On “Shooting” a Moving Vehicle with Data Flows

Seokhoon Yoon, Hung Q. Ngo, and Chunming Qiao

Computer Science and Engineering
State University of New York at Buffalo
Email: {syoon4,hungngo,qiao}@cse.buffalo.edu

***Abstract*— This paper proposes an ITS system that uses not only integrated cellular and ad hoc relaying technologies, but also users’ mobility profiles. It focuses on a subsystem where cellular bandwidth is used mostly for control signals, and an ad hoc distribution network (ADN) is used for file downloading and video streaming. The paper formulates an optimization problem with the objective being to maximize the amount of data to be delivered to a moving vehicles via the ADN, and thereby minimizing the usage of the costly cellular bandwidth for data transfer. Three approaches based on non-linear and linear programs are described and compared, and simulation results indicate that a satisfying performance for file download applications can be achieved.**

I. Introduction

We envision an architecture for ITS in metropolitan areas that integrates the cellular and ad hoc relaying technologies. The proposed architecture, called iCAR-MOPA [1], will provide various services to vehicles, which include file downloading, stored or real-time video streaming, and customized traffic alert as well as adaptive navigation.

In this work, we focus on a subsystem for the file download services, but the concepts and techniques discussed can be adapted for other types of applications such as video streaming. In addition to exploiting the integrated cellular and ad hoc relaying technologies, another unique feature of the proposed architecture is that it incorporates knowledge of mobility profiles of vehicles. Hereafter, we refer to the subsystem under consideration as *mobility profile aided downloading systems* or iCAR-MOPADS.

In the envisioned iCAR-MOPADS (and iCAR-MOPA), vehicles are equipped with both cellular (e.g., 3G WCDMA) and ad hoc relaying (e.g., 802.11) interfaces as well as GPS. One or more server stations (SS) at distributed locations have connections to a high-bandwidth and relatively low-cost-per-bit backbone (e.g., a wireline network). The cellular network is also integrated with the backbone so the files on the SS are accessible through the cellular channel.

In addition, many access points based on low-cost wireless technologies (e.g. 802.11) equipped with storage, which are similar to the infostations in [1], [2], are placed at strategic locations such as major intersections throughout the metropolitan areas. Unlike most of the proposed system, these APs are not directly connected to the same wireline backbone as the SS to avoid the expected high cost. Instead, some of the APs can be connected with each other (and also with the SS) via a wireless mesh network. Other APs may form a part of an intermittently connected mobile ad hoc network (ICMAN [3]) where data from one AP (or SS) to another AP must be carried by vehicles moving from the former toward the latter. Furthermore, with mobility profile information of the subscriber, a vehicle can also download data for another vehicle by acting as a mobile ”AP” delivering the data at their encounter. In either case, delivering data through such a network, hereafter referred to as the *ad hoc distribution network* or ADN, is expected to provide a higher bandwidth at a lower cost per bit than using a cellular channels. Fig. 1 shows an example of an ADN where a vehicle downloads data via and/or the cellular channel.

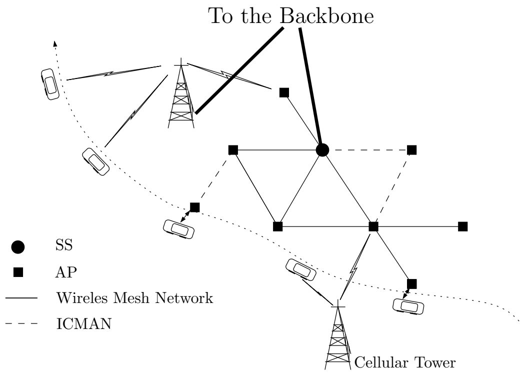


Fig. 1. An Ad hoc Distribution Network

A basic idea behind the envisioned iCAR-MOPADS is that vehicles (or in general other mobiles) can download a file stored on the SS or cached at some APs when the vehicle passes by an AP using the ad hoc (e.g., 802.11 based) interface, instead of the cellular interface. Vehicles can also download a file using their cellular interfaces but the cost per bit using a cellular channel is expected to be much higher and thus the usage of the cellular channel for file downloading is to be minimized. Nevertheless, the APs and vehicles will use cellular channels for the transmission and reception of control signals which require a relatively low bandwidth.

