IEEE Open Journal of
Vehicular Technology

Received 10 January 2022; accepted 1 February 2022. Date of publication 10 February 2022;
date of current version 24 March 2022. The review of this article was coordinated by Editor Derrick Wing Kwan Ng.

Digital Object Identifier 10.1109/OJVT.2022.3150241

Mobility-Aware Edge Caching for Minimizing Latency in Vehicular Networks

YOUSEF ALNAGAR [1], RAMY H. GOHARY[1], SAMEH HOSNY[2], AND AMR A. EL-SHERIF [3] (Senior Member, IEEE)

[1]Department of Systems and Computer Engineering, Carleton University, Ottawa, ON K1S 5B6, Canada
[2]Department of Electrical and Computer Engineering, Nile University, Cairo 12588, Egypt
[3]School of Engineering, New Giza University, Giza Governorate 2388, Egypt

CORRESPONDING AUTHOR: YOUSEF ALNAGAR (e-mail: yousefalnagar@cmail.carleton.ca)

This work supported in part by an ENCQOR OCI Grant and in part by a CRD NSERC Grant.

ABSTRACT This work proposes novel proactive caching schemes for minimizing the communication latency in Vehicular *Ad Hoc* Networks (VANETs) under freeway and city mobility models. The main philosophy that underlies these schemes is to exploit information that may be available *a priori* for vehicles' demands and mobility patterns. We consider two paradigms: cooperative, wherein multiple Roadside Units (RSUs) collaborate to expedite the transfer of information to the intended user, and non-cooperative, wherein each RSU operates independently of other RSUs in the network. To develop the proposed schemes, for each of the considered models we formulate optimization problems that expose the impact of velocity and demand profile of the vehicle on the optimal caching decision. Unfortunately, the developed formulations are NP-hard, and hence difficult to solve for moderate-to-large problems. To circumvent this difficulty, we use the insight developed through these formulations to develop practical caching algorithms based on the knapsack problem and suboptimal relaxations. These algorithms are shown to yield close-to-optimal solutions at much lower computation costs than the corresponding exhaustive search. Our numerical investigations suggest that the proposed proactive caching schemes yield substantial gains over the traditional caching policy, and their respective reactive baselines with no caching. Furthermore, the cooperative schemes are significantly more advantageous than their non-cooperative counterparts.

INDEX TERMS Knapsack, optimization, proactive caching, VANETs.

I. INTRODUCTION

Vehicular *Ad-Hoc* Networks (VANETs) comprise a particular class of wireless networks in which the communicating nodes are moving vehicles. Relying on Vehicle-to-Vehicle (V2V) and Vehicle-to-Infrastructure (V2I) communications, VANETs are prospected to be a key enabler for a plethora of delay-sensitive industrial and civil applications, including intelligent transportation, emergency services, self-driving vehicles, and information sharing and entertainment services [1]. Such applications place stringent latency constraints, which may not be possible to meet in VANETs due to delays inherent in transferring data from backhaul servers. An effective approach to alleviate this difficulty is to invoke caching techniques, which are expected to enhance energy efficiency, data accessibility and spectrum utilization of the network [2].

In conventional caching, popular files are transferred to intermediate nodes within the proximity of prospective users prior to being requested. This approach reduces the delay of transferring files from remote servers and its effectiveness is determined by the probability of cache hits, i.e., the probability that a cached file is indeed requested by the users.

To enhance the probability of cache hits, a class of proactive caching schemes has been introduced [3]. In contrast with its conventional counterpart, which tends to depend on collective demand patterns, proactive caching relies on identifying demand patterns of individual users. In particular, in proactive caching the service provider predicts the users' demand on user-by-user basis and caches information proactively either to the base station (BS) to which the user is connected or directly to the user's terminal. To achieve this

goal, proactive caching takes advantage of the information available about individual users, which may include the user's contexts of interest, and temporal and spatial mobility patterns. Acquiring these patterns requires the network to track, learn and build mobility and demand profiles of individual users. Incorporating such patterns in the design of caching policies yields substantial quality of service (QoS) gains in emerging 5G VANETs [4], [5]. In line with this vision, our goal in this paper is to develop proactive caching methodologies that harness the information available about individual users to minimize latency under various mobility and traffic models.

A. RELATED WORK

The use of caching techniques have been investigated in the context of cellular communications under various mobility conditions [6]–[9]. For instance, the case in which the users and the BSs are fixed was considered in [6]. Therein it was shown that optimizing the files to be cached to minimize overall latency constitutes an NP-complete problem. To alleviate this difficulty a computationally-efficient greedy algorithm was developed in [6]. The solution yielded by this algorithm is shown to be within a constant factor from the optimal one.

The case in which users migrate between cells was considered in [7]–[9] under various migration models. In these papers, the network coordinator uses prior information about the location of individual users to develop a caching technique that accounts for the changes in the network topology induced by users' migration. Unfortunately, the models in [7]–[9] do not make use of the information available on the user's velocities and travelling directions nor do they consider the constraints imposed by the particular roads travelled by the users. Incorporating this information can significantly enhance the communication effectiveness of VANETs.

Caching in VANETs depends on a combination of V2V and V2I communications. In V2V, caching depends on the cooperation between neighbouring vehicles [10], whereas in V2I, caching is effected using roadside units (RSUs), which provide connectivity support to passing vehicles and act as gateways to serve vehicles requesting data files from the backhaul network. Caching effectiveness depends on the contact time between the vehicles and the RSUs, and the periodicity with which the cached items are refreshed [11], [12].

The use of RSUs in caching is known to have a significant impact on the delay performance of the network [13], [14]. For instance, using RSUs with large storage capacities to cache popular files has been proposed in [13] to tackle latency problems in VANETs. Further delay reductions were sought in [14] by introducing a split-content caching method. In this method, large data files are partitioned into smaller chunks that are distributed across multiple RSUs. Although this method increases the probability of cache hits, its implementation requires the backhaul network to have an area controller to manage caching placement across the RSUs. Subsequent development in content caching was introduced in [15]. Therein file caching is performed jointly across vehicles and RSUs. This technique increases the storage capacity of the network and offers the potential of making effective reduction in latency. However, this technique places stringent constraints on the vehicles, requiring them to coordinate their mobility and directions, which can be reasonable in freeway mobility models, but not in urban ones.

In addition to the abstract models in [13]–[15], other models that account for high mobility freeway and urban scenarios, bi-directional routes, traffic lights and building structures have been developed. For instance, the freeway model considered in [16] and [17] is commonly used to describe the mobility patterns in VANETs. In this model, the freeway is assumed to be in the the free-flow state and the traffic density is assumed to be low. In this case, the speeds of the vehicles can be assumed to be independent and identically distributed random variables which can be generated through appropriately truncated Gaussian distribution. Another model that is based on the actual traces obtained through the global positioning system (GPS) was developed in [18] and later enhanced in [19] to include stop signs and traffic lights. To capture building structures and bi-directional routes in urban environments, the so-called Manhattan mobility model was introduced in [19]. In this model, the city is organized in a grid of orthogonal streets along which the vehicles are allowed to travel. At intersections, the vehicles may turn in either direction or they may continue straight ahead with prescribed probabilities.

In forthcoming developments, we will consider the, commonly used, freeway and city traffic models [16]–[21]. Despite their simplicity, these models enable us to draw insight into the complexity of cache allocation and the impact of caching on latency. In fact, these model will result in NP-hard optimization problems, which are difficult to solve. Although more complicated models might be more accurate to describe practical traffic, such models are likely to result in even more cumbersome optimization problems, and may be difficult to extract insight from.

Although existing work incorporates mobility patterns in developing caching schemes [13]–[15], no work seems to incorporate the demand history and mobility patterns of individual users in these schemes. In particular, existing caching schemes do not exploit the information available about people's daily routines reported in the literature, e.g., commuting to work and visiting particular websites [22]–[24]. These routines render the behaviour of individuals rather predictable, and provides valuable side information to enhance the effectiveness of caching.

A methodology for identifying the mobility patterns of multiple users was developed in [22] and [23]. This methodology relies on collecting statistical information from the BSs along the route taken by each user. This information enables the network coordinator to build a profile to help it predict future routes of each user. Analogously, the history of the websites visited by individuals can be tracked to build a demand profile. Such an approach was considered in [24] to enhance a

recommendation system that predicts future users' demands. Our goal in this paper is to invoke the mobility patterns and demand prediction techniques in proactive caching to reduce communication latency in VANETs.

B. MOTIVATION AND CONTRIBUTIONS

Our motivation is to invoke the mobility patterns and demand profiles of mobile users to alleviate the stress on the communication resources. Freeing up resources not only improves the quality of experience of the end user, but also contributes to reducing cost, energy and bandwidth, allowing for critical delay-sensitive information to be reliably communicated. As such, the caching framework considered herein can be viewed as a means of trading, the typically abundant, memory for scarce spectral, temporal and energy resources.

The main contribution of this work lies in the development of a realistic mathematical model of the latency experienced by users in caching-assisted mobile networks, taking into consideration their actual parameters, including velocity, demand profiles and rate requirements. This is in contrast with the development in [13] and [14], which only consider caching of popular files without taking these parameters into account.

The developed framework characterizes the latency in the freeway model, and subsequently in the more involved Manhattan city model. This framework is then used to develop optimization algorithms to determine the caching policies that reduce the overall latency experienced by the users of the network. To the best of our knowledge, our framework is the first to study the impact of the users' velocities, demands and rate requirements on latency and the caching policies that minimize it. As a by-product, our framework provides performance assurances, whereby the maximum latency experienced by a user of the system is guaranteed to fall below prescribed thresholds. This aspect of making caching decisions has not been considered in related literature on caching in VANETs and RSUs [11], [12], [15].

Our main contributions are summarized as follows.

- We propose two novel caching paradigms: non-cooperative, wherein each RSU operates independently of other RSUs in the network, and cooperative, wherein multiple RSUs collaborate to serve the intended users. Such schemes have not been reported previously in the VANETs literature.
- For each of the proposed models, we formulate optimization problems to minimize the latency in the network under freeway and city mobility models.
- We show that special cases of the proposed formulations represent instances of the knapsack problem. Hence, we concluded that these formulations are NP-hard, and hence difficult to solve for moderate-to-large problems. This connection does not appear in the current VANET literature and is instrumental in developing efficient cache placement algorithms.
- We develop practical caching algorithms based on the identified connection with the knapsack problem and suboptimal relaxations thereof. These algorithms yield close-to-optimal solutions and have a much lower computational complexity than the corresponding exhaustive search.
- Our numerical results reveal that proactive and cooperative caching schemes yield significant gains over their reactive and non-cooperative counterparts.

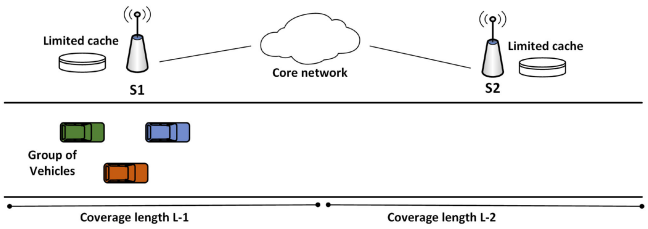

FIGURE 1. Freeway Model.

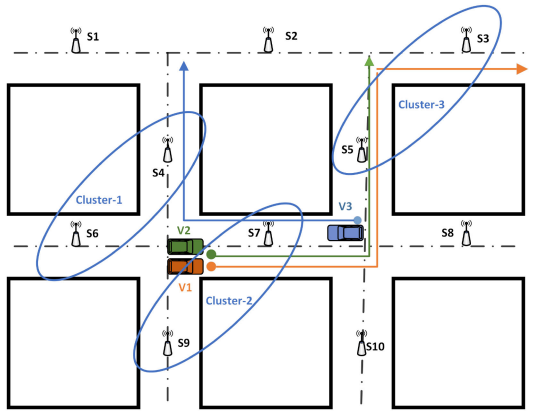

FIGURE 2. City Model.

The rest of the paper is organized as follows. In Section II, we state the system model and the main assumptions. The problem is formulated in Section III, wherein reactive and proactive scenarios and different caching schemes are developed for the freeway and city models. In Section IV we discuss the complexity of the proposed algorithms and in Section V we present and discuss our numerical results. The paper is concluded in Section VI.

II. SYSTEM MODEL

We consider two classes of proactive caching schemes in two common traffic models, the freeway model [18], [25] and the city model [19], [26]. Despite their simplicity, these models will enable us to draw insight into the complexity of cache allocation and the impact of caching on latency. In both models, we consider a set of S RSUs, $\mathcal{S} = \{1, \ldots, S\}$, connected by optical-fiber links, cf. Figs. 1 and 2, and a set of V vehicles, $\mathcal{V} = \{1, \ldots, V\}$. Each RSU $s \in \mathcal{S}$ is equipped with a cache of size Z_s and covers a circular area with diameter L_s. The vehicles are interested in a set of M uncorrelated data items, $\mathcal{M} = \{1, \ldots, M\}$, where the size of the m-th data item is C_m bytes. Each RSU $s \in \mathcal{S}$ serves a number of vehicles within

its coverage zone. The s-th RSU communicates with the v-th vehicle at a data rate of r_{sv} bytes/sec. Due to mobility, a vehicle will be connected to an RSU for a certain amount of time before handing over to the next RSU along its travelling direction. We denote the contact time vector by $\boldsymbol{\tau}^s = \left(\tau_1^s, \ldots, \tau_v^s\right)$, where τ_v^s is the contact time between vehicle v and RSU s in seconds.

