The Effect of Vehicle Route Uncertainty in Green Roadside Communication

Naby Nikookaran and Terence D. Todd
Department of Electrical and Computer Engineering
McMaster University
Hamilton, Ontario, CANADA
Email: {nikookn,todd}@mcmaster.ca

Shiqiang Zhang and George Karakostas
Department of Computing and Software
McMaster University
Hamilton, Ontario, CANADA
Email: {zhans51,karakos}@mcmaster.ca

***Abstract*—This paper addresses the problem of scheduling transmission requests in vehicular networks so that long-term road side unit (RSU) energy costs are minimized. We demonstrate that knowledge of vehicular routes greatly improves the energy service costs and request drop ratio of RSU transmission. At the same time, simple and fast prediction algorithms can recover a significant portion of the loss incurred by a lack of vehicle route knowledge. The proposed algorithms use recent historical traffic data and simple calculations, such as Bayesian estimates, to predict the next few routing decisions by a vehicle, in order to load balance the scheduling of its requests over the RSU network. Our simulation results show that, while the common assumption in the literature of knowing the vehicle routes is indeed crucial for achieving good performance, simple algorithms can be used in cases where vehicle routes are not known ahead of time, in order to achieve comparable costs and loss ratios.**

***Index Terms*—vehicular networks, energy efficiency, scheduling, unknown vehicle routes, route prediction.**

I. Introduction

A key component of future vehicular systems are roadside units (RSUs) that provide a local bridge between fixed infrastructure and the communicating vehicles. Studies have shown that in large scale deployments, a significant fraction of the RSUs must be solar powered, due to the unavailability of power grid connectivity. For this reason, energy efficiency at the RSUs is an important consideration and must be taken into account during packet scheduling. This is the focus of this paper.

In many applications, competing vehicular job requests may span multiple RSU coverage areas as vehicles travel through the coverage region [1]. When vehicle routes are known by the scheduler [2], [3], this information can be incorporated into the scheduling in a straightforward manner. This may eventually be possible once self-driving vehicle technology becomes widespread, since route information may be communicated directly to the roadside infrastructure [4]. However, when vehicle routes are unknown, which is currently the case, the scheduling problem becomes significantly more complicated.

In this paper, we consider the problem of roadside unit job scheduling when vehicle routes are unknown. The scheduler is given the topology of an urban road network and historical traffic traces that are used to extract vehicular motion statistics. Given this information, the scheduler must process online vehicular job requests that are subject to hard deadline constraints. The objective is to perform this scheduling such that the long-term energy service costs of the RSUs are minimized. Two schedulers are introduced, and referred to as the Route Coverage Expectation Scheduler (RCES) and the Bayesian Route Predictor (BRP). RCES and BRP use historical traffic trace inputs to estimate vehicular motion and the associated energy communication costs. BRP uses this information to first predict vehicle routes and then schedules the job requests across RSUs located on the predicted route. RCES uses this information to schedule job requests across multiple RSUs whenever possible.

A wide variety of results are presented that show the performance of the proposed schedulers. In particular, we compare RCES and BRP to optimal offline scheduling, where routes are assumed to be known in advance. We also compare them with a simple greedy online scheduler, which also knows vehicle routes at the time of scheduling. This algorithm greedily assigns each job request to the energy-minimum time-slots among the RSUs with available capacity that the vehicle will encounter. RCES and BRP are also compared to the scheduler in [5], which assigns requests to the RSUs on the current street segment using an earlier-deadline-first (EDF) scheduling policy. The results show that deploying RCES and BRP when vehicle routes are unknown achieves a drop ratio similar to the drop ratio achieved when these routes are known, with only a modest increase in energy cost.

II. Related Work

Previous work has considered the problem of roadside unit job scheduling when vehicle routes are unknown. Unfortunately, much of this work is not helpful in our case, since it focuses on improved network capacity rather than on RSU energy savings with hard packet deadline constraints. The most relevant references are briefly discussed in what follows.

Reference [6] introduces a motion prediction-based scheduling scheme in which RSUs cooperatively balance their loads by transferring part of their requests to nearby RSUs. If a request in the current queue cannot be served before its deadline, it is transferred to the subsequent RSUs in decreasing order of its probability to encounter. Reference [5] extends the model introduced in [6], referred to as the cooperative load balancing (CLB) scheduler. It prioritizes the current street RSUs before considering those in the next encountered intersection. Reference [3] modifies the CLB scheduler in [5] by changing the assumption to the case that the vehicle routes are known at the time of scheduling. It can therefore achieve better load balancing than the CLB scheduler, but at each request transfer, the scheduler also transfers the request to the RSU at the next intersection, in case the vehicle deviates from its route. Note that the above studies do not focus on reducing RSU energy expenditure, as is the case in our paper.

