efficiency for VANETs however, has typically not been an issue, as vehicles are usually assumed to have unlimited energy reserves. Moreover, from the roadside infrastructure point of view, most work assumes urban settings where wired power is available at reasonable cost.

Traffic scheduling at the roadside unit has been considered in [17] where simple schedulers are used based on data size and deadline but without considering the energy consumption of the infrastructure. In [18], an optimization is used to maximize the total throughput of a roadside RSU given the locations and velocities of the vehicles in range. A scheduler was proposed which is suitable for use in the contention free

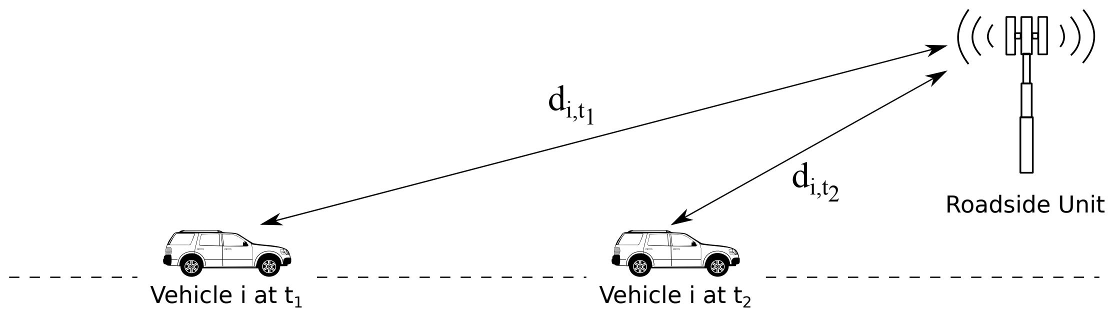

Fig. 1. Roadside Unit (RSU) Example. Vehicle i is shown at two different times, t_1 and t_2, at distances d_{i,t_1} and d_{i,t_2} from the RSU, respectively.

period of IEEE 802.11e. The energy consumption for the RSU however, was not taken into consideration. It is this aspect that is considered in this paper.

II. System Description

A roadside unit (RSU) is considered which is serving vehicles passing by in one direction as shown in Figure 1[1] The figure shows a single Vehicle i at two different times, t_1 and t_2, and at distances d_{i,t_1} and d_{i,t_2} from the RSU, respectively. We consider the energy consumption of the RSU radio interface that is used to communicate with the vehicles in the downlink (i.e., RSU-to-vehicle) direction[2]. Note that since vehicle radios are powered by the car engines, energy efficiency is not an issue on the vehicular side. We assume that the RSU uses transmit power control so that a constant bit rate reception is achieved at each vehicle with which the RSU communicates[3]. For this reason, the power consumption needed when the RSU communicates to a nearby vehicle can be significantly lower than when it communicates with a more distant vehicle. In the example shown in Figure 1, the RSU power consumption needed for communicating with Vehicle i may be significantly less at time t_2 compared with that at time t_1 since $d_{i,t_1} >> d_{i,t_2}$. For this reason, communication with Vehicle i at time t_2 is preferable compared to time t_1.

The RSU would like to minimize its long-term power consumption subject to satisfying the communication requests associated with the passing vehicles. We assume that a particular vehicular communication may occur any time throughout the vehicle's RSU transit time, i.e., the communications is delay tolerant for the period of time during which a vehicle is within the RSU coverage range. In the results in Section V, it is also assumed that a given vehicle travels at a constant speed when moving through the coverage area of the RSU, which is typically the case in highway situations [19], however, different vehicles may be

[1]The methods and results in this paper are equally applicable to two-way vehicular traffic.

[2]Due to the coverage range normally associated with RSUs, the average power consumption of an energy efficient RSU design may be strongly dominated by downlink transmission power.

[3]We have also considered the fixed transmit power, variable bit rate case, which will be published in future paper.

moving at different speeds. Accordingly, we assume that the vehicles do not interact significantly within the RSU coverage range considered [20]. When a vehicle first enters the coverage range area of the RSU, it communicates its current position and speed to the RSU, information which may be obtained via GPS inputs for example [21], and can be used to determine the vehicle's position.

III. Offline Energy Bounds

In this section we formulate lower bounds for the total RSU energy needed for downlink radio transmission to serve a finite set of vehicular arrival demands. The bound formulation consists of deriving the energy-optimal *offline schedule* where the entire vehicular arrival process and associated transmission demands are made available to the scheduler. For this reason these bounds are not generally achievable in practice since the scheduler has non-causal knowledge of future vehicular inputs. However, they are important in that they establish limits on what can be achieved in practice and are compared with online scheduling algorithms later in the paper.

Two cases are considered. The first assumes *Packet-Based Scheduling*. In this case contiguous downlink time is allocated to satisfy each vehicular communication requirement. It is shown that this can be formulated as a generalization of classic single machine job scheduling and a proof of NP-completeness is given via a reduction to the well-known PARTITION problem [22]. The second case assumes *Timeslot-Based Scheduling* where the each vehicle's transmission requirement can be independently allocated across non-contiguous timeslots. We present a mixed integer linear program and we give an algorithm based on solving a minimum cost flow problem that runs in time which is polynomial in the number of timeslots.

A. Packet-Based Scheduling

In this section we will show that this case can be modeled using a classical single machine job scheduling problem with deadlines [23]. Machine scheduling is first described, then our problem is formulated as a generalization of this well-known problem.

1) Notation and Framework: In machine scheduling, n jobs are submitted for processing on m machines. The subscript i refers to a job while subscript j refers to a machine. The pair (i, j) refers to processing *job* i on *machine* j. Processing time is represented by p_{ij} if it is machine dependent or p_i if it is machine independent. The Release Date r_i is the arrival time of job i to the system. The due date, u_i, is the job deadline, and a penalty will occur if job i is completed after this time. The completion time of job i is denoted by C_i and the objective is also a function of the due date u_i. The *lateness* of job i is defined as

$$L_i = C_i - u_i\,, \tag{1}$$

and the *tardiness* is defined by

$$T_i = \max(C_i - u_i, 0) = \max(L_i, 0). \tag{2}$$

Lateness and tardiness are two of the basic due date related penalty functions [23].

We are interested in the *total weighted tardiness*, given by $\sum_{i=1}^{n} w_i T_i$, i.e., the sum of tardiness of all processed jobs, weighted by a weight w_i. The *earliness* of job i is defined in a symmetric way as

$$E_i = \max(u_i - C_i, 0). \tag{3}$$

Again, we are interested in the *total weighted earliness*, i.e., $\sum_{i=1}^{n} w_i E_i$, where the weights are the same as those used for the total weighted tardiness above. When tardiness and earliness are combined in one objective, the scheduler will try to have each job serviced exactly at its due date, otherwise a penalty of w_i will be paid by each job i for every unit of earliness or tardiness. In our case every vehicle defines a job whose deadline is its arrival time at the RSU, and we would like to have it serviced at that time, i.e., using the minimum estimated RSU transmit power.

2) Earliness-Tardiness Single Machine Scheduling: We now formulate the minimum energy scheduling problem as a generalization of single machine scheduling. Each vehicle has an associated request which is equal to its job size in units of slot times. If we represent the RSU as a single machine, in order to promote serving vehicles closest to the RSU, we can choose the objective function to be the minimization of a combination of earliness and tardiness. In order to do this, we represent the due date u_i for each communication slot as the time at which vehicle i arrives *exactly* at a position closest to the RSU. Executing job i before reaching the RSU is penalized (energy-wise) by the earliness component E_i, while executing the job after leaving the RSU location is penalized (energy-wise) by the tardiness component, T_i, of the objective function. Therefore, we can think of our problem as a schedule on a single machine that minimizes the total weighted earliness and tardiness, i.e., $\sum_{i=1}^{n} w_i(E_i + T_i)$, or, if we use the standard scheduling notation of Graham et. al. [23], $1||\sum_{i=1}^{n} w_i(E_i + T_i)$.

In our case, earliness and tardiness in time corresponds to increases in the power consumption with distance from the RSU. In this case the energy dependence of the objective on E_i and T_i is not linear but is governed by RSU-to-vehicle path loss. For example, if we assume a standard distance-dependent exponential radio path loss propagation model [24] the relationship between the distance and the required transmission power needed to overcome path loss when the RSU is communicating with vehicle i at time t is given by

$$P_{i,t} = \rho P_0 \left(\frac{d_{i,t}}{d_0}\right)^{\alpha} , \tag{4}$$

where P_0 is a reference power, α is the assumed propagation path loss exponent, d_0 is a reference distance, and $d_{i,t}$ is the distance between vehicle i and the RSU at time t [24].

To properly model the energy distance dependence in this case, we define for each vehicle i, $\hat{d}(i)$ as the distance that the vehicle is from the RSU at the time that the *middle* of it's message transmission occurs. This can be expressed as

$$\hat{d}_i = \bar{d}_i + d_0 + \tilde{d}_i, \tag{5}$$

where $\bar{d}_i$ is a normalized distance from the RSU to vehicle i at the time that the middle of the message is being transmitted, e.g., if $\bar{d} = 0$, the middle of the message is being transmitted at the point closest to the RSU (i.e., the optimum power position). $\tilde{d}_i$ is the additional distance that the vehicle was from when the middle of the message was transmitted to where the vehicle was when the edge of the message was transmitted (it could be the front or back edge of the message). Therefore,

$$\bar{d}_i = v_i|C_i - u_i|. \tag{6}$$

i.e., C_i and u_i are defined as above in units of seconds referenced to the middle of the packet, i.e., $|C_i - u_i|$ is how long in seconds the packet was early or tardy as previously discussed. v_i is vehicle i's velocity. Also,

$$\tilde{d}_i = v_i p_i/2, \tag{7}$$

where p_i is vehicle i's packet transmission time. Then the transmit power needed for this packet is given by

$$P_t = \rho P_0 \left(\frac{\bar{d}_i + d_0 + \tilde{d}_i}{d_0}\right)^{\alpha} \tag{8}$$

$$= \rho P_0 \left(\frac{|C_i - u_i|v_i + v_i p_i/2 + d_0}{d_0}\right)^{\alpha} \tag{9}$$

$$= \quad h_i\left(|C_i - u_i| + r_i\right)^{\alpha}, \tag{10}$$

where h_i and r_i are additional constants specific to vehicle i. The total energy required for this packet transmission is therefore $p_i P_t$. The total energy needed to communicate to all n vehicles is therefore given by

$$\sum_{i=1}^{n} p_i h_i (|C_i - u_i| + r_i)^{\alpha}. \tag{11}$$

Note that a minimum energy scheduler must minimize Equation 11. The scheduling system described above can be represented in the standard form of Graham et. al. [23] as

$$1||\sum_{i=1}^{n} y_i(|C_i - u_i| + r_i)^{\alpha}, \tag{12}$$

where $y_i = p_i h_i$. It can be seen that when $\alpha = 1$ and $r_i = 0$ for all i, Equation 12 reduces to conventional earliness/tardiness as discussed in Section III-A1. This is therefore a generalization of the classical Earliness-Tardiness single machine scheduling problem, which we refer to as α-Earliness-Tardiness.