We assume that the RSUs can track, learn and predict the behaviour of each vehicle within its coverage zone. Hence, each RSU constructs a demand profile for each vehicle based on its history, and communicated this information to the network coordinator. As such, the network coordinator treats the demands as deterministic, rather than random, quantities. Each profile can be described by the probability with which a user will request a certain file. In particular, let the demand profile of vehicle v be denoted by $\boldsymbol{p}_v := (p_v^1, \ldots, p_v^M)$, where p_v^m is the probability that vehicle v requests data item m. The RSUs are assumed to cache data independently to each vehicle. In other words, the demands of one user at a particular time instant are independent of the demands of other users and the demands of the same user at other time instants.

Let x_s^m be the caching decision of data item m at RSU s, that is,

$$x_s^m \in \{0, 1\}, \forall m \in \mathcal{M}, \forall s \in \mathcal{S}. \tag{1}$$

With this notation setting $x_s^m = 1$ implies that the data item m is cached by the s-th RSU, whereas setting $x_s^m = 0$ implies that this data item is not cached by the s-th RSU.

The size of the cached items must satisfy a storage constraint, whereby the total size of these items must be less than or equal to the total RSU storage Z_s. Therefore, we have

$$\sum_{m=1}^{M} C_m x_s^m \leq Z_s, \qquad \forall s \in \mathcal{S}. \tag{2}$$

We note that, since herein the RSUs are assumed to be connected by optical fiber links, which can support significantly higher rates than their wireless counterparts, we ignore the capacity constraints imposed by the inter-RSUs connections. Indeed, the disparity between the capacity of optical fiber and wireless links typically renders the rate constraints imposed by the inter-RSU links inactive, see e.g., [27]–[29].

Having described the general caching framework considered in this paper. In the next section, we will provide a mathematical characterization of the latency experienced by individual vehicles in the freeway and city models.

III. PROBLEM FORMULATION

In this section we formulate the average latency of the network as an optimization problem for the freeway and city models. A key assumption in this work is that each vehicle can be served by exactly one RSU. This assumption is realistic in VANETs, especially when the RSUs operate over high frequencies and their coverage regions are small to avoid interference. To characterize the latency with which the m-th data item is delivered to a vehicle in either the freeway or the city model, we note that the time required to deliver a data item of size C_m when the transmission rate between the s-th RSU and the v-th vehicle is r_{sv} is given by $\frac{C_m}{r_{sv}}$ if the data item is available in the cache of the RSU and $\left(\frac{C_m}{r_{sv}} + \Delta_m\right)$ if this item is not available in the cache, and must be fetched from the network backhaul with an additional delay Δ_m. Since each RSU has a finite cache size, our objective is to find the optimal caching policy $\{x_s^{m*}\}_{m=1}^M$ to determine the data items to be be cached at each RSU to ensure minimal latency in the network.

For the freeway model, we propose non-cooperative and cooperative caching schemes. In the first scheme, each RSU finds its optimal caching decision independently of the decision of the other RSUs. In contrast, the second caching scheme, the RSUs cooperate to reach collective caching decisions by updating the users' profiles containing information about the direction, demand and velocity of individual vehicles. The implementation of these schemes assumes that the roads are unidirectional and follow the free-flow model. Hence, these schemes are not readily implementable in the city model, wherein the roads are bidirectional with nonzero probabilities of being congested. In that case, we develop two caching methodologies, non-cooperative and cooperative. Both schemes are based on the non-cooperative scheme developed for the freeway model. The non-cooperative approach takes into account congestion probabilities and the route taken by the vehicle. In the cooperative approach, multiple RSUs cooperate to deliver data items from within a cluster of RSUs without resorting to the backhaul network. Numerical simulations show that these approaches result in significant reduction in the average latency of the network.

A. FREEWAY SYSTEM MODEL AND FORMULATION

The freeway model is depicted in Fig. 1. For simplicity, we consider the case of $S = 2$ RSUs, but our approach can be readily extended to $S > 2$ RSUs. To characterize the caching framework in this model, let θ_v be the probability that the v-th vehicle enters the freeway. This probability can be obtained from the BSs to enable the network coordinator to build the mobility profile of each user [22]. In this model, the vehicles flow in one direction, implying that a vehicle lying within the coverage area of the RSU at the entry point, is guaranteed to lie within the coverage area of subsequent RSUs along the freeway.

For ease of exposition we consider the case in which the freeway is in the free-flow state, whereby the traffic density is low and the velocities of the vehicles can be assumed to be independent and identically distributed. In this case, the random velocity $\mathbf{u}_v \in [u_{\min}, u_{\max}]$ of any vehicle $v \in \mathcal{V}$ can be modelled with a truncated Gaussian distribution [16], [30]

with mean μ and variance σ^2. This distribution can be expressed as

$$f_{\mathbf{u}_v}(u) = \begin{cases} \frac{2\exp\left(\frac{-(u-\mu)^2}{2\sigma^2}\right)}{\sqrt{2\pi\sigma^2}\left(\mathrm{erf}\left(\frac{u_{\max}-\mu}{\sigma\sqrt{2}}\right)-\mathrm{erf}\left(\frac{u_{\min}-\mu}{\sigma\sqrt{2}}\right)\right)}, \\ \qquad u_{\min} \le u \le u_{\max}, \\ 0, \qquad \text{otherwise.} \end{cases} \tag{3}$$

The freeway model is a special case of the city model, wherein multiple rather than one road is considered and each road is bidirectional. This model will be described in Section III-B.

We consider the freeway model with two types of networks, viz., proactive and reactive.

1) PROACTIVE NETWORK

In this type of networks, each RSU exploits its available memory to proactively cache elect data items based on the mobility and demand profiles of individual users. The goal is to identify the optimal choice of the data items that minimizes the expected overall delay in the network. We note that in the reactive network model, where the users' requests are served directly from the network backhaul without caching at the RSUs. The delay to deliver the m-th data item to the v-th vehicle is $\left(\frac{C_m}{r_{sv}} + \Delta_m\right)$. However, when this item is available in the cache, the delay reduces to $\frac{C_m}{r_{sv}}$. The total number of items available in the cache of the s-th RSU can be expressed as $\sum_{m=1}^{M} x_s^m$, cf. (1).

Towards characterizing proactive networks, let M_v^s be the maximum number of data items that can be guaranteed to be received by the v-th vehicle from the s-th RSU. This number depends on the contact time between the vehicle and the RSU, τ_v^s, the sizes of the data items in the library, $\{C_m\}_{m=1}^M$, and the time delay required to retrieve these items form the backhaul network, $\{\Delta_m\}_{m=1}^M$. Since the m-th item takes an overall time $(\frac{C_m}{r_{sv}} + \Delta_m)$ to be delivered, in the absence of caching, we have

$$M_v^s = \min\left(\left\lfloor \frac{\tau_v^s}{\max_{m\in\mathcal{M}}\left(\frac{C_m}{r_{sv}} + \Delta_m\right)} \right\rfloor, M\right). \tag{4}$$

In other words, the v-th vehicle is guaranteed to receive k data items from the s-th RSU, provided that k lies in the set $\{1, \ldots, M_v^s\}$. The demand profile of the v-th vehicle comprises the probabilities $\{\rho_v^k\}_{k=1}^M$ that it requests $k \in \{1, \ldots, M\}$ data items, in addition to the probability of demand of the m-th data item, p_v^m, $m = 1, \ldots, M$. In contrast with traditional approaches, e.g., [13], [15], the probabilities $\{p_v^m\}$ enable us to consider an enhanced demand profile wherein not only the probability that a particular vehicle demands k items is given, but also the probabilities that a particular combination of k items is requested by this vehicle. To characterize this profile, we note that, for each number of demands k, there are $G_k = \binom{M}{k}$ combinations of data items that the v-th vehicle may request. Let the i-th combination of the k requested files be indexed by $\mathcal{A}_{ik} = \{l_{i_1}, l_{i_2}, \ldots, l_{i_k}\}$, and let q_{vk}^i denote the conditional probability that the v-th vehicle requests the items in $\mathcal{A}_{ik}$, $i \in \{1, \ldots, G_k\}$. Assuming that the requests are independent yields

$$q_{vk}^i = \prod_{l\in\mathcal{A}_{ik}} p_v^l \prod_{r\notin\mathcal{A}_{ik}} \left(1 - p_v^r\right). \tag{5}$$

Now, since the v-th vehicle will be connected to the s-th RSU for τ_v^s seconds, the maximum number of data items guaranteed to be delivered to the v-th vehicle by this RSU when all requested data items are available in the cache is

$$\hat{M}_v^s = \min\left(\left\lfloor \frac{\tau_v^s}{\max_{m\in\mathcal{M}}\left(\frac{C_m}{r_{sv}}\right)} \right\rfloor, \sum_{m=1}^{M} x_s^m\right). \tag{6}$$

In addition to cached items, the v-th vehicle may request items that are not available in the cache. The RSU will be able to deliver those items only if the contact time τ_v^s exceeds the time required to deliver the $\hat{M}_v^s$ cached items, i.e., $\tau_v^s > \hat{M}_v^s \max_{m\in\mathcal{M}}\left(\frac{C_m}{r_{sv}}\right)$. In this case, the maximum number of uncached data items that can be delivered to the v-th vehicle can be expressed as

$$\tilde{M}_v^s = \left\lfloor \frac{\tau_v^s - \hat{M}_v^s \max_{m\in\mathcal{M}}\left(\frac{C_m}{r_{sv}}\right)}{\max_{m\in\mathcal{M}}\left(\frac{C_m}{r_{sv}} + \Delta_m\right)} \right\rfloor. \tag{7}$$

Note that since $\left\lfloor \frac{\tau_v^s}{\max_{m\in\mathcal{M}}\left(\frac{C_m}{r_{sv}}\right)} \right\rfloor$ is an upper bound on $\hat{M}_v^s$, $\tilde{M}_v^s \ge 0$.

Combining (6) with (7) yields that the number of guaranteed data items that each vehicle can receive is at most

$$\bar{M}_v^s = \min\left(\hat{M}_v^s + \tilde{M}_v^s, M\right). \tag{8}$$

Using the notation in (5) and (8), we obtain an expression for the expected total delay in the proactive model. Doing so, yields

$$W_s^{\mathcal{P}} = \sum_{v=1}^{V}\left(\theta_v \sum_{k=1}^{\bar{M}_v^s} \rho_v^k \sum_{i=1}^{G_k} q_{vk}^i \sum_{l\in\mathcal{A}_{ik}}\left(\frac{C_l}{r_{sv}} + \Delta_l(1 - x_s^l)\right)\right), \tag{9}$$

where the superscript $\mathcal{P}$ is used to identify the proactive model. In the expression in (9), the inner summation accounts for the expected delay incurred by the requests of the v-th vehicle, averaged over all combinations of data items and demand profiles whereas the outer summation accounts for the overall expected delay observed by the vehicles served by the s-th RSU. We note that the backhaul delay Δ_m is multiplied by an indicator of whether the m-th item is cached.

Our goal is to find the optimal caching policy, $\{x_s^{m*}\}_{m\in\mathcal{M}}$, for each RSU $s \in \mathcal{S}$ to ensure that the overall latency is minimized. Towards that end, we will propose two caching schemes, namely, non-cooperative and cooperative. In the first scheme, each RSU determines its optimal caching decision independently of the decisions of other RSUs. In contrast, in the second scheme, the RSUs cooperate to reach collective

caching decisions by updating the users' profiles by incorporating information about their directions, demands and velocities. Numerical results show that the cooperative scheme significantly outperforms its non-cooperative counterpart.

a) Non-Cooperative Caching Scheme: In this scheme, each RSU finds its optimal caching decision independently of the decisions of other RSUs. This is possible because each vehicle is assumed to have a unique association with a serving RSU. Hence, for the s-th RSU to determine its optimal caching policy, it solves the following optimization problem:

$$\begin{aligned} \min_{x_s^m} \quad & W_s^{\mathcal{P}}, \\ \text{subject to} \quad & x_s^m \in \{0,1\}, \forall m \in \mathcal{M}, \\ & \sum_{m=1}^{M} C_m x_s^m \leq Z_s. \end{aligned} \tag{10}$$

This problem is in the form of an integer linear program, that is generally difficult to solve. We note that the quantities $\{\bar{M}_v^s\}$ depend on the (binary) decision variables $\{x_s^m\}$ as shown in (8). Hence, it can be seen that this problem is not affine in the decision variables. In fact, we have the following result, which is formally proved in Appendix A.