To the best of our knowledge, our paper is the first that considers roadside unit schedulers that operate with unknown routes and with the objective of minimizing long-term energy operating costs under job deadlines and with small packet loss.

III. System Model

We study the scheduling of transmission requests made to a network $\mathcal{N}$ of RSUs by a set $\mathcal{V}$ of vehicles where their routes are *unknown*. Nevertheless, the scheduler can use the network topology and historical data (e.g., past traffic traces) to deduce vehicle mobility patterns, such as turning probabilities of vehicles at intersections, average travel times of streets, etc.

It is assumed that the RSUs use downlink power control when communicating with the vehicles, i.e., they adapt their transmit power in order to maintain a constant bit rate [1], [7]. Each vehicle $v \in \mathcal{V}$ generates a set $\mathcal{R}_v$ of requests, for a total of $\mathcal{R} = \cup_{v\in\mathcal{V}}\mathcal{R}_v$ requests. Each one of these requests has a release date, i.e., the time when the request is generated, and a due date, i.e., the deadline of the associated RSU response. These dates define a set of time slots during which the request can be served; if it is not, the request is *dropped*. We make the typical quality-of-service assumption, that a vehicle generating a request, communicates its size, release, and due dates to the first RSU it encounters, and, therefore, the system is aware of these parameters for scheduling purposes. Moreover, we assume that a request with a size of l time slots is split into l unit-size requests that can be scheduled on different RSUs. Note that we assume *non-preemptive* scheduling of requests, i.e., the assignment of a request to a specific RSU during a certain time slot cannot be changed or preempted once it is made.

Let $\mathcal{T}$ be the set of time slots; within a time slot, RSU n has the capacity to transmit to at most u_n vehicles, and a vehicle can communicate with at most one RSU during a time slot. To schedule the requests, we use decision variables X_{ntr}, so that $X_{ntr} = 1$ if RSU n serves request r of vehicle v during time slot t, and $X_{ntr} = 0$ otherwise. When vehicle v is within the coverage area of RSU n during time-slot t, the energy cost for servicing request r (referred to as its *energy service cost*) is c_{ntr}, which depends on the RSU-vehicle distance (and other propagation effects) in time slot t. This is done by first computing the transmit power needed to overcome the path loss from RSU n to request r at time t, such that a target SNR is achieved that supports the chosen data rate. This power is added to the quiescent radio power consumption, and the total energy is computed by multiplying by the time slot duration [7]. In order to enforce the servicing of all requests, if possible, we use a decision variable Z_r for each request r, so that $Z_r = 1$ if r is dropped, incurring a large cost D_r. As a result, in the optimization defined below, the scheduler will never drop a request, unless there is a capacity constraint violation. Since this part of the objective function is an artifice to ensure service, it will not be included in the energy service cost we present in the results obtained.

Given the above assumptions and for a given input traffic trace, we formulate the optimum cost as an integer linear program. This provides a lower bound on the energy service cost. Since the scheduler is given non-causal knowledge of future vehicular inputs, this bound is not generally achievable in practice. The optimization is given as follows and discussed below.

$$\min_{X,Z} \sum_{n\in\mathcal{N}}\sum_{t\in\mathcal{T}}\sum_{r\in\mathcal{R}} c_{ntr}X_{ntr} + \sum_{r\in\mathcal{R}} D_r Z_r \quad \text{s.t.} \tag{ILP}$$

$$Z_r + \sum_{n\in\mathcal{N}}\sum_{t\in\mathcal{T}} X_{ntr} = 1 \qquad \forall r \in \mathcal{R} \tag{1}$$

$$\sum_{r\in\mathcal{R}} X_{ntr} \le u_n \qquad \forall n \in \mathcal{N}, t \in \mathcal{T} \tag{2}$$

$$\sum_{n\in\mathcal{N}}\sum_{r\in\mathcal{R}_v} X_{ntr} \le 1 \qquad \forall t \in \mathcal{T}, v \in \mathcal{V} \tag{3}$$

$$X_{ntr}, Z_r \in \{0,1\} \qquad \forall n \in \mathcal{N}, t \in \mathcal{T}, r \in \mathcal{R} \tag{4}$$

Constraint (1) guarantees that requests are satisfied. Note that the requests of size of more than one are split into multiple requests of unit-size that can be served by different RSUs. Constraint (2) enforces the capacity constraint for the RSUs and constraint (3) implies that only one request of vehicle v can be serviced during time slot t.