In the Appendix we provide a proof of the complexity of this problem showing its NP-completeness by a reduction to the PARTITION problem [22]. This result establishes the complexity of optimal packet-based energy scheduling even when propagation is governed by simple distance dependent exponential path loss.[4] The proof given can also be easily extended to less restrictive cases than this by specifying arbitrary path loss values and by modifying the weights used in defining the instance of PARTITION.

In the remainder of the paper we focus on the timeslot-based scheduling case. This is more practical from a general scheduling viewpoint, and it will be shown that polynomial algorithms are possible even in the offline scheduling case.

B. Timeslot-Based Scheduling

In this section we consider the timeslot-based scheduling case where packets that make up a vehicle's transmission requirement can be independently assigned to (non-contiguous) time slots. A mixed integer linear program is first presented and a polynomial complexity algorithm is given based on solving a minimum cost flow graph.

1) Mixed Integer Linear Program Formulation: An offline MILP optimization is formulated whose output gives a schedule that achieves minimum energy consumption and satisfies RSU-to-vehicle com-

[4]Note that this result is true for all $\alpha \geq 1$ so we do not need to know α's exact value in practice in order to establish the NP-completeness.

munication requirements. We are given an input vehicle traffic trace consisting of N vehicles indexed by the set $\mathcal{N} = \{1, ..., N\}$. Each vehicle has a communication requirement, R_i bits, for vehicle i and the vehicles pass completely by the RSU during the time period $\mathcal{T} = \{1, ..., T\}$. Our objective is to minimize the total downlink energy needed to process the vehicular requests. This can be easily written as mixed integer linear program (MILP), i.e.,

$$\underset{K_{i,t}}{\text{minimize}} \quad \sum_{t=1}^{T} \sum_{i=1}^{N} \rho \mathcal{L}_{i,t} K_{i,t} \tag{13}$$

subject to

$$\sum_{t \in \mathcal{T}} K_{i,t} \geq R_i/B, \quad \forall i \in \mathcal{N} \tag{14}$$

$$\sum_{i=1}^{N} K_{i,t} \leq 1, \qquad \forall t \in \mathcal{T} \tag{15}$$

$$K_{i,t} \in \{0, 1\}, \qquad \forall \{i, t | i \in \mathcal{N}, t \in \mathcal{T}, \mathcal{L}_{i,t} \leq \mathcal{L}_{max}\} \tag{16}$$

$$K_{i,t} = 0, \qquad \forall \{i, t | i \in \mathcal{N}, t \in \mathcal{T}, \mathcal{L}_{i,t} > \mathcal{L}_{max}\}. \tag{17}$$

In Equation 13, $K_{i,t}$ is a binary decision variable equal to 1 if the RSU transmits to Vehicle i at time t and 0 otherwise. The objective uses the propagation path loss, $\mathcal{L}_{i,t}$, for Vehicle i at time t to calculate the total downlink energy needed by the RSU to serve the demands of the vehicles and ρ is an energy scaling factor. Constraint 14 ensures that the scheduler satisfies the communication demands of all the vehicles where B is the number of packet (payload) bits carried per timeslot. Constraint 15 and the restrictions on $K_{i,t}$ ensure that the RSU communicates with at most a single vehicle during each time slot. Note that in the above optimization, when a vehicle is outside of the maximum coverage range of the RSU, which corresponds to a path loss exceeding $\mathcal{L}_{max}$, the associated values of $K_{i,t}$ are set to zero.

The above model ensures that vehicle requirements are met with minimum expended energy and provides a lower bound on the energy needed by any realizable scheduling algorithm. Although we have solved this optimization directly for small problem sizes[5], solutions in general may require exponential time-complexity due to the MILP formulation. In the next section we present a polynomial complexity algorithm that can be used to perform this optimization instead, based on a minimum cost flow graph formulation.

[5]This can be done using standard techniques such as branch-and-bound methods [5].

2) Minimum Cost Flow Graph Formulation: In this model, we modify the above MILP formulation, without any loss of generality, to a graph problem that can be solved using a minimum cost flow algorithm. Letting

$$U_{i,t} = \rho \mathcal{L}_{i,t}, \tag{18}$$

we can write the objective as

$$\underset{K_{i,t}}{\text{minimize}} \sum_{t=1}^{T} \sum_{i=1}^{N} U_{i,t} K_{i,t}. \tag{19}$$

Note that the RHS of Constraint 14 can be made integral without changing the feasibility set, and therefore can be written as

$$\sum_{t=1}^{T} K_{i,t} \geq \lceil R_i / B \rceil, \quad \forall i \in \mathcal{N}, \tag{20}$$

and as we are seeking the minimum, Equation 20 can be tightened to

$$\sum_{t=1}^{T} K_{i,t} = \lceil R_i / B \rceil, \quad \forall i \in \mathcal{N}. \tag{21}$$

The problem after modification can now be written as

$$\underset{K_{i,t}}{\text{minimize}} \quad \sum_{t=1}^{T} \sum_{i=1}^{N} U_{i,t} K_{i,t} \tag{22}$$

$$\text{subject to} \tag{23}$$

$$\sum_{t=1}^{T} K_{i,t} = \lceil R_i / B \rceil, \quad \forall i \in \mathcal{N} \tag{24}$$

$$\sum_{i=1}^{N_v} K_{i,t} \leq 1, \qquad \forall t \in \mathcal{T} \tag{25}$$

$$K_{i,t} \in \{0, 1\}, \qquad \forall \{i, t | i \in \mathcal{N}, t \in \mathcal{T}, \mathcal{L}_{i,t} \leq \mathcal{L}_{max}\} \tag{26}$$

$$K_{i,t} = 0, \qquad \forall \{i, t | i \in \mathcal{N}, t \in \mathcal{T}, \mathcal{L}_{i,t} > \mathcal{L}_{max}\} \tag{27}$$

$$U_{i,t} \geq 0, \qquad \forall i \in \mathcal{N}, t \in \mathcal{T} \tag{28}$$

In this form the optimization can be viewed as a standard Minimum Cost Flow Problem [25]. This is shown in the graph of Figure 2, where $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is defined by a set $\mathcal{V}$ of vertices (nodes) and a set $\mathcal{E}$ of edges (arcs) connecting the nodes. For each edge $(i, j) \in \mathcal{E}$ we associate a capacity $C_{i,j}$ that denotes the maximum flow on the edge. Each edge (i, j) also has an associated cost, $U_{i,j}$, that denotes the cost per

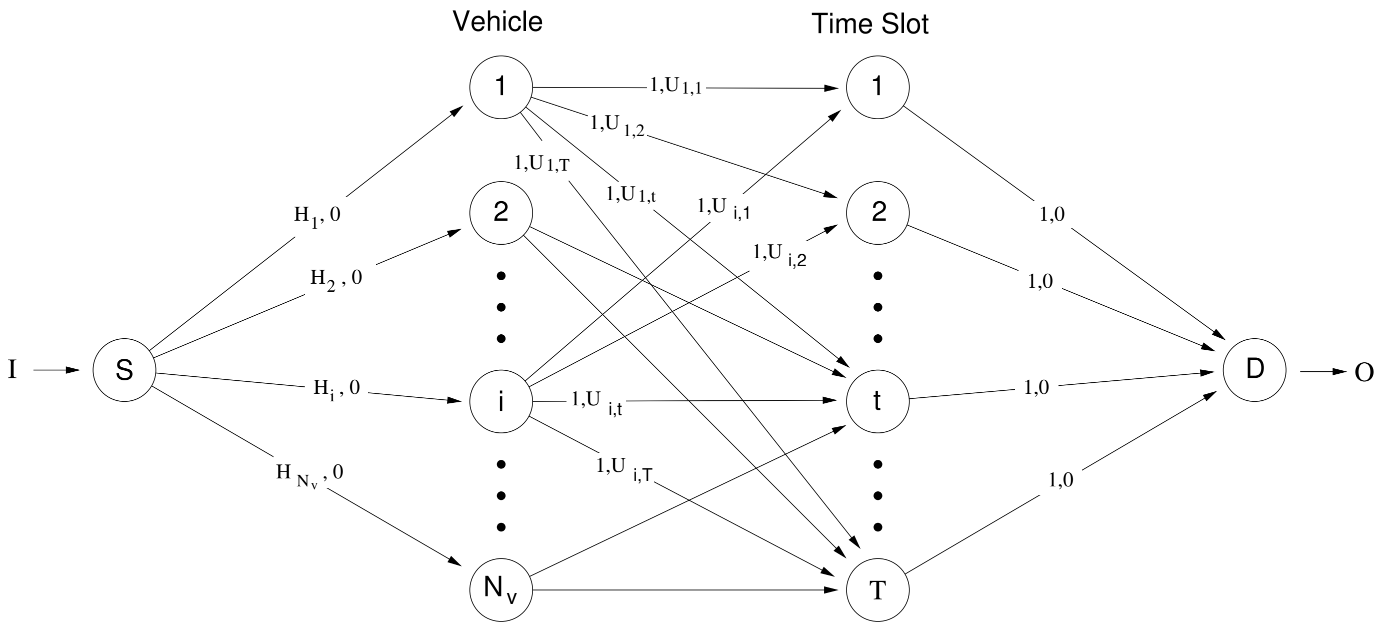

Fig. 2. Minimum Cost Network Flow Graph Representation, $\mathcal{G}$. Each edge is labeled with an ordered pair, $(C_{i,j}, U_{i,j})$, where $C_{i,j}$ and $U_{i,j}$ are the capacity and cost of using edge (i, j). The input and output links, I and O, carry a flow of $\sum_{i=1}^{N} H_i$ with a 0 edge cost.

unit flow on that edge. These are written as ordered pairs, $(C_{i,j}, U_{i,j})$, on each graph edge in Figure 2. For example, the capacity and cost of edge $(S, 1)$ in the figure is given by H_1 and 0, respectively.

