FleaNet: A Virtual Market Place on Vehicular Networks

Uichin Lee, *Member, IEEE*, Jiyeon Lee, Joon-Sang Park, *Member, IEEE*, and Mario Gerla, *Fellow, IEEE*

***Abstract*—In this study, we introduce the concept of a virtual "flea market" over a Vehicular Ad Hoc Network (VANET) called FleaNet. FleaNet customers express their demands/offers to buy/sell items via radio queries. These queries are opportunistically disseminated, exploiting the mobility of other customers, until a matching customer/vendor is found. We identify the key performance metrics, namely, query resolution latency, scalability, mobility, and churning, and evaluate their impact on performance using analytic and simulation models. The results show that FleaNet can function as an effective virtual marketplace.**

***Index Terms*—Mobility, query dissemination, routing, vehicular ad hoc networks (VANETs), vehicular applications.**

I. Introduction

Wireless mobile devices, such as smart phones, personal digital assistants (PDAs), and laptops, have become ubiquitous in our daily lives. We use them at home, while we walk, and while we drive. In the future, every vehicle will be equipped with wireless communication devices (dedicated short-range communication (DSRC)/802.11p (WAVE) [2], [3]) that will enable communications with roadside objects and with other vehicles. These devices can usher in a new era of pervasive computing in which seamless access to information sources will be provided. When traveling or shopping, for instance, we can search the *web* to obtain directions or to locate specific products. In fact, not only can such devices empower us with ubiquitous Internet access, but they can also create a new environment where opportunistic cooperation can emerge among users with shared interests/goals, e.g., drivers exchanging safety-related information, shoppers/sellers trading goods, etc. [4]. Thus, using wireless devices, drivers will be able to communicate to a greater extent and with a greater flexibility than ever before.

The research described in this paper is related to an emerging body of work on vehicular wireless connectivity. Previous works include PeopleNet [5], which enables wireless users to form a virtual social network that mimics the way people seek information via social networking through direct contacts. In PeopleNet, devices communicate with each other on behalf of their owners to exchange or obtain needed information. For example, an individual who wants to buy/sell a ticket for a Yankees/Red Sox baseball game for which tickets are sold out can simply "post" a buy/sell ticket query (for a stated price) near Yankee Stadium. To permit the exchange of information over relatively large geographic areas, PeopleNet uses fixed infrastructure (e.g., cellular networks, mesh networks, etc.) to post a query to its geographically pertinent place called a bazaar, e.g., the Yankees' stadium. Within this bazaar, it takes advantage of the *free-of-charge* but intermittent connectivity provided by short-range radio technologies, such as Bluetooth, resulting in epidemic query dissemination. In other words, a query is propagated from one device to another without restriction whenever a wireless connection can "opportunistically" be established. If a match to the query is found, then the user who initially placed the query is eventually informed of the match (e.g., via e-mail or short message service).

In this paper, we propose FleaNet, which is a "virtual flea market" service that works in urban vehicular networks to facilitate communications between buyers and sellers of goods (or information) and to efficiently find matches of interest, potentially leading to transactions. FleaNet provides an excellent platform for mobile location-aware classified ads, such as those related to selling/buying baseball tickets to/from interested mobile users in a local area. FleaNet can also be used for the retrieval of more generic location-aware (sensor) data, such as gas prices or traffic information. The mobile, virtual flea market concept extends PeopleNet in many ways. First, intervehicular communication enables an extended geographic coverage, since DSRC/WAVE [2], [3] has a longer radio range, and vehicles provide a means for faster mobility (driving versus walking). Second, FleaNet relaxes the concept of a bazaar, allowing users to define their own bazaar by specifying the region of interest, such as the area where they reside. Third, FleaNet can operate on vehicular ad hoc networks (VANETs) without any infrastructure support because it uses intervehicular communication for query dissemination, matching, and notification.

In fact, the urban vehicular network is large scale, highly mobile, and quite dense. Occasional intermittent network connectivity is overcome by mobility. The FleaNet design must

Manuscript received December 29, 2008; revised May 22, 2009 and August 6, 2009. First published August 25, 2009; current version published January 20, 2010. This work was supported in part through participation in the International Technology Alliance by the U.S. Army Research Laboratory and the U.K. Ministry of Defense under Agreement W911NF-06-3-0001 and in part by ARMY Multi-disciplinary University Research Initiative (MURI) under Grant W911NF0510246. This paper was presented in part at the Second International Workshop on Vehicle to Vehicle Communications in 2006 [1]. Part of this work was done while U. Lee, J. Lee, and J.-S. Park were affiliated with the University of California at Los Angeles. The review of this paper was coordinated by Dr. L. Cai.

U. Lee is with Bell Labs, Alcatel-Lucent, Holmdel, NJ 07733 USA (e-mail: uclee@ieee.org).

J. Lee is with Cisco Systems, Inc., San Jose, CA 95134 USA (e-mail: jayle@cisco.com).

J.-S. Park is with the Department of Computer Science, Hongik University, Seoul 121-791, Korea (e-mail: jsp@hongik.ac.kr).

M. Gerla is with the Department of Computer Science, University of California at Los Angeles, Los Angeles, CA 90095 USA (e-mail: gerla@cs.ucla.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TVT.2009.2030892

0018-9545/$26.00 © 2009 IEEE

Authorized licensed use limited to: Korea Advanced Inst of Science & Tech - KAIST. Downloaded on June 27,2023 at 04:26:16 UTC from IEEE Xplore. Restrictions apply.

be scalable and must tolerate intermittent connectivity. It must not interfere with safety messaging, the primary motivation for V2V communications. For these reasons, the unrestricted epidemic dissemination of PeopleNet and other such schemes [6]–[10] is unsuitable in FleaNet because of the risk of flooding and the overhead of many redundant matches.

FleaNet remedies this problem through mobility-assisted query dissemination, where the query "originator" periodically advertises a query only to a few hop neighbors. Each neighbor then stores the advertisement (i.e., query) in its local database without any further relaying; thus, the query spreads only because of vehicle motion. Upon receiving a query, a node tries to locally resolve it in its database; in case of a match, the originator will automatically be informed. Since a match only happens in its neighbors, redundant match notification can be minimized. The match could lead to an actual transaction; FleaNet also provides a routing mechanism for the notification with last encounter routing (LER) [11]—a geo-routing method that combines a location service and routing service, which is extended to cope with intermittent network connectivity.

This paper contributes new concepts/results to the existing body of research in this area as follows.

1) We introduce a new model for a virtual marketplace over vehicular networks and propose FleaNet, which a novel epidemic-based architecture that supports query dissemination, matching, and notification routing and scales up to thousands of nodes without disrupting existing services. FleaNet handles the issue of nonuniform popularity of queries by providing different levels of query dissemination strategies, ranging from single-hop dissemination to multiuser cooperative dissemination.
2) The FleaNet protocols are extensively evaluated through both analysis and simulation. Our results show that a random query can be resolved, in most cases, within a tolerable amount of time and with minimal bandwidth, storage, and processing/routing overheads. If the advertiser, i.e., Adstation, is stationary, then the query resolution time is critically dependent on its location. In addition, FleaNet tolerates a reasonable degree of churning (i.e., nodes with queries leave the network area).

This paper is organized as follows: In Section II, we review related research. In Section III, we present the FleaNet system architecture. In Section IV, we introduce the FleaNet system. This is followed by a simple analysis in Section V. In Section VI, we evaluate our protocols through extensive simulations. Finally, we present the conclusions of the research in Section VII.

II. Related Work

A. Query Dissemination

Mobile-to-mobile information exchange with infrastructure support has been addressed in PeopleNet [5]. It forms a virtual wireless social network through portable devices with multiple network interfaces, such as cellular and Bluetooth interfaces. In PeopleNet, a given area is divided into nonoverlapping regions called bazaars, each of which is dedicated to handling certain types of queries placed by users. A query (either sell or buy) is propagated through the network infrastructure, typically the cellular network, to k randomly selected users in the associated bazaar; this acts to generate the initial seeds for data dissemination. To spread queries in a sparse network in a given bazaar, PeopleNet uses epidemic query dissemination; a node randomly swaps queries with one of its neighbors because encounters between people may be rare and the size of the buffers is relatively small.

As mentioned earlier, PeopleNet is less efficient when it operates in a dense large-scale network, such as a VANET, because the dissemination policy results in network-wide flooding; this is also true for other epidemic dissemination protocols [6]–[10]. One might argue that this problem can be alleviated by introducing probabilistic forwarding and a large advertisement interval (e.g., a received query could be disseminated with low probability), but such a policy is intrinsically not scalable, because a node could potentially receive a number of queries proportional to the size of the network.[1] In addition, this epidemic dissemination may potentially lead to many "redundant" matches. In the worst case, given that a sell query has already disseminated to all users, for example, N, once a buy query starts spreading from the originator, it will generate $N-1$ redundant matches. The originator will be notified of these redundant matches, which will severely degrade the performance. For these reasons, PeopleNet or other similar schemes [6]–[10] cannot directly be applied to vehicular networks.