When a iCAR-MOPADS service subscriber starts a trip, he/she may request via a cellular channel for a file (a new piece of software or a video for example). Ideally, iCAR-MOPADS can learn, based on direct input or mobility profile of the user,

[1] where iCAR and MOPA stand for Integrated Cellular and Ad Hoc Relaying Systems and Mobility Profiled Assistance respectively

the route to be taken by the user and calculate its estimated time to arrival (ETA) at each and every major intersection along the route[2]. iCAR-MOPADS determines the appropriate amount of data to be "pushed" from one or more SS to the APs at those intersections so that the vehicle can download the data when it arrives (and passes by).

For a given AP i and a vehicle that passes by the AP, the SS determines the optimal amount of data, denoted by x_i, to be pushed so that the expected amount of data downloaded by the vehicle is maximized. The expected amount of data that the vehicle could download at the AP depends on the probability of the vehicle passing by the corresponding intersection, the expected duration of "AP association time" and the expected download rate of the AP. Since the data has to arrive at an AP before the vehicle does, x_i also depends on the status of the ADN, including its link transmission speed and latency, current queuing delays and information about other vehicles (including their requests for file download and routes).

If the vehicle fails to download all the data of a file from an AP originally planned by iCAR-MOPADS, the cellular channel bandwidth (in addition to that devoted to control signals) is used to deliver the rest of the data as needed.

To simplify our presentation in this paper, we will focus on the problem of delivering a maximum amount of data to a vehicle via the ADN (wireless mesh networks of APs and ICMAN), thereby minimizing the cost associated with the use of cellular channel for data. Moreover, we will only consider a simple routing strategy in ADN where SS uses a fixed, pre-computed path to route data to each AP, obviously leaving many open problems related to routing in a wireless mesh network or via relaying by other vehicles (as needed in an ICMAN network) for future works. The simplified version of the problem is thus essentially analogous to the problem of shooting a moving vehicle to maximize the number of "bullets" hitting it. We believe that this is the first time that such a problem is being investigated.

The rest of the paper is organized as follows. We first formally describe the problem in Section II. We then propose three approaches to formulate the optimization problem in Section III. Section IV presents our simulation results, and Sections V and VI discuss related studies and concludes the paper respectively.

II. Problem Description

Consider a vehicle whose route includes APs $v_1, v_2, \ldots, v_m$ in that order which can be estimated from the mobility profile of the vehicle. Assume that, at time t=0, the vehicle makes a request for a file of size F bytes stored at SS via a cellular channel. Let T_i be the (estimated) time that the vehicle will be at v_i. Note that in practice the number of bytes the vehicle can download from an AP varies, and has a certain upper-bound say b_i. The basic idea of the envisioned iCAR-MOPADS is to partition the file into several parts, $x_i <= b_i$, where $i = 1, 2 \cdots, m$ and $\sum_{i=1}^{m} x_i \leq F$, and try to push x_i bytes from SS to v_i before the vehicle arrives at T_i.

Problem 1 (Shooting a Moving Vehicle – SMV)**.** *Find file segmentation and transmission schedule at SS that maximizes the expected amount of data that a vehicle will receive on its trip.*

Let X_i be the random variable denoting the time (starting from 0) at which AP v_i receives the x_i bytes sent from SS. Note that X_i not only is a function of $x_1, \cdots, x_m$, but also depends on the order in which SS sends these messages out (since the routes these messages take may overlap). Note also that depending on various practical conditions (e.g. traffic light, driving speed in the AP coverage zone including the likelyhood that the vehicle will stop at a red-light), a vehicle may only be able to download data available for it from v_i with probability p_i. Let B (a random variable) be the total number of bytes the vehicle receives on this trip, then our objective function is

$$\mathrm{E}[B] = \sum_{i=1}^{m} x_i p_i \,\mathrm{Prob}[X_i \leq T_i], \qquad (1)$$

subject to $0 \leq x_i \leq b_i$ for all i, and $\sum_{i=1}^{m} x_i \leq F$.