We associate with each vertex $i \in \mathcal{V}$ a number b_i which represents the supply/demand of the vertex. If $b_i > 0$, the node is a supply node; if $b_i < 0$ node i is a demand node and node i is a transshipment node if b_i is zero. In our problem, the set of nodes is given by $\mathcal{V} = \{S\} \cup \mathcal{N} \cup \mathcal{T} \cup \{D\}$. The flow enters and exits the graph at dummy nodes S and D, respectively and all other nodes are transshipment. The first column of nodes represents all vehicles in $\mathcal{N}$ and the second column represents all time slots in $\mathcal{T}$. Each vehicle node has edges connected to the time slot nodes during which the vehicle is inside the RSU coverage range. For this reason, slower moving vehicles will have larger numbers of vehicle-to-timeslot graph edges, i.e., higher total RSU-to-vehicle capacity. The capacity for an edge from the source S to a vehicle node $i \in \mathcal{N}$ is the communication requirement for vehicle i denoted H_i where

$$H_i = \lceil \frac{R_i}{B} \rceil. \tag{29}$$

The capacity for an edge from any time slot node to the destination D is 1. This capacity prevents time slots from being used more than once. The edges between a vehicle $i \in \mathcal{N}$ and a time slot $t \in \mathcal{T}$ also has a capacity of 1. This ensures that only one unit of transmission requirement can be assigned to a given time slot.

The cost for using the edges originating from Node S and terminating at Node D is zero as these are dummy flow collection nodes. The cost of using the edges between nodes $i \in \mathcal{N}$ and $t \in \mathcal{T}$ is given by $U_{i,t}$ which can be computed from Equation 18. Finding the minimum cost flow for graph $\mathcal{G}$ provides the minimum energy the RSU must consume to schedule vehicle transmission requirements for the input traffic trace.

Now the importance of transforming the data in Constraints 14 to the integral data in Constraints 20 becomes apparent, because we can used the Integrality Property Theorem (e.g., Theorem 9.10 in K. Ahuja et. al. [25]), which states that "If all edge capacities and supplies/demands of nodes are integer, the Minimum Cost Flow problem always has an integer minimum cost flow". Accordingly, the resulting flow between the vehicle nodes $\mathcal{N}$ and time nodes $\mathcal{T}$ is integer. Coupled with the fact that the capacity of these edges is 1, the resulting flow is a binary matrix, $K_{i,t}$, for $i \in \mathcal{N}$ and $t \in \mathcal{T}$. When $K_{i,t} = 1$, the RSU communicates with vehicle i at time t and when $K_{i,t} = 0$ the slot t is not used for this communication. Since S and D are dummy nodes, the $K_{i,t}$ part of the flow is the vehicle schedule that we can now compute using standard flow algorithms that run in time polynomial in T and N.

IV. Online Timeslot-Based Scheduling Algorithms

A. Motivation and Notation

The results in Section III-B give a lower bound on the downlink RSU energy needed to fulfil vehicular packet requirements. In order to compute these bounds, the energy costs associated with a given packet transmission must be known. Although it is difficult to precisely know this information in general situations, in certain scenarios excellent estimates of this cost can be readily made [26][27]. Accordingly, we consider a highway scenario where vehicles travel at a constant speed through the RSU coverage area [19]. When vehicles enter the RSU coverage area, they announce their location and speed, information that can subsequently be used to estimate future energy transmission costs[6]. The results we present in Section V show that provided that the deterministic components of path loss are dominant, large improvements in performance are possible[7]. This would typically be the case in highway situations.

Three online algorithms are introduced which attempt to reduce total energy and which have different processing time complexity. In the following sections, t' is used to denote the current time slot. The

[6]This information can be obtained from GPS readings at the vehicle, for example.

[7]We assume that downlink power control is used during RSU-to-vehicle communication. This can be accomplished in a variety of ways such as using a short two-way handshake prior to user data packet transmission.

vehicle arriving at the RSU coverage range at time t' is denoted v'. The algorithms are applicable for more than one vehicle arriving at the same time slot but for simplicity of the presentation we assume one vehicle can arrive during a time slot. The set $\mathcal{N}'$ contains all vehicles inside the RSU coverage range at t' with unsatisfied communication requirements. Vehicle $i \in \mathcal{N}'$ unsatisfied demand at time t' is denoted H'_i. The energy cost that the RSU expends to communicate with vehicle i at time slot t is denoted $U_{i,t}$ and is computed according to Equation 18. As each vehicle has a different arrival time to the RSU and different speed, the time slot representing its departure time t''_i from the RSU coverage can be different. For $i \in \mathcal{N}'$ the set of time slots between t' and t''_i is called $\mathcal{T}_i$. The time elapsed between current time and the departure of last vehicle of $\mathcal{N}'$ is $\mathcal{T}' = \bigcup_{i \in \mathcal{N}'} \mathcal{T}_i$.

B. *Greedy Minimum Cost Flow (GMCF)*

In Greedy Minimum Cost Flow (GMCF), the vehicle schedule is obtained using a greedy version of the bound formulation from Section III-B2. However, unlike the bound which incorporates all vehicle communication requirements at once, GMCF constructs a graph similar to the one in Figure 2 but limited to those vehicles that are currently inside the RSU coverage range.

GMCF is executed upon arrival of a new vehicle v' into the RSU range at time t'. The directed graph constructed for GMCF is similar to that of Figure 2 with $\mathcal{N}'$ replacing the vehicle column and $\mathcal{T}'$ replacing the time slot column. The capacity between Node S and Node $i \in \mathcal{N}'$ is given by H'_i which represents Vehicle i's *unsatisfied* communications requirements at time t'. The supply, I', to Node S is the demand O' from Node D, i.e.,

$$I' = -O' = \sum_{i \in \mathcal{N}'} H'_i. \tag{30}$$

Let this graph be denoted $G'(V', E')$, where $V' = \{S\} \cup \mathcal{N}' \cup \mathcal{T}' \cup \{D\}$ and E' is the set of edges between S,$\mathcal{N}'$, $\mathcal{T}'$ and D. Let flow $\mathcal{F}'$ be the minimum cost flow for graph $G'(V', E')$ and let the part of $\mathcal{F}'$ representing flows between vehicle nodes $i \in \mathcal{N}'$ and time slots $t \in \mathcal{T}'$ be $K'_{i,t}$. Then, $K'_{i,t}$ is the schedule for vehicles $i \in \mathcal{N}'$ during the time period $\mathcal{T}'$ or until a new vehicle enters the RSU range. In the latter case, the remaining unexecuted part of $K'_{i,t}$ is ignored and the algorithm is restarted. Whenever Vehicle i is served, its corresponding unsatisfied demand H'_i is reduced by 1.

Reference [25] provides several algorithms for solving the minimum cost flow problem in polynomial time. Although algorithms such as Capacity Scaling, Cost Scaling and Double Scaling are polynomial, they are not strongly polynomial. For example, the Double Scaling Algorithm solves the problem in

$O(nm \log U \log(nC))$, where n is number of nodes, m number of edges, C is the maximum cost and U is the maximum capacity. In GMCF, the cost over an edge can be quite large. Algorithms such as Repeated Capacity Scaling and Enhanced Capacity Scaling provide strongly polynomial time execution. For example, the complexity for Enhanced Capacity Scaling is $O((m \log n)(m + n \log n))$ is an improvement as they do not depend on C or U.

In GMCF the number of nodes, n is $v+t$, where v is the number of vehicles inside the coverage range and t is the number of time slots needed to exit the RSU coverage range. As for the number of edges, m it can be assumed to be $m = v \times t$ as if $\mathcal{N}'$ and $\mathcal{T}'$ are fully connected. Substituting these values in the Enhanced Capacity Scaling complexity, the GMCF complexity can be expressed as $O(v^2 t^2 \log(v + t))$

Although the GMCF algorithm achieves good energy usage, running the algorithm can be time consuming. In the following sections we introduce two heuristics, i.e., Static Scheduler (SS) and Nearest Fastest Set (NFS) which are more efficient from a time complexity viewpoint.

C. Static Scheduler (SS)

The basic idea in SS is to sort vehicles according to the energy they would use if they were served at energy optimal positions. The algorithm is static in the sense that these weights do not change as the vehicle propagates through the RSU coverage range. SS serves vehicles with high energy costs first in order to reduce the total energy required. For example, if two vehicles with equal demands are travelling a different speeds it is better to serve the faster vehicle first in order to avoid the extra energy costs of serving it at higher distances. SS would allocate the faster vehicle all its requested time slots and then search for other time slots to assign to the slower vehicles.

SS is executed upon the arrival of a new vehicle v' into RSU range at time t'. The algorithm consists of two phases, namely, weight computation and scheduling. In the weight computation phase, the weight W_i' for each vehicle $i \in \mathcal{N}'$ is computed by finding its optimal energy cost. This can be best described by using a minimum cost flow graph as in Figure 2 but restricted solely to the vehicle in question. We denote that flow graph as G_i for Vehicle i. An example of this is shown in Figure 3, and contains one source node i which generates a flow equal to Vehicle i's remaining demand H_i' at time t'. It also contains node D which is a dummy destination node with demand equal to $-H_i'$. The intermediate nodes represent time slots of the set $\mathcal{T}_i$, starting at t' and ending with t_i''. The edge capacity between the vehicle i node and time node $t \in \mathcal{T}_i$ is set to 1. The cost $U_{i,t}$ over these edges is computed according to Equation 18. The capacity and cost for edges between time node $t \in \mathcal{T}_i$ and dummy destination D are 1 and 0 respectively.