Theorem 1: The caching placement problem in (10) is a generalization of the knapsack problem.

Proof: See Appendix A. ■

The NP-hardness of this problem implies that finding the global optimal solution $\{x_s^{m*}\}_{m\in\mathcal{M}}$ of (10) is only possible for small-to-medium size problems, but computationally infeasible for problems with practical sizes. In particular, using exhaustive search to solve (10) can be readily shown to require a number of operations that grows super exponentially with M and V. To show that, we note that in the exhaustive search, the total expected latency for each candidate solution is calculated. The algorithm tests all possible combinations of cached items from the library $\mathcal{M}$, which requires 2^M iterations. Moreover, the algorithm tests all possible demand combinations for all users which requires $\sum_{v=1}^{V}\sum_{i=1}^{\bar{M}_v^s}\binom{M}{i}$ operations. Hence, the overall complexity of solving (10) with exhaustive search is $\mathcal{O}(2^M \sum_{v=1}^{V}\sum_{i=1}^{\bar{M}_v^s}\binom{M}{i})$.

To overcome the computational difficulty arising from the NP-hardness of (10), we exploit the structure of (10) to propose an efficient greedy algorithm which is shown via numerical simulations to yield close to the optimal caching policy. The use of the greedy algorithm is motivated by the connection between the knapsack problem and the optimization problem in (10). Towards finding this algorithm, we rewrite the expected total delay of the s-th RSU in (9) as follows:

$$\begin{aligned} W_s^{\mathcal{P}} &= \sum_{v=1}^{V}\left(\theta_v \sum_{k=1}^{\bar{M}_v^s}\rho_v^k \sum_{i=1}^{G_k} q_{vk}^i \sum_{l\in\mathcal{A}_{ik}}\left(\frac{C_l}{r_{sv}}+\Delta_l\right)\right) \\ &\quad - \sum_{v=1}^{V}\left(\theta_v \sum_{k=1}^{\bar{M}_v^s}\rho_v^k \sum_{i=1}^{G_k} q_{vk}^i \sum_{l\in\mathcal{A}_{ik}}\Delta_l x_s^l\right), \end{aligned} \tag{11}$$

$$\begin{aligned} &= \sum_{v=1}^{V}\left(\theta_v \sum_{k=1}^{\bar{M}_v^s}\rho_v^k \sum_{i=1}^{G_k} q_{vk}^i \sum_{l\in\mathcal{A}_{ik}}\left(\frac{C_l}{r_{sv}}+\Delta_l\right)\right) \\ &\quad - \sum_{m=1}^{M}\left(\sum_{v=1}^{V}\theta_v \sum_{k=1}^{\bar{M}_v^s}\rho_v^k \sum_{i=1}^{G_k}\mathbb{I}_{\mathcal{U}}(m) q_{vk}^i\right)\Delta_m x_s^m, \end{aligned} \tag{12}$$

where for an arbitrary set $\mathcal{U}$ we define

$$\mathbb{I}_{\mathcal{U}}(m) = \begin{cases} 1, & m \in \mathcal{U}, \\ 0, & \text{otherwise.} \end{cases}, \tag{13}$$

and in (12) we use $\mathcal{U} = \mathcal{A}_{ik}$. To obtain (12) from (11), we noted that in the last summation in (11), the caching decision variable of the l-th data item, x_s^l, is weighted by the sum of all the combination probabilities that contain this item.

In Appendix A, we observed that, for a given value of $\bar{M}_v^s$, the minimization in (10) is equivalent to a problem in which $\sum_{m=1}^{M}(\sum_{v=1}^{V}\theta_v\sum_{k=1}^{\bar{M}_v^s}\rho_v^k\sum_{i=1}^{G_k}\mathbb{I}_{\mathcal{A}_{lk}}(m)q_{vk}^i)\Delta_m x_s^m$ is maximized, thereby giving rise to a standard (0-1)-knapsack problem. In this problem, a set $\mathcal{M}$ containing M items is given. Associated with each item, $m \in \mathcal{M}$, are a value, v_m, and a weight, w_m, that account for the benefit and the cost of adding this item to the knapsack, respectively. The goal in this problem is to maximize the total value of the selected items in the knapsack without exceeding its capacity. As shown in Appendix A, this problem maps directly to the problem in (10) when $\bar{M}_v^s$ is fixed. To see this, in Appendix A we set $v_m = (\sum_{v=1}^{V}\theta_v\sum_{k=1}^{\bar{M}_v^s}\rho_v^k\sum_{i=1}^{G_k}\mathbb{I}_{\mathcal{A}_{ik}}(m)q_{vk}^i)\Delta_m$ and $w_m = C_m$. This observation will enable us to apply standard approaches for the knapsack problem [31], [32] to obtain good solutions for the problem in (10).

Among the effective approaches to obtain solutions for the (0-1)-knapsack problem is the greedy technique. This technique features low complexity and is known to yield reliable, albeit not necessarily optimal, solutions for large scale problems which are likely to arise in the practical caching scenarios considered herein.

The philosophy of the greedy technique, which is commonly used in solving the knapsack problem [31], is to make caching decisions successively, starting from the item that has the largest gain-to-cost ratio. Applying this philosophy in the current context implies that, at the i-th iteration of this technique there are $M-i$ members in the set of items to be considered and the m-th item in this set is included in the knapsack if it has the largest value of $\pi^m = v_m/w_m$. This process continues until no more items can be included in the knapsack without exceeding its capacity. The greedy technique is known to yield locally optimal solutions [31]. Unfortunately, the fact that the value of $\bar{M}_v^s$ depends on the caching decision, prevents this technique from being directly applied to the problem in (10).

To circumvent this difficulty, we propose an iterative algorithm wherein a feasible vector of caching decisions is assumed to be given, e.g., $x_s^m = 0$ for every s and $m \in \mathcal{M}$. For these decisions, the value of $\bar{M}_v^s$ is calculated and assumed

Algorithm 1 A Non-cooperative caching policy—The Freeway model.

Initialize $q^i_{vk}, \rho^k_v, \theta_v, C_m, \Delta_m, Z_s, \forall v \in \mathcal{V}, \forall m \in \mathcal{M}$
Set Convergence=False
while *Convergence=False* **do**
  Evaluate $\pi^m_s := \frac{1}{C_m}\left(\sum_{v=1}^{V} \theta_v \sum_{k=1}^{\bar{M}^s_v} \rho^k_v \sum_{i=1}^{G_k} \mathbb{I}_{\mathcal{A}_{ik}}(m) q^i_{vk}\right) \Delta_m$
  Sort π^m_s descendingly
  Fill the memory without exceeding the capacity Z_s
  Output x^m_s
  Update $\bar{M}^s_v, \forall v \in \mathcal{V}$
  if x^m_s *does not change* **then**
    Convergence = True
  end
end
Output x^m_s.

constant. The greedy algorithm is then applied to the problem corresponding to the fixed value of $\bar{M}^s_v$ and a new a set of caching decisions is obtained. These caching decisions are then used to update the value of $\bar{M}^s_v$ and the greedy algorithm is applied again to the new problem. The algorithm continues until convergence.

In Section V we will show that the proposed algorithm yields close-to-optimal caching decisions, but with a significantly higher computational efficiency in comparison with the exhaustive search needed to obtain the optimal decision. In fact, whereas exhaustive search requires an exponential number of computations, the proposed algorithm requires only $\mathcal{O}(MV + M\log M)$ per iteration, where MV is the number of computations required to evaluate each entry of the vector $\{\pi^m\}_{m=1}^{M}$, and $M\log M$ is the complexity of the algorithm required for sorting this vector. For completeness, the details of the proposed technique are given in Algorithm 1 above.

b) Cooperative Caching Scheme: In contrast with the non-cooperative caching scheme, the cooperative counterpart assumes that the RSUs along the freeway cooperate by exchanging information signals. In particular, in this model, the RSUs update the location, direction and demand profiles of each vehicle using information received from preceding RSUs. This mechanism incurs negligible communication delay because of the high speed optical fiber inter-RSUs connections, and can hence be ignored. The main advantage of this scheme is that it avoids redundant caching. To see this, we note that in the non-cooperative scheme each RSU runs its caching algorithm independently of other RSUs. Hence, a data item may be cached in a given RSU even though it has been already downloaded by all users from prior RSUs. Additionally, the cooperative caching scheme invokes the updated profiles to modify the caching priority. For instance, if a data item has already been downloaded by the majority of users from prior RSU, its caching gain in subsequent RSUs is reduced.

The main assumption in the freeway model is that each user enters the freeway with a probability θ_v, i.e., θ_v is the probability that the v-th vehicle was in the coverage area of the first RSU. In the cooperative scheme, the probability that the v-th vehicle lies in the coverage area of subsequent RSUs is deterministic with Boolean values. Enabling this feature increases the certainty of the vehicles' locations, and improves the caching efficiency. That is, once a vehicle enters the freeway, it is expected to continue travelling along it indefinitely. Note that this model does not consider the possibilities that vehicles enter or exit the freeway between consecutive RSUs. These possibilities will be addressed in the City Model considered in the next section.

To characterize the cooperative caching scheme, we define the indicator function $C_v(s)$ to be

$$C_v(s) = \begin{cases} 1, & \text{if the v-th vehicle lies in coverage} \\ & \text{area of the s-th RSU} \\ 0, & \text{otherwise .} \end{cases} \tag{14}$$

Note that the freeway assumption implies that $C_v(s) = C_v(s-1)$ for all $s \in \{2, \ldots, S\}$. We also define the set of the data items downloaded by the v-th vehicle from the s-th RSU as

$$\mathcal{M}^s_v = \left\{m | \text{the } v\text{-th vehicle received item } m \text{ from RSU s } \right\}.$$

To provide the s-th RSU with information about the prior demands of the vehicles, we use $\boldsymbol{p}_{vs}$ to denote the updated demand profile of vehicle $v \in \mathcal{V}$ when it lies within the coverage region $C_v(s)$. To obtain this profile, the s-th RSU updates the demand profile from $\boldsymbol{p}_{v(s-1)}$ to $\boldsymbol{p}_{vs}$, the RSU assumes that a data item that has already been downloaded by the v-th vehicle by the preceding RSU is unlikely to be requested by the same vehicle from the current RSU. This assumption can be mathematically characterized by setting $p^m_{vs} = 0, \forall m \in \mathcal{M}^{s-1}_v$, and leaving the remaining entries of $\boldsymbol{p}_{vs}$ unchanged, apart from necessary scaling, that is, we set

$$p^m_{vs} = \gamma_{vs} p^m_{v(s-1)}, \qquad \forall m \in \mathcal{M} \setminus \cup_{r=1}^{s-1} \mathcal{M}^r_v, \tag{15}$$

where γ_{vs} is a scaling factor that ensures that

$$\sum_{m \in \mathcal{M} \setminus \cup_{r=1}^{s-1} \mathcal{M}^r_v} p^m_{vs} = 1. \tag{16}$$

Note that the profile updating procedure described in (15) takes into consideration all the data items that have been downloaded by each vehicle $v \in \mathcal{V}$ since it entered the freeway.

For (16) to be satisfied, it can be readily seen that the value of γ_{vs} in (15) must be given by

$$\gamma_{vs} = \frac{1}{1 - \sum_{m \in \mathcal{M} \setminus \cup_{r=1}^{s-1} \mathcal{M}^r_v} p^m_{v(s-1)}}.$$

Hence, we have

$$p^m_{vs} = \frac{p^m_{v(s-1)}}{1 - \sum_{m \in \mathcal{M} \setminus \cup_{r=1}^{s-1} \mathcal{M}^r_v} p^m_{v(s-1)}}. \tag{17}$$

From (17) it can be inferred that the demand profile updating procedure will assign higher priority to the caching of data items that have not been previously downloaded, thereby avoiding the caching redundancy that arises in the non-cooperative scheme described in Section III-A1a.

Using the updated demand profiles (17), we obtain conditional probabilities for the possible combinations of data items analogous to the one given in (5). In particular, the updated conditional probabilities for the s-th RSU can be expressed as

$$q^i_{vsk} = \prod_{l\in\mathcal{A}_{ik}} p^l_{vs} \prod_{r\notin\mathcal{A}_{ik}} \left(1 - p^r_{vs}\right). \tag{18}$$

Note that, unlike its non-cooperative counterpart, in the cooperative caching scheme the conditional probabilities are indexed by s, that is the formulation is not identical for each RSU. Proceeding as in the non-cooperative case, we have for the reactive cooperative model, the total expected delay is given by

$$W^{CO-\mathcal{R}}_s = \sum_{v=1}^{V} C_v(s-1) \sum_{k=1}^{M^s_v} \rho^k_v \sum_{i=1}^{G_k} q^i_{vsk} \sum_{l\in\mathcal{A}_{ik}} \left(\frac{C_l}{r_{sv}} + \Delta_l\right), \tag{19}$$

whereas for the proactive cooperative model, the total expected delay is given by

$$W^{CO-\mathcal{P}}_s = \sum_{v=1}^{V} C_v(s-1) \sum_{k=1}^{\bar{M}^s_v} \rho^k_v \sum_{i=1}^{G_k} q^i_{vsk} \times \sum_{l\in\mathcal{A}_{ik}} \left(\frac{C_l}{r_{sv}} + \Delta_l(1-x^l_s)\right). \tag{20}$$

Note that $W^{CO-\mathcal{R}}_s$ and $W^{CO-\mathcal{P}}_s$ are the respective analogs of their non-cooperative counterparts in (21) and (9), but with θ_v replaced with $C_v(s-1)$ and q^i_{vk} replaced with q^i_{vsk}.