Instead, we utilize the two-hop *multicopy* protocol, which is a mobility-assisted data dissemination technique [12] wherein the source node only copies a message to its direct contact nodes (i.e., to nodes within its communication range) and those nodes may then forward it to the destination node. It is important to note that such a protocol is "scalable," as shown in our earlier paper [1]. The key difference from the *multicopy* protocol is that FleaNet does not assume any particular destination for a given message; rather, the message is disseminated over the network, and intermediate nodes with a matched query of shared interests are the potential destinations of the message.

In AdTorrent [13], static wireless digital billboards on the roadside are used for advertising. Digital billboards only push advertising content to the vehicles passing by. Such content is potentially large in size, e.g., hotel virtual tours or show previews. Each node gossips about its content availability to neighbors to facilitate the search. Mobile users search for content of interest by querying neighbors via multihop querying and then download the advertising content. When a network suffers from intermittent connectivity, such multihop querying/downloading may not be feasible. The overall download delay could be very large as a user has to encounter other users with data of interest. Given this, Zhang *et al.* [14] proposed Roadcast, which is a popularity-aware content-sharing scheme. Instead of downloading a data item that is highly relevant to a user's interest, Roadcast trades relevance for download delay by reranking data items based on (local access) popularity to give more opportunities to spread data items with high popularity,

[1] Readers can find the detailed analytic results in our earlier paper [1].

Authorized licensed use limited to: Korea Advanced Inst of Science & Tech - KAIST. Downloaded on June 27,2023 at 04:26:16 UTC from IEEE Xplore. Restrictions apply.

thus increasing data accessibility in the future. FleaNet is more focused on disseminating and resolving spatiotemporal queries through which users satiate their market demands, such as buying or selling an item. Since there is equilibrium between supply and demand in a typical market, we could say that FleaNet is basically a "balanced" push/pull system. Moreover, the concept of popularity in FleaNet is different from that in Roadcast because FleaNet users are interested in a certain "good" and not a specific query (e.g., there could be many queries from different users for the same good).

Note that we assume that the nodes are willing to consume their resources to help other nodes by, for instance, disseminating the advertisements of other nodes. In our companion paper [15], we investigated vehicular networks with non-cooperative nodes and proposed mechanisms to induce or enforce cooperation among nodes.

B. Geographic Routing

Georouting in VANET has been extensively investigated for scalable delivery. Georouting works well in dense networks. However, if the vehicle density is low, for example, during nonrush hour periods and in peripheral areas, then the vehicle connectivity is intermittent. In this case, it is possible to exploit the predictable mobility in a VANET to "assist" georouting with *carry and forward*: A vehicle carries packets and forwards them to a newly found vehicle that is moving toward their destinations. This works well only for *delay-tolerant* applications. Reference [16] uses the knowledge of the relative velocities and directions of one's neighbors to make forwarding decisions. A mobility-centric data dissemination algorithm for vehicular networks (MDDV) [17] utilizes vehicle traffic history data and routes packets based on a predetermined trajectory using a digital map.

A prerequisite of geographic routing is a location service that informs the location of the destination. Devising efficient, scalable, and robust location services has been an active area of research in recent years [18]. Unfortunately, the frequent changes experienced in a vehicle network topology make an accurate location service costly. An elegant way of reducing this cost is by exploiting the spatial–temporal correlation that exists in most realistic mobility patterns, i.e., the distance between two nodes is more or less correlated with the time elapsed since their last encounter. This observation brought forth LER [11]. In FleaNet, since each vehicle can piggyback its current position into its query advertisement, LER can be supported at no extra cost. LER, however, does not address intermittent connectivity. In this paper, we enhance LER by providing the carry-and-forward functionality. As we will see later, enhanced LER plays a key role in FleaNet when the query source needs to notify the owner of the chosen match of its decision.

C. Publish and Subscribe Systems

The publish/subscribe (or pub/sub) paradigm has widely been used to provide selective information dissemination where "publishers" act as information providers, "subscribers" act as information consumers, and a "broker" mechanism routes relevant publications to the subscribers of interest. Existing pub/sub systems for mobile networks usually use subscription messages to establish routing structure, such as [21], or to learn mobility patterns for delay-tolerant routing [22], which is then used to route publication messages to the subscribers. In FleaNet, however, a user's query is symmetric, since it can be both a publication and a subscription. Thus, we use hop-limited opportunistic data dissemination for both types of queries, where nodes broadcast their own queries to k-hop neighboring nodes, and queries are opportunistically diffused to the network as a result of vehicular mobility. During this hop-limited query broadcasting, a match of interest can be found by comparing incoming queries with the queries stored in one's storage. Like the context of relevance or interest in Frey and Roman's work [23], FleaNet users specify the regions of interest to which their queries will be disseminated. Note that FleaNet differs from existing spatial/temporal pub/sub systems [23], [25] in that it uses opportunistic data dissemination, employing vehicular mobility patterns instead of geocasting, which typically assumes network connectivity; the proposed dissemination strategy is scalable and nonintrusive to other existing protocols such as periodic safety message broadcasting.

III. FleaNet System Architecture

A. System Model

FleaNet is based on urban vehicular networks in which vehicles communicate through a wireless interface, such as DSRC [2] and 802.11p (WAVE) [3]. DSRC/WAVE operates in the 5.9-GHz band having multiple channels, some of which are allocated for non–safety-related applications, such as FleaNet. Vehicles, as well as static roadside *Adstations*, generate and propagate advertisements or queries. As we will see, our dissemination is quite simple: a node periodically broadcasts its query to neighboring nodes, and thus, a query is dispersed in the network via "mobility."

Advertisements in FleaNet are location sensitive because a publisher specifies the region of interest in the advertisements (e.g., within 10 mi from the publisher's home location). This relaxes the concept of a bazaar. The region of interest could be centered on the user's current location or the location of a popular place. For instance, a mobile user who wants to sell baseball tickets near the stadium can set his/her region of interest to be centered at the stadium, and a roadside gas station can set its region of interest to be centered at the store location. Thus, FleaNet is suited for mobile classified ads (or bazaars) for selling/buying items to/from local area users and also for advertising and searching for more generic location-aware data, such as gas prices and special offers in a certain area.

One of the main distinctions between FleaNet nodes and typical personal mobile devices, such as PDAs and smartphones, is that FleaNet nodes (e.g., in-car computers or desktop machines in a building) are not strictly restricted by energy constraints and possess fairly high processing power and storage. FleaNet does not consider Internet connections through the hotspots available on urban streets or through second- or third-generation services; rather, it focuses on information

Authorized licensed use limited to: Korea Advanced Inst of Science & Tech - KAIST. Downloaded on June 27,2023 at 04:26:16 UTC from IEEE Xplore. Restrictions apply.

dissemination using "free-of-charge" vehicle-to-vehicle communications. Moreover, we assume that users with the same goal are willing to cooperate.

B. FleaNet Messages

In FleaNet, every message has a common message header that includes a message type field followed by a user ID (UID) and encoded positions. There are three message types: 1) query; 2) match; and 3) transaction. First, a query message is used for representing information about the goods that a person possesses or seeks. Second, a match message is used to inform users that a node has found queries with a matching interest. Finally, a transaction message is used whenever a user wants to make a transaction request or respond to such a request.

A query includes the sequence number, query type, notification flag (N), maximum number of matches (N_{MAX}), reference location/radius (R-Loc/Radius), expiration time, detailed user description, and match sensitivity threshold. The sequence number, along with the UID, is used to uniquely identify a query. There could potentially be many types of queries, but for the sake of simplicity, we assume that there are two query types, namely, buy and sell, for the same item. A notification flag is set if a user wants to receive the query resolution results. By setting N_{MAX}, a user can explicitly set the maximum number of matches that he/she wants to receive. R-Loc contains the user's reference location and the dissemination radius. A user can define an arbitrary location (e.g., his/her real home address) as a home location and accordingly set the dissemination radius. The expiration time for a query can be set through Exp. The user's detailed description is in the user description field. The N_{MAX}, or the expiration time field, permits nodes to automatically dispose of the query after either receiving N_{MAX} matches or passing the expiration time. The query size is relatively small, and a node typically packs a number of queries into a single packet.