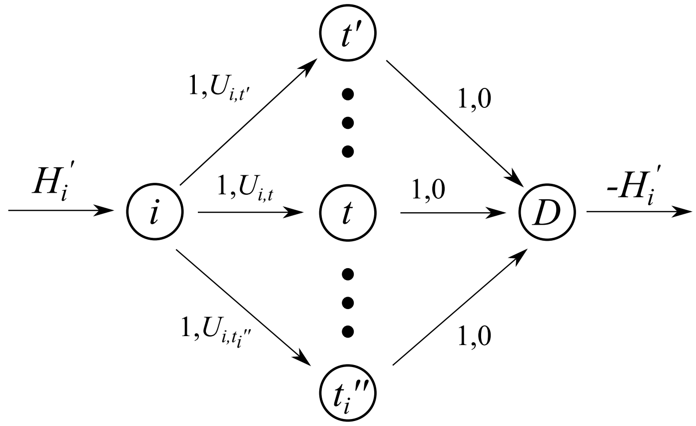

Fig. 3. Example Flow Graph, G_i, for the SS Algorithm. This is used in the weight calculation phase.

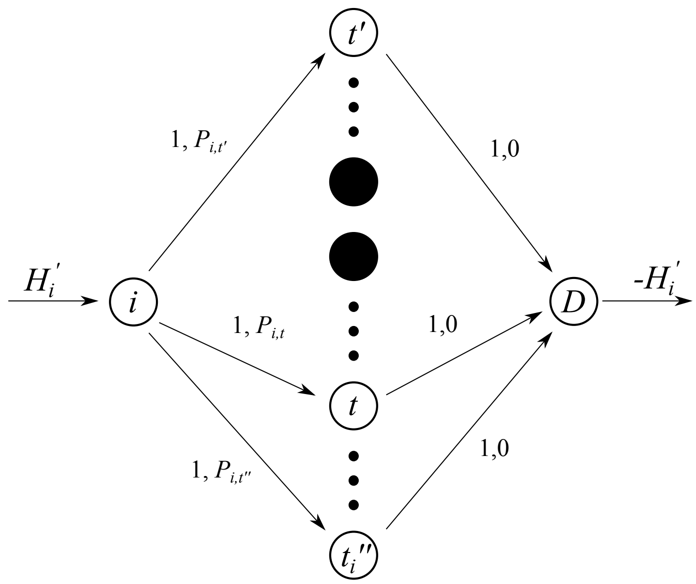

Fig. 4. Example Flow Graph, M, for the SS Algorithm. This is used in the scheduling phase. Shaded slots have already been assigned for communication with higher priority vehicles.

The minimum cost flow for graph G_i is computed and the cost associated with this flow is the weight W_i' for Vehicle i.

In the scheduling phase a single graph, M, is formed as shown in Figure 4. The graph consists of one vehicle source node, time nodes and a dummy destination node D. Initially, the time nodes consists of all elements of the set $\mathcal{T}'$. The edge capacity and cost from any time node to the D node are 1 and 0 respectively. Vehicles are sorted according to their weights W_i' where $i \in \mathcal{N}'$. In weight descending order, schedules are computed for each vehicle one at a time. Let vehicle $i \in \mathcal{N}'$ be the currently selected vehicle. The supply to the vehicle node i is set to be H_i' and demand by D node is set to $-H_i'$. The capacity of edges between vehicle i node and time $t \in \mathcal{T}'$ is set to 1. The cost, $P_{i,t}$, of these edges is set as

$$P_{i,t} = \begin{cases} d_{i,t}, & \text{if } t \in \mathcal{T}_i \\ \infty, & \text{otherwise.} \end{cases} \tag{31}$$

Setting the cost to ∞ for time slots when Vehicle i will be outside RSU coverage range prevents selecting them in the schedule. Let the flow $\mathcal{F}$ be the flow that minimizes the cost for graph M. Let the part of $\mathcal{F}$ specifying the flows between vehicle i and nodes $t \in \mathcal{T}'$ be $K'_{i,t}$. $K'_{i,t}$ is the schedule for vehicle i. The set of time nodes in graph M is updated according to

$$\mathcal{T}' = \mathcal{T}' - \{t | K'_{i,t} = 1, t \in \mathcal{T}'\}. \tag{32}$$

This removes the time slots already scheduled to serve Vehicle i from the set of available time slots for remaining vehicles. The vehicle following Vehicle i in weight order is selected. Graph M supply, demand, and edge costs are updated according to the following vehicle requirements and distances in the same way like vehicle i. The flow that minimizes the cost for the updated M is found for the new vehicle and the process is repeated until all vehicles are scheduled. After the scheduling phase has been executed for all vehicles $i \in \mathcal{N}'$, the schedule for all vehicles would be $K'_{i,t}$ where $i \in \mathcal{N}'$ and $t \in \mathcal{T}'$. This schedule will execute until the last vehicle of $\mathcal{N}'$ exits or a new vehicle arrives into the RSU range.

In determining the complexity of SS, we will use the same notation as in the complexity analysis of GMCF, where v is the number of vehicles inside the range and t is the number of time slots needed for them to exit the RSU range and we add to them H_m as the maximum number of slots a vehicle can demand.

SS is invoked upon the arrival of each new vehicle. In the weight computation phase, the process of finding the weight is executed for each vehicle. Finding the weight is equivalent in complexity to finding the minimum of an array of length t. As we find the minimum H_m number of times for each v vehicle, the complexity of weight computation phase can be stated as $O(H_m vt)$.

In the scheduling phase, the search for the highest weight among vehicles is $O(v)$. Scheduling the vehicle with the highest weight is similar to finding the minimum of an array of length t repeated H_m times. As this is repeated for each vehicle, the complexity of the scheduling phase can be stated as $O(H_m vt)$. Thus the total complexity of SS is $O(H_m vt)$. This is a big improvement over GMCF provided that $H_m < vt$, which is safe to assume.

D. Nearest Fastest Set (NFS) Scheduler

The Nearest Fastest Set (NFS) Scheduler uses vehicle inputs in a simpler and more dynamic way than SS. The motivation is to dynamically change the weight of the vehicles according to remaining demands. If a vehicle is selected for communication from the RSU at the current time slot, its weight is reduced while the weight of other vehicles is increased. The notion of "fastest" comes from the role that vehicle speed plays in weight computation. Consider the case where two vehicles are together and moving away from the RSU. If they are moving at different speeds, then serving the faster one first will lead to lower overall energy consumption. This is clearly due to the fact that in the next time step the faster vehicle will be farther away from the RSU. NFS uses this by embedding the effect of vehicle proximity and velocity in the weight calculation when considering which vehicle to serve in a given time slot. The execution details of NFS are explained below.

NFS consists of preparation, execution and updating phases. The preparation phase is invoked upon the arrival of a new vehicle v' into RSU range at time t'. The time slots during which vehicle v' will be closest to the RSU are identified. This is done using a graph $G_{v'}$ similar to the one used in the weight computation phase of SS algorithm and shown in Figure 3. Vehicle v' requirements $H'_{v'}$ constitutes the supply to the vehicle node. Dummy destination node demand is set to $-H'_{v'}$. The time slot nodes in G'_v represent the time slots in the set $\mathcal{T}_{v'}$. The capacities for all the edges in graph $G_{v'}$ are set to 1. The costs over the edges between vehicle v' node and time node $t \in \mathcal{T}_{v'}$ is set to $U_{v',t}$ which is computed according to Equation 18. The costs over edges ending in node D are set to 0.

Let $\mathcal{F}$ be the flow that minimizes the cost for graph $G_{v'}$. The weight $W_{v',t'}$ for vehicle v' is the cost of the flow $\mathcal{F}$. Let the array $Z_{v',t}$ represent the part of the flow $\mathcal{F}$ that specifies the flow between the vehicle v' node and time slots $t \in \mathcal{T}_{v'}$. As all supply, demand, capacities and cost of graph $G_{v'}$ are integers, and according to the Integrality Theorem, the flow $\mathcal{F}$ consists of integer values. As the maximum capacity of any edge in $G_{v'}$ is 1, the $Z_{v',t}$ is a binary array. Since the flow $\mathcal{F}$ minimizes the energy cost, in $Z_{v',t}$, the time slots during which vehicle v' is closest to the RSU are set to 1. These are the candidate time slots during which vehicle v' would like to communicate with the RSU.

As the preparation phase is executed for every vehicle $i \in \mathcal{N}'$ and $t \in \mathcal{T}_i$ currently inside the RSU range upon their respective arrivals, there is already a weight $W_{i,t'}$ and separate $Z_{i,t}$ for each vehicle $i \in \mathcal{N}'$ identifying the time slots each vehicle would like to use.

The execution phase happens every time slot. Let the current time be t'. If there is no vehicle $i \in \mathcal{N}'$

that requires the current time slot, then there is nothing to schedule and the execution phase and update phase are terminated. But if not, let the set of contending vehicles be $\mathcal{E}$. The weights of these vehicles $W_{i,t'}$ are compared and the vehicle with the highest weight is allocated the current time slot t'. Let the vehicle with the highest demand be x. Then the remaining demand for vehicle x is decreased by 1 as it has been scheduled for downlink transmission in the current time slot. The candidate time slot array for vehicle x is updated by setting $Z_{x,t'} = 0$ so its weight will be reduced.

The update phase is for the vehicles that contended for time slot t'. A new set of candidate time slots and weights $W_{i,t'+1}$ for $i \in \mathcal{E}$ is computed. The start time is $t' + 1$ instead of t' because they will be contending for future time slots following t'. These new candidate time slots and vehicle weights are computed in the same way as in the preparation phase.