Similar to the non-cooperative model, our goal in the cooperative case is to determine the optimal caching policy, $\{x^{m*}_s\}_{m\in\mathcal{M}}$. However, in this model, the RSUs solve their respective optimization problems sequentially, using updated profiles, rather than independently using the original profiles. In other words, in this model, every RSU keeps a record of the data items downloaded by each vehicle and passes this information to the following RSU along the freeway.

The objective of the optimization problem of each RSU is given by the total expected delay in (20), with constraints identical to the ones in (1) and (2). Analogous to the non-cooperative case, these optimization problems possess the knapsack structure when M^s_v is fixed, cf. Theorem (1), and can hence be solved using Algorithm 1. Using this observation we now propose a caching policy for the cooperative caching scheme. In this policy, the first RSU on the freeway, i.e., $s = 1$, uses Algorithm 1 to find its caching policy. For the remaining RSUs, each uses Algorithm 1 but with the profiles it updated using the information it received from the previous RSU. This sequential scheme is summarized in Algorithm 2.

2) REACTIVE NETWORKS

A reactive network can be considered as a special case of the corresponding proactive network when the cache memory is set to zero, i.e., $x^l_s = 0$. Using (9), the total delay of all vehicles when served by the s-th RSU is given by

$$W^{\mathcal{R}}_s = \sum_{v=1}^{V} \left(\theta_v \sum_{k=1}^{M^s_v} \rho^k_v \sum_{i=1}^{G_k} q^i_{vk} \sum_{l\in\mathcal{A}_{ik}} \left(\frac{C_l}{r_{sv}} + \Delta_l\right)\right), \tag{21}$$

where $\mathcal{R}$ is used to identify the reactive model. The inner summation is analogous to its counterpart in the proactive model in (9), but with $x^l_s = 0$, and M^s_v is defined in (4).

Algorithm 2 A cooperative caching policy—The Freeway model.

The first RSU $s = 1$ runs Algorithm 1.
Output $\boldsymbol{x}^m_1$
for *s=2 to S* **do**
  Find $C_v(s-1)$, p^m_{vs} in (17) and q^i_{vsk} in (18)
  Run Algorithm 1 to solve the problem in (20) by replacing θ_v to $C_v(s-1)$ and q^i_{vk} to q^i_{vsk}
  Output $\boldsymbol{x}^m_s$
end

B. CITY SYSTEM MODEL AND FORMULATION

The freeway model can be considered as a special case of the city model. Alternatively, a path in the city model can be represented as a concatenation of freeways, each with its own velocity, contact time, and number of data items that can be reliably delivered, cf. Section III-A. Using this representation will enable us to carry forward the methodology developed for the freeway model to the relatively involved Manhattan city model, allowing users to have different velocities and contact times on each road. To characterize the city model, we consider a city in which the vehicles follow a Manhattan city model, cf. Fig. 2 wherein the vehicles traverse a uniform grid of horizontal and vertical bidirectional streets [33]. Herein, we assume that the vehicles use the global positioning system (GPS) to determine the fastest path to travel to their prescribed destinations. For each street in the grid there are S RSUs, which are assumed to be equidistantly spaced. When the streets are in the free-flow state, the velocities of the vehicles can be modelled as random variables that follow the truncated Gaussian distribution in (3), analogous to the freeway model. However, in the city model, the streets are prone to congestion, causing the velocities of the vehicles to depend on each other. In this case, all vehicles in the zone with congested flow can be assumed to have the same velocity which is uniformly distributed over $[\bar{u}_{\min}, \bar{u}_{\max}]$ [34]. We assume that each vehicle maintains its assigned velocity u_v while traveling across the coverage area of the closest RSU. However, u_v can be different for each street depending on the traffic state [17]. Each RSU receives statistical information from the network coordinator about each vehicle that will be passing by its coverage zone. Such information may include the direction, velocity, and demand profile of each vehicle. Having described the model under consideration, in the following sections, two types of caching schemes will be considered, viz., non-cooperative and cooperative.

1) A NON-COOPERATIVE (NON-CLUSTERED) CACHING SCHEME

Analogous to the freeway model, we develop a non-cooperative scheme for the Manhattan city model considered in this section. A distinguishing feature of this model is the bidirectionality of its roads and the fact that each vehicle can select one of multiple routes between the starting and end points. These routes may have different traffic conditions which usually influence the way in which the vehicle favours a particular route. Information pertaining to traffic conditions are provided to each vehicle by the GPS, and this information is subsequently relayed to the network coordinator. For simplicity, in the current model, we assume that once a vehicle chooses a path, it stays in this path until it reaches the destination. Unlike the freeway model, the intersections of routes in the city model give rise to conflicting objectives at each RSU. For instance, due to channel conditions, an RSU may be inclined to cache particular data items for vehicles traversing its coverage region in the East-West direction, even though vehicles traversing the same region in the North-South direction may have already experienced intolerable delays along their paths. The goal in this section is to account for such conflicts in making caching decisions at each RSU.

Towards accounting for decision conflicts at the RSUs, we assume that the network coordinator uses the information about the path chosen by each vehicle to determine its expected velocity and contact time with the RSUs along its path. For instance, in congested sections, the slow speed of vehicles will result in large contact times with the RSUs lying in this section. The channel conditions in such sections can be favourable, resulting in high data rates, or poor, resulting in outages.

To alleviate caching decision conflicts, we will focus on minimizing the worst case latency experienced by each vehicle along its entire path. In particular, using $C_v(s)$ in (14), the path of the v-th vehicle, $\mathcal{S}_v$, $v \in \mathcal{V}$ can be characterized by the RSUs lying along this path, where

$$\mathcal{S}_v = \{s | C_v(s) = 1\}, \qquad \forall v \in \mathcal{V}.$$

To obtain expressions for the expected latency experienced by the v-th vehicle, we modify the expressions obtained in Sections III-A2 and III-A1 to account for entire path of each vehicle. For reactive networks, this latency can be expressed as $D_v^{\mathcal{R}} = \sum_{s\in\mathcal{S}_v} \sum_{k=1}^{\bar{M}_v^s} \rho_v^k \sum_{i=1}^{G_k} q_{vk}^i \sum_{l\in\mathcal{A}_{ik}} (\frac{C_l}{r_{sv}} + \Delta_l)$ (cf. (21)), whereas for proactive networks, this latency can be expressed as (cf. (9))

$$D_v^{\mathcal{P}} = \sum_{s\in\mathcal{S}_v} \sum_{k=1}^{\bar{M}_v^s} \rho_v^k \sum_{i=1}^{G_k} q_{vk}^i \sum_{l\in\mathcal{A}_{ik}} \left(\frac{C_l}{r_{sv}} + \Delta_l (1 - x_s^l) \right)$$

$$= \sum_{s\in\mathcal{S}_v} \left(\sum_{k=1}^{\bar{M}_v^s} \rho_v^k \sum_{i=1}^{G_k} q_{vk}^i \sum_{l\in\mathcal{A}_{ik}} \left(\frac{C_l}{r_{sv}} + \Delta_l \right) \right)$$

$$- \sum_{m=1}^{M} \sum_{s\in\mathcal{S}_v} \left(\sum_{k=1}^{\bar{M}_v^s} \rho_v^k \sum_{i=1}^{G_k} \mathbb{I}_{\mathcal{U}}(m) q_{vk}^i \right) \Delta_m x_s^m \tag{22}$$

$$= \sum_{s\in\mathcal{S}_v} \left(\sum_{k=1}^{\bar{M}_v^s} \rho_v^k \sum_{i=1}^{G_k} q_{vk}^i \sum_{l\in\mathcal{A}_{ik}} \left(\frac{C_l}{r_{sv}} + \Delta_l \right) \right) - \mathrm{Tr}(B_v^T X), \tag{23}$$

where $\mathbb{I}_{\mathcal{U}}(m)$ in (22) is defined in (13). In (23) X is the matrix containing the caching decisions x_s^m, $\forall m \in \mathcal{M}$, $s \in \mathcal{S}$ and the sm-th element of the matrix B_v is given by

$$b_{vs}^m = \left(\sum_{k=1}^{\bar{M}_v^s} \rho_v^k \sum_{i=1}^{G_k} \mathbb{I}_{\mathcal{U}}(m) q_{vk}^i \right) \Delta_m. \tag{24}$$

Note that, in the expressions of $D_v^{\mathcal{R}}$ and $D_v^{\mathcal{P}}$, the location probabilities $\{\theta_v\}$ are set to 1, since in the city model, we assume that the entire path, $\mathcal{S}_v$, of every vehicle $v \in \mathcal{V}$ is known *a priori*.

In the scheme described in this section, the RSUs do not cooperate, whence each RSU either serves its connected vehicles from its local cache, or delivers the requested data items from the network backhaul with higher latencies. Analogous to the case of the freeway model, our goal in the city model is to determine the optimal caching policy, $\{x_s^{m*}\}_{m\in\mathcal{M}, s\in\mathcal{S}}$. However, unlike the case of the freeway model, in the city model, the goal is to minimize the maximum overall latency expected by any vehicle, thereby providing a performance guarantee for the entire system. This goal can be achieved by minimizing a tight upper bound on the expected delay of any vehicle. This can be expressed using the following optimization problem:

$$\begin{aligned} \min_{t,X} \quad & t, \\ \text{subject to} \quad & t \geq D_v^{\mathcal{P}}, v = \{1, \ldots, V\}, \\ & x_s^m \in \{0, 1\}, \forall m \in \mathcal{M}, \forall s \in \mathcal{S}, \\ & \sum_{m=1}^{M} C_m x_s^m \leq Z_s, \forall s \in \mathcal{S}. \end{aligned} \tag{25}$$

In this formulation, the variable t acts as the upper bound on all the latencies in the network. That is, after solving this problem, the optimal t will be equal to the delay of the vehicle with maximum latency. Unfortunately, this problem constitutes a mixed integer linear program, which generally difficult to solve. In fact, in Appendix B we prove the following lemma.

Lemma 1: The caching placement problem in (25) is a generalization of multidimensional knapsack problem.

Proof: See Appendix B. ■

To overcome the computational difficulty arising from the NP-hardness of (25). We resort to the convex relaxation whereby the Boolean constraint on x_s^m in (25) is removed, i.e., $x_s^m \in [0, 1]$. When $\bar{M}_v^s$ is given and $\{x_s^m\}$ are assumed continuous, the problem in (25) is linear, and can be solved using standard solvers, e.g., CVX [35]. This relaxation has been commonly used for solving integer linear programs, e.g., [36],

Algorithm 3 A Non-cooperative caching policy—The City model.

Initialize $\mathcal{S}_v, q^i_{vk}, \rho^k_v, \tau^s_v, C_m, \Delta_m, r_{sv}, Z_s, \forall v \in \mathcal{V}, \forall m \in \mathcal{M}, \forall s \in \mathcal{S}$.
Set Convergence=False.
Set X= $\mathbf{0}_{M\times S}$.
while *Convergence=False* **do**
  Solve (25) with the relaxed constraint $x^m_s \in [0,1]$.
  Output $\hat{X}, X = \lfloor \hat{X} \rceil$.
  Use (8) to find $\bar{M}^s_v \forall v \in \mathcal{V}$.
  if $\bar{M}^s_v$ *does not change* **then**
    Convergence = True
  end
end
Output X.

and readily yields both a feasible solution and a lower bound on the maximum delay of the network. Although the lower bound is not necessarily achievable it provides a benchmark on the minimum maximum delay.

To find a set of good feasible caching decisions, we propose an iterative algorithm analogous to the ones used in the freeway model. We begin from an initial feasible vector of caching decisions, e.g., $x^m_s = 0, \forall s \in \mathcal{S}, m \in \mathcal{M}$, and use (8) to determine the corresponding value of $\bar{M}^s_v$. Now, we fix $\bar{M}^s_v$ and solve (25), but with the Boolean constrained relaxed. The continuous solutions generated by the relaxed problem are rounded and violated constraints are eliminated by setting the corresponding caching decision to zero. The remaining caching decisions will be feasible. The value of $\bar{M}^s_v$ is updated. This procedure is summarized in the following algorithm. In each iteration of Algorithm 3, the continuous solution generated by the solver is denoted by $\hat{X}$, whereas the rounded solution is denoted by $X = \lfloor \hat{X} \rceil$. We note that, for a given $\bar{M}^s_v$, the solution of the relaxed problem constitutes a lower bound on the minimum maximum latency, whereas the latency corresponding to the rounded caching decisions constitutes an upper bound on this latency. Hence, a small gap between these latencies implies that the rounded decisions are close to optimal. However, the converse is not necessarily true.