C. Query Similarity Measure

In FleaNet, we say that query q_1 matches query q_2 when the degree of similarity is above the threshold defined in the query $q_1(q_1.MST)$; i.e., $sim(q_1, q_2) \geq q_1.MST$, where the function sim is a vector-based similarity metric. To be precise, a query vector $\vec{q}$ is defined as $\vec{q} = (w_{1,q}, w_{2,q}, \ldots, w_{t,q})$, where t is the total number of index terms for those queries. The vector model evaluates the degree of similarity between q_1 and q_2 as the correlation between these two vectors. In FleaNet, we quantify the correlation by the *cosine of the angle* between q_1 and q_2 [26]. That is

$$sim(q_1, q_2) = \frac{\vec{q_1} \cdot \vec{q_2}}{|\vec{q_1} \times \vec{q_2}|} \tag{1}$$

$$= \frac{\sum_{i=1}^{t} w_{i,q_1} w_{i,q_2}}{\sqrt{\sum_{i=1}^{t} w_{i,q_1}^2}\sqrt{\sum_{i=1}^{t} w_{i,q_2}^2}} \tag{2}$$

$w_{i,\cdot} \geq 0$, $sim(q_1, q_2)$ varies from 0 to 1. Thus, instead of attempting to predict whether a query is relevant or not, the vector model ranks the queries according to their *degree of similarity* to the original query. In FleaNet, the query originator establishes a match sensitivity threshold for $sim(q_1, q_2)$, and a match only occurs when a degree of similarity is above that threshold. Note that more advanced methods, such as the inverse document frequency, can be used to deal with the situation where terms appearing in many queries are not very useful for quantifying the relevance [26]. When a match occurs, a node sends a match message. This match message includes the UID and the sequence number of the matched query. After this, a person can make a final decision by sending a transaction request, which includes the UID and the sequence number of the matching query. Upon receiving this request, the owner of the matching query responds with a transaction response, through which the owner provides notice of his/her decision to accept or reject the offer.

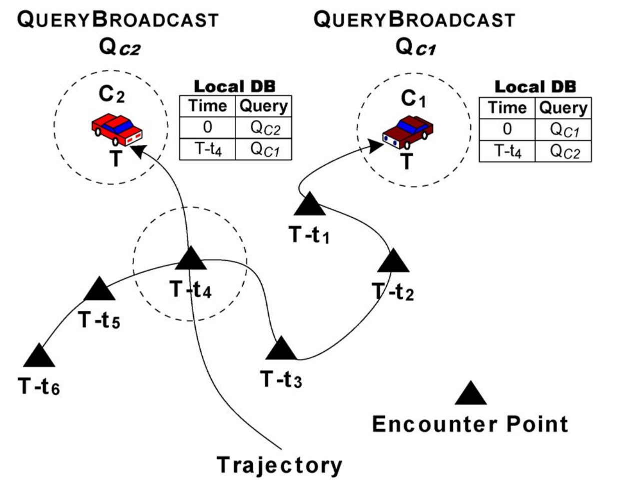


Fig. 1. Query dissemination.

IV. FleaNet PROTOCOL

A. Query Dissemination

Using QUERYBROADCAST, a node (mobile/static) periodically broadcasts its query to its one-hop neighbors by default. We use random jitter to avoid packet collisions due to synchronized broadcasting among neighbors.[2] Each node listens to its neighbors' query broadcasts and stores the received queries in its local database. Owing to the nodes' mobility, queries are opportunistically disseminated through the entire network. Fig. 1 depicts the case of two mobile nodes (i.e., C_1 and C_2) that encounter other nodes over time. A black triangle with a timestamp in the figure represents an encounter between C_1 and other nodes that are within the communication range of C_1 at that time. We denote the query generated by node C_k as Q_{C_k}. Since C_1 and C_2 periodically advertise queries Q_{C_1} and Q_{C_2}, respectively, when they meet each other at time $T - t_4$, they can receive and store each other's query. In other words, C_1

[2]Since collisions could still occur even with jitter due to wireless interference, the broadcast period should carefully be configured to limit the impact. When the average contact duration is t and the period is τ, our dissemination strategy is equivalent to retransmitting the query $r = t/\tau$ times.

Authorized licensed use limited to: Korea Advanced Inst of Science & Tech - KAIST. Downloaded on June 27,2023 at 04:26:16 UTC from IEEE Xplore. Restrictions apply.

carries Q_{C_2} and C_2 carries Q_{C_1} in their local databases after time $T - t_4$.

Queries may slowly propagate due to restricted urban mobility patterns. One of the ways to improve the query dissemination speed is to increase the range of the QUERYBROADCAST to m-hop neighboring nodes, particularly when the network is sufficiently dense to support multihop broadcasting. To reduce redundant packet transmissions, we use probabilistic forwarding, in which a received broadcast message is forwarded with a predefined probability.

As an alternative or a complementary means to m-hop dissemination, for a given query, we can introduce the concept of "proxies," which disseminate queries on behalf of other nodes. In our scheme, only the query originator periodically broadcasts queries to its neighbors by using QUERYBROADCAST. When we deploy k additional proxy nodes, there will be a total of $k + 1$ advertisers that periodically advertise a query to their neighboring nodes in parallel. FleaNet uses the following *proxy query advertiser selection policies*: Random Walk (RW) and Neighbor Split (NS). In RW, to select k proxy advertisers, a query originator first chooses one of its neighboring nodes and sends an ADVERTISERSELECT message with the number of remaining proxy advertisers (initially k). The ID of the selected advertiser is appended in the message (called an advertiser list) to prevent duplicate selection. Similarly, the current ADVERTISERSELECT message holder randomly selects one of its neighboring nodes (that is not on the advertiser list) and passes the message after decreasing the number of remaining advertisers to $k - 1$. This process repeats itself until the number of remaining advertisers becomes zero.

In NS, a query originator equally splits k proxy advertisers among its neighboring nodes. Assuming that there are r neighboring nodes, $r - 1$ nodes will have $\lfloor k/r \rfloor$, and the remaining node will have $k - \lfloor k/r \rfloor \times (r - 1)$. This information is then embedded in the ADVERTISERSELECT message and is broadcast to one's neighboring nodes. Upon receiving this message, each node first checks whether it has previously received the message, and the node decreases its share in the message by one only if it has not received the message before. After this step, we now have r nodes that will split their share of selecting new advertisers, as previously described. The splitting process repeats itself until one's share becomes zero.

B. Match and Transaction Notification

Every incoming query is resolved from the local database. If a node C_{RES} finds a set of matched queries $Q_{\acute{C}_1}, Q_{\acute{C}_2}, \ldots, Q_{\acute{C}_k}$ to an incoming query $Q_{C_{\mathrm{IN}}}$, then this set of matched queries will then be sent using LOCALMATCH to the query originator C_{IN}. The resolver sends a notification of the results only to C_{IN}; it does not notify the originators of the matched queries, i.e., $\acute{C}_1, \acute{C}_2, \ldots, \acute{C}_k$. If the number of matched queries k is larger than N_{MAX} (i.e., the maximum number of matches that C_{IN} wants to receive), then the resolver will randomly pick N_{MAX} matches and send them to C_{IN}. Otherwise, k matched queries are returned to C_{IN}. After this, the query originator C_{IN} updates the N_{MAX} field of the query by subtracting k. If $N_{\max}$ goes below zero, then the query will be discarded. Finally, the user in C_{IN} will choose one of the matched queries and notify the originator of the chosen matched query, for example, $\acute{C}_\ell$, of his/her decision by sending a transaction request message, i.e., TRANXREQ. If $\acute{C}_\ell$ accepts the transaction, then he/she will respond with a transaction reply message, i.e., TRANXREP.

In FleaNet, queries are symmetric: both seller and buyer disseminate their queries to their neighboring nodes, and their queries are diffused through the network via mobility. During the periodic broadcast phase, a match could happen when a buyer finds a neighboring node that is storing a sell query or *vice versa*.

If there are multiple neighboring nodes, then it is nontrivial to control N_{MAX}. Assuming that there are two neighboring nodes, and each node finds more than N_{MAX} matches, they will all return N_{MAX} matches to the query originator. It will receive up to $2 \times N_{\mathrm{MAX}}$ matches from its neighbors, as there could be some overlapping matches. In this case, the node will only select N_{MAX} matches from the list of returned matches, which does not violate the user requirements. The overhead of redundant packet transmission is minimal since the match size is small and multiple queries can be fit into a single packet.

Note that, in vehicular networks, packets are routed according to geographic forwarding schemes. However, a prerequisite of geographic routing is a location service that allows a source node to obtain the location of a destination before data traffic is initiated. In contrast to this approach, FleaNet exploits a hybrid approach that intertwines location service and routing using node encounter history called LER [11]. The underlying concept is that the node encounter history, or the last encounter information, generally provides a rough, yet useful, estimate of the current network topology (or the destination location). Since the speed of multihop packet forwarding is much faster than the mobility of users, routing based on encounter history can quickly direct the packet toward the up-to-date location of the destination. In FleaNet, the last encounter information can be published when nodes periodically send a QUERYBROADCAST. However, the original LER does not support intermittent connectivity, i.e., if a node fails to find any fresher encounter information, then routing fails. We extend LER to cope with intermittent connectivity; when a node fails to find a next hop node, it stores the packet in its buffer and waits for other nodes with more up-to-date encounter information.

