2014 12th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt)

A Vehicular Backbone Network (VBN) with Joint Transportation-Wireless Capacity Utilization

Bo Tan, Jubin Jose, Xinzhou Wu
Qualcomm Research
Bridgewater, NJ 08807, USA
Email: {bot, jjose, xinzhouw}@qti.qualcomm.com

Lei Ying
ECEE, Arizona State University
Tempe, AZ 85287, USA
Email: lei.ying.2@asu.edu

***Abstract*—A vehicular backbone network (VBN) has the potential to augment the Internet with high-throughput data flows for delay-tolerant traffic. High-throughput flows require a joint utilization of transportation capacity for carrying data packets through physical mobility and wireless capacity for switching data packets from one route to another. This paper establishes a model that incorporates both transportation mobility and wireless switching. Then, it characterizes the network capacity based on flow conservation, wireless communication capacity constraints and data storage limits, and solves a convex optimization that results in joint routing and congestion control. A variant with cost minimization reduces delay while maximizing throughput. Next, this paper develops a distributed algorithm that achieves the global objective with limited infrastructure support. Lastly, a packet-level simulation platform using real-world road map and traffic statistics is used to evaluate the distributed algorithm, and demonstrate the significant performance enhancement achieved.**

I. Introduction

Big data challenges are not limited to data processing and analysis. It has become a challenging issue to transfer large volumes of data for storage, and the broadband Internet alone is not sustainable for big data transfers. As reported in [1], 2.5 quintillion (10^{18}) bytes of data were created daily in 2012. While new technologies are necessary to address this challenge, in this paper, we advocate a combination of "old" and "new" methods to alleviate this immediate issue — a high-throughput backbone network based on the existing transportation networks upgraded with any wireless communication technology. Below is a simple example that demonstrates the potential data rate that a vehicular network may provide.

Example: Consider a two-lane national highway with a volume of 360 cars/hour, i.e., 0.1 cars/s. If each car has a 1 TB hard drive, which costs around $100 today, the aggregated rate of the data flows carried by the cars is 800 Gbps. □

We note that only a small fraction of the Internet backbone network today has bandwidth of 100 Gbps. So with a rather minimal investment on each car (a hard drive and a wireless card), a nationwide *vehicular backbone network (VBN)* can be used for big data transfers with much higher aggregate throughput than today's Internet. While this vehicular backbone cannot replace the broadband Internet because the transmission latency is orders of magnitude larger (hours versus milliseconds), it is a valuable alternative for delay-tolerant data transfers and can be used to deliver high quality videos to households, upload large volumes of data created by businesses/enterprises to cloud computing clusters, or transfer data between data centers at different geographical locations.

Apart from providing an alternative high-throughput network, a vehicular backbone network has a rather unique advantage of being a reliable post-disaster communication network. After disasters such as earthquakes and hurricanes, vast geographical areas could be in power outage with possible damages to communication equipment including routers and wireless base stations. For example, during hurricane Sandy, the most destructive hurricane of 2012 Atlantic hurricane season, millions were left without power for many days. In such scenarios, since vehicles have high capacity batteries and mostly use petroleum for fuel, we can rely on a VBN to provide a high-throughput and reliable data-delivery network. Again, this may not be appropriate for delay-critical traffic.

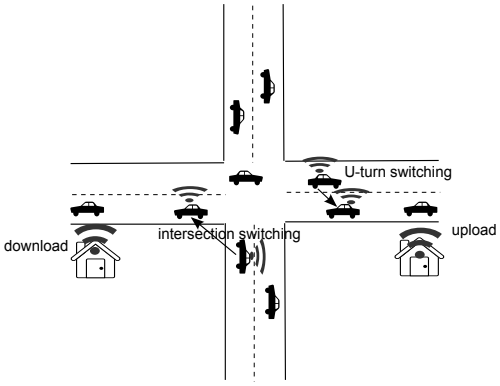


Figure 1: The basic architecture of a VBN

One possible approach to utilize this transportation-based capacity is to have dedicated vehicles (e.g., trucks from FedEx or UPS) to deliver high capacity disks from sources to destinations. However, this approach incurs a significant overhead (fuel, labor, maintenance, etc.), and only utilizes a small fraction of the vehicles on the road, resulting in significant under-utilization of available transportation capacity. In this paper, we propose an architecture as shown in Figure 1. In this architecture, end-users upload and download their data to and from vehicles using wireless communication. These vehicles are not dedicated for data transfers so their destinations and travel routes are determined by their own transportation needs. Since the mobility of these vehicles is uncontrollable from the perspective of data transfers, the vehicle which first receives the data may not be able to deliver them to their destination if it is not on the route of this vehicle. Hence, if we rely only on the vehicle mobility to transfer data, the packet loss rate will be very high. We propose to solve this problem by *wireless switching*. The wireless switching contains two components: intersection switching and U-turn switching as shown in Figure 1. Under the proposed architecture, a vehicle maintains a routing decision table for each destination and each

978-3-901882-63-0/2014 - Copyright is with IFIP

intersection. These routing decision tables are computed based on vehicle traffic and data traffic statistics, and are distributed to vehicles periodically in a relatively slow time scale. When a vehicle passes a road intersection, it will look up its routing decision table and possibly hand off data to the vehicles on neighboring road segments chosen by the routing decision, called intersection switching. When a vehicle is moving away from the data destination, it may also hand off packets based on routing decisions to the vehicles that are moving in the opposite direction, called U-turn switching.