III. Approaches

We present three approaches for formulating the aforementioned SMV problem with non-linear and linear programs (NLP and LP). To simplify the problem, we assume that there is only one server station SS which has the data requested by users and routes data to each of the n APs using a fixed path. The union of the routing paths forms a directed tree T rooted at SS. In addition, in all three approaches, we will assume that the transmissions of messages from SS are in the order $x_1, \ldots, x_m$ (though this assumption may not lead to optimal solutions).

Approach 1. Solve (1) directly with an NLP optimization software. In order to realize this approach, an explicit formula for Prob$[X_i \leq T_i]$ has to be derived.

As a first approximation, we assume that each AP operates as an M/M/1 queue whose service time is exponential with mean $1/\mu$. In addition, data generated at SS in response to downloading requests from vehicles is Poissonian with rate λ. At equilibrium, we assume that a file segment coming into a node v has equal probabilities of going out on any outgoing edge of v. The entire network is thus an open Jackson network. The data arrival rate on each edge is thus Poissonian and can be recursively computed using flow conservation equations. In particular, if the incoming data to node v (through edge e_v) is Poissonian with rate λ_v, then the flow through each outgoing edge of v is also Poissonian with rate $\lambda_v/\text{out-deg}(v)$.

Let P_i be the path in the routing tree from SS to v_i. Let k_i be P_i's length. Denote the nodes (i.e. APs) on this path as follows: SS $= a_0^i, a_1^i, \ldots, a_{k_i}^i = v_i$. For each node a_j^i along P_i, let λ_j^i be the mean incoming flow rate for its queue, and $\rho_j^i = \lambda_j^i/\mu$ be the traffic intensity, both of which depend on

[2] This is straightforward if the iCAR-MOPA also provides an adaptive navigation service that the user subscribes to.

x_i; also let W_j^i be the random variable denoting the queuing time at the node. Then, it is standard that

$$\begin{aligned} \text{Prob}[W_j^i \leq t] &= 1 - \rho_j^i e^{-(\mu - \lambda_j^i)t}, \quad t \geq 0, && (2) \\ \text{E}[W_j^i] &= \frac{\rho_j^i}{\mu - \lambda_j^i}, && (3) \\ \text{Var}\left[W_j^i\right] &= \frac{(2-\rho_j^i)\rho_j^i}{(\mu - \lambda_j^i)^2}. && (4) \end{aligned}$$

Also, for $1 \leq j \leq k_i$, let d_j^i be the propagation delay from a_{j-1}^i to a_j^i. Obviously, we must have $\sum_{j=1}^{k_i} d_j^i < T_i$ in order for the data to have any chance to arrive at v_i in time. When $\sum_{j=1}^{k_i} d_j^i >= T_i$, iCAR-MOPADS will simply set $x_i = 0$, namely no data will be sent to v_i for this request. Let r be the average transmission speed of links along P_i. Then,

$$X_i = \frac{x_i k_i}{r} + \sum_{j=1}^{k_i} d_j^i + \sum_{j=0}^{k_i} W_j^i \qquad (5)$$

Note that the first term $\frac{x_i k_i}{r}$ is simply the sum of the transmission times of all nodes before v_i. However, X_i will be longer if there are other data to some other node v_j sent earlier by the SS which also use the first few hops along P_i (i.e. when P_i and P_j share a prefix).

Even if one finds a way to take into account the effect of previously transmitted data x_j $(j < i)$ on X_i, in order to compute Prob$[X_i \leq T_i]$, we need to compute the distribution of the sum $\sum_{j=0}^{k_i} W_j^i$. This is computable in closed form, but the formula is fairly complex, involving the sum of various exponential functions on the x_i. Accordingly, the problem has an NLP formulation. Since solving this problem may be costly in time, we present two other simpler approaches as follows.

Approach 2. To avoid the costly computation needed to optimize the complicated, non-linear objective function in (1), in this approach we try to approximate the optimization problem by putting a threshold δ on each failure probability Prob$[X_i > T_i]$, and then optimize the expected number of bytes received by the vehicle. This approach is quite conservative. We expect it to work better when the delays fluctuate widely. In this case, the optimization problem becomes

$$\begin{aligned} \max \quad & \sum_{i=1}^{k} p_i x_i \\ \text{subject to} \quad & \text{Prob}[X_i > T_i] \leq \delta, \quad \forall i \quad 1 \leq i \leq m \qquad (6) \\ & 0 \leq x_i \leq b_i, \quad \forall i \quad 1 \leq i \leq m \\ & \sum_{i=1}^{m} x_i \leq F. \end{aligned}$$