V. FleaNet PERFORMANCE ANALYSIS

We develop an analytical model for the notification latency. FleaNet's scalability results can be found in our earlier paper [1].

Let us assume that in an area of $L \times L$ meters square, there are N nodes, each of which communicates with other nodes within a radio range of R meters. For simplicity, we also assume that there exists a single target query of interest (e.g., one buyer and one seller). Given that nodes are moving based on random waypoint (RWP), random direction, or Manhattan mobility, Groenevelt *et al.* [12] show that the intermeeting time between two mobile nodes (or the pairwise intermeeting time) follows an exponential distribution.

Authorized licensed use limited to: Korea Advanced Inst of Science & Tech - KAIST. Downloaded on June 27,2023 at 04:26:16 UTC from IEEE Xplore. Restrictions apply.

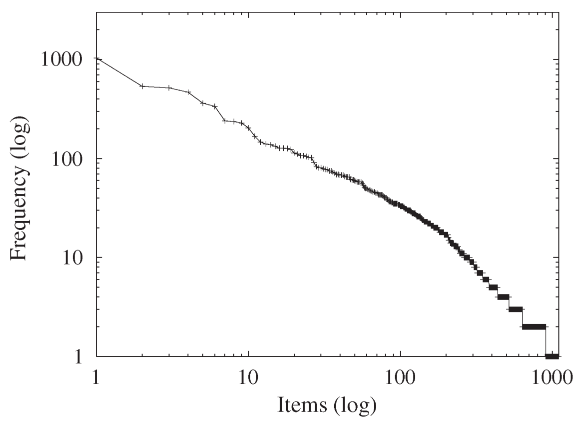


Fig. 2. Posted item popularity distribution (log-log plot).

Let us first characterize the matching latency, which can be defined as the time needed for a random seller to meet one of the nodes with a matching query, i.e., either the buyer itself or any node that encountered the buyer. The same definition holds for a buyer seeking sellers. According to [12], the average latency D can be expressed as

$$\mathbb{E}[D] = \frac{1}{\lambda}\left(\sqrt{\frac{\pi}{2N}} + \mathcal{O}\left(\frac{1}{N}\right)\right). \tag{3}$$

Here, λ is a *pairwise intermeeting rate* and depends on the mobility pattern. It is defined as $\lambda \approx (2\alpha r\mathbb{E}[\mathbb{V}^*])/L^2$, where $\mathbb{E}[\mathbb{V}^*]$ is the average relative speed between two mobile nodes. From (3), we see that the latency is inversely proportional to the number of nodes and to the rate λ (and thus the average relative speed). The fast mobility of an urban vehicular network reduces the average latency. However, as we will see in Section VI-A, the restricted mobility patterns of an urban environment severely decrease the odds of nodes meeting each other (compared with random mobility), thus offsetting the benefits of fast mobility.

In reality, it is likely that there will be many people with the same interest. Fig. 2 shows the popularity distribution of 16 862 postings (make + model) in the vehicle ad section of Craigslist[3] during March 1–7, 2006. The plot approximately follows the power-law distribution. The top 100 items make up 60% of the total advertisements.

Let us analyze the impact of popularity. Assume that there are K users with the same interest, or K query advertisers (e.g., selling an "iPod nano"). We generalize the model in [12] to the case of K query advertisers. Fig. 3 shows a Markov chain where each state denotes the number of users who have at least one advertisement. In state K, an advertiser encounters the remaining $M-K$ users at the rate of $(M-K)\lambda$. Since there are K advertisers, the aggregate rate from state K to $K+1$ is $(M-K)K\lambda$. A match happens when the target node T encounters any of the K advertisers. Thus, the rate from state K to T is $K\lambda$. This way, we can find the rates for other states, as shown in Fig. 3.

We now calculate the average delay. When a match occurs at state K, the average delay is $1/K\lambda$. This happens

[3]http://www.craigslist.org/.

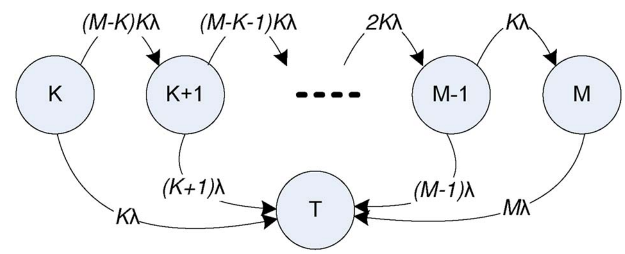


Fig. 3. Markov chain with K users of the same interest. Each state denotes the number of users having at least one advertisement. State T denotes the event where node T finds a match, and we have $M = N - 1$ (i.e., excluding target node T).

with probability

$$P(K \to T) = \frac{K}{(M-K)K + K}. \tag{4}$$

Similarly, when a match happens at state $K+1$ (i.e., the state transition is $K \to K+1 \to T$), the average delay is simply the summation of the delays at state K and the current state $K+1$, i.e.,

$$\frac{1}{K\lambda} + \frac{1}{(K+1)\lambda}. \tag{5}$$

This event happens with probability

$$P(K \to K+1)P(K+1 \to T) = \frac{(M-K)K}{(M-K)K+K}\,\frac{K+1}{(M-K-1)K+K+1}. \tag{6}$$

This way, we can find the probability and average delay of the case where a set of transitions is given as $K \to K+1 \to \cdots \to K+i \to T$. Thus, the average delay is given as

$$\mathbb{E}[D] = \sum_{i=0}^{M-K} E[D_{K\to\cdots\to K+i\to T}]P(K \to \cdots \to K+i \to T). \tag{7}$$

Here, we have

$$E[D_{K\to\cdots\to K+i\to T}] = \sum_{j=0}^{i} \frac{1}{(K+j)\lambda} \tag{8}$$

and

$$\Pr\{K \to \cdots \to K+i \to T\} = \left[\prod_{j=0}^{i-1} \frac{(M-K-j)K}{(M-K-j)K+K+j}\right] \frac{K+i}{(M-K-i)K+K+i}. \tag{9}$$

Note that unlike the case of $K=1$, it is nontrivial to derive the closed-form solution.

To illustrate the impact of popularity, we consider the scenarios with $N = 100$ and two different pairwise intermeeting rates: $\lambda = 1/300$ and $1/500$. We solve the preceding equation and plot the results in Fig. 4. This figure clearly shows that the average delay decreases as the popularity increases.

Authorized licensed use limited to: Korea Advanced Inst of Science & Tech - KAIST. Downloaded on June 27,2023 at 04:26:16 UTC from IEEE Xplore. Restrictions apply.

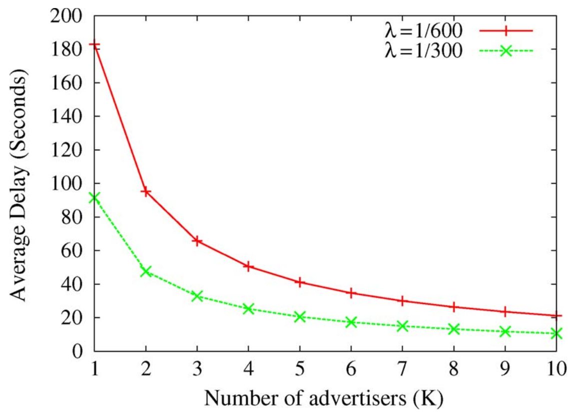


Fig. 4. Average delay with different number of advertisers.

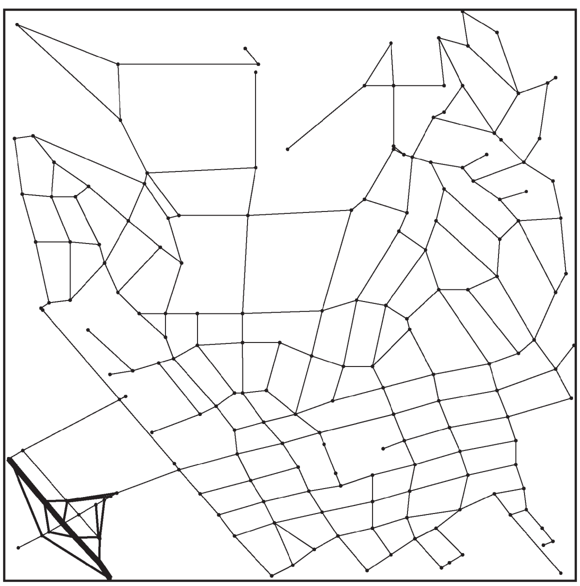


Fig. 5. Westwood area in the vicinity of UCLA.