In determining the complexity of NFS, we again use the notation from before where v is the number of vehicles inside the range and t is the number of time slots needed to exit the RSU coverage area and H_m as the maximum vehicle requirement. The NFS preparation phase is executed upon the arrival of a new vehicle. Unlike SS, this is executed not for all vehicles inside the range but only for the newly arriving vehicle. Thus, the complexity of this phase is $O(H_m t)$. In the execution phase, determining if one or more vehicles requires communication with the RSU is an addition operation across all vehicle candidate slot arrays. This is equivalent to $O(v)$. Only when there are multiple candidates for a given time slot is the update phase executed. The update phase complexity is equivalent to the preparation phase, but it involves other vehicles inside the RSU, thus the update phase complexity is $O(H_m vt)$. When compared SS, NFS complexity is less. Only during the times of strong contention will NFS complexity become equal to that of SS.

V. Performance Evaluation

In this section, the performance of the proposed algorithms is investigated. The theoretical bound for energy required by the RSU to serve the input vehicle requirements as derived in Section III-B2 is referred to as *Bound* in the graphs. The bound is compared to the online algorithms (GMCF, SS and NFS) proposed in Section IV.

The online algorithms use knowledge of vehicle position and associated estimates of downlink transmission energy costs. For this reason two sets of results will be presented. The first assumes that an accurate prediction of energy costs is possible based on a deterministic path loss scenario using a distance dependent exponential path loss model. These results will give an indication of the best-case potential for energy savings at RSU using the proposed algorithms. In many practical systems however, there will be dominant deterministic propagation with random components due to effects such as shadowing. For this reason we also present results which include errors due to strong shadowing components. It is well-known that in highway scenarios with predominantly line-of-sight propagation, that average path loss has strong deterministic components. A recent measurement based paper has confirmed this even for the vehicle-to-vehicle case where average antenna heights are low compared with the RSU-to-vehicle case [27][8].

In [28] and [29] vehicular traffic models were surveyed including those for intra-city and outside city scenarios. Highway traffic models are known to vary greatly from those in urban settings. For example, Reference [20] showed that car-to-car interaction was a minimum requirement for realistic highway traffic flow models. A special characteristic of the highway environment is the tendency to maintain constant speed for long durations. Reference [26] for example, models vehicles as belong to different classes with different uniformly distributed speed ranges and in [19][30], vehicle arrivals were taken to be Poisson distributed. The mobility model proposed in this paper is from [19] and [26]. The highway consists of several lanes, which permit vehicles to pass each other without a change in speed. Vehicles traveling on the highway belong to one of several classes. Each class has a Poisson arrival model and a desired speed at which the vehicles travel. Vehicles from each class do not overtake each other but can overtake vehicles from the other classes. There is a single RSU in the tested highway segment and vehicles travel in one direction.

In the following two sections we present the results of experiments conducted using two and three classes of vehicles. Large scale path-loss using a distance dependent exponential path loss model is used in the first set of results. Following these results, we include random log-normal shadowing for two different levels of shadowing. The random shadowing components are unknown to the scheduler which bases its decisions on deterministic positional information. In the graphs, each plotted point is an average of multiple runs to ensure the results are true representation of the performance of the algorithms. The value of the points are normalized to the first point of the Bound graph in each figure.

In the first experiment the traffic consists of two classes of vehicles with arrival rates of $\lambda_1 = 1/22$ and $\lambda_2 = 1/22$ vehicles/sec, respectively. The communication requirement for each vehicle was tested for low and medium fixed values. The results are presented in Figure 5 for low demand and in Figure 6 for high

[8]The scenario that we consider also has the advantage that since the RSU is stationary, it can therefore measure and learn the propagation environment, which would lead to increased accuracy.

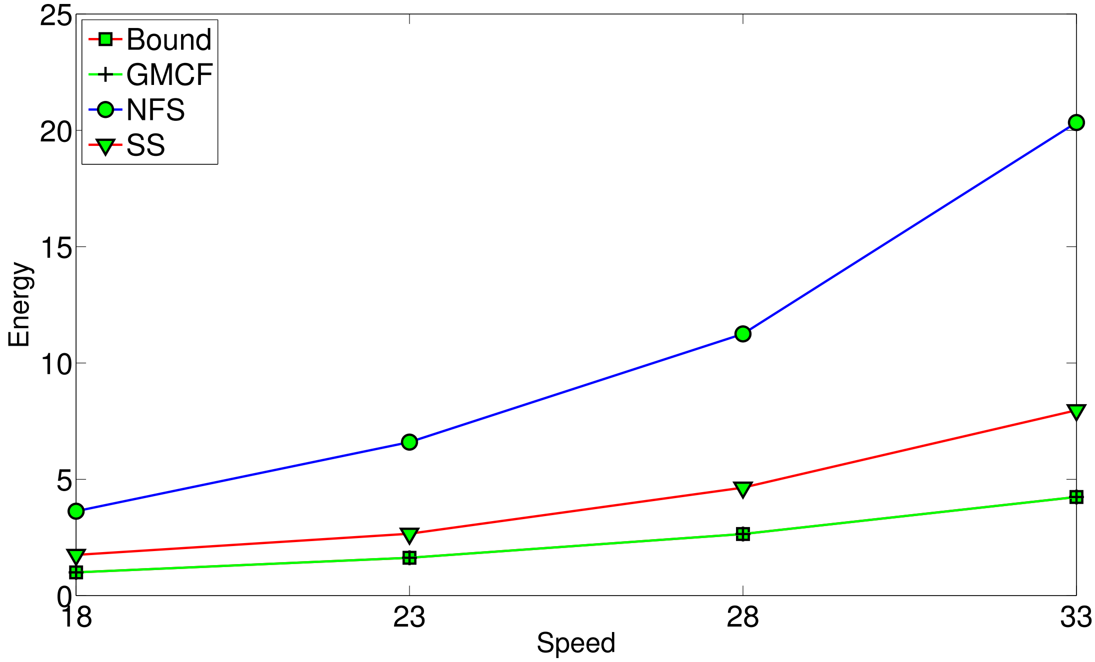

Fig. 5. Performance of GMCF, SS and NFS for Two Vehicle Classes under Light Loading. No Shadowing Components.

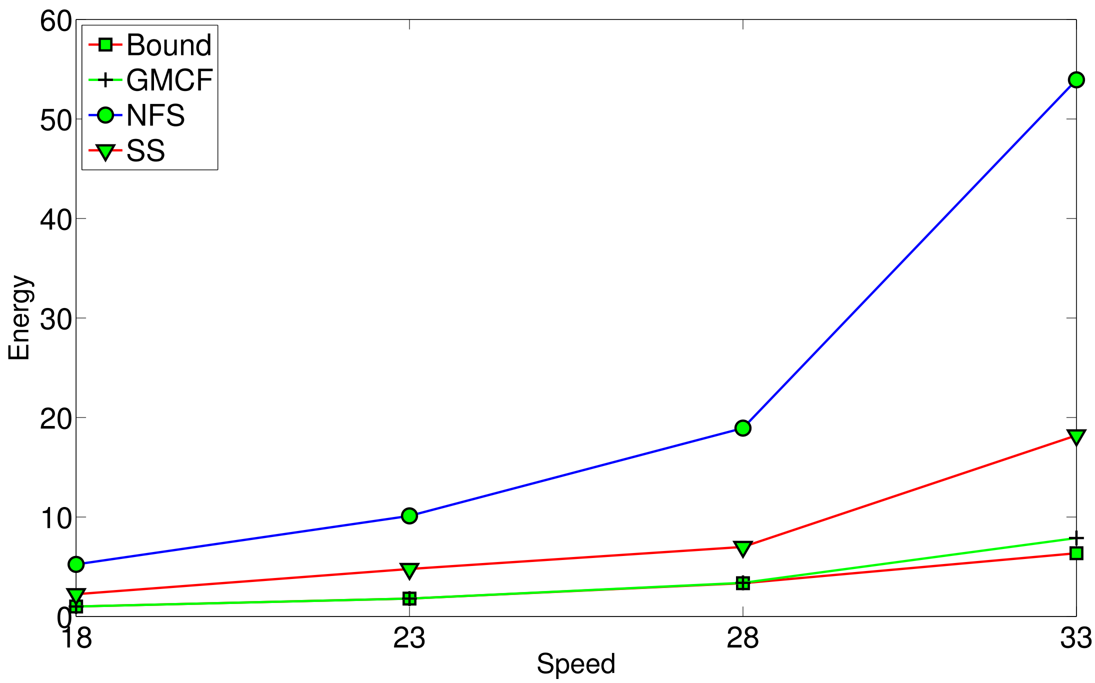

Fig. 6. Performance of GMCF, SS and NFS for Two Vehicle Classes under Medium Loading. No Shadowing Components.

demand. Class 1 speed is maintained at 18 m/sec and Class 2 speed is tested for velocities: $18, 23, 28$ and 33 m/sec. Each point is the average of 17 runs.

As vehicular speed increases, the amount of time slots the faster moving vehicles, i.e., Class 2, can spend inside the communication range decreases. This forces the algorithms to communicate with the vehicles at more distant locations from the RSU in order to satisfy their communication requirements. So as expected, this shows that the total energy needed is an increasing function of average vehicle speed. Also, it can be seen in Figures 5 and 6 that the performance of the GMCF algorithm is very close to that obtained from the lower bound. Under low demand levels as in Figure 5, even with the increase of Class 2 speed which allows vehicles to pass Class 1 vehicles, these periods of scheduling contention do not seem to be long or frequent enough to result in significant increases in energy compared with the bound. Under heavier loading, as in Figure 6, with Class 2 speed at 33 m/sec, periods of contention become longer, thus forcing the GMCF algorithm to communicate with vehicles at larger distances. The price paid for GMCF's good performance is a relatively long computation time needed to solve the associated minimum cost flow problem.