This is not yet a linear program since Prob$[X_i > T_i]$ is non linear on the x_i. To "turn" it into an LP, we can upperbound Prob$[X_i > T_i]$ using the one-sided Chebyshev's inequality, and then enforce the threshold δ indirectly on the upperbound instead of directly on Prob$[X_i > T_i]$. This is doable since X_i is a linear function of the waiting times in the queue along the path to v_i, and since our queueing network is an acyclic open Jackson network [4]–[6], implying the waiting times are independent. Thus we can easily compute both E$[X_i]$ and Var$[X_i]$ using (2), (3), and (4). The upperbound of Prob$[X_i > T_i]$ will then be a linear combination of the variables $x_1, \dots, x_m$.

As an example of how to find an upperbound for Prob$[X_i > T_i]$, consider a path P_i which shares no prefix with any other P_j, implying X_i can be expressed as in (5). We have,

$$\begin{aligned} \nu_i := \text{Var}\left[\sum_{j=0}^{k_i} W_j^i\right] &= \sum_{j=0}^{k_i} \frac{(2-\rho_j^i)\rho_j^i}{(\mu - \lambda_j^i)^2} \\ \mu_i := \text{E}\left[\sum_{j=0}^{k_i} W_j^i\right] &= \sum_{j=0}^{k_i} \frac{\rho_j^i}{\mu - \lambda_j^i} \end{aligned}$$

We will only consider the case when $T_i \geq \mu_i + \sum_{j=1}^{k_i} d_j^i$, otherwise we set $x_i = 0$. By Chebyshev's inequality, we have

$$\begin{aligned} \text{Prob}[X_i > T_i] &= \text{Prob}\left[\sum_{j=0}^{k_i} W_j^i > T_i - \frac{x_i k_i}{r} - \sum_{j=1}^{k_i} d_j^i\right] \\ &\leq \frac{\nu_i}{\nu_i + \left(T_i - \frac{x_i k_i}{r} - \sum_{j=1}^{k_i} d_j^i - \mu_i\right)^2} \end{aligned}$$

Consequently, Prob$[X_i > T_i] \leq \delta$ will hold if

$$x_i \leq \frac{r}{k_i}\left(T_i - \sum_{j=1}^{k_i} d_j^i - \mu_i - \sqrt{\nu_i(\frac{1}{\delta} - 1)}\right) \qquad (7)$$

This constraint can then be used to substitute the constraint in (6) (as an approximation), which results in an LP formulation (whose detailed description is omitted due to space limitation).

Approach 3. A potential problem with Approach 2 is that the Chebyshev inequality used to upper bound Prob$[X_i > T_i]$ makes it quite conservative. In this Approach 3, we think of the waiting times at all queues as constants. More specifically, we assume that each node a_j^i along the fixed routing path from SS to AP v_i periodically sends its queuing delay to the SS using a cellular channel. The SS uses the mean q_j^i of the reported queuing delays for node a_j^i to approximate the actual queuing delay that packets will experience. This way, the SS can estimate the approximate value of X_i as a linear function of the x_i. Then, we can formulate a linear program as follows:

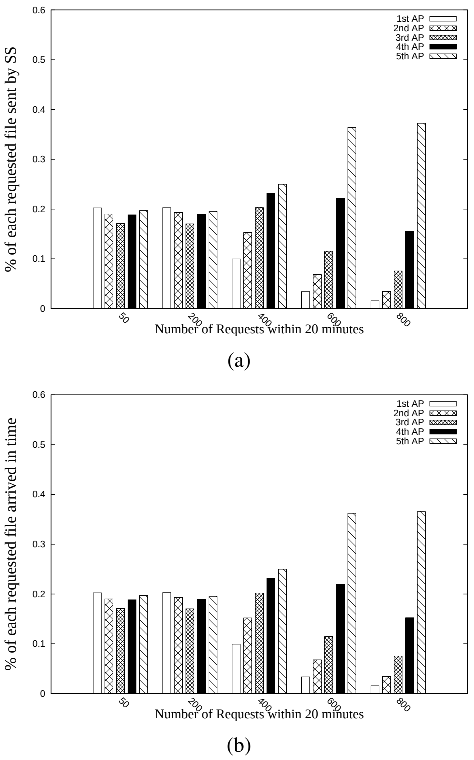
$$
\begin{aligned}
\max \quad & \sum_{i=1}^{m} p_i x_i \\
\text{subject to} \quad & \sum_{j=1}^{k_i} (d_j^i + q_{j-1}^i + \frac{x_i}{r}) + && (8) \\
& \sum_{j=0}^{k_i-1} \sum_{u=0}^{i-1} \frac{x_u}{r} I(j,i,u) \leq T_i - \delta, \quad 1 \leq i \leq m && (9) \\
& I(j,i,u) = \begin{cases} 1 & \text{if } a_j^i = a_j^u; \\ 0 & \text{otherwise} \end{cases} && (10) \\
& 0 \leq x_i \leq b_i, \qquad 1 \leq i \leq m && (11) \\
& \sum_{i=0}^{m} x_i \leq F && (12)
\end{aligned}
$$

where δ is a system parameter which provides a safety margin to ensure data arrives earlier than the vehicle, and F is total amount of data requested by a user.

In the constraints of the linear program formulation, constraint (9) states that the data x_i should be delivered to v_i before the estimated arrival time of the vehicle at the AP minus the safety margin δ. Note that if the actual delay time distributions are highly concentrated with a small tail, a small value of δ is sufficient to mimic the threshold constraint in (6) without having to become conservative as in Approach 2. The first term of (9), i.e. term (8), is the sum of the propagation, transmission, and queueing delay along the path to v_i. The next term of (9) approximates the total additional queueing delay for x_i caused by the previous transmission of data, $\{x_1, \cdots, x_{i-1}\}$. Note that, $I(j,i,u)$ in (9) and (10) indicates whether or not P_i and P_u share an AP. The constraint (11) keeps the SS from transmitting data more than the maximum amount of data a vehicle can download at v_i. Also, the sum of x_i can not exceed the total amount of data requested as represented in (12). One may be able to send more data than requested (with some sort of coding/replication algorithm such as erasure codes) to increase the expected amount of delivered data. This approach is an open research direction.

IV. Numerical Results

In this section, we evaluate Approach 3 (selected mainly for its feasibility) using simulation. The basic idea is that for each request, we solve the LP in Approach 3 to determine the optimal x_i ($1 \leq i \leq m$). The resulting $x_1, \ldots, x_m$ are then fed into a simulator, along with other input parameters, to obtain performance results. The basic simulation set-up is as follows. A total of 25 APs are deployed forming a 5-by-5 grid with a distance of 750m between horizontally and vertically adjacent APs to mimic a 3km ×3km section of central Manhattan. They are connected to the SS located at the center of the region with a tree topology (different tree topologies are simulated), and in each of such tree topologies, only two geographically neighboring APs in the grid may be connected with a tree branch. In all the tree topologies simulated, SS is connected to four close-by APs with one transmitter (and queue) per AP. Each AP may be connected to 0 to 3 other downstream APs but has only one transmitter. We assume that an AP can receive (from upstream) and transmit (to downstream) data simultaneously using two separate channels.



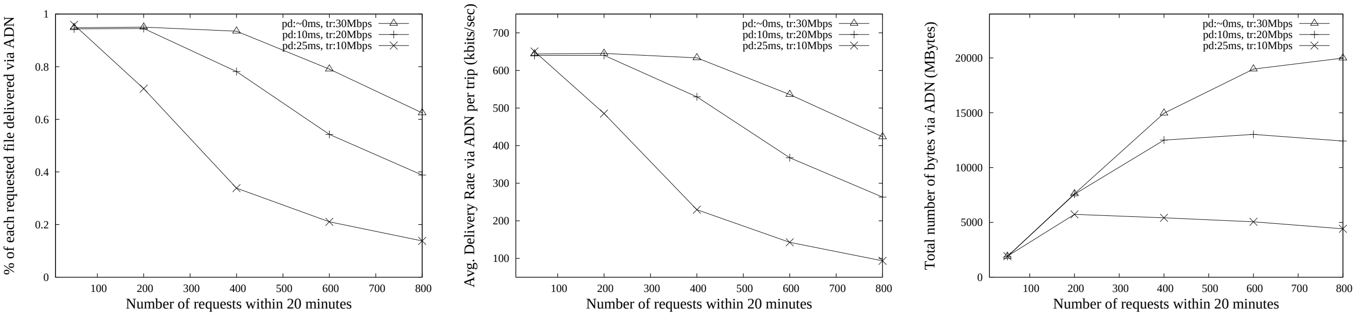
(a)