2) A COOPERATIVE (CLUSTERED) CACHING SCHEME

Unlike its non-cooperative counterpart, in the cooperative caching scheme the RSUs share their cached data items with an elect group of RSUs. In particular, in this scheme the RSUs in the system are partitioned into non-overlapping clusters. An RSU collaborates with other RSUs in the cluster to serve not only the vehicles directly connected to it, but also the vehicles connected to other members of the cluster. The assumption that underlies this scheme is that the time latency required to deliver a data item available in the cluster is less than the latency required to deliver the item from the backhaul network without any caching.

As an example, we consider a cluster that consists of three RSUs, viz., $\{s_1, s_2, s_3\}$. Suppose now that the v-th vehicle, which is connected to RSU s_1, requests the m-th data item. First, RSU s_1 consults its local cache for the requested item. If this item is found, RSU s_1 delivers it with latency $\frac{C_m}{r_{sv}}$. If, however, RSU s_1 does not find the data item in its local cache, it consults the caches of the other RSUs in the cluster, i.e., s_2 and s_3. If found at this stage, the item is delivered with latency $\left(\frac{C_m}{r_{sv}} + d_m\right)$, where d_m is the time needed to fetch the m-th data item from the common storage of the cluster. Finally, if the requested data item is not found in the cache of any of the members of the cluster, it will be delivered to the vehicle from the backhaul network, incurring a latency of $\left(\frac{C_m}{r_{sv}} + \Delta_m\right)$, where $\Delta_m > d_m$. We note that in this scheme, the RSUs share data traffic among each other. Hence, in this case, the communication delay, d_m, between the RSUs cannot be ignored. This is in contrast with the cooperative scheme in the freeway model in which the RSUs share only traffic information, rather than data, signals. We also note that the delay, d_m, automatically takes into consideration the energy consumed for inter-RSU communication. To see this, we note that d_m can be expressed as the quotient of the size of the desired data item and the data communication rate, which is a function of consumed energy and the modulation schemes used for communication.

To characterize this model, we assume that the system consists of N_c clusters, which are denoted by Φ_i, $i = 1, \ldots, N_c$. The number of RSUs in the i-th cluster is denoted by $|\Phi_i|$. Let $\alpha^{s_j}_1$ denote the maximum latency incurred by delivering any data item directly from RSU s_j to the v-th vehicle, that is, $\alpha^{s_j}_1 = \max_{m\in\mathcal{M}}\left(\frac{C_m}{r_{s_jv}}\right)$. Analogously, we let $\alpha^{s_j}_2 = \max_{m\in\mathcal{M}}\left(\frac{C_m}{r_{s_jv}} + d_m\right)$ denote the maximum latency incurred by delivering any data item from a cluster member other than the RSU to which the v-th vehicle is connected. Finally, we let $\alpha^{s_j}_3 = \max_{m\in\mathcal{M}}\left(\frac{C_m}{r_{s_jv}} + \Delta_m\right)$ denote the maximum latency incurred by delivering any data item from the backhaul network. Using these expressions, we obtain the largest number of guaranteed data items that can be delivered to each vehicle.

As in previous scenarios, we assume that the contact time between the v-th vehicle and RSU s_j is $\tau^{s_j}_v$. Hence, the maximum number of data items that can be guaranteed to be delivered to the v-th vehicle from the cache of RSU s_j is given by

$$\hat{N}^{s_j}_v = \min\left(\left\lfloor \frac{\tau^{s_j}_v}{\alpha^{s_j}_1} \right\rfloor, \sum_{m=1}^{M} x^m_{s_j}\right), \tag{26}$$

where the first term accounts for the number of data items that can be transferred during the contact time and the second term accounts for the number of items available in the cache of RSU s_j. Hence, the time required to transfer the $\hat{N}^{s_j}_v$ data items is $\hat{N}^{s_j}_v \alpha^{s_j}_1$. If $\tau^{s_j}_v > \hat{N}^{s_j}_v \alpha^{s_j}_1$, the RSU will be able to deliver

at most $\lfloor \frac{\tau_v^{s_j} - \hat{N}_v^{s_j}\alpha_1^{s_j}}{\alpha_2^{s_j}} \rfloor$ data items from the cluster, provided that this number is less than the number of items cached in the cluster, excluding RSU s_j. Hence, the maximum number of data items to be transferred from the cluster Φ_i can be expressed as

$$\bar{N}_v^{s_j} = \min\left(\left\lfloor \frac{\tau_v^{s_j} - \hat{N}_v^{s_j}\alpha_1^{s_j}}{\alpha_2^{s_j}} \right\rfloor, \sum_{\Phi_i \setminus s_j} \sum_{m=1}^{M} x_s^m\right). \tag{27}$$

Finally, if the contact time $\tau_v^{s_j}$ exceeds the required time to deliver the $(\hat{N}_v^{s_j} + \bar{N}_v^{s_j})$ cached data items, i.e., $\tau_v^{s_j} > (\hat{N}_v^{s_j}\alpha_1^{s_j} + \bar{N}_v^{s_j}\alpha_2^{s_j})$, the maximum number of uncached data items that can be delivered from the backhaul network to the v-th vehicle can be expressed as

$$\tilde{N}_v^{s_j} = \max\left(\left\lfloor \frac{\tau_v^{s_j} - (\hat{N}_v^{s_j}\alpha_1^{s_j} + \bar{N}_v^{s_j}\alpha_2^{s_j})}{\alpha_3^{s_j}} \right\rfloor, 0\right). \tag{28}$$

Combining (26), (27) and (28), it can be seen that the maximum number of data items guaranteed to be delivered to the v-th vehicle when connected to the RSU s_j in cluster Φ_i is given by

$$N_v^{s_j} = \min\left(\hat{N}_v^{s_j} + \bar{N}_v^{s_j} + \tilde{N}_v^{s_j}, M\right). \tag{29}$$

To obtain expressions for the expected latency experienced by the v-th vehicle, we modify the expressions obtained in Section III-B1 to account for the entire path of each vehicle. For reactive networks, this latency can be expressed as in the non-cooperative scheme in the city model in Section III-B1, that is, $D_v^{\mathcal{R}} = \sum_{s\in\mathcal{S}_v} \sum_{k=1}^{M_v^s} \rho_v^k \sum_{i=1}^{G_k} q_{vk}^i \sum_{l\in\mathcal{A}_{ik}} (\frac{C_l}{r_{sv}} + \Delta_l)$. In contrast, for proactive networks, this latency is given by

$$D_v^{Cl-\mathcal{P}} = \sum_{s\in\mathcal{S}_v} \sum_{k=1}^{N_v^s} \rho_v^k \sum_{i=1}^{G_k} q_{vk}^i t_{i,k}^s, \tag{30}$$

$$t_{i,k}^s = \sum_{l\in\mathcal{A}_{ik}} \left(x_s^l \left(\frac{C_l}{r_{sv}}\right) + \left(1 - x_s^l\right) \mathbb{I}_{\mathcal{U}}\left(\sum_{\Phi\setminus s} x_s^l\right) \left(\frac{C_l}{r_{sv}} + d_l\right) + \left(1 - \mathbb{I}_{\mathcal{U}}\left(\sum_{\Phi} x_s^l\right)\right)\left(\frac{C_l}{r_{sv}} + \Delta_l\right)\right), \tag{31}$$

where $\mathbb{I}_{\mathcal{U}}(m)$ is the indicator function defined in (13), and here we set $\mathcal{U} = [1, \infty)$. The first term in (31) accounts for the time required to transfer the l-th data item in the i-th combination in $\mathcal{A}_{ik}$ to the v-th vehicle, when the item is cached at the s-th RSU to which the vehicle is connected. The second term accounts for the time required to transfer the l-th data item to the v-th vehicle, when a copy of this item is available at the cluster, Φ, but not at the s-th RSU to which the vehicle is connected. In this term, the function $\mathbb{I}_{\mathcal{U}}(\sum_{\Phi\setminus s})$ serves as an indicator which yields a value 1 if a copy of the requested resides in the set $\Phi \setminus s$, and yields a value 0 otherwise. Finally, the last term represents the case where no RSU in the cluster has the requested data item and the item should be fetched from the backhaul network with more latency.

a) The Case of Two RSUs per Cluster: The expressions in (30) and (31) capture the latency involved in transferring data items in a system with an arbitrary number of clusters and an arbitrary number of RSUs in each cluster. This expression can be significantly simplified if each cluster contains two RSUs, say s and $\bar{s}$, where s is the RSU connected directly to the vehicle under consideration and $\bar{s}$ is other RSU in the cluster, i.e., $s \cup \bar{s} = \Phi$. In particular, suppose that a given cluster contains the RSUs s_i and s_{i+1}. Then for the path of vehicle v, $\mathcal{S}_v$, we may have $s = s_i$ and $\bar{s} = s_{i+1}$, whereas for the path of vehicle v', $\mathcal{S}_{v'}$, we may have $s = s_{i+1}$ and $\bar{s} = s_i$. Using this notation, it can be readily verified that the expression in (30) reduces to:

$$D_v^{Cl-\mathcal{P}} = \sum_{s\in\mathcal{S}_v} \sum_{k=1}^{N_v^s} \rho_v^k \sum_{i=1}^{G_k} q_{vk}^i \sum_{l\in\mathcal{A}_{ik}} \Big(-\Delta_l x_s^l - (\Delta_l - d_l)x_{\bar{s}}^l + (\Delta_l - d_l)x_s^l x_{\bar{s}}^l + \left(\frac{C_l}{r_{sv}} + \Delta_l\right)\Big) \tag{32}$$

$$\begin{aligned} &= \sum_{s\in\mathcal{S}_v} \left(\sum_{k=1}^{N_v^s} \rho_v^k \sum_{i=1}^{G_k} q_{vk}^i \sum_{l\in\mathcal{A}_{ik}} \left(\frac{C_l}{r_{sv}} + \Delta_l\right)\right) \\ &\quad - \sum_{m=1}^{M} \sum_{s\in\mathcal{S}_v} \left(\sum_{k=1}^{N_v^s} \rho_v^k \sum_{i=1}^{G_k} \mathbb{I}_{\mathcal{U}}(m) q_{vk}^i\right) \Delta_m x_s^m \\ &\quad - \sum_{m=1}^{M} \sum_{s\in\mathcal{S}_v} \Big(\sum_{k=1}^{N_v^s} \rho_v^k \sum_{i=1}^{G_k} \mathbb{I}_{\mathcal{U}}(m) q_{vk}^i\Big)(\Delta_m - d_m)x_{\bar{s}}^m \\ &\quad + \sum_{m=1}^{M} \sum_{s\in\mathcal{S}_v} \Big(\sum_{k=1}^{N_v^s} \rho_v^k \sum_{i=1}^{G_k} \mathbb{I}_{\mathcal{U}}(m) q_{vk}^i\Big) \\ &\quad \times (\Delta_m - d_m)x_s^m x_{\bar{s}}^m, \end{aligned} \tag{33}$$

where $\mathbb{I}_{\mathcal{U}}(m)$ in (33) is defined in (13). Let the coefficient of x_s^m be denoted by b_{1vs}^m, where

$$b_{1vs}^m = \left(\sum_{k=1}^{N_v^s} \rho_v^k \sum_{i=1}^{G_k} \mathbb{I}_{\mathcal{U}}(m) q_{vk}^i\right) \Delta_m, \tag{34}$$

and the coefficient of $x_{\bar{s}}^m$ be denoted by $b_{2v\bar{s}}^m$, where

$$b_{2v\bar{s}}^m = \left(\sum_{k=1}^{N_v^s} \rho_v^k \sum_{i=1}^{G_k} \mathbb{I}_{\mathcal{U}}(m) q_{vk}^i\right) (\Delta_m - d_m). \tag{35}$$

The cache-sharing feature underlying the proposed clustering scheme enables the RSUs to cooperate to serve the vehicles connected to the cluster. Consequently, the latency of delivering data items to each vehicle is reduced in comparison with the non-cooperative scheme, i.e., when each cluster contains one RSU. The cooperation in this scheme gives rise to a conflict of interest wherein each RSU must decide whether

to favor serving the vehicles directly connected to it or the vehicles in the coverage area of the other RSUs in the cluster.

Analogous to the case of the non-cooperative scheme in the city model, our goal herein is to determine the optimum caching policy that minimizes the maximum overall latency expected by any vehicle. Hence, we will use an optimization framework analogous to the one used in the non-cooperative scheme in (25), but with the latency expression in (33). In other words, the set of constraints in (25) will be replaced with the following set:

$$t \geq D_v^{Cl-\mathcal{P}}, \qquad v = \{1, \ldots, V\}. \tag{36}$$

Analogous to the argument used to prove Lemma 1, it can be verified that the optimization problem resulting from replacing (25) with (36) is NP-hard. In particular, when $x_{\bar{s}}^m = 0$, $\forall m \in \mathcal{M}$, the cooperative scheme corresponding to the set of constraints in (36) reduces to the non-cooperative scheme corresponding to (25), which is shown in Lemma 1 to be NP-hard.