IV. The Route Coverage Expectation Scheduler (RCES) and the Bayesian Route Predictor (BRP)

The two algorithms we propose are probably the simplest ones, in terms of the computations needed for their route predictions. By studying them, we demonstrate that in settings where their deployment is practically desirable or unavoidable, more sophisticated prediction methods (e.g., using Markov chains as in [8]) are bound to do even better.

Our first proposed algorithm RCES (Algorithm 1) takes as its input a road network with RSUs installed, vehicular traffic statistics (mean traveling time of streets and turning probabilities at intersections) based on historical traffic data, and a communication cost matrix. New vehicular job requests that need to be scheduled arrive in an on-line fashion. Algorithm RCES greedily finds the time-slots of minimum communication cost to serve requests, trying to minimize the total energy cost of scheduling. Note that the algorithm does not make any assumptions about the vehicle routes. RCES schedules part of a request on the RSU which currently covers the vehicle's current location (if such an RSU exists), and leaves the rest to be serviced by RSUs that the vehicle may encounter in the future. Any portion of the first part that could not be scheduled before the vehicle leaves an RSU coverage area, will be postponed along with the second part. The decision on how requests should be divided is based on the expected value of free capacity the vehicle will encounter in the next $\xi_{\max}$ streets or before its request expires (whichever happens first), where $\xi_{\max}$ is a preset algorithm parameter upper-bounding its lookahead. The expectation is calculated over all possible routes within these limits, using the probabilities given as an input. In our experiments we set $\xi_{\max} = 3$.

More specifically, set $\mathcal{R}^t$ contains the requests released during time-slot t, together with all other requests that were previously postponed (line 2). If any request expires before it can be served, it will be dropped, otherwise the request will be added to $\mathcal{R}^t$ when the vehicle announces its arrival to an RSU coverage area. In lines 4-12, RCES goes through all requests in $\mathcal{R}^t$ and either schedules them or postpones them. We use an earlier-deadline-first policy to pick all requests in $\mathcal{R}^t$ with the same earliest deadline that were submitted to the same RSU by the same vehicle (line 5). In line 6, γ_n is the overall available capacity of the RSU currently covering the vehicle, while in line 7, Γ_i is the expected overall available capacity in route i following the current RSU. By 'overall available capacity' we mean the total number of time slots of capacity available for the request we are currently serving (for example, if the vehicle will be in an RSU's coverage area during time slots 3, 4, 5, and 6, and during these time slots the RSU has available capacity 2, then the overall available capacity in this RSU is 8). In line 8, we calculate the ratio $ratio_n$ by which we partition the set $\mathcal{R}_d^t$ into two parts, $\mathcal{P}_d$ and $\mathcal{R}_d^t \backslash \mathcal{P}_d$. We postpone the requests in $\mathcal{R}_d^t \backslash \mathcal{P}_d$, and schedule the ones in $\mathcal{P}_d$, if possible (lines 10-11). Ratio $ratio_n$ (line 8) is the ratio of the overall capacity of the current RSU γ_n over the expected overall capacity in all possible routes a vehicle may follow, including the current RSU. To calculate the denominator, we enumerate all possible routes of length at most $\xi_{\max}$ or the request deadline (whichever comes first), and use the given traffic statistics. The intuition behind this ratio, is the obvious motivation of assigning to an RSU a portion of total remaining requests that is proportional to the available capacity of this RSU over the expected available capacity the vehicle is (or is going to be) encountering.

Our second algorithm BRP (Algorithm 2) uses a simple Bayesian estimator to estimate the most probable route for a vehicle, and schedules its requests accordingly. The Bayesian estimator consists of a route predictor and a travel time estimator. Using a vehicle's immediate traffic history, the route predictor calculates the most probable street that it will encounter next. More specifically, given the sequence $S = (s_1, s_2, \ldots, s_{i-1})$ of streets already traversed by the vehicle (with s_{i-1} being the latest and s_1 the $(i-1)$-th previous one), the next street s_i is chosen iff $i = \arg\max_k p(s_k|S) = \arg\max_k \frac{p(S|s_k)p(s_k)}{p(S)}$. The predictor "predicts" only the next few (typically 2 or 3) streets of a vehicle's future route. In order to estimate the street travel times, we employ a travel time estimator, which uses historical traffic data, such as average travel time of different parts of streets.