(b)

Fig. 2. Percentage of the requested file sent to and received by $v_1 \cdots v_5$

Note that although the simulation set-up models the ADN mainly with a wireless mesh (with directional antenna), it can also be adapted, by changing the assumptions on the propagation delay and link transmission speed, to simulate other ADNs, e.g., an ICMAN where data from one AP can be relayed to another AP in a multi-hop fashion using other vehicles. In this study, the time-varying transmission rate (in Mbps) of each AP ranges from high, medium and low, modeled using a normal distribution with $(\mu, \sigma) = \{(30,5),(20,3.4),(10,1.7)\}$ respectively. The propagation delay of each link ranges from low ($\sim 0ms$ representing the time for a signal to travel between two APs at light speed), medium ($10ms$ on average, taking into consideration the delay caused by 5-hop store-and-forward relay among vehicles between two APs that are located $750m$ apart, assuming each store-and-forward delay is $2ms$) and high ($25ms$ on average, taking into consideration about 12-hop relay).

Within a total simulation time of 20 minutes, a number of requests (up to 800), one from each vehicle, were generated at random intervals. Each vehicle travels along a shortest path that includes exactly 5 APs $(v_1, \ldots, v_5)$, where v_1 and v_5 are selected arbitrarily, and making a random decision at each possible intersection (e.g., a vehicle going North-East diagonally will randomly choose North or East at an



(a) Percent of requested data delivered via ADN

(b) Effective delivery rate via ADN

(c) Total number of bytes delivered via ADN

Fig. 3. Simulation Results

intersection as long as it will get closer to the destination). The moving speed between two APs (in meters per second) follows a normal distribution with $(\mu, \sigma) = (10, 3)$, and the sojourn time of a vehicle at an AP (in seconds) also follows a normal distribution with $(\mu, \sigma) = (30, 3)$. This means that on average, a trip will take 525 seconds. We also assume that each vehicle requests for a file whose size is exactly $F = 40Mbytes$, and that, for each request, SS will send at most $b_i = 20Mbytes$ to each AP (v_i) to be visited by the vehicle in an attempt to avoid overloading certain links and APs in the ADN. As for performance metrics, we are interested in knowing the following values averaged over all vehicles and all simulation runs: (a) the percentage of each requested file sent by SS to each v_i (according to the LP solver), which is x_i over 40Mbytes; (b) the percentage of each requested file received by each v_i; (c) the percentage of each requested file delivered to all the five APs in time and delivered to a vehicle via the ADN; (d) the effective delivery rate and the total number of bytes downloaded to a vehicle via ADN.

Fig. 2 shows the results obtained under high transmission rate (30Mbps) and low link propagation delay (1.6×10^{-6} seconds). From Fig. 2(a), we can see that at a low load (e.g., 50 requests within 20 minutes), SS evenly distributes the amount of data (x_i) to APs, because SS has enough time to transmit data to all APs including APs close to the current location of the vehicle. Also, almost all data seems to get delivered by the time a vehicle reaches its fifth AP. These results have positive implications for video streaming applications (as a reference point, 40Mbytes of video data can be played for 640 seconds at the playing speed of $500kbps$ which gives a decent picture quality/size).

As the load increases (e.g. 400 requests), however, SS is forced to send more data to the APs farther along the path to be taken by a vehicle due to the increased delivery delay (mainly resulted by the high queuing delay). At a very high load (800 requests), SS sends most data to the last AP of the vehicle's journey. Also note that if SS is not restricted to send data in the order of $x_1, \ldots, x_5$, more data may be delivered to the fifth and fourth APs along a vehicle's paths, and the total amount of data delivered *may* also increase.

As shown in Fig 2 (b), the distribution of the received data by $v_1 \cdots v_5$ is almost same as that of sent data from SS. This indicates that most data sent by SS arrive at APs before vehicles do, which enables the vehicles to download the data.