In these results the SS scheduler consistently requires more energy than the Bound or GMCF. But as shown earlier, its computational requirements are much less than that required by GMCF. Under light vehicular load, the energy required by SS is almost twice that required by the bound as shown in Figure 5 and grows linearly. Under heavier loading, as in Figure 6, the increase becomes more dramatic. This is due to the static nature of the algorithm. As the difference between Class 1 and Class 2 velocities increase, many more vehicle interactions occur, creating longer periods of scheduling contention which are more difficult for the scheduler to efficiently resolve. In these periods, unlike the GMCF algorithm which seeks minimum energy for all the vehicles currently inside the RSU coverage range, SS tries to minimize the total energy by seeking local minima for individual vehicles in order of their weightings. This drawback of static scheduling is illustrated next with the following example. Suppose two vehicles V_1 and V_2 are inside the RSU coverage range, and V_1's speed is must lower than that of V_2, and the remaining demands are 2 and 1 units of timeslots, respectively. Assume also that both vehicles are closest to the RSU at the same time. Since the candidate location for V_2 is at the same minimum RSU distance, SS serves V_1 at this location, causing V_2 to be served at a more distant location, significantly raising the energy cost for serving V_2, and resulting in a higher overall energy cost. If V_2 had been served at the location the lowest distance, the total energy would have been lower than what SS would choose.

As for the NFS scheduler, although it is a dynamic algorithm compared to SS in terms of continuously reevaluating weights for contending vehicles after assigning a timeslot, scheduling contention is not addressed until they actually happen. This way, if a contention arises, it is resolved by assigning a slot in the future, which prevents more past energy efficient slots from being used. This explains the increased energy required by NFS compared with the others.

We have also simulated results using a first-come-first-served, i.e., FCFS, scheduler. In this algorithm the RSU places each vehicle request in a FCFS queue and immediately begins service in that order. As a result, FCFS tends to serve vehicles at the outer edge of the RSU coverage area, resulting in very high power consumption values. When compared with the results of our algorithms, FCFS usually results in total energy requirements which are orders of magnitude higher than that of the algorithms shown in our graphs. For this reason we have not presented these results, but this method is clearly a poor approach when energy efficiency is desired.

We now present results for three classes of vehicles. The first class maintains a speed of 18 m/sec across all experiments. The second class is tested for four different speeds, $18, 20, 22$ and 24 m/sec. The third class is tested also for four different speeds, $18, 23, 28$ and 33 m/sec. The arrival rate, λ, for each class is $1/30$ vehicles/second. The results presented here are the average of 10 independent experiments.

In Figure 7, all three classes travel at 18 m/sec. The total energy needed by the four algorithms are compared against different levels of normalized demand, ranging from 4 to 10. It can be seen that as before, the GMCF algorithm is very close to the lower bound. This is partially due to the uniformity of the traffic, as all vehicles from all three classes are traveling at the same speed. The scheduling contention tends to be minimal, so there is less value in the bound's use of non-causal information. The behavior of the SS algorithm is similar to the former experiment where under low demand levels, the energy requirement growth is almost a linear relation to that of GMCF. But at high demand levels, the increase in SS energy demand starts to increase with higher rates than that of GMCF and the bound. The weakness of the NFS algorithm is more evident here. As the demand increases, scheduling contention increases and NFS can only solve it by allocating slots in the forward direction, wasting more energy efficient slots in the backward direction.

In Figure 8, Class 1 vehicles are traveling at 18 m/sec, Class 2 at 24 m/sec and Class 3 are traveling at 33 m/sec. This causes the vehicles to interact with each other creating longer and more complex periods of scheduling contention not envisioned by the online heuristics compared to the bound computation. This is why we can see in this figure that when the demand is high, there is a clear difference between the bound and the GMCF algorithm. The NFS and SS behavior is close to that of Figure 7, but as the contention periods are longer and more frequent, the increase in energy requirements as demand increases is steeper than that of Figure 7.

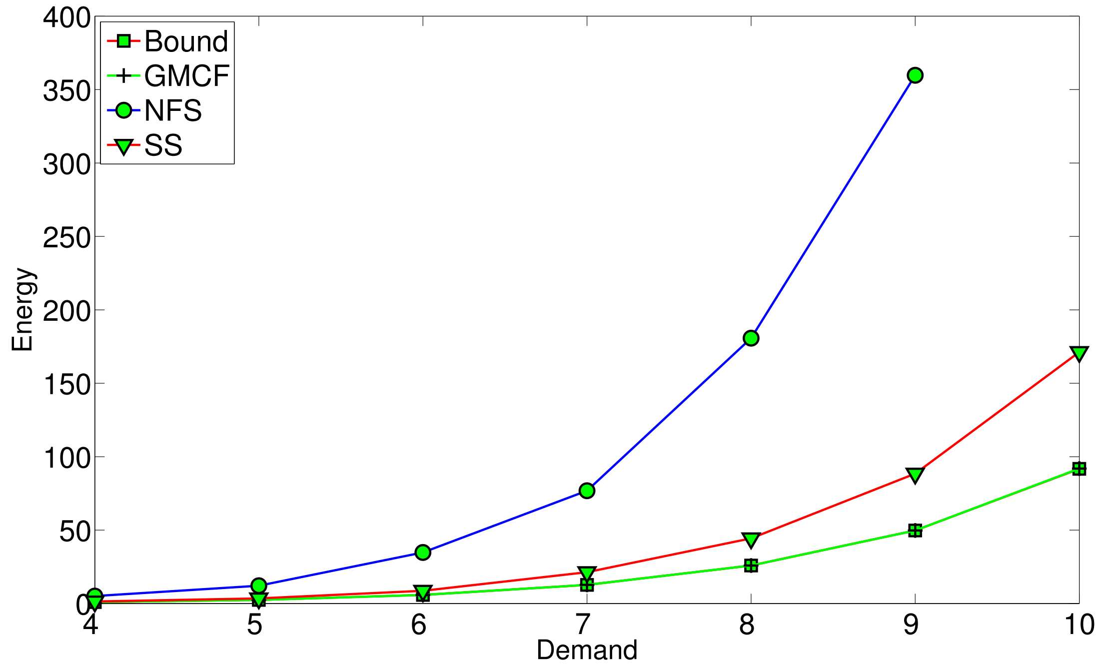

Fig. 7. Energy Requirements for Three Vehicle Classes Traveling at the same Speed. No Shadowing Components.

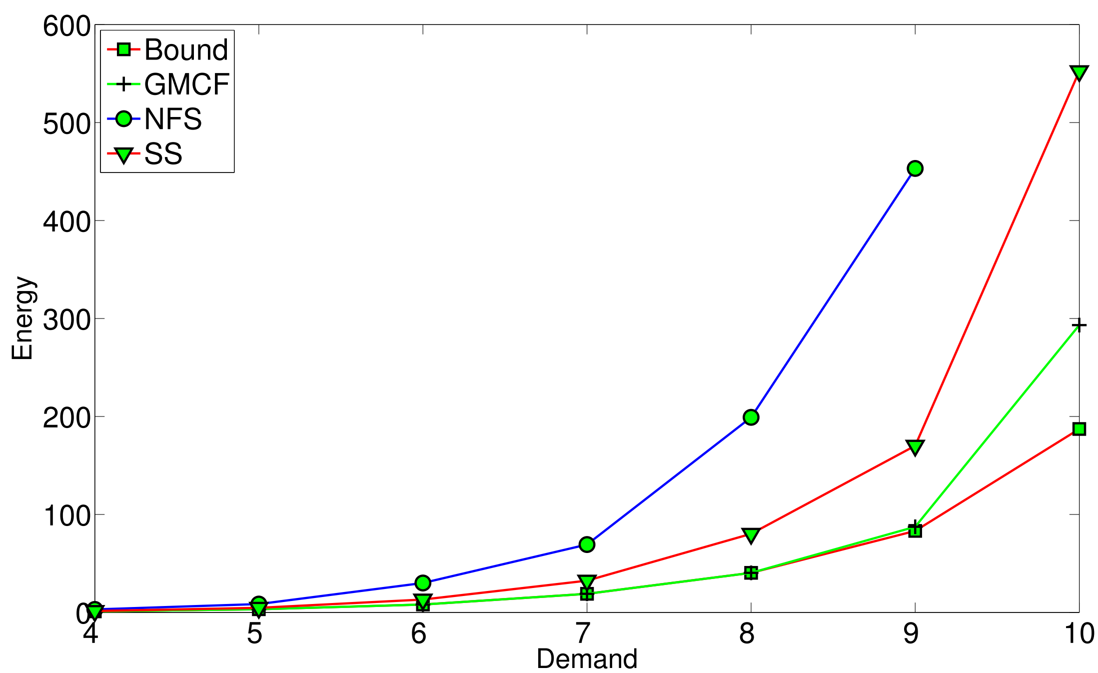

Fig. 8. Energy Requirements for Three Vehicle Classes Traveling at Different Speeds. No Shadowing Components.

The results shown above consider the case where the schedulers are able to accurately predict the energy costs of each vehicle communication. These results clearly show that it is possible to significantly decrease the energy costs of RSU downlink transmission. In the next set of results we include propagation shadowing effects which results in unpredictable randomness in these estimates. A conventional log-normal shadowing model is used with zero mean and standard deviations of $\sigma \in \{4, 12\}$ dB [31] [32]. The simulation results include three classes of vehicles. The performance of the different algorithms has been tested for various demand levels and different mixes of vehicle class speeds are examined. Class 1 always maintains a speed of 18 m/s for all experiments. Class 2 speeds are 18, 20, 22 and 24 m/s. Class 3 speeds are 18, 23, 28 and 33 m/s. The presented results are the average of 8 random iterations. The energy required by the two shadowing cases is given by the solid and dashed lines, respectively. In the graph legend, "-s1" and "-s2" are appended to the algorithm names for the two shadowing cases. Note that for these two scenarios we have recomputed the lower bound using knowledge of the path loss values, which gives an absolute bound on system performance.