Algorithm 1 Route Coverage Expectation Scheduler (RCES)

Input:
- Deployed RSUs $\mathcal{N}$, and their capacities $\mathcal{U}$
- Incoming requests $\mathcal{R}$, time slots $\mathcal{T}$, and communication costs for all requests, RSUs and time-slots
- City graph G, street traveling times, intersection turning probability matrix P

Output: Online schedule of all non-dropped requests

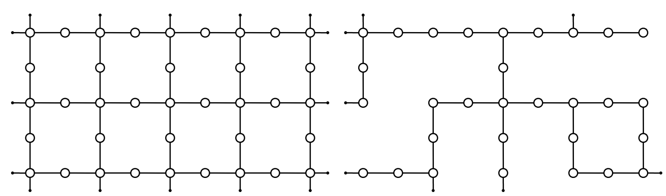
1: **for all** $t \in \mathcal{T}$ **do**
2: $\quad \mathcal{R}^t$ = set of postponed or released at time t requests
3: $\quad$ Drop from $\mathcal{R}^t$ requests whose deadlines cannot be met
4: $\quad$ **while** $\mathcal{R}^t \neq \emptyset$ **do**
5: $\quad\quad \mathcal{R}_d^t := \{$same vehicle, RSU requests in $\mathcal{R}^t$ with smallest deadline$\}$
6: $\quad\quad \gamma_n :=$ overall available capacity in RSU n
7: $\quad\quad \Gamma_i :=$ expected overall available capacity in route i, $\forall$ routes i
8: $\quad\quad ratio_n := \frac{\gamma_n}{\gamma_n + \sum_{\text{route } i} \Gamma_i}$
9: $\quad\quad \mathcal{P}_d :=$ first $(ratio_n \cdot |\mathcal{R}_d^t|)$ requests in $\mathcal{R}_d^t$
10: $\quad\quad$ Schedule requests in $\mathcal{P}_d$ in current RSU
11: $\quad\quad$ Postpone requests in $\mathcal{R}_d^t \backslash \mathcal{P}_d$
12: $\quad$ **end while**
13: **end for**

Algorithm 2 Bayesian Route Predictor (BRP) Scheduler

Input:
Same as Algorithm 1, plus street travel time estimates

Output: Online schedule of all non-dropped requests

1: Lines 1-6 from Algorithm 1
2: $\Gamma_{\max}$:= overall available capacity in the route that is more likely to be chosen by the vehicle
3: $ratio_n := \frac{\gamma_n}{\gamma_n + \Gamma_{\max}}$
4: Lines 9-13 from Algorithm 1



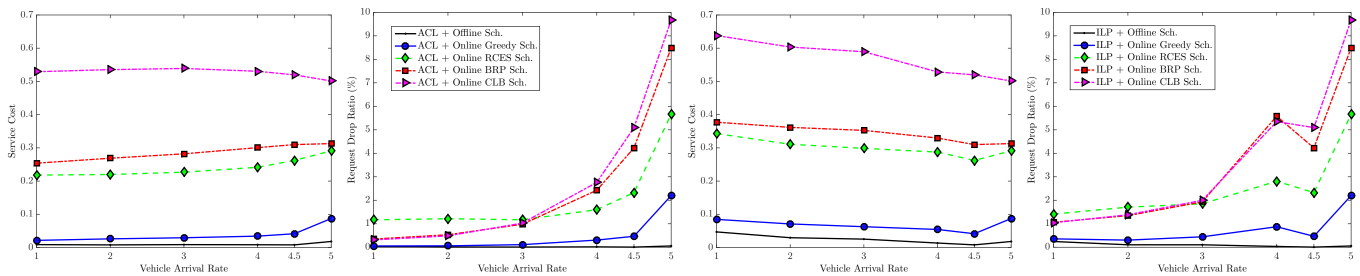
(a) Full Grid (b) Reduced Grid

Fig. 1. Grid Road Networks with RSU Candidate Locations.

V. Performance Results

In order to compare the performance of scheduling algorithms when the vehicle routes are known against the performance of such algorithms when the routes are unknown, we perform simulations on a variety of topologies.