Figure 2 shows a simulation result on a "Manhattan map." The network throughput under joint utilization of transportation and wireless capacities (solid line) increases significantly as the wireless capacity increases, and it largely outperforms the schemes which only use the wireless capacity (dotted line as an upper bound). Details will be described in Section IV.

The focus of this paper is to propose a basic architecture for a VBN. The main contributions of this paper include:

- We establish a practical and holistic model that incorporates both transportation mobility and data packet routing through wireless. In particular, we characterize the capacity of a vehicular backbone network based on flow conservation constraints, data storage limits and wireless capacity, and then formulate a convex optimization problem for joint routing and congestion control.
- We further introduce a cost minimization problem. The joint congestion control/routing algorithm for cost minimization improves delay performance while guaranteeing throughput and fairness.
- Based on the optimization formulation, we develop a distributed routing/congestion control algorithm which is capacity achieving, guarantees fairness among different flows and has good delay performance.
- We build a packet-level simulator with real-world road map and traffic statistics to evaluate distributed algorithms against the benchmark, and demonstrate the significant performance improvement of the developed algorithms.

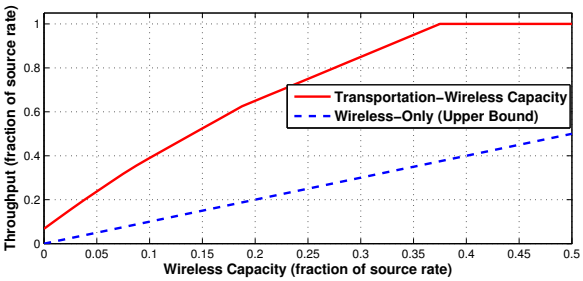


Figure 2: Throughput affected by wireless capacity

A. Related Work

Many aspects of routing in mobile ad hoc networks (MANETs) [2] and delay-tolerant networks (DTNs) [3] apply to vehicular ad hoc networks (VANETs), but the challenges and objectives could be different. A comprehensive survey on distributed routing algorithms in VANETs can be found in [4].

Capacity scaling results show that mobility can significantly increase the overall capacity for multiple unicasts [5] and multicasts [6]. Physical mobility, i.e., the ability to carry data from one location to another, is not fully exploited by many routing algorithms. However, depending on vehicular mobility alone for data delivery could result in very large packet delays and packet drops. Hence, both physical mobility and wireless capacity have to be jointly utilized to achieve higher throughput with reasonable delay performance. Wireless capacity could be used for switching data packets from one route to another in an intelligent manner. Since typical routing algorithms do not solve this problem of joint utilization of physical mobility and wireless capacity, new framework and algorithms are required. In this paper, we develop a new framework that exploits multiple paths for each flow in an optimal manner based on source rates, road topology, aggregate traffic parameters and source-destination locations. We design distributed algorithms based on this framework.

We envision the utilization of the vehicular network as a backbone network for multiple flows between fixed sources and destinations. In contrast to data delivery to vehicles, a vehicular backbone network demands very high throughput. Hence, flooding-based approaches used for data delivery such as gossip [7] [8] may not be suitable. Gossip-based routing is primarily for geocast or multicast and could result in a broadcast storm. Hence, broadcast storm mitigation techniques are required [9]. In this paper, we focus on unicast and hence there are no packet duplications. Packet drops are expected to be handled by adaptive source coding or rateless codes [10]. There are also similarities with the problem of data delivery to or from vehicles [11] [12]. We utilize a mobility-centric approach as in [11] and intersection switching as in [12]. However, our primary objective is throughput maximization instead of delay minimization used in [11] and [12]. In [13], the authors provide a throughput optimization framework that is vehicle-centric in contrast to our mobility-centric approach. In [14], the authors propose a delay-optimal scheduling policy under a simple two-hop vehicular relay network model. There also exist prior wireless systems like "Infostations" [15] and "Daknet" [16], which are relevant to our VBN but mainly focus on improving connectivity and coverage for mobile terminals.

In [17], static nodes are used to temporarily store packets in the network, which could be a fairly stringent assumption. In our distributed algorithm, we assume the presence of road-side units or limited cellular connectivity for exchanging control messages, i.e., limited infrastructure support. This assumption is easier to realize in practice due to the low overhead for control messages compared to data packets.