One approach to obtain a good feasible solution of the optimization problem corresponding to the cooperative scheme is to relax the binary constraints, solve the resulting optimization problem and round the decision variables, as in the non-cooperative scheme. However, this approach is difficult to implement due to the non-convexity of the relaxed optimization problem. To circumvent this difficulty, we consider all possible cases of x_s^m and $x_{\bar{s}}^m$ in (33). When $x_s^m = x_{\bar{s}}^m = 0$, there is no caching for the m-th data item and latency assumes the largest possible value. When $x_s^m = 1$ and $x_{\bar{s}}^m = 0$, the m-th data item is cached by the RSU directly connected to the v-th vehicle, and the latency is given by the first two terms. When $x_s^m = 0$ and $x_{\bar{s}}^m = 1$, the m-th data item is cached by the RSU not directly connected to the v-th vehicle, and the latency is given by the first and third terms. Finally, when $x_s^m = x_{\bar{s}}^m = 1$, the last two terms will cancel, and the latency is given by the first two terms, as in the case with $x_s^m = 1$ and $x_{\bar{s}}^m = 0$, however with the penalty that the m-th data item is cached in the cluster, but without contributing to latency reduction. From this observation, it can be seen that without loss of optimality, it suffices to restrict ourselves to the case in which $x_s^m x_{\bar{s}}^m = 0$, which implies that each file is at most cached once in each cluster. Since $x_s^m, x_{\bar{s}}^m \in \{0, 1\}$, this constraint can be expressed in the alternate form $x_s^m + x_{\bar{s}}^m \leq 1$, which is linear and significantly easier to incorporate in the optimization framework. Summarizing, the expression in (33) can be written as

$$\bar{D}_v^{Cl-\mathcal{P}} = G(N_v^s) - \sum_{m=1}^{M} \sum_{s \in \mathcal{S}_v} b_{1vs}^m x_s^m - \sum_{m=1}^{M} \sum_{s \in \mathcal{S}_v} b_{2v\bar{s}}^m x_{\bar{s}}^m, \; x_s^m + x_{\bar{s}}^m \leq 1, \tag{37}$$

$$G(N_v^s) = \sum_{s \in \mathcal{S}_v} \left(\sum_{k=1}^{N_v^s} \rho_v^k \sum_{i=1}^{G_k} q_{vk}^i \sum_{l \in \mathcal{A}_{ik}} \left(\frac{C_l}{r_{sv}} + \Delta_l \right) \right). \tag{38}$$

Algorithm 4 A cooperative caching policy—The City model.

Initialize $\mathcal{S}_v, q_{vk}^i, \rho_v^k, \tau_v^s, C_m, \Delta_m, r_{sv}, Z_s, \forall v \in \mathcal{V}, \forall m \in \mathcal{M}, \forall s \in \mathcal{S}$.
Set Convergence=False.
Set X= $\mathbf{0}_{M \times S}$.
while *Convergence=False* **do**
  Solve (39) using CVX
  Output $\hat{X}$, $X = \lfloor \hat{X} \rceil$.
  Use (29) to find $N_v^s \; \forall v \in \mathcal{V}$.
  if N_v^s *does not change* **then**
    Convergence = True
  end
end
Output X.

Now assuming that N_v^s is given renders the problem in (37) linear and in the form of the multidimensional knapsack problem. Towards minimizing the average latency, we proceed in a fashion similar to the one used in the non-cooperative caching scheme in the city model III-B1. We relax the Boolean constraint on x_s^m in (37) to be $x_s^m \in [0, 1]$, and assume that N_v^s is given. The resulting linear program is solved using standard solvers and the solutions are rounded to yield feasible caching decisions.

$$\begin{aligned} \min_{t,X} \quad & t, \\ \text{subject to} \quad & t \geq \bar{D}_v^{Cl-\mathcal{P}}, v = \{1, \ldots, V\}, \\ & x_s^m \in [0, 1], \forall m \in \mathcal{M}, \forall s \in \mathcal{S}, \\ & x_s^m + x_{\bar{s}}^m \leq 1, \forall m \in \mathcal{M}, \forall s, \bar{s} \in \Phi \\ & \sum_{m=1}^{M} C_m x_s^m \leq Z_s, \forall s \in \mathcal{S}. \end{aligned} \tag{39}$$

Similar to the iterative algorithm proposed in the non-cooperative scheme, we begin with an initial caching placement of $x_s^m = 0, \forall m \in \mathcal{M}, \forall s \in \mathcal{S}$. Using (29), we determine the corresponding value of N_v^s. Using CVX, a set of caching decisions $\{x_s^m\}$ is obtained. These caching decisions are then used to update the value of N_v^s and the algorithm is applied again to solve the problem. The details of this algorithm are summarized in Algorithm 4.

IV. COMPLEXITY ANALYSIS

In this section, we provide the complexity of the considered algorithms. The complexity of all exhaustive search algorithms is exponential. To see that, we note that finding the optimum solutions for the non-cooperative scheme in (10) and the cooperative scheme in (20) requires testing all possible combinations of cached items from the library $\mathcal{M}$, which requires 2^M iterations, and all possible demand combinations for all users, which requires $\sum_{v=1}^{V} \sum_{i=1}^{\tilde{M}_v^s} \binom{M}{i}$ operations. Hence, the overall complexity of solving (10) or (20) with

TABLE 1. Complexity of Exhaustive Search Algorithms

Scenario	Complexity
Non-cooperative	$\mathcal{O}\left(2^M \sum_{v=1}^{V} \sum_{i=1}^{\bar{M}_v^s} \binom{M}{i}\right)$
Cooperative	$\mathcal{O}\left(2^M \sum_{v=1}^{V} \sum_{i=1}^{\bar{M}_v^s} \binom{M}{i}\right)$
Non-clustering	$\mathcal{O}\left(2^M S \sum_{v=1}^{V} \sum_{i=1}^{\bar{M}_v^s} \binom{M}{i}\right)$
Clustering	$\mathcal{O}\left(2^M 2^{\lvert\Phi\rvert} \sum_{v=1}^{V} \sum_{i=1}^{\bar{M}_v^s} \binom{M}{i}\right)$

TABLE 2. Complexity of Proposed Algorithms

Algorithm	Complexity
1	$\mathcal{O}(MV + M \log M)$
2	$\mathcal{O}(MV + M \log M)$
3	$\mathcal{O}\left((M+1)^2 (V + M + S)\right)$
4	$\mathcal{O}\left((M+1)^2 (V + 2M + S)\right)$

exhaustive search is $\mathcal{O}(2^M \sum_{v=1}^{V} \sum_{i=1}^{\bar{M}_v^s} \binom{M}{i})$. For the non-clustering scheme, this complexity is multiplied by S, whereas for the clustering scheme, exhaustive search requires testing all possible caching decisions within the cluster. Hence, the previous complexity is multiplied by $2^{|\Phi|}$. These complexities are summarized in Table 1.

The complexity of the proposed (sub-optimal) algorithms are given in Table 2. These algorithms have polynomial complexity and hence practical for medium-to-large networks. In particular, the greedy algorithms, i.e., Algorithms 1 and 2, can be readily seen to require only $\mathcal{O}(MV + M \log M)$ per iteration, where MV is the number of computations required to evaluate each entry of the vector $\{\pi^m\}_{m=1}^{M}$, and $M \log M$ is the complexity of the algorithm required to sort this vector. For the relaxation-based algorithms, i.e., Algorithms 3 and 4, every iteration requires a number of multiplications equal to the square of the number of variables multiplied by number of constrains [35], which yields the last two entries of Table 2.

V. NUMERICAL RESULTS AND DISCUSSION

In this section, we compare the performance of the schemes proposed in Sections III-A and III-B with the corresponding reactive baseline scheme. For each scenario, we use exhaustive search to obtain the optimum solution. Throughout, the library is assumed to contain $M = 20$ data items. In each instance, we assume that the size, C_m, of the m-th item is a random variable uniformly distributed over [100,1000] Mbytes. The rate provided by the s-th RSU to the v-th vehicle, r_{sv}, is assumed to be a random variable that is uniformly distributed over the interval [100,1000] Mbytes/sec. The time Δ_m needed to fetch the m-th data item from the backhaul network is generated randomly from the interval [0.1,5] seconds. The demand profiles, $\boldsymbol{p}_v$, $v \in \mathcal{V}$ and the probabilities $\{\rho_v^k\}_{k=1}^{M}$ that the v-th vehicle requests $k \in \{1, \ldots, M\}$ data items are modelled using the Zipf distribution [37]. Data items generated by this distribution are randomly permuted to model the non-identical interests of the vehicles in the system.

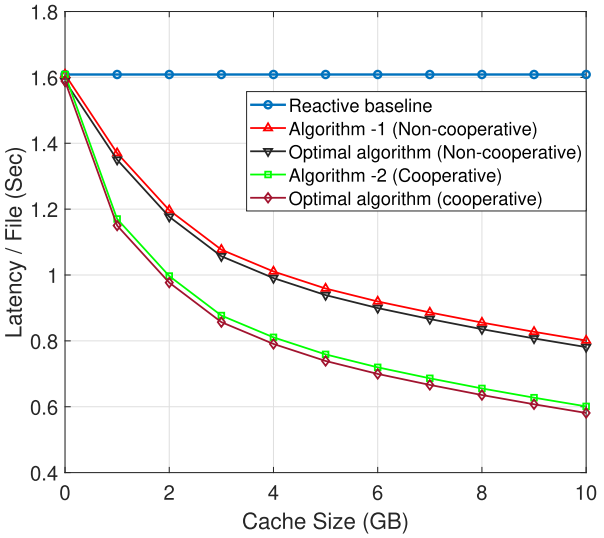

FIGURE 3. Non-cooperative and cooperative caching in the freeway model. Impact of velocity on rate neglected.

A. FREEWAY SCENARIO

In this example, we consider a case with $S = 2$ RSUs, each with a coverage distance $L_s = 50$ m and serving $V = 5$ vehicles. The probabilities of entering the freeway, $\{\theta_v\}$, are random and uniformly distributed over [0, 1]. The velocities of the vehicles u_v, $v \in \mathcal{V}$ are generated using the truncated Gaussian distribution in (3) with mean $\mu = 65$, variance $\sigma^2 = 10$, minimum velocity $u_{\min} = 10$ km/h and maximum velocity $u_{\max} = 120$ km/h. The entries of the contact time vectors $\boldsymbol{\tau}^s$, $\forall s \in \mathcal{S}$, can be readily computed. In particular, $\tau_v^s = \frac{L_s}{u_v}$.

1) NON-COOPERATIVE VS. COOPERATIVE CACHING

Fig. 3 shows the expected time per file for different schemes versus the cache size Z_s. The reactive baseline scenario has the worst performance due to absence of caching. For the non-cooperative caching scheme, each RSU takes its decision independently, neglecting previous information about demand and location statistics. The figure shows that the expected time decreases as the RSU cache size increases. For instance, at cache size of 4 GB, the caching gain is 37.5%. This gain increases to 50% when the cache size increases to 10 GB. For the cooperative caching scheme, each RSU collaborates with the previous RSUs to update its information about the mobility patterns of the vehicles and demand history. This collaboration allows the RSUs to increase the certainty about the users behaviours and enhance their caching decisions. It is noticeable that cooperative caching outperforms its non-cooperative counterpart at all cache sizes. For instance, at a cache size of 4 GB, the cooperative caching scheme achieves a caching gain of 50% in comparison with the 37.5% achieved by the non-cooperative scheme at the same cache size. Finally, we note that the proposed sub-optimal greedy algorithms, viz., Algorithms 1 and 2 achieve a performance close to that of the

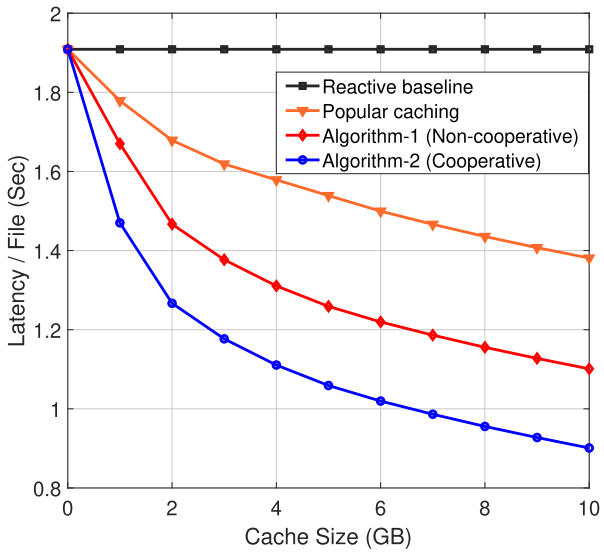

FIGURE 4. Proposed algorithms and popular caching policy in the freeway model. Impact of velocity on rate neglected.