Figure 9 shows the performance when all classes have a speed of 18 m/s and Figure 10 when Class 1, 2 and 3 has speeds of 18, 24 and 33 m/s, respectively. To simplify these two graphs we have not included results for the lower bound. Figure 11 shows the required energy to serve vehicles with a communication requirement of 8 timeslots under different velocity mixes. In this graph we have also included the lower bound calculation. An interesting phenomenon occurs when random shadowing components are included. Since the bound is aware of these values, it routinely schedules packet transmissions at greater distances from the RSU, provided that the shadowing gives a favourable path loss. This does not happen in our algorithms and for this reason the power consumption bound *decreases* when randomness is introduced.

From these figures, it can be seen than increasing the random shadowing component significantly increases the energy required by the RSU compared to the non-random case, as would be expected. However, the relative performance between the different algorithms is maintained, i.e., GMCF is still close to the bound except in situations where there is heavy demand and high differences in vehicle speeds.

When comparing the energy needed to satisfy vehicle communication requirements with and without random shadowing components we find that significant increases in power consumption results from the scheduler's unawareness of these values. However, Figures 9, 10 and 11 indicate that provided there is a known dominant component of path loss, then there is a high value in using the algorithms. These and other

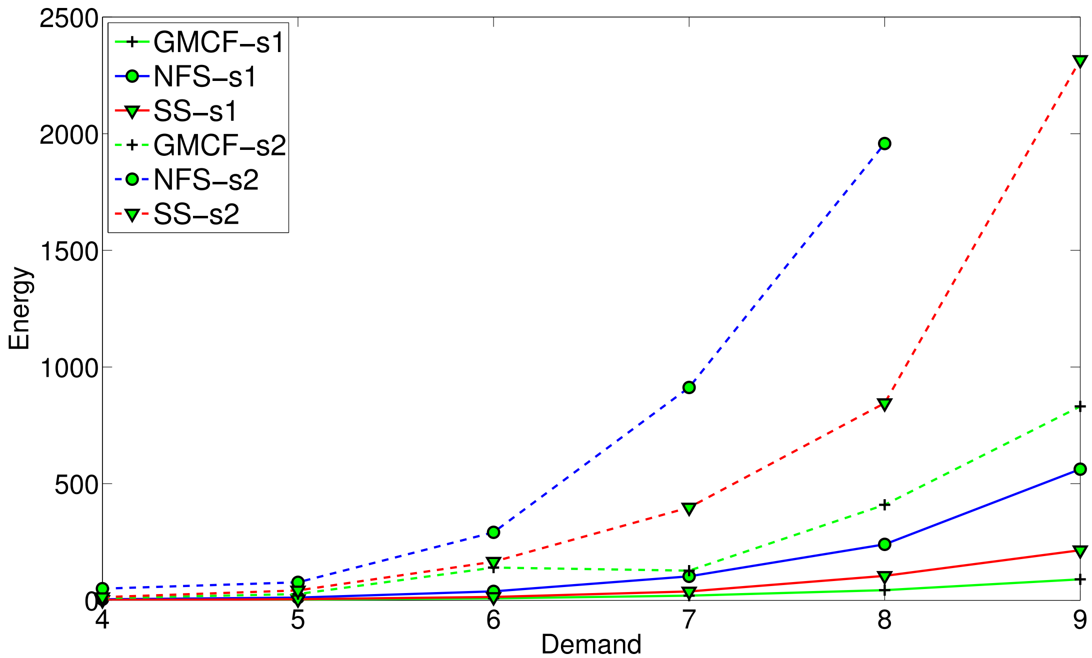

Fig. 9. Scheduling Performance with Two Levels of Log-normal Shadowing.

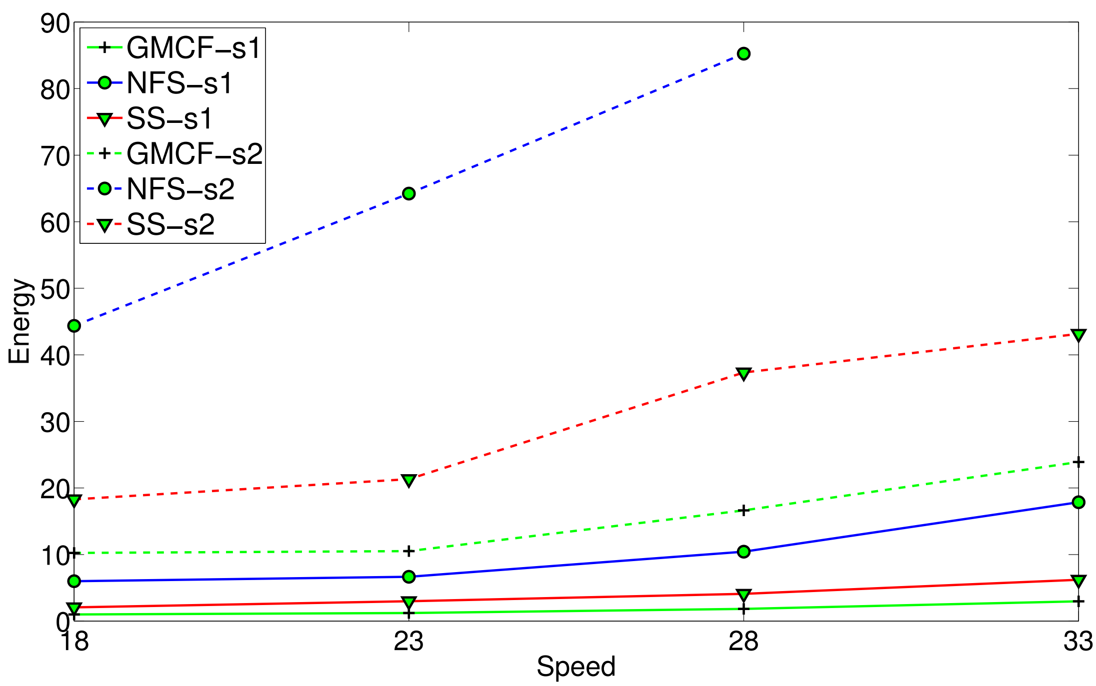

Fig. 10. Scheduling Performance with Two Levels of Log-normal Shadowing.

results that we have obtained show that although the overall energy costs increase with increases in energy cost uncertainly, in practical highway scenarios there will clearly be a dominant enough deterministic path loss component that can be used to significant increase energy savings using the proposed algorithms.

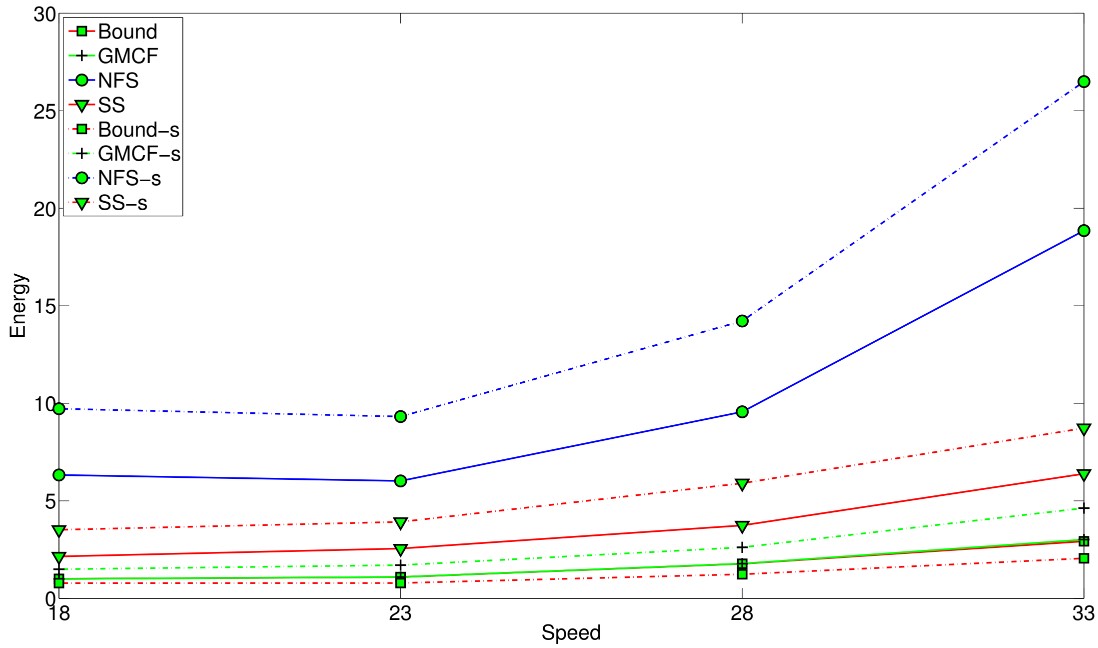

Fig. 11. Scheduling Performance with Shadowing. Normalized vehicle demand of 8.

VI. Conclusions

Roadside infrastructure can be used to provide a wide variety of commercial services for vehicular networks. One particular challenge is that of providing roadside radio coverage in highway locations where wired electricity is not available. In this case, roadside units (RSUs) powered by renewable energy such as solar or wind power, is a viable alternative. The cost of provisioning this type of roadside infrastructure is dependent on the average power consumption of the RSU, and can be significantly reduced by energy efficient scheduling. In this paper, we have considered the problem of satisfying vehicular communication requirements while minimizing the downlink energy needed by the roadside unit.

The associated scheduling can be either *packet-based* or *timeslot-based*. We first showed that for packet-based scheduling the offline problem can be formulated as a generalization of the classical single-machine job scheduling problem with earliness and tardiness penalties, referred to as α-Earliness-Tardiness. A proof was given which shows that the problem is NP-complete even under a simple distance-dependent exponential radio path loss assumption. In the timeslot-based scheduling case, we formulate the problem as a Mixed Integer Linear Program (MILP) which was shown to be solvable in time which is polynomial in the number of timeslots using a minimum cost flow graph construction. These formulations provide lower bounds on the energy requirements needed to satisfy arriving vehicular communication requests.