VI. Evaluation

In this section, we evaluate FleaNet's performance through extensive simulations using Ns-2 [27].

A. Simulation Setup

Each node is considered to have 802.11b connectivity, with a bandwidth of 11 Mb/s and a radio range of 250 m. For the radio propagation model, the two-ray ground reflection model is assumed. For realistic mobility generation, we use VanetMobiSim [28] in combination with a Westwood vicinity street map obtained from the U.S. Census Bureau's TIGER (TGR06037, Los Angeles) (see Fig. 5). Hereinafter, we refer to this mobility as the Westwood Mobility (WM). VanetMobiSim can simulate macromobility and micromobility patterns in urban environments. Macromobility deals with road topology/structure and traffic signs (stop signs, traffic lights, and speed limits), and micromobility models the speed and acceleration of each vehicle. For macromobility, we use a random trip generation module: each vehicle randomly picks starting and ending points in the given map and moves along the shortest path while considering traffic congestion. For micromobility patterns, we use the intelligent driver model to determine the distance between two consecutive vehicles for a given speed. To contrast WM mobility with a random mobility pattern, we also test our protocol using RWP, in which each node finds a random position in the terrain area and travels toward that position with a random speed. In both mobility models, the simulations consider a vehicular network with the number of nodes varying between 100 and 400 in a 2400 m $\times$ 2400 m area. Vehicles travel at an average speed of between 5 and 25 m/s; e.g., if we set WM with $v_{\min} = 0$ and $v_{\max} = 10$, we assume an average speed of 5 m/s.

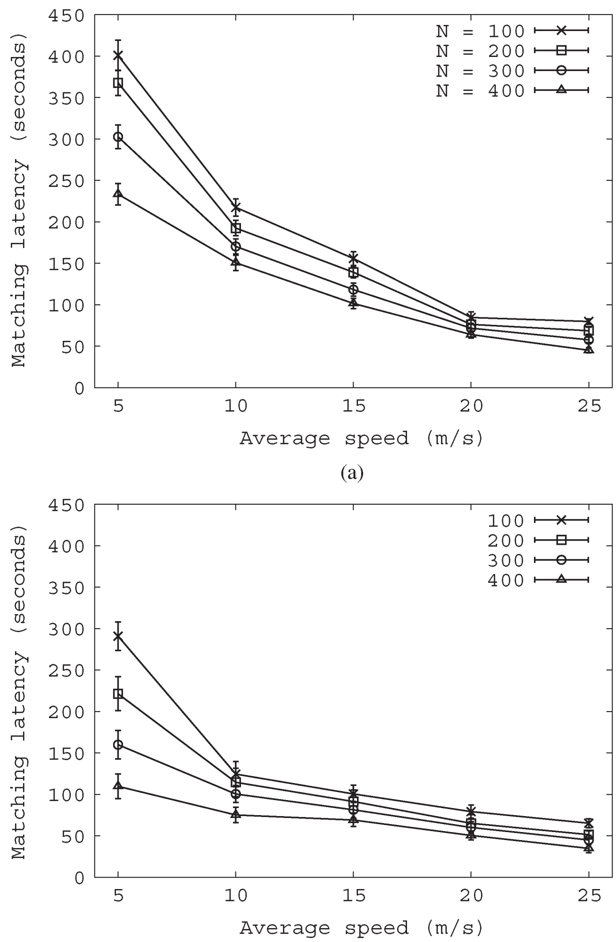
As a metric for the evaluation of system performance, we mainly use the average latency of transaction completion, which comprises the matching latency and the routing latency. For a given query, the *matching latency* measures the time for a node to receive a match after initiating a query. When a node encounters another node that has a matching query (i.e., LOCALMATCH), TRANXREQ is sent to the query originator using the extended LER (e.g., from a seller to a buyer or *vice versa*), and the *routing latency* measures the time needed to deliver the notification to the destination. In our simulations, we assume that a user immediately sends a transaction request to the other party. Thus, the latency of transaction completion is the sum of the matching latency and the routing latency. These latencies are dependent on many parameters, specifically density/speed, query popularity, and mobility. This section focuses on our investigation of the impacts of such parameters.

For each mobility scenario (density and speed), we generate 30 mobility scenario files using VanetMobiSim. For each protocol configuration, we select a random pair of nodes from a mobility scenario and measure the latency. We repeat this five times for each mobility scenario file; thus, we run 150 simulations for each protocol configuration. Unless otherwise mentioned, we report the average latency with a 95% confidence interval.

B. Impact of Mobility

The analysis discussed in Section V shows that the average latency is a function of node density and speed. In this section, we study the effect of these mobility parameters on the transaction completion latency using both WM and RWP mobility models. Fig. 6 shows the latency as a function of density and speed in each node density with a buyer. For each node density, a node is randomly selected as a seller to evaluate how node density affects the matching latency depending only on the speed. In the figure, the x- and y-axes represent the average vehicle speed and the transaction completion latency, respectively. From this figure, we see that both density and speed of vehicles are important factors in determining the latency. As the average speed or the number of nodes increases, the matching latency decreases. This is not surprising, since as the average speed or number of nodes increases, a node has a higher chance of meeting other nodes, which translates into more rapid dissemination of the query and also a greater chance of finding a match.

Authorized licensed use limited to: Korea Advanced Inst of Science & Tech - KAIST. Downloaded on June 27,2023 at 04:26:16 UTC from IEEE Xplore. Restrictions apply.



(a) (b)

Fig. 6. Average latency as a function of speed. (a) WM. (b) RWP.

This trend in our simulation results is consistent with the analytic results given in Section V, indicating that the average speed is the dominating factor in the matching latency. We have shown that the latency is inversely proportional to the average speed and to the square root of the number of nodes [see (3)]. In addition, the advantage of increasing the speed exists even with different node densities. In other words, increasing the speed is equally beneficial in all node density cases. We can clearly see that the total latency improves by 80.08% for 100 nodes and by 80.67% for 400 nodes, which are about the same.

We also observe that the average total latency is much lower in RWP than in WM. For example, the latency at the lowest speed $(s = 5 \text{ m/s})$ in RWP is 41.8% lower than the latency in WM. On the basis of this observation, we conclude that WM can be regarded as a worst-case scenario, and therefore, we use WM mobility for the rest of our experiments to evaluate the proposed protocol in a more realistic urban environment.

C. Impact of Query Popularity

The overall latency is heavily dependent on the query popularity. We can easily see that if many people are interested in a specific item, then notification will occur quickly. To show this, we plot the latency as a function of popularity in a single-buyer k-seller case, as shown in Fig. 7. We increase the ratio

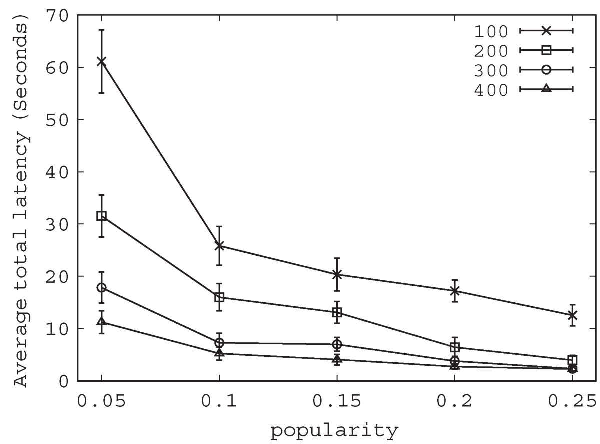


Fig. 7. Impact of delay with respect to popularity.