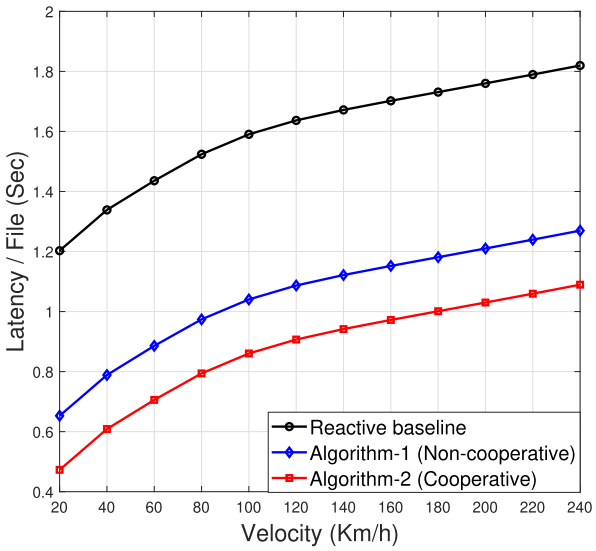

FIGURE 5. Velocity vs latency at cache size 4 GB in the freeway model.

optimal, albeit computationally intensive, exhaustive search algorithms.

2) PROPOSED ALGORITHMS VS. TRADITIONAL CACHING POLICY

In this example, we compare the performance of the proposed algorithms with that of the widely used popular files caching policy [13] in which the RSUs ignore traffic conditions and base their caching decisions solely on prior users' requests [13]. Towards that end, in Fig. 4 we plot the expected latency versus the cache size when the number of users is $V = 40$. For this number of users, exhaustive search is computationally infeasible. Hence, a comparison with its performance is not presented in this figure. From Fig. 4 it can be seen that, all the considered algorithms feature a latency decrease as the cache size increases. However, the proposed algorithms significantly outperform the popular files caching policy. This is largely because the proposed algorithms take into consideration, not only the demand history of the users, as does the popular file caching policy, but also the mobility patterns and traffic conditions. For instance, Fig. 4 shows that the popular files caching policy has a latency advantage of about 15.8% over the reactive baseline when the cache size is 4 GB. In contrast, the advantage of the proposed non-cooperative (Algorithm 1) and cooperative (Algorithm 2) schemes at the same cache size 4 GB is about 30.5% and 42%, respectively.

3) VELOCITY VS. LATENCY

To incorporate the impact of velocity on latency in the presence of caching, we use the simulation-based results reported in [38] pertaining to achievable throughput of LTE-based systems operating in high mobility environment. The impact of caching in this case are given in Fig. 5. In this figure, the expected latency per file is plotted against the velocity when the cache size is 4 GB for the proposed non-cooperative and cooperative algorithms. For comparison, the latency of the reactive baseline is also plotted. It can be seen from Fig. 5 that the latency of all algorithms increase with velocity. For instance, for the non-cooperative algorithm, the caching gain over the reactive baseline is around 38.5% when the velocity is 80 km/h. This gain increases to 48% for the cooperative scheme. This figure highlights the fact that negative impact of velocity on latency can be alleviated by using proactive caching at the RSUs. In other words, caching at the RSUs reduces the delay associated with transferring files from remote servers, thereby compensating for the loss in throughput induced by mobility.

B. CITY MODEL SCENARIO

In this example we consider two scenarios: a non-clustered caching scheme wherein each cluster contains 1 RSU, and a clustered caching scheme wherein each cluster contains 2 RSUs. The RSUs have equal coverage distances of $L_s = 50$ m over each road in the city. The time d_m to fetch the m-th data item form the cluster is generated randomly from the interval [0.1, 2] seconds. For simplicity, we consider the presence of 3 vehicles, v_1, v_2 and v_3, travelling between given starting and end points, as shown in Fig. 2. Three clusters, Φ_1, Φ_2 and Φ_3, marked by blue ellipsoids are shown in this figure. Vehicles v_1 and v_2 move in the same direction but the destination of v_1 is one coverage area ahead of the destination of v_2. Vehicle v_3 moves in the opposite direction. The network coordinator suggests the optimal routes, corresponding to the minimum travel time, based on the GPS and traffic information. In the free flow state we assume that vehicles velocities are generated by a truncated Gaussian distribution with the same parameters as the freeway model. While in case of the congested flow, we assume that vehicles velocities are uniformly distributed over the interval [0, 20] km/h.

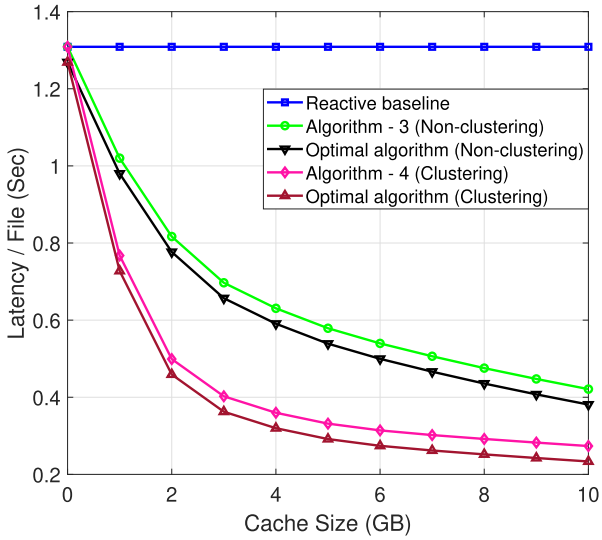

FIGURE 6. Non-clustered and clustered caching schemes in the city model. Impact of velocity on rate neglected.

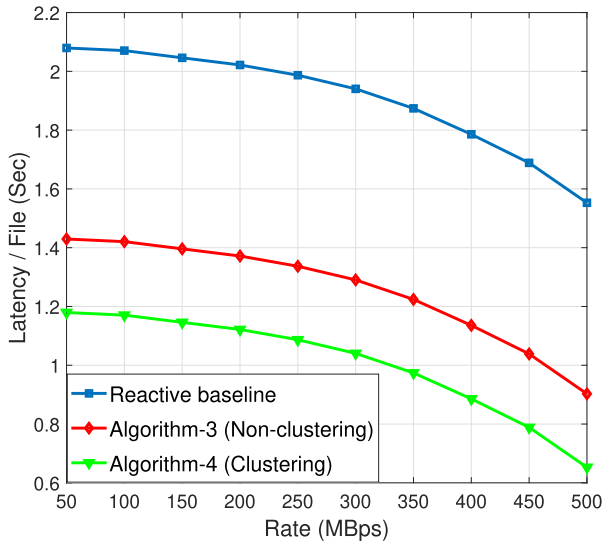

FIGURE 7. Rate and latency in the city model. Impact of velocity on rate neglected.

1) NON-CLUSTERED VS. CLUSTERED CACHING SCENARIOS

Fig. 6 illustrates the expected delay per file for various caching schemes versus the cache size Z_s. The reactive baseline is also shown for comparison. In the case of the non-clustered caching scheme, i.e. with 1 RSU per cluster, each RSU serves the connected vehicles through its local cache if the requested data items are cached. Otherwise, the data items are fetched from the backhaul server with higher latency.

The figure shows that the expected time decreases as the RSU cache size increases to give a caching gain of 51% at a cache size of 4 GB for the non-clustered caching scheme. In contrast, in the case of the clustered scheme, cluster members can share their local caches to serve the connected vehicles in the coverage area of the cluster, thereby giving the clustered scheme an advantage over its non-clustered counterpart. For example, the clustered scheme attains a caching gain of 72% at a cache size of 4 GB, in comparison with the 51% gain achieved by the non-clustered scheme. Numerical results show that the caching gain increases with the number of RSUs per cluster. However, the gain declines as the number of cached files approaches the size of the library $\mathcal{M}$. Finally, for all numerical experiments, the proposed sub-optimal relaxation-based algorithms exhibited close-to-optimal performance albeit with a much smaller complexity.

2) RATE VS. LATENCY

Fig. 7 shows the expected latency per file versus the data rate at a fixed cache memory of 4 GB for the reactive baseline, and the non-clustering and clustering algorithms proposed herein. As seen in Fig. 7, for all considered algorithms, the latency decreases as the data rate increases. For instance, for the non-clustering algorithm, the caching gain over the reactive baseline is about 31.5% when the data rate is 250 MBps. This gain increases to 45% when non-clustering algorithm is replaced with the clustering one.

A key message that can be inferred from this figure is that to maintain a certain level of latency in the reactive model, the data rate must be increased. Unfortunately, this approach stresses the scarce spectral resources of the network, and may not be even feasible. Alternatively, the network can employ caching banks, which are relatively abundant, along with intelligent caching strategies that take into consideration the users' demand profiles and traffic conditions, as per the philosophy of the proposed algorithms. In other words, this figure highlights the inherent trade-off between the cache memory resources and the spectral resources of the network.

3) DIFFERENT TRAFFIC SCENARIOS

In this example, we use the results in [38] to compare the expected latency of Algorithm 3 (non-clustering) at two different velocities, i.e., 30 and 60 km/h. This comparison is depicted in Fig. 8, along with expected latency of the reactive baseline. This figure shows that, as in case of the freeway model, the expected latency of the network increases with velocity. For example, at a cache size of 5GB and a velocity of 60 km/hr, the expected latency obtained by using Algorithm 3 is about 15.44% higher than the corresponding latency at a velocity of 30 km/h. For both velocities, the expected latency decreases with cache size.

C. COMPARISON BETWEEN FREEWAY AND CITY MODELS

We provide a comparative discussion on the impact of caching in the freeway and city models depicted in Fig. 3 and Fig. 6, respectively. In these figures, the impact of velocity on throughput is assumed negligible and the average size of the library is assumed to be 10 GB. Examining the slope of the latency curves of the clustered algorithm in Fig. 6, it

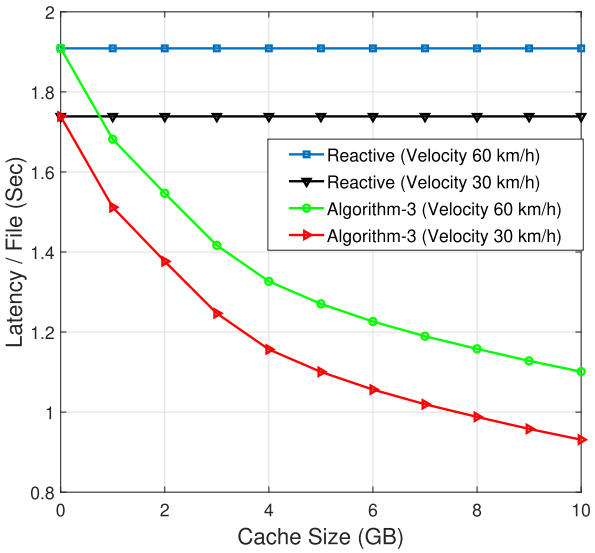

FIGURE 8. Different traffic scenarios in the city model.

can be seen that latency exhibits a relatively sharp slope at small cache sizes, but as the cache size increases, the decrease in latency becomes marginal. In contrast, Fig. 3 shows that although the impact of caching on latency is less than its counterpart in the city model, caching continues to provide substantial improvement in latency as the cache size increases. This phenomenon can be attributed to the fact that in the city model, the impact of caching becomes marginal after most of the data items have been cached in the cluster. For instance, when each RSU has cached about 4 GB of the 10 GB data available in the library. This mechanism is not present in the freeway model; therein caching is performed by individual RSUs, which will not be able to cache a significant portion of the library, whence increasing the cache size in that case continues to exhibit tangible improvement in latency. Hence, in conclusion, when the impact of velocity on throughput is negligible, it is more beneficial to allocate more (clustered) caching to the city model, but as the number of caching module increases, it is more beneficial to allocate these modules to the freeway model.

In contrast with the case in which the impact of velocity on the data rate is negligible, when this impact is significant, the opposite behaviour is observed. In particular, Fig. 5 shows that caching on freeways is more beneficial than caching in cities. This can be attributed to the fact that the high velocities typical of freeways have a negative impact on the data rate. Hence, using caching at the RSUs can reduce the delay associated with transferring files from remote servers, thereby compensating for the loss in throughput. Hence, it can be concluded that when velocity has a significant impact on rate, it is more beneficial for the network to invest in caching on freeways than in cities.

VI. CONCLUSION

We developed non-cooperative and cooperative proactive caching schemes that incorporate user demands and mobility profiles to minimize the communication latency in VANETs. In the non-cooperative schemes, the RSUs operate independently of each other, whereas in the cooperative schemes multiple RSUs collaborate to serve the intended users. Our formulations show that finding the optimal caching decision is NP-hard. However, the insight gained through these formulations enables us to devise efficient caching algorithms, which yield close-to-optimal solutions, but at a much lower computational cost than the corresponding exhaustive search. Numerical results show that the proactive and cooperative caching schemes yield significant gains over their reactive and non-cooperative counterparts.