Fig. 3 compares the performance of ADN with different combinations of the propagation delay and transmission rate of the links. As shown in Fig. 3, the system has acceptable performance with less than 400 requests within 20 minutes at a low to medium load, high to medium transmission rate, and low to medium propagation delay. For example, Fig. 3(a) shows more than 80% of each requested file can be delivered through the ADN when the load is low to medium and the propagation delay and transmission rate are 10ms and 20Mbps respectively. As the load increases, the percent decreases due to a longer queuing delay within the ADN from the network congestion.

As shown in Fig. 3(b), the file delivery rate via ADN is more than 500 kbits/sec up to 400 users when (propagation delay, transmission rate) = (10ms, 20Mbps) or better, which is enough for most Internet video streaming services.

Fig. 3(c) shows that the total number of bytes vehicles receive via ADN. At a low to medium load, the number of bytes a vehicle can download during a 525 seconds trip is more than 30 MBytes on average when transmission rate and propagation delay are 20Mbps and 10ms or better. However, as the load grows, the total number of bytes decreases due to the congestion.

V. Related Works

Existing architectures for ITS can roughly be categorized into three kinds: purely infrastructure-based, purely ad hoc, or heterogeneous. The first kind is mainly for vehicle-to-Infra-structure or infra-structure-to-vehicle communications, and typically involves cellular towers or base stations (BS), or Access Points (APs) that are connected to the Internet. Vehicles communicate indirectly with servers or other vehicles via BSs using medium-to-long range wireless cellular interfaces (e.g., 3G or WiMAX), or via APs using short-range wireless interfaces (e.g., 802.11 based). [7]–[9]

In the second kind of architectures, which are mainly for vehicle-to-vehicle communications, vehicles are typically equipped with various sensors and communicate mainly with each other using short-range wireless interfaces [10]–[13]. Communications between two remote vehicles may be via relays by other vehicles and road-side access points (APs). (Note that unlike the BSs or APs in the first kind of architectures, these APs, when used in this context, are not connected to the Internet.)

In the third kind of architectures, heterogeneous wireless technologies such as cellular, WiMAX, 802.11 and DSRC [14] are used. There are various approaches with different research foci. For example, each vehicle may download a small piece of the same file from a road-side AP (connected to the Internet) during the limited period of AP association time, and then share with other vehicles [15]. The research focus here is to emulate the Bit-Torrent in a vehicular environment. In another approach, a vehicle in an area covered by multiple BS and APs may choose one of several possible connections with different costs and QoS performance characteristics [16]. The research focus here is to minimize the cost subject to certain QoS constraint. Yet another approach is to enable a vehicle, far from any BS or AP, to use an ad hoc interface to communicate with another vehicle acting as a relay or gateway to a BS/AP (and then the Internet).

The proposed iCAR-MOPA architecture is somewhat similar to those heterogeneous architectures (i.e., the third kind), especially the work in [16], but with significant differences from the existing approaches. For instance, the subsystem iCAR-MOPADS described in this paper allows a vehicle to download most data via the Ad hoc Distribution Network (ADN) and use the cellular channels to send and receive control signals and a minimal amount of data only if it is necessary. This is to address high cost and low bandwidth issues associated with cellular based architecture (the first kind) and the limited scalability issues associated with the purely ad hoc architecture (the second kind), and allow iCAR-MOPADS to provide quality and scalable services (to large number of potential subscribers) at affordable price. In iCAR-MOPADS, unlike in [16] and other existing approaches, the content server(s) is (are) a part of the ADN, instead of residing in the backbone. Accordingly, our research focus is on how to partition a requested file into different segments and route them through the multi-hop ADN to different APs to be visited by a vehicle under certain delay and bandwidth constraints such that a maximum amount of data can be delivered via the ADN.

VI. Concluding Remarks

In this work, we have proposed a new architecture called iCAR-MOPA that integrates these heterogeneous wireless/mobile technologies by utilizing both a cellular system and an ad hoc distribution network (ADN), and in addition, takes advantage of users' mobility profiles in order to provide good scalability, QoS support, and cost-efficiency. We have investigated a subsystem that provides file download services called iCAR-MOPADS, formulated an optimization problem called SMV (Shooting a Moving Vehicle) using NLP and LP, evaluated a simple LP approach via simulations, and obtained encouraging results. The work will be extended to developing other subsystems providing services such as video streaming, customized real-time traffic alerts, and adaptive navigation under the proposed architecture.