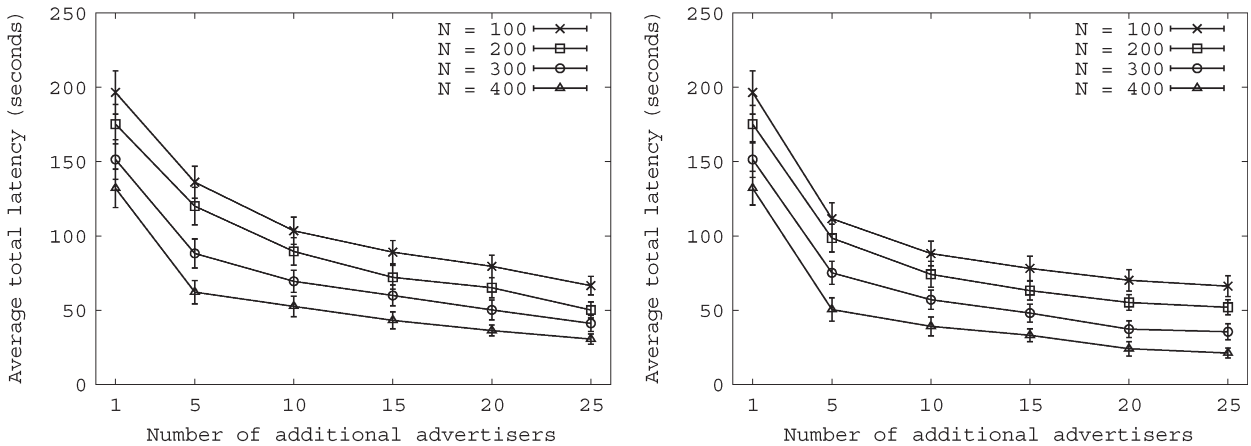
of sellers in the network from 5% to 25% in increments of 5%, as shown on the x-axis, and vary the number of nodes (i.e., 100–400 nodes). Given a single buyer, $x\%$ sellers are randomly selected, and the latency for transaction notification is measured. We limit ourselves to the single-buyer case to clearly understand the impact of query popularity on latency. The figure clearly confirms our intuition about the impact of popularity on the latency: As the popularity increases, the latency decreases. The latency improvement rate decreases in a relative way as the percentage of sellers increases. For example, at a node density of 100, the delay improves by 57.7% from 5% sellers to 10% sellers. However, the improvement rate gradually decreases to 26.2% when the number of sellers reaches 25% of the total node population. The trend is also observed in the 400-node scenario, with the rate of performance deteriorating from 53.8% to 17.6%. The figure also shows that the total latency improvement is 79.5% from 5% sellers to 25% sellers for a node density of 100. The performance-increasing rate remains consistent for other node densities.

D. Proxy Query Advertiser Selection Strategies

A query originator can expedite query dissemination by using m-hop dissemination and/or employing k proxy concurrent advertisers. We increase the number of proxy advertisers that advertise a query on a seller's behalf using different advertiser selection strategies, namely, RW and NS. Recall from Section IV that, in RW, the *AdvertiserSelect* message randomly hops from one node to another to select k proxy advertisers, and in NS, the value k is equally divided among one's neighboring nodes, and advertiser selection occurs in parallel. In our simulation, latency improvement is observed when the number of proxy advertisers is increased from 1 to 25. To understand the impact of increasing the number of proxy advertisers alone, we use a single-seller/buyer scenario. In the simulations, nodes move at an average speed of 10 m/s, and queries are disseminated to either single-hop or three-hop neighbors.

RW: Fig. 8 shows what happens when we increase the number of proxy advertisers k from 1 to 25. As the number of proxy advertisers increases, the performance of the system improves. The latency improvement peaks when we have $k = 5$

Authorized licensed use limited to: Korea Advanced Inst of Science & Tech - KAIST. Downloaded on June 27,2023 at 04:26:16 UTC from IEEE Xplore. Restrictions apply.



(a) (b)

Fig. 8. Impact of delay with respect to the number of additional advertisers. (a) RW. (b) NS.

and tapers off after $k = 15$. For example, in the 100-node scenario, increasing the number of proxy advertisers from 1 to 5 results in a 38.78% latency improvement, whereas that from 10 to 15 improves the delay only in 11.45%. This trend becomes obvious as the network size increases from 100 to 400 nodes. In the 400-node scenario, latency improves by 58.88% when the number of additional advertisers increases to 5, but only a 15% improvement is obtained when the number of proxy advertisers increases from 10 to 15. The latency improvement, however, becomes larger as the number of proxy advertisers grows from 15 to 25 for node densities of 100 and 200. On the other hand, in the 300- and 400-node cases, no significant benefit is observed by increasing the number of proxy advertisers to more than 15. Since there is only one seller in each simulation setting, we expect a steady delay improvement as the number of proxy advertisers increases to over 25. Although the overall latency improvement obtained from increasing the number of proxy advertisers in RW is consistent and steady, the network size often affects the delay. When the vehicles are dense (e.g., $N = 300$ and $N = 400$) in the network, adding a few proxy advertisers provides a significant delay improvement, whereas the benefit is less obvious in low-density scenarios (e.g., $N = 100$ and $N = 200$). In the extended version of this paper [24], we also report the results with three-hop dissemination, showing that multi-hop dissemination significantly improves the latency, yet the degree of improvement decreases as the node density increases.

NS: We use the same simulation setting as in RW. Similar to RW, the performance of the system improves as the number of proxy advertisers increases. However, the latency improvement tapers off after it reaches a certain number. For example, we obtain about a 52.74% improvement by employing five proxy advertisers, whereas the performance improvement reduces to 4.76% when the number of proxy advertisers increases from 20 to 25. Furthermore, the performance improvement grows with node density. For example, the latency improves by 48.69% when we increase the number of proxy advertisers to five in the 100-node scenario, and it improves to 59.9% in the 400-node scenario.

The differences in latency improvement between RW and NS can be observed in Fig. 8. Although both RW and NS improve the latency, we see that NS is slightly better than RW. The better performance of NS with a small k can be attributed to its query dissemination strategy. Compared with RW, NS can quickly increase the number of proxy advertisers, as it can logarithmically decrease the size using the neighbor size as the base, whereas it takes linear time for RW to increase the number of additional advertisers.

Note that using proxy advertisers can be interpreted as an increase in query popularity. In the previous section, we observe that the total latency is directly influenced by the query popularity. Taking the result as a reference for delay improvement, we compare the performance improvement by increasing the number of proxy advertisers as opposed to increasing the query popularity (i.e., by increasing the number of sellers) in the network. Considering the different query popularity conditions of the simulations, we compare the total latency with the same number of query disseminators (e.g., multiple sellers versus multiple advertisers). Our results show that increasing the number of sellers results in better performance than increasing the number of additional advertisers. The latency in the case of ten sellers is 3.5 times lower than that in the case of $k = 10$ proxy advertisers in the 100-node scenario. This gap increases with the number of sellers or the number of proxy advertisers. For example, with 25 sellers, the delay is 4.47 times lower. This is simply because of the extra delay involved in finding proxy advertisers. In other words, queries can be disseminated by sellers without any "start-up" delay, whereas queries cannot be spread until proxy advertisers are chosen.

E. Routing Delay Analysis

Fig. 9 shows how node density/speed influences routing delay. Recall that routing is carried out by extended LER. As shown in this figure, the routing latency decreases as the node density increases. In the 100-node scenario, the average routing delay is 3.52 s. When the number of nodes is 300, the routing delay becomes 1.36 s, which is 2.16 s less than that in the 100-node scenario. In addition, the routing delay decreases as the average speed increases. For instance, the routing delay gradually reduces from 5.79 to 2.18 s as the average speed

Authorized licensed use limited to: Korea Advanced Inst of Science & Tech - KAIST. Downloaded on June 27,2023 at 04:26:16 UTC from IEEE Xplore. Restrictions apply.

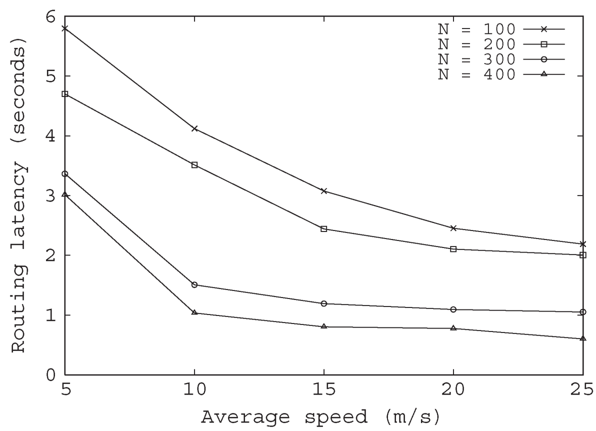


Fig. 9. Routing latency of delay tolerant LER.

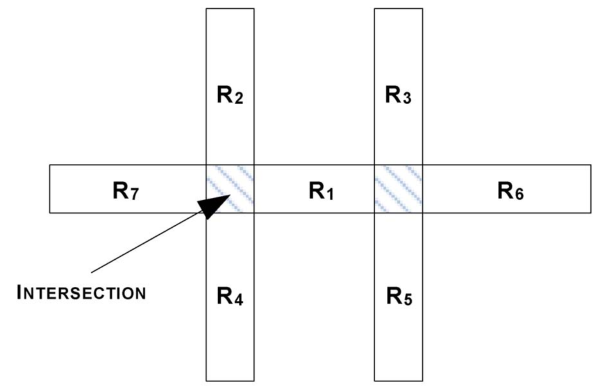
increases from 5 to 25 m/s in the 100-node scenario. Comparing the results in the earlier section, the routing delay is an order of magnitude smaller than the matching latency. Note that the packet delivery ratio used is 100% in all the scenarios.