More specifically, for the case of *known* vehicle routes, we consider the greedy online scheduler which assigns each job request to the minimum energy RSU time slots remaining that meet the request deadline, as well as the solution of (ILP), which is a lower bound for *any* causal online scheduling algorithm, since it is fractional, offline, and non-causal. This group of schedulers is compared with online schedulers that are *unaware* of vehicle routes, namely, our RCES and BRP algorithms of Section IV, and the algorithm of [5], which assigns requests to the current RSU and transfers unassigned



(a) ACL Service Cost (b) ACL Drop Ratio (c) ILP Service Cost (d) ILP Drop Ratio

Fig. 2. The Performance of the Scheduling Algorithms at the Full Grid Road Network.

requests only if they cannot be served at the current RSU using earlier-deadline-first, using a time window of 5 time slots, $\beta = 0.25$ for their exponential weighted moving average, and modified to fit our non-preemptive setting. We will refer to the latter algorithm as *Cooperative Load Balancing (CLB)*.

The two groups of algorithms are compared on three different road network topologies. The first is the Manhattan grid of bidirectional streets shown in Figure 1a. The second street from the top and the third street from the left are 5-lanes with 60 km/h speed limits. The rest are 4-lanes with a 50 km/h speed limit. All intersections are controlled by traffic lights. The smallest block has a 1 km^2 area, which gives a total region of 11.25 km^2. Figure 1 shows the locations where RSUs can be placed for the two topologies. All deployed RSUs have a capacity of 4 downlink radio channels and a coverage range of 250 m on each side. The second topology is the network in Figure 1b, produced from the full grid by removing streets (and leaving the rest as before). The last topology is a 10-km bidirectional highway with multiple entrances and exits and no traffic lights, 5 lanes, and a 60 km/h speed limit. The RSU candidate locations are uniformly placed along the highway, and the distance between two consecutive locations is twice the RSU coverage range.

The performance evaluation is done using 10 30-min vehicular traffic traces as input. The first trace is used as (known) historical data used by the algorithms for their estimates, such as turning probabilities at intersections and mean street traveling times. The remaining nine traces are used to evaluate the performance of the algorithms. In each trace, traffic is generated by using SUMO [9], with each vehicle following the shortest path between its uniformly at random picked origin and destination. Vehicular arrivals are generated by a Poisson process. Each vehicle generates transmission requests according to a Poisson process, with mean arrival rate uniformly selected between 0.01 and 0.02 per time slot, deadlines chosen uniformly at random between 80 to 160 time slots from their release time, and sizes generated by an exponential distribution, with mean value selected uniformly between 2 and 4 time slots.

A distance dependent exponential path-loss model with log-normal shadowing [10] is used to determine the transmit power needed over a given link. The transmission power between a transmitter and a receiver, $P_{t,r}$, can be expressed by $P_{t,r} = P_{t,0} P_{sh} \left(\frac{d_{t,r}}{d_{t,0}}\right)^{\alpha}$, where $d_{t,0}$ is the reference distance, $P_{t,0}$ is the reference power at the reference distance, P_{sh} is a random variable that models the shadowing effect of the channel, α is the path loss exponent, and $d_{t,r}$ is the distance between the transmitter and the receiver. The shadowing effect of the radio channel can be modeled as a random variable with log-normal distribution which has a zero mean (in dB) and a standard deviation of $\sigma_{\text{dB}} = 4$. This model is used to compute the energy service cost of each downlink transmission [1], [7].

The algorithms are evaluated with two different RSU placements: The All Candidate Locations (ACL) placement places an RSU in all candidate locations of the network, while the ILP placement is done according to the offline Integer Linear Program of [11]. In all experiments, we run the optimal offline scheduler, the online greedy scheduler with known routes, and algorithms RCES, BRP and CLB.

Figure 2 shows results for the full Manhattan grid road network. Figures 2a and 2c show the energy service cost and Figures 2b and 2d show the request drop ratio of ACL and ILP, respectively. In these results, we increase the vehicle arrival rate from 1 to 5 vehicles per time slot. There are 37 RSU candidate locations. ACL places RSUs at all these locations, while ILP places 22, 26, 29, 33, 37, and 37 RSUs for vehicle arrival rates of 1, 2, 3, 4, 4.5, and 5, respectively.