The paper then introduced energy efficient online traffic scheduling algorithms. The first was motivated by a greedy implementation of the MILP formulation which optimizes over finite overlapping time windows. This is referred to as Greedy Minimum Cost Flow (GMCF). Two other algorithms with reduced complexity compared with GMCF were then proposed. The Nearest Fastest Set (NFS) scheduler uses vehicle location and velocity inputs to dynamically schedule communication activity. The Static Scheduler (SS) performs the same task using a simple position-based weighting function. Results from a variety of experiments show that the proposed scheduling algorithms perform well when compared to the energy lower bounds in vehicular situations where path loss has strong deterministic components so that energy costs can be estimated. Our results also show that near-optimal results are possible but come with increased computation times compared to our heuristic algorithms.

APPENDIX

In this section we show the NP-completeness of the α-Earliness-Tardiness Problem formulated in Section III-A2. This is done by a reduction from the well-known PARTITION PROBLEM [22]. In the decision version of PARTITION we are given a set of n integers and must answer the question: "Can we divide the given set into two subsets such that the sum of the numbers in each set are equal?" Our proof is by reduction of PARTITION to the following version of α-EARLINESS-TARDINESS where $\alpha \geq 1$.

INPUT: n jobs, each job $i = 1, \ldots, n$ with its own processing time p_i, weight w_i, shift amount r_i, and due date D_i.

OUTPUT: A single-machine (non-preemptive) schedule of the jobs that minimizes

$$\sum_{i=1}^{n} w_i(|C_i - D_i| + r_i)^{\alpha}$$

where C_i is the completion time of job i in the schedule. Note that we assume that all our input data are integers and that jobs can only start at integral time points.

Theorem 1. α-EARLINESS-TARDINESS *is NP-complete for any real* $\alpha \geq 1$.

Proof: Suppose we are given an instance of PARTITION with n objects, each with a value $p_i,\ i = 1, \ldots, n$. Let $P := \sum_{i=1}^{n} p_i$. We define an instance of α-EARLINESS-TARDINESS as follows:

- We have $n + 3$ jobs: Jobs $1, 2, \ldots, n$ correspond to the PARTITION objects; each has processing time p_i, weight 1, and due date D_1. Job $n + 1$ has processing time $p_{n+1} = 1$, weight w and due date

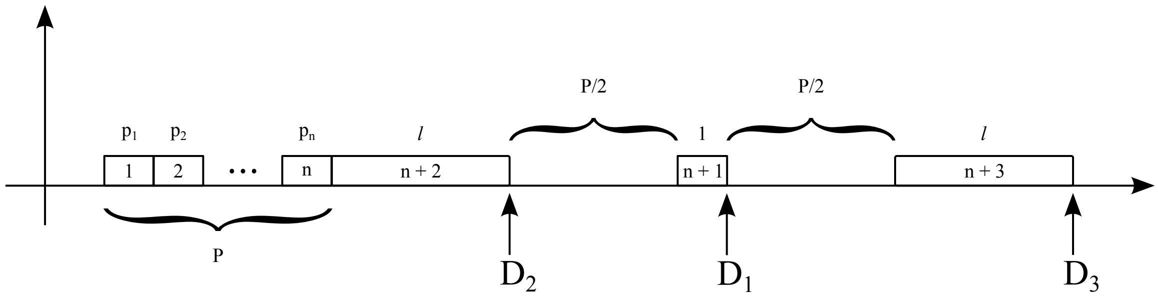

Fig. 12. Schedule $\mathcal{S}'$.

D_1 (as jobs $1, \ldots, n$). Job $n+2$ has processing time $p_{n+2} = l$ and weight w, defined as follows:

$$l := n^{\frac{1}{\alpha}}(1+P), \quad w := n(2P+l+1)^{\alpha} + 1$$

and due date D_2. Job $n+3$ has processing time $p_{n+3} = l$ and weight w (just like job $n+1$), but due date D_3. Parameters D_1, D_2, D_3 are defined next.

- All the jobs will have one of the following three due dates:

$$D_2 := P + l, \quad D_1 := D_2 + \frac{P}{2} + 1, \quad D_3 := D_1 + \frac{P}{2} + l.$$

- Every job j has $r_j = \frac{p_j}{2}$.

First we show that any optimal schedule for this instance must schedule jobs $n+1, n+2, n+3$ so that they finish exactly on their due date. Indeed, suppose that there is an optimal schedule $\mathcal{S}$ such that at least one of these three jobs finishes earlier or later than its due date by at least 1 time unit. Then let $\mathcal{S}'$ be a schedule where these three jobs are neither early nor late, and the rest n jobs are scheduled right before job $n+2$ in any order (say in order $1, 2, 3, \ldots, n$). So $\mathcal{S}'$ is as shown in Figure 12. Going from schedule $\mathcal{S}$ to schedule $\mathcal{S}'$, the following will happen.

- Due to jobs $n+1, n+2, n+3$, the cost decreases by at least w: Assume that job $n+1$ doesn't finish on its due date (the other cases are similar, or even better for our argument in case more than one of jobs $n+1, n+2, n+3$ don't finish on their due date). Then the cost decrease in $\mathcal{S}'$ due to the special jobs is at least

$$w(1 + \frac{p_{n+1}}{2})^{\alpha} - w(\frac{p_{n+1}}{2})^{\alpha} \geq w.$$

- Due to jobs $1, 2, \ldots, n$, the cost increases by at most $\sum_{j=1}^{n}(3P/2 + l + 1 + \frac{p_j}{2})^{\alpha} \leq n \cdot (2P + l + 1)^{\alpha}$, since we are going from earliness/tardiness penalty of at least 0 in $\mathcal{S}$ (actually $\frac{p_j}{2}$ but 0 suffices for our purposes) to at most $(2P + l + 1)^\alpha$ in $\mathcal{S}'$ for each such job.

Therefore

$$cost(\mathcal{S}') - cost(\mathcal{S}) \leq n \cdot (2P + l + 1)^\alpha - w < 0,$$

which contradicts the optimality of $\mathcal{S}$. Hence, in any optimal schedule jobs $n + 1, n + 2, n + 3$ finish exactly at times D_1, D_2, D_3 respectively. Next, we look at the optimal cost difference between the cases of the existence of a partition and the non-existence of such a partition.

- If a partition exists, then there is a schedule that doesn't schedule any of jobs $1, 2, \ldots, n$ before or after jobs $n + 2$ and $n + 3$. In any such schedule, each of jobs $1, 2, \ldots, n$, say job j, incurs a cost of at most $(1 + P/2 + \frac{p_j}{2})^\alpha \leq (1 + P)^\alpha$, for a total cost of the optimal $c_{YES} < n(1 + P)^\alpha$ (note that $n \geq 2$, so not all processing times can be P).
- If a partition doesn't exist, at least one of jobs $1, 2, \ldots, n$, say job j, must be scheduled before job $n + 2$ or after job $n + 3$. Therefore, the cost of the optimal schedule will have a cost $c_{NO} \geq (l + P/2 + \frac{p_j}{2})^\alpha \geq l^\alpha = n(1 + P)^\alpha$.

By our choice of l, we have $c_{YES} < n(1 + P)^\alpha \leq c_{NO}$, and the decision problem "Is there a partition?" has the same answer as "Is there a schedule with cost smaller than $n(l + P)^\alpha$?"

■

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] A. Farbod and T. Todd, "Resource Allocation and Outage Control for Solar-Powered WLAN Mesh Networks," *IEEE Transactions on Mobile Computing*, vol. 6, no. 8, pp. 960–970, 2006.

[4] G. H. Badawy, A. A. Sayegh, and T. D. Todd, "Energy Provisioning in Solar-Powered Wireless Mesh Networks," *IEEE Transactions on Vehicular Technology*, vol. 59, no. 8, October 2010.

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

[6] 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.

[7] 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.

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

[9] 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.

[10] 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.

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

[12] M. F. Jhang and W. Liao, "On Cooperative and Opportunistic Channel Access for Vehicle to Roadside (V2R) Communications," *GLOBECOM*, Dec. 2008.

[13] 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.

[14] A. Nandan, S. Das, G. Pau, M. Gerla, and M. Sanadidi, "Co-operative Downloading in Vehicular Ad-Hoc Wireless Networks," *WONS*, Jan. 2005.

[15] 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.

[16] 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.

[17] 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*, 2007.

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

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

[20] D. Helbing, “Traffic and Related Self-driven Many-particle Systems,” *Rev. Mod. Phys.*, vol. 73, pp. 1067–1141, Dec 2001. [Online]. Available: http://link.aps.org/doi/10.1103/RevModPhys.73.1067

[21] Y. Zhang, J. Zhao, and G. Cao, “Service Scheduling of Vehicle-Roadside Data Access,” *Mobile Networks and Applications*, vol. 15, no. 1, pp. 83–96, 2010.

[22] M. Garey and D. Johnson, *Computers and Intractability; A Guide to the Theory of NP-Completeness*. New York, NY, USA: W. H. Freeman & Co., 1990.

[23] R. Graham, E. Lawler, J. Lenstra, and A. Kan, “Optimization and Approximation in Deterministic Sequencing and Scheduling: A Survey,” *Discrete Optimization: Proceedings*, vol. 1, p. 287, 1979.

[24] T. Rappaport, *Wireless Communications : Principles and Practice*. Upper Saddle River, N.J. : Prentice Hall PTR, 1996.

[25] 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.

[26] 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.

[27] 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.

[28] J. Harri, F. Filali, and C. Bonnet, “Mobility Models for Vehicular Ad Hoc Networks: A Survey and Taxonomy,” *Communications Surveys & Tutorials, IEEE*, vol. 11, no. 4, pp. 19–41, 2009.

[29] F. Martinez, C. Toh, J. Cano, C. Calafate, and P. Manzoni, “A Survey and Comparative Study of Simulators for Vehicular Ad Hoc Networks (VANETs),” *Wireless Communications and Mobile Computing*, vol. 11, no. 7, pp. 813–828, 2011.

[30] 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.

[31] J. Gozalvez, M. Sepulcre, and R. Bauza, “Impact of the Radio Channel Modelling on the Performance of VANET Communication Protocols,” *Telecommunication Systems*, 2010.

[32] T. Rappaport and S. B. O. (Firme), *Wireless Communications: Principles and Practice*. Prentice Hall PTR New Jersey, 1996, vol. 2.