Acknowledgement

The work of Hung Q. Ngo was supported in part by NSF CAREER Award CCF-0347565. The work of S. Yoon and C. Qiao was supported in part by NSF SGER grant CNS-0553273

References

[1] T. Small and Z. Haas, "The shared wireless infostation model - a new ad hoc networking paradigm (or where there is a whale, there is a way)," in *In Proceedings of IEEE MobiHoc 2003*, 2003.

[2] R. Frenkiel, B. Badrinath, J. Borras, and R. Yates, "The infostations challenge: Balancing cost and ubiquity in delivering wireless data," *EEE Personal Communications*, vol. 7, no. 2, pp. 66–71, 2000.

[3] T. Spyropoulos, K. Psounis, and C. Raghavendra, "Spray and Wait: An efficient routing scheme for intermittently connected mobile networks," in *Proceeding of the 2005 ACM SIGCOMM workshop on Delaytolerant networking*, August 2005.

[4] B. Simon and R. D. Foley, "Some results on sojourn times in acyclic Jackson networks," *Management Sci.*, vol. 25, no. 10, pp. 1027–1034 (1980), 1979.

[5] A. J. Lemoine, "Networks of queues—a survey of equilibrium analysis," *Management Sci.*, vol. 24, no. 4, pp. 464–481, 1977/78.

[6] E. Reich, "Note on queues in tandem," *Ann. Math. Statist.*, vol. 34, pp. 338–341, 1963.

[7] N. Imai, H. Morikawa, and T. Aoyama, "Prefetching architecture for hot-spotted networks," in *IEEE International Conference on Communications (ICC 2001)*, vol. 7, Helsinki, Finland, June 2001.

[8] B. Hull, V. Bychkovsky, K. Chen, M. Goraczko, A. Miu, E. Shih, Y. Zhang, H. Balakrishnan, and S. Madden, "Cartel: A distributed mobile sensor computing system," in *Proceedings of the 4th ACM Conference on Embedded Networked Sensor Systems (SenSys)*, Colorado, USA, 2006.

[9] J. Ott and D. Kutscher, "Drive-thru internet: Ieee 802.11b for 'automobile' users," in *Proceedings of IEEE INFOCOM 2004*, 2004.

[10] Q. Xu, T. Mark, J. Ko, and R. Sengupta, "Vehicle-to-vehicle safety messaging in dsrc," in *Proceedings of the First ACM Workshop on Vehicular Ad Hoc Networks (VANET 2004)*, PA, USA, October 2004.

[11] L. Briesemeister, L. Schafers, and G. Hommel, "Disseminating messages among highly mobile hosts based on inter-vehicle communication," in *Proceedings of the IEEE Intelligent Vehicles Symposium*, MI, USA, October 2000.

[12] J. Zhao and G. Cao, "VADD: Vehicle-assisted data delivery in vehicular ad hoc networks," in *Proceedings of IEEE INFOCOM*, 2006.

[13] X. Yang, J. Liu, F. Zhao, and N. Vaidya, "Vehicle-to-vehicle communication protocol for cooperative collision warning," in *Proceedings of MobiQuitous 2004*, August 2004.

[14] "Standard specification for telecommunications and information exchange between roadside and vehicle systems - 5ghz band dedicated short range communications (dsrc) medium access control (mac) and physical layer (phy) specifications," in *ASTM E2213-03*, SEPTEMBER 2003.

[15] A. Nandan, S. Das, G. P. M. Sanadidi, and M. Gerla, "Cooperative downloading in vehicular ad hoc wireless networks," in *Proceedings of IEEE/IFIP International Conference on Wireless On demand Network Systems and Services*, St. Moritz, Switzerland, January 2005, pp. 32–41.

[16] H. Wu, C. Qiao, S. De, , and O. Tonguz, "Integrated cellular and ad hoc relaying systems: icar," *IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS*, vol. 19, no. 10, pp. 2105–2115, OCTOBER 2001.