As shown in Figures 2b and 2d, increasing the vehicle arrival rate increases the request drop ratio for all algorithms, as expected. As mentioned above, the offline scheduler provides a lower bound for the drop ratio (as well as the energy service cost) of any online scheduler, but the greedy scheduler is almost as good. However, algorithms RCES, BRP, and CLB (which do not have vehicle route knowledge) have higher drop ratios than the greedy algorithm (or the bound), due to lack of capacity. Nevertheless, RCES is seen to have a lower drop ratio than BRP and CLB for vehicle arrival rates over about 3. This is not surprising for CLB, since it tries to have the current RSU fulfill as many requests as it can locally, therefore capacity may be depleted for subsequent arrivals. Note that BRP performs similarly to CLB, due to its similarly greedy tactic of relying only on the route with the highest probability, ignoring the other route choices.

Figures 2a and 2c for energy service cost, show that there are large differences between the greedy scheduler and those without vehicle route knowledge, i.e., RCES, BRP, and CLB. At the higher arrival rates, for example, the energy service cost is up to about 500% higher for the CLB algorithm. The use of traffic statistics by RCES and BRP, however, leads to a much lower service cost in comparison to CLB, i.e., they are about 50% of that for CLB. This illustrates that while route uncertainly significantly increases the service cost, the use of historical statistics greatly improves the energy performance. Note that the service cost differences slightly decrease with increases in vehicle arrival rate due to the increase in the drop ratio.

Finally, the comparison between ACL and ILP results in Figure 2 shows a slight difference between the energy service costs in Figures 2a and 2c, since ACL opens all RSUs. This provides more flexibility for the scheduler to meet the request deadlines.

Similar results are shown for the reduced grid road network in Figure 3. In these experiments, we vary the vehicle arrival rate over a smaller range (0.5 to 2) due to the reduced RSU capacity of the network. In this case there are 30 RSU candidate locations, all used by ACL, while ILP opens only 20, 21, 23, 29, and 25 RSUs for vehicle arrival rate of 0.5, 1, 1.5, 1.75, and 2, respectively.

Figure 3 shows that the relative behavior for the different algorithms is similar to that previously observed, i.e., RCES and BRP both outperform CLB in terms of energy service cost. However, it can be seen that the differences in energy service cost performance have decreased with this topology, i.e., the energy service cost range is less than about a factor of 3 under heavy vehicle arrival rate. This is attributed to the fact that in the reduced grid topology, route uncertainty is smaller than that in the full grid. This tends to somewhat reduce the value of vehicular route knowledge. Note that the drop ratio of the RCES and BRP algorithms is higher than that of CLB. This is due to the traffic congestion on some of the roads, which degrades the travel time estimation accuracy of RCES and BRP algorithms.

Figure 4 shows the results of ACL and ILP placement for the highway network topology. There are 20 RSU candidate locations that are used by ACL, while ILP opens 14, 17, 20, 20, and 20 RSUs for vehicle arrival rates of 1, 2, 3, 4, and 5, respectively.

As shown in Figure 4, RCES and BRP both outperform the CLB algorithm in both energy service cost and drop ratio. The drop ratio of RCES, BRP, and the greedy algorithm is close to the lower bound (see Figures 4b and 4d). However, there is a large difference between these algorithms and CLB. The reason for this is that CLB transfers requests to RSUs that should be among those before the next vehicle turning point or exit. Since there is a possibility of a vehicle leaving the highway at the next exit, the CLB algorithm does not transfer a vehicle's requests to the RSUs beyond the next exit. On other hand, RCES and BRP consider those RSUs.

As one would expect, the CLB algorithm has a higher energy service cost than the other schedulers (see Figures 4a and 4c). However, what is surprising here is that CLB has a relatively higher service cost than RCES and BRP despite its higher drop ratio. This is due the its policy to schedule as many requests as possible at the current RSU and to transfer requests only when there is no remaining capacity. This often leads to the higher service costs.