APPENDIX A
PROOF OF THEOREM 1

We will show that the problem in (10) is a generalization of the the knapsack problem which is known to be NP-Complete [31]. The knapsack problem is an integer linear program, which can be cast in the following form.

$$\begin{aligned} \max_{x^m} \quad & \sum_{m=1}^{M} v_m x_m, \\ \text{subject to} \quad & \sum_{m=1}^{M} w_m x_m \leq W, \\ & x_m \in \{0,1\}, \forall m \in \{1,\ldots,M\}, \end{aligned} \tag{40}$$

where v_m and w_m are nonnegative weights for all $m \in \{1,\ldots,M\}$.

To prove Theorem 1, it suffices to show that a special case reduces to (40). We consider the case in which $\bar{M}_v^s = M$ in (8) for all $v \in \mathcal{V}$.

In this case, $W_s^{\mathcal{P}}$ in (12) can be expressed as

$$\begin{aligned} W_s^{\mathcal{P}} = & \sum_{v=1}^{V} \left(\theta_v \sum_{k=1}^{M} \rho_v^k \sum_{i=1}^{G_k} q_{vk}^i \sum_{l \in \mathcal{A}_{ik}} \left(\frac{C_l}{r_{sv}} + \Delta_l \right) \right) \\ & - \sum_{m=1}^{M} \left(\sum_{v=1}^{V} \theta_v \sum_{k=1}^{M} \rho_v^k \sum_{i=1}^{G_k} \mathbb{I}_{\mathcal{A}_{ik}}(m) q_{vk}^i \right) \Delta_m x_s^m. \end{aligned} \tag{41}$$

Using the expression in (41), it can be readily seen that the optimization problem in (10) is equivalent to the following optimization problem:

$$\begin{aligned} \max_{x_s^m} \quad & \sum_{i=1}^{M} v_m x_s^m, \\ \text{subject to} \quad & \sum_{m=1}^{M} C_m x_s^m \leq Z_s, \forall s \in \mathcal{S}, \\ & x_s^m \in \{0,1\}, \forall m \in \mathcal{M}, \forall s \in \mathcal{S}, \end{aligned} \tag{42}$$

where $v_m = (\sum_{v=1}^{V} \theta_v \sum_{k=1}^{M} \rho_v^k \sum_{i=1}^{G_k} \mathbb{I}_{\mathcal{A}_{ik}}(m) q_{vk}^i) \Delta_m$. Noting that v_m, C_m and Z_s are non-negative establishes the equivalence of the special case with the knapsack problem in (40) by considering $w_m = C_m$ and $W = Z_s$. and completes the proof.

APPENDIX B
PROOF OF LEMMA 1

To prove this lemma, we will show that a special case of (25) is equivalent to the multidimensional knapsack problem which is known to be NP-hard [39]. Towards that end, we note that, upon solving (25), the variable t can take V possible values, each corresponding to the latency of one of the V vehicles in the system. Hence, assuming, without loss of generality, that the vehicle v_1 has the maximum delay, i.e., $D^{\mathcal{P}}_{v_1} \geq D^{\mathcal{P}}_{v \in \mathcal{V} \setminus v_1}$, (25) can be written as:

$$\begin{aligned} \min_{X} \quad & D^{\mathcal{P}}_{v_1}, \\ \text{subject to} \quad & D^{\mathcal{P}}_{v_1} \geq D^{\mathcal{P}}_{v \in \mathcal{V} \setminus v_1}, \\ & x^m_s \in \{0, 1\}, \forall m \in \mathcal{M}, \forall s \in \mathcal{S}, \\ & \sum_{m=1}^{M} C_m x^m_s \leq Z_s, \forall s \in \mathcal{S}. \end{aligned} \tag{43}$$

When $\bar{M}^s_v = M, \forall v \in \mathcal{V}$ yields that the constraints are linear in the binary decision variables, which can be readily cast in the form of the standard multidimensional knapsack problem.

REFERENCES

[1] H. Hartenstein and L. P. Laberteaux, "A tutorial survey on vehicular Ad Hoc networks," *IEEE Commun. Mag.*, vol. 46, no. 6, pp. 164–171, Jun. 2008.

[2] G. S. Paschos, G. Iosifidis, M. Tao, D. Towsley, and G. Caire, "The role of caching in future communication systems and networks," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 6, pp. 1111–1125, Jun. 2018.

[3] J. Tadrous and A. Eryilmaz, "On optimal proactive caching for mobile networks with demand uncertainties," *IEEE/ACM Trans. Netw.*, vol. 24, no. 5, pp. 2715–2727, Oct. 2016.

[4] E. Bastug, M. Bennis, and M. Debbah, "Living on the edge: The role of proactive caching in 5G wireless networks," *IEEE Commun. Mag.*, vol. 52, no. 8, pp. 82–89, Aug. 2016.

[5] K. Zhang, S. Leng, Y. He, S. Maharjan, and Y. Zhang, "Cooperative content caching in 5G networks with mobile edge computing," *IEEE Wireless Commun.*, vol. 25, no. 3, pp. 80–87, Jun. 2018.

[6] N. Golrezaei, K. Shanmugam, A. G. Dimakis, A. F. Molisch, and G. Caire, "Femtocaching: Wireless video content delivery through distributed caching helpers," in *Proc. IEEE Int. Conf. Comput. Commun.*, 2012, pp. 1107–1115.

[7] Y. Guan, Y. Xiao, H. Feng, C. C. Shen, and L. J. Cimini, "Mobicacher: Mobility-aware content caching in small-cell networks," in *Proc. IEEE Glob. Commun. Conf.*, 2014, pp. 4537–4542.

[8] S. Hosny, A. Eryilmaz, and H. E. Gamal, "Impact of user mobility on D2D caching networks," in *Proc. IEEE Glob. Commun. Conf.*, 2016, pp. 1–6.

[9] B. Gabr, S. Hosny, and M. Nafie, "Content delivery in mobility-aware D2D caching networks," in *Proc. IEEE Glob. Commun. Workshops*, 2018, pp. 1–7.

[10] O. Attia and T. ElBatt, "On the role of vehicular mobility in cooperative content caching," in *Proc. IEEE Wireless Commun. Netw. Conf. Workshop*, 2012, pp. 350–354.

[11] F. Wang, Z. Wang, Z. Yang, and S. Chen, "Contact duration aware cache refreshing for mobile opportunistic networks," *IET Netw.*, vol. 5, no. 4, pp. 93–103, Jul. 2016.

[12] Y. Li, D. Jin, P. Hui, and S. Chen, "Contact-aware data replication in roadside unit aided vehicular delay tolerant networks," *IEEE Trans. Mobile Comput.*, vol. 15, no. 2, pp. 306–321, Feb. 2016.

[13] R. Ding, T. Wang, L. Song, Z. Han, and J. Wu, "Roadside-unit caching in vehicular ad hoc networks for efficient popular content delivery," in *Proc. IEEE Wireless Commun. Netw. Conf.*, 2015, pp. 1207–1212.

[14] A. Mahmood, C. Casetti, C. F. Chiasserini, P. Giaccone, and J. Harri, "Mobility-aware edge caching for connected cars," in *Proc. Wireless On-Demand Netw. Syst. Serv. Conf.*, 2016, pp. 1–8.

[15] J. Ma, J. Wang, G. Liu, and P. Fan, "Low latency caching placement policy for cloud-based VANET with both vehicle caches and RSU caches," in *Proc. IEEE Glob. Commun. Workshops*, 2017, pp. 1–6.

[16] S. M. Abuelenin and A. Y. Abul-Magd, "Empirical study of traffic velocity distribution and its effect on VANETs connectivity," in *Proc. Int. Conf. Connected Veh. Expo.*, 2014, pp. 391–395.

[17] Y. Zhang, H. Zhang, W. Sun, and C. Pan, "Connectivity analysis for vehicular Ad Hoc network based on the exponential random geometric graphs," in *Proc. IEEE Intell. Veh. Symp.*, 2014, pp. 993–998.

[18] S. Madi and H. Al-Qamzi, "A survey on realistic mobility models for vehicular Ad Hoc networks VANETs," in *Proc. Int. Conf. Netw., Sens. Control*, 2013, pp. 333–339.

[19] A. Mahajan, N. Potnis, K. Gopalan, and A.-I. Wang, "Urban mobility models for VANETs," in *Proc. IEEE Workshop Next Gener. Wireless Netw.*, 2006, pp. 333–339.

[20] J. Harri, F. Filali, and C. Bonnet, "Mobility models for vehicular Ad Hoc networks: A survey and taxonomy," *IEEE Commun. Surv. Tut.*, vol. 11, no. 4, pp. 19–41, Oct.–Dec. 2009.

[21] D. Karamshuk, C. Boldrini, M. Conti, and A. Passarella, "Human mobility models for opportunistic networks," *IEEE Commun. Mag.*, vol. 49, no. 12, pp. 157–165, Dec. 2011.

[22] C. Song, Z. Qu, N. Blumm, and A.-L. Barabási, "Limits of predictability in human mobility," *Science*, vol. 327, no. 5968, pp. 1018–1021, Dec. 2010. [Online]. Available: http://science.sciencemag.org/content/327/5968/1018

[23] I. F. Akyildiz and W. Wang, "The predictive user mobility profile framework for wireless multimedia networks," *IEEE/ACM Trans. Netw.*, vol. 12, no. 6, pp. 1021–1035, Dec. 2004.

[24] O. B. Fikir, I. O. Yaz, and T. Özyer, "A movie rating prediction algorithm with collaborative filtering," in *Proc. Int. Conf. Adv. Social Netw. Anal. Mining*, 2010, pp. 321–325.

[25] Y. AlNagar, S. Hosny, and A. A. El-Sherif, "Towards mobility-aware proactive caching for vehicular ad hoc networks," in *Proc. IEEE Wireless Commun. Netw. Conf. Workshop*, 2019, pp. 1–6.

[26] Y. AlNagar, S. Hosny, and A. A. El-Sherif, "Proactive caching for vehicular ad hoc networks using the city model," in *Proc. IEEE Wireless Commun. Netw. Conf. Workshop*, 2019, pp. 1–7.

[27] J.-i. Kani, S. Kuwano, and J. Terada, "Options for future mobile Backhaul and Fronthaul," *Opt. Fiber Technol.*, vol. 26, pp. 42–49, Dec. 2015.

[28] X. Liu and F. Effenberger, "Emerging optical access network technologies for 5G wireless," *J. Opt. Commun. Netw.*, vol. 8, no. 12, pp. B70–B79, Dec. 2016.

[29] A. Amari, O. A. Dobre, R. Venkatesan, O. S. S. Kumar, P. Ciblat, and Y. Jaouën, "A survey on fiber nonlinearity compensation for 400 Gb/s and beyond optical communication systems," *IEEE Commun. Surv. Tut.*, vol. 19, no. 4, pp. 3097–3113, Oct.–Dec. 2017.

[30] S. Yousefi, E. Altman, R. El-Azouzi, and M. Fathy, "Analytical model for connectivity in vehicular ad hoc networks," *IEEE Trans. Veh. Technol.*, vol. 57, no. 6, pp. 3341–3356, Nov. 2008.

[31] H. R. Lewis and C. H. Papadimitriou, "Elements of the theory of computation," *ACM SIGACT News*, vol. 29, no. 3, pp. 62–78, Apr. 1998.

[32] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*, 3rd ed. Cambridge, MA, USA: MIT Press, 2009.

[33] M. Alam, M. Sher, and S. A. Husain,"Integrated mobility model (IMM) for VANETs simulation and its impact," in *Proc. Int. Conf. Emerg. Technol.*, 2009, pp. 452–456.

[34] A. Y. Abul-Magd, "Modeling highway-traffic headway distributions using superstatistics," *Phys. Rev. E*, vol. 76, Nov. 2007, Art. no. 057101.

[35] M. Grant and S. Boyd, "CVX: Matlab software for disciplined convex programming, version 2.1," 2014.

[36] K. Genova and V. Guliashki, "Linear integer programming methods and approaches—A survey," *J. Cybern. Inf. Technol.*, vol. 11, no. 1, pp. 1–25, Jan. 2011.

[37] L. Breslau, P. Cao, L. Fan, G. Phillips, and S. Shenker, "Web caching and Zipf-like distributions: Evidence and implications," in *Proc. IEEE Int. Conf. Comput. Commun.*, vol. 1, pp. 126–134, 1999.

[38] M. Lerch, J. Rodriguez-Pineiro, J. A. Garcia-Naya, and L. Castedo, "Methods to perform high velocity LTE experiments at low velocities," in *Proc. IEEE Veh. Technol. Conf.*, 2016, pp. 1–5.

[39] A. Fréville, "The multidimensional 0-1 knapsack problem: An overview," *Eur. J. Oper. Res.*, vol. 155, no. 1, pp. 1–21, 2004.