F. Impact of Location

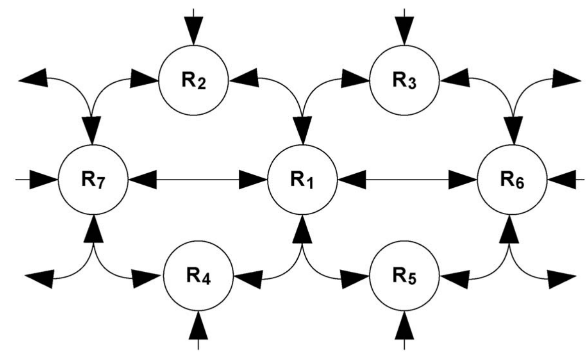
Nodes can be static, e.g., Adstation in FleaNet. In this section, we show how a stationary node affects the average notification latency, i.e., the impact of its location. We can consider that since the average relative speed between two nodes is higher when both are moving, a mobile node has a higher chance of meeting more nodes than a stationary node, which results in faster query dissemination. The restricted mobility of the WM model worsens the situation for a stationary node, because nodes tend to stay longer in an area where roads are densely clustered together.

This can be better understood by modeling the travel of a vehicle through the urban grid as a Markov chain, where each state in the Markov chain represents a vehicle occupying a given road segment (see Fig. 10). The vehicle moves from one segment to another. The transition probability P_{ij} from state R_i to R_j can be calculated from the mobility trace as follows. Given that, for a given period of time τ, we observe a total of γ nodes leaving segment R_i, the transition probability P_{ij} is defined as the fraction of nodes making the transition to segment R_j. From this Markov chain, we can find the stationary distribution (or the spatial node distribution) π, where π_i is the probability that a node is in road segment R_i.

In reality, the length of a road segment can vary. Cars will traverse short segments faster than long segments, and thus, the "residence" time (or sojourn time) of each state depends on the length of a road segment. Unlike the preceding discrete-time Markov chain, the transition in this case is determined by both the current state and the length of time that a node has spent in the current state. This can be modeled using a semi-Markov process $\{Z(t) : t \geq 0\}$ describing the state at time t. Its embedded Markov chain is the same as the preceding Markov chain (see Fig. 10). The probability of being at state R_i is proportional to $\pi_i E[T_i]$, where π_i is the stationary distribution calculated from the embedded Markov chain. In other words, the longer the length of a road segment, the higher the probability that



(a)



(b)

Fig. 10. Finding the spatial node distribution using a Markov chain. R_i denotes the ith road segment. Each state R_i in the Markov chain corresponds to road segment R_i. (a) Road layout. (b) Markov chain.

a node is observed at that road segment. Thus, the occupancy distribution at state R_i is given as

$$p_i = \frac{\pi_i E[T_i]}{\sum_j \pi_j E[T_j]} \tag{10}$$

where $E[T_j]$ denotes the mean time that a node spends at road segment j for all j.

For ease of illustration, we assume that each segment has the same number of forks and that the average length of a road segment is fixed (i.e., $E[T_i] = E[T_j]$ for all $i \neq j$). Assuming that a vehicle at the end of a segment randomly chooses its direction among k forks, i.e., the current segment is connected to k different road segments, the transition probability from road segment i to j is simply $P_{ij} = 1/k$. Then, the stationary probability that a vehicle stays in road segment R_i is $p_i = 1/N$, where N is the number of road segments. Now, the probability that a vehicle stays in an area with "m" segments is simply m/N. Without loss of generality, the thicker the concentration of segments in the area (e.g., historic downtown), the higher the probability that a vehicle stays in that area. Moreover, when the length of a road segment is not fixed (i.e., $E[T_i] \neq E[T_j]$ for all $i \neq j$), this probability also depends on the total length of the road segments; thus, the longer the total distance, the higher the probability.

To clearly understand how the position of a stationary node affects the latency, we use a scenario based on the WM model, with 100 nodes moving at an average speed of 25 m/s without

Authorized licensed use limited to: Korea Advanced Inst of Science & Tech - KAIST. Downloaded on June 27,2023 at 04:26:16 UTC from IEEE Xplore. Restrictions apply.

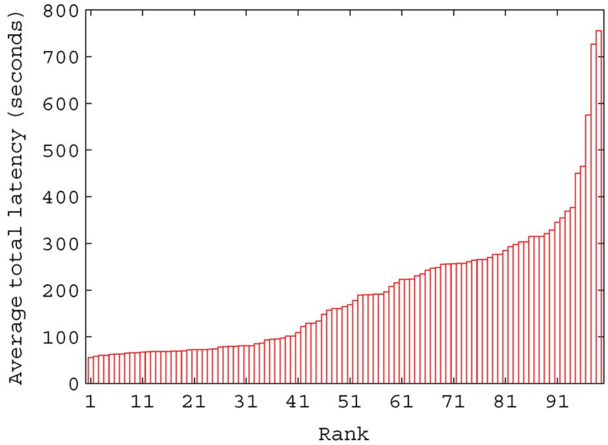


Fig. 11. Impact of location on total average latency.

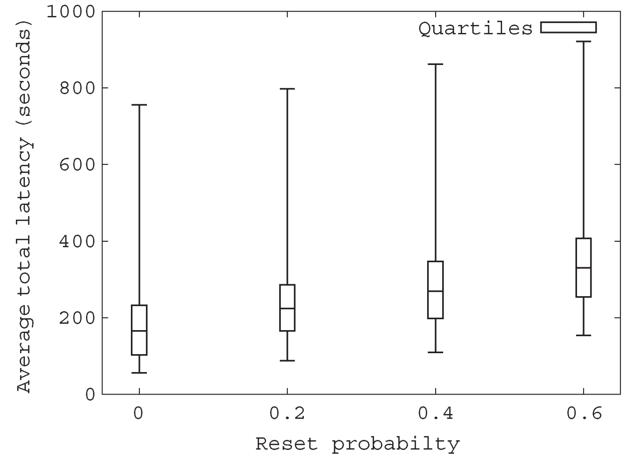


Fig. 12. Impact of location on total average latency with churn rate.

churning and with a static buyer. We perform 100 trials with random node placements, where one of the nodes is randomly chosen as a seller. Queries are broadcast within three-hop neighbors, and no proxy query advertiser is applied in the scenario. In Fig. 11, the total latency distribution is presented in ascending order with the rank of the delay on the x-axis. The ith index of the x-axis represents a node with the ith largest latency. For example, the smallest latency (rank 1) is 55.4 s, whereas the largest latency (rank 100) is 755.9 s. As can be seen, the largest latency is 13.6 times higher than the smallest latency. Thus, the latency heavily depends on the location. By examining the location of the static node in each experiment, we found that the first 10% of the stationary nodes (Rank 1 to Rank 10) are located on the southwest side of the map, where the entrance to the highway is located. In fact, there is heavy traffic around the highway entrance and the loop to the highway in the scenario. Therefore, it is likely that the stationary node in the area has the highest probability of encounters with other nodes. On the other hand, the last 10% of the stationary nodes (Rank 91-Rank 100) are placed either at the border of the map or in the northwest area, where the roads are sparse.

G. Impact of Churning

To simulate a more realistic mobile environment, we measure the latency of stationary nodes in the presence of churning, where nodes move out of the network area (or out of the region of interest). When a node reaches the border area of the network (width of 100 m), we reset the node's buffer to probability p, with a higher probability representing a higher churning rate. We vary the reset probability p from 0 to 0.2, in increments of 0.05, to show the impact of churning. Fig. 12 shows the delay distribution using a box-and-whisker plot. As shown in the graph, the median value increases with the reset probability. The median increases from 165.34 to 319.79 s as the reset probability increases to 0.6. Similarly, the maximum delay with a 0.6 reset probability is about 196 s longer than that without churning. Manual examination of the results reveals that some stationary nodes located at the border area have a noticeable delay increment of 22.6%. In an earlier section, we showed that stationary nodes with high delay are placed either at the border of the map or in the northwest area, where roads are sparse. Nodes visit these sites less frequently than the other sites due to road layout. Moreover, those nodes traveling near the border area are more likely to suffer from churning. Thus, as churning increases, stationary nodes near the border area experience a much higher delay.

VII. Conclusion

In this paper, we have proposed a novel concept of a virtual marketplace called FleaNet. Using FleaNet, mobile and stationary users can carry out buy/sell transactions (or other activities for matching common interests) on a vehicular network. This concept is implemented through the FleaNet architecture, which defines the details of FleaNet components, such as FleaNet nodes (e.g., Adstations) and query formats. Query dissemination and resolution are carried out through a FleaNet protocol suite, which is scalable to thousands of nodes and nonintrusive with regard to other existing services. We have evaluated the proposed protocols by means of mathematical analyses and simulations. In particular, our simulation results showed that, in most cases, a random query could be resolved within a tolerable amount of time.