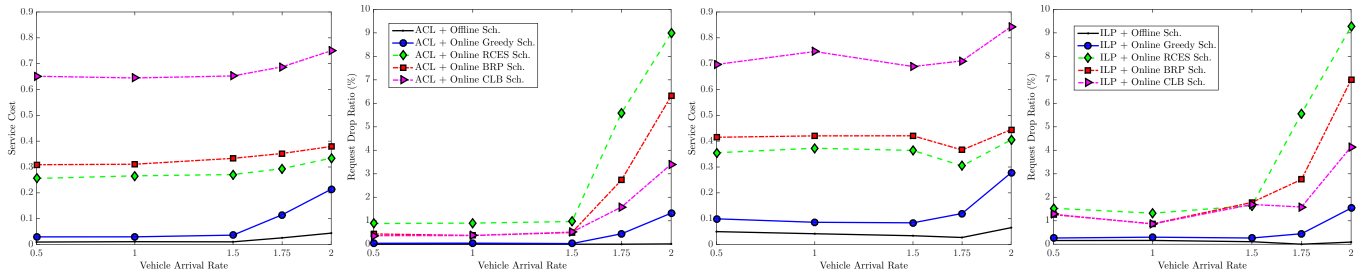
As in the previous results, the ILP placement leads to a higher energy service cost and drop ratio than ACL placement in Figure 4 for the vehicle arrival rates of less than about 3, since ILP opens fewer RSUs. However, when the vehicular load reaches the point that requires all RSUs to be opened (i.e., vehicle arrival rate above about 3), we see a slowly growing trend in both the service cost and the drop ratio. The greedy scheduler that knows the routes has a slightly larger drop ratio than RCES for vehicle arrival rates of less than about 3. This is due to the objective of the former algorithm, which is the minimization of service cost without regard to how many requests are dropped.

We show that the trends observed in the previous experiments persist in the more general case of knowing a fraction of vehicle routes, with an obvious direct dependence on this fraction. More specifically, we consider knowing 0%, 30%, 50%, 70%, and 100% of all routes. If a vehicle submits a route, the request is scheduled using the online greedy scheduler; otherwise, RCES and BRP schedulers are used to schedule the request. Since RCES and BRP schedulers have a better performance than CLB, in this section, we do not include the CLB algorithm in the presented results. In the interests of brevity, we only consider the reduced grid topology from Figure 1b with the ILP placement of RSUs.

Figures 5a, and 5b show the energy service costs and drop ratios for RCES, and Figures 5c, 5d are the corresponding curves for BRP. As seen in Figure 5, the RSU energy service cost and the drop ratio both decrease as more information about vehicle routes becomes available to the proposed schedulers. These results are a strong indication that the proposed algorithms can accommodate varying levels of route knowledge, and can significantly improve RSU energy and loss rates when this information is available. Under moderate loading for example, the service costs drop by about 50% as the route knowledge goes from zero to 70%. The improvements are roughly the same for both ILP and ACL RSU placements.

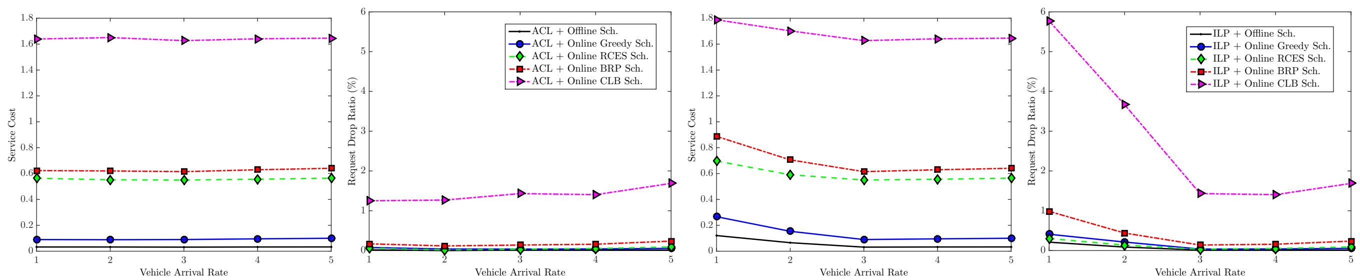
VI. CONCLUSIONS

The purpose of this paper is to demonstrate that, for the general setting of Section III, (i) prior knowledge of vehicular routes greatly improves the energy service cost and drop ratio of RSU transmission, and (ii) simple and fast prediction algorithms like RCES and BRP (Section IV) can recover a significant portion of the loss incurred by our lack of vehicle routing knowledge. The simulations of Section V show that, while the common assumption in the literature of knowing the vehicle routes is indeed crucial for achieving good performance, simple algorithms can be used in cases where vehicle routes cannot be known ahead of time, in order to achieve comparable costs and loss ratios. Future research can



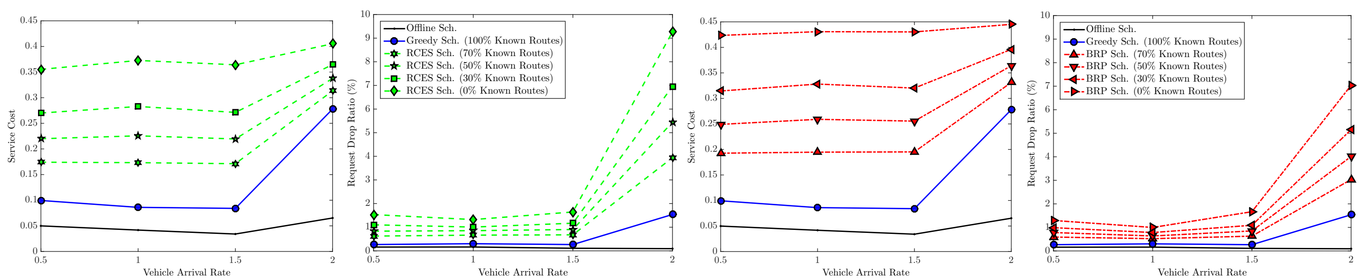
(a) ACL Service Cost (b) ACL Drop Ratio (c) ILP Service Cost (d) ILP Drop Ratio

Fig. 3. The Performance of the Scheduling Algorithms at the Reduced Grid Road Network.



(a) ACL Service Cost (b) ACL Drop Ratio (c) ILP Service Cost (d) ILP Drop Ratio

Fig. 4. The Performance of the Scheduling Algorithms at the Highway Road Network.



(a) (b) (c) (d)

Fig. 5. Scheduling Algorithm Performance with Known Route Fraction (Reduced Grid Road Network).

focus on more sophisticated (but still practical) prediction and RSU scheduling algorithms, which can render prior knowledge of vehicle routes unnecessary for good performance.

REFERENCES

[1] A. Khezrian, T. Todd, G. Karakostas, and M. Azimifar, "Energy-Efficient Scheduling in Green Vehicular Infrastructure With Multiple Roadside Units," *IEEE Trans. on Veh.Tech.*, vol. 64, no. 5, pp. 1942–1957, 2015.

[2] 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 *Proc. 5th ACM Int. Conf. on Ubiquitous Information Management and Communication*, 2011, pp. 1–8.

[3] G. N. Ali, P. H. J. Chong, S. K. Samantha, and E. Chan, "Efficient Data Dissemination in Cooperative Multi-RSU Vehicular Ad Hoc Networks (VANETs)," *J. of Systems and Software*, vol. 117, pp. 508 – 527, 2016.

[4] B. Paden, M. Čáp, S. Z. Yong, D. Yershov, and E. Frazzoli, "A Survey of Motion Planning and Control Techniques for Self-Driving Urban Vehicles," *IEEE Trans. on Intel. Vehicles*, vol. 1, no. 1, pp. 33–55, 2016.

[5] G. G. M. N. Ali, E. Chan, and W. Li, "On Scheduling Data Access with Cooperative Load Balancing in Vehicular Ad Hoc Networks (VANETs)," *The Journal of Supercomputing*, vol. 67, no. 2, pp. 438–468, 2014.

[6] Y. Gui and E. Chan, "Data Scheduling for Multi-item Requests in Vehicle-Roadside Data Access with Motion Prediction Based Workload Transfer," in *Proc. of 26th Int. Conf. on Advanced Information Networking and Applications(WAINA)*, 2012, pp. 569–574.

[7] A. Hammad, T. Todd, G. Karakostas, and D. Zhao, "Downlink traffic scheduling in green vehicular roadside infrastructure," *IEEE Trans. on Veh. Tech.*, vol. 62, no. 3, pp. 1289–1302, 2013.

[8] J. Krumm, "A Markov Model for Driver Turn Prediction," in *SAE Technical Paper*. SAE International, 04 2008.

[9] J. Song, Y. Wu, Z. Xu, and X. Lin, "Research on Car-Following Model Based on SUMO," in *Advanced Infocomm Technology (ICAIT), 2014 IEEE 7th International Conference on*, Nov 2014, pp. 47–55.

[10] T. S. Rappaport, *Wireless Communications: Principles and Practice*, 2nd ed. Prentice Hall, 2001.

[11] N. Nikookaran, G. Karakostas, and T. Todd, "Combining Capital and Operating Expenditure Costs in Vehicular Roadside Unit Placement," *IEEE Trans.on Veh. Tech.*, vol. 66, no. 8, pp. 7317–7331, 2017.