References

[1] U. Lee, J.-S. Park, E. Amir, and M. Gerla, "FleaNet: A virtual market place on vehicular networks," in *Proc. V2VCOM*, San Jose, CA, Jul. 2006.
[2] D. Jiang, V. Taliwal, A. Meier, W. Holfelder, and R. Herrtwich, "Design of 5.9 GHz DSRC-based vehicular safety communication," *IEEE Wireless Commun.*, vol. 13, no. 5, pp. 36–43, Oct. 2006.
[3] IEEE P802.11—Task Group p. [Online]. Available: http://grouper.ieee.org/groups/802/11/Reports/tgp_update.htm
[4] H. Rheingold, *Smart Mobs: The Next Social Revolution*. New York: Basic, 2003.
[5] M. Motani, V. Srinivasan, and P. S. Nuggehalli, "PeopleNet: Engineering a wireless virtual social network," in *Proc. MobiCom*, Aug. 2005, pp. 243–257.
[6] A. Vahdat and D. Becker, "Epidemic routing for partially-connected ad hoc networks," Duke Univ., Durham, NC, Tech. Rep. CS-200006, Apr. 2000.
[7] P. Juang, H. Oki, Y. Wang, M. Martonosi, L.-S. Peh, and D. Rubenstein, "Energy-efficient computing for wildlife tracking: Design tradeoffs and early experiences with ZebraNet," in *Proc. ACM ASPLOS-X*, San Jose, CA, Oct. 2002.
[8] T. Small and Z. J. Haas, "The shared wireless infostation model: A new ad hoc networking paradigm (or where there is a whale, there is a way)," in *Proc. ACM MOBIHOC*, Annapolis, MD, Jun. 2003, pp. 233–244.

Authorized licensed use limited to: Korea Advanced Inst of Science & Tech - KAIST. Downloaded on June 27,2023 at 04:26:16 UTC from IEEE Xplore. Restrictions apply.

[9] A. Datta, "Autonomous gossiping: A self-organizing epidemic algorithm for selective information dissemination in wireless mobile ad hoc networks," in *Proc. ICDCS Doctoral Symp.*, Providence, RI, May 2003.

[10] B. Xu, A. Ouksel, and O. Wolfson, "Opportunistic resource exchange in inter-vehicle ad-hoc networks," in *Proc. IEEE/ACM MDM*, Berkeley, CA, Jan. 2004.

[11] M. Grossglauser and M. Vetterli, "Locating nodes with EASE: Mobility diffusion of last encounters in ad hoc networks," in *Proc. IEEE INFOCOM*, San Francisco, CA, Mar./Apr. 2003, pp. 1954–1964.

[12] R. Groenevelt, P. Nain, and G. Koole, "Message delay in mobile ad hoc networks," in *Proc. Performance*, Juan-les-Pins, France, Oct. 2005.

[13] A. Nandan, S. Das, S. Tewari, M. Gerla, and L. Klienrock, "AdTorrent: Delivering location cognizant advertisements to car networks," in *Proc. WONS*, Les Menuires, France, Jan. 2006.

[14] Y. Zhang, J. Zhao, and G. Cao, "Roadcast: A popularity aware content sharing scheme in VANETs," in *Proc. ICDCS*, Montréal, QC, Canada, Jun. 2009, pp. 223–230.

[15] S. Lee, G. Pan, J. Park, M. Gerla, and S. Lu, "Secure incentives for commercial ad dissemination in vehicular networks," in *Proc. ACM MOBIHOC*, 2007, pp. 150–159.

[16] J. LeBrun, C.-N. Chuah, D. Ghosal, and M. Zhang, "Knowledge-based opportunistic forwarding in vehicular wireless ad hoc networks," in *Proc. IEEE VTC*, Dallas, TX, Sep. 2005, pp. 2289–2293.

[17] H. Wu, R. Fujimoto, R. Guensler, and M. Hunter, "MDDV: A mobility-centric data dissemination algorithm for vehicular networks," in *Proc. VANET*, Philadelphia, PA, Oct. 2004, pp. 47–56.

[18] Y. Yu, G.-H. Lu, and Z.-L. Zhang, "Enhancing location service scalability with HIGH-GRADE," in *Proc. IEEE MASS*, Fort Lauderdale, FL, Oct. 2004, pp. 164–173.

[19] E. Yoneki and J. Bacon, "Distributed multicast grouping for publish/subscribe over mobile ad hoc networks," in *Proc. WCNC*, New Orleans, LA, Mar. 2005, pp. 2293–2299.

[20] Y. Huang and H. Garcia-Molina, "Publish/subscribe tree construction in wireless ad-hoc networks," in *Proc. MDM*, Melbourne, Australia, Jan. 2003, pp. 122–140.

[21] P. Costa and G. P. Picco, "Semi-probabilistic content-based publish–subscribe," in *Proc. ICDCS*, Columbus, OH, Jun. 2005, pp. 575–585.

[22] E. Yoneki, P. Hui, S. Chan, and J. Crowcroft, "A socio-aware overlay for publish/subscribe communication in delay tolerant networks," in *Proc. MSWIM*, Crete Island, Greece, Oct. 2007, pp. 225–234.

[23] D. Frey and G.-C. Roman, "Context-aware publish subscribe in mobile ad hoc networks," Washington Univ., St. Louis, MO, 2007.

[24] U. Lee, J. Lee, J.-S. Park, and M. Gerla, "Fleanet: A virtual market place on vehicular networks," UCLA, Tech. Rep., Sep. 2009.

[25] M. D. Dikaiakos, S. Iqbal, T. Nadeem, and L. Iftode, "VITP: An information transfer protocol for vehicular computing," in *Proc. ACM VANET*, Cologne, Germany, Sep. 2005, pp. 30–39.

[26] R. Baeza-Yates, *Modern Information Retrieval*. Reading, MA: Addison-Wesley, 2001.

[27] ns-2—(The Network Simulator). [Online]. Available: http://www.isi.edu/nsnam/ns/

[28] J. Härri, M. Fiore, F. Fethi, and C. Bonnet, "VanetMobiSim: Generating realistic mobility patterns for VANETs," in *Proc. VANET*, Los Angeles, CA, Sep. 2006, pp. 96–97.

Uichin Lee (M'09) received the B.S. degree in computer engineering from Chonbuk National University, Jeonju, Korea, in 2001, the M.S. degree in computer science from the Korea Advanced Institute of Science and Technology, Daejeon, Korea, in 2003, and the Ph.D. degree in computer science from the University of California at Los Angeles, in 2008.

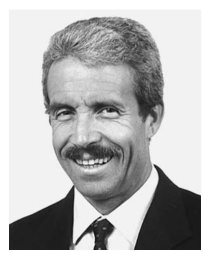
He is currently a Member of Technical Staff with Bell Labs, Alcatel-Lucent, Holmdel, NJ. His research interests include distributed systems, mobile wireless networking systems, and performance modeling/evaluation.

Jiyeon Lee received the B.S. degree in computer engineering from Oregon State University, Corvallis, in 2007 and the M.S. degree in computer science from the University of California at Los Angeles, in 2009.

She is currently a Software Engineer with the Server Access Virtualization Department, Cisco Systems, Inc., San Diego, CA.

Joon-Sang Park (M'06) received the M.S. degree in computer science from the University of Southern California, Los Angeles, in 2001 and the Ph.D. degree in computer science from the University of California at Los Angeles (UCLA), in 2006.

From 2006 to 2007, he was a Postdoctoral Researcher with UCLA. He is currently with the Department of Computer Engineering, Hongik University, Seoul, Korea. His research interests include mobile ad hoc and sensor networks, multiple-input–multiple-output, and network coding.



Mario Gerla (F'02) received the Engineering degree from Politecnico di Milano, Milan, Italy, and the Ph.D. degree from the University of California at Los Angeles (UCLA).

From 1973 to 1976, he was with Network Analysis Corporation, Glen Cove, NY, where he helped transfer Advanced Research Projects Agency Network (ARPANET) technology to government and commercial networks. In 1976, he joined the UCLA Faculty. He is currently a Professor in computer science with UCLA, where he was part of the team that developed the early ARPANET protocols under the guidance of Prof. L. Kleinrock. He has also designed and implemented network protocols, including ad hoc wireless clustering, multicast On-Demand Multicast Routing Protocol (ODMRP), CodeCast, and Internet transport (TCP Westwood). He has lead the $12 million six-year Office of Naval Research MINUTEMAN project, designing next-generation scalable airborne Internet for tactical and homeland defense scenarios. He is now leading two advanced wireless network projects under Army and IBM funding. His team is developing a vehicular testbed for safe navigation, urban sensing, and intelligent transport. A parallel research activity explores personal communications for cooperative networked medical monitoring (see http://www.cs.ucla.edu/NRL for recent publications).

Authorized licensed use limited to: Korea Advanced Inst of Science & Tech - KAIST. Downloaded on June 27,2023 at 04:26:16 UTC from IEEE Xplore. Restrictions apply.