II. Model and System Description

A. Road Network: A Graph-Theoretic Description

A road network is defined as a directed graph $G(\mathcal{V}, \mathcal{E})$ where $\mathcal{V}$ and $\mathcal{E}$ are respectively the sets of vertices and edges. This graph G models real-world roads through the following abstraction. Each road-intersection (*intersection* for short) is represented as a vertex $u \in \mathcal{V}$ and each directed road-segment (*segment* for short) between any two intersections is represented as an edge $i \in \mathcal{E}$.

Next, we introduce some notations that are used in the rest of this paper. Let ω_i and $\bar{\omega}_i$ denote the start vertex and end vertex of edge i. Since many roads have two directions[1], we define $\bar{i}$ as the reverse edge of edge i, i.e., $\omega_i = \bar{\omega}_{\bar{i}}$, $\omega_{\bar{i}} = \bar{\omega}_i$,

[1]Typically, each direction could have several lanes. Since the width of a road is much smaller than the communication range, all the lanes can be regarded as one "aggregate" lane.

and $\bar{\bar{i}} = i$ always hold. Note that for a one-way road i, $\bar{i}$ does not exist. In all the formulations we use in this paper, $\bar{i}$ can be treated as a null index if it does not exist. We define $\mathcal{O}_u$ as the set of outbound edges at each $u \in \mathcal{V}$. Furthermore, $\mathcal{N}_i$ is defined as the set of potential "next-hop" edges of edge $i \in \mathcal{E}$, i.e., $\bar{\omega}_i = \omega_j$ for all $j \in \mathcal{N}_i$. In a dual-way road, $\bar{i} \in \mathcal{N}_i$ holds.

In short, the road network describes the edge connectivity of the roads. However, connectivity is only a necessary condition. There may not be any vehicle that go directly from i to an edge in $\mathcal{N}_i$ if the probability of this event is zero. Possible examples include temporarily blocked roads and illegal U-turns. This is captured by a mobility model that is described next.

B. Markovian Mobility Model for Vehicles

Real-world traffic patterns are quite complicated to model using simple mobility models. Even though traffic traces could be used for simulation purposes, it is difficult to use such traces for analysis and algorithm design. Hence, we make a few assumptions to arrive at a simple mobility model. All vehicles are assumed to be homogeneous and independently follow the same mobility pattern. Each vehicle's random movement at every intersection follows a Markov chain. The state of this Markov chain is the current edge the vehicle is staying on, and the transition matrix is denoted as $\mathbf{P} \triangleq [p_{ij}]$, i.e., p_{ij} is the probability of switching from edge i to edge j when this vehicle reaches the end of edge i, such that $\sum_{j \in \mathcal{N}_i} p_{ij} = 1$. Markovian mobility assumptions are used frequently in the VANET research (e.g. [18] [19]). These assumptions are also motivated by the need for a causal and memoryless model to develop algorithms that are robust.

Remark 1: The modeling of mobility patterns can be easily generalized to include heterogeneity by dividing vehicles into different classes. Vehicles of the same class follow their own mobility pattern. This approach can also be used to model buses and shuttles with fixed routes. The Markovian model applies to them if we set all the transition probabilities along the fixed route (of a certain class) as 1.

C. Traffic Splitting of Transportation Flows

We utilize a flow-level model for throughput analysis and optimization. Given the simplicity of the Markovian mobility model, it is fairly straightforward to obtain the flow-level model. Instead of treating $\{p_{ij}\}$ as transition probabilities, under the flow-level model, $\{p_{ij}\}$ can be regarded as the splitting fractions of the transportation flows (which "fluidizes" the real-world vehicles). Defining the transportation traffic rate on edge i as ν_i (in vehicle/sec), the rate of the traffic switching to edge $j \in \mathcal{N}_i$ will be $p_{ij}\nu_i$ under this splitting rule.

D. Data Delivery Model

Next, we describe the data delivery model built upon the transportation network composed of moving vehicles. There are M independent data flows[2] (*flows* for short) between M sources and M sinks with arbitrary but fixed locations. These flows are assumed to be long-term data flows. All sources and sinks can transfer data to/from nearby vehicles through wireless connectivity, for example, Wi-Fi (802.11n, 802.11ac and so on). Data transmission from a source to nearby vehicles is referred to as *loading* data and from nearby vehicles to a destination is referred to as *unloading* data. In addition, vehicles can also talk to nearby vehicles using wireless connectivity, for example, DSRC (i.e., 802.11p). Hence, data packets can move from one geographical location to another using both physical mobility and wireless transmissions.

We assume that packet transmissions use unicast mode, i.e., although all the vehicles in the transmission range can receive the transmitted packet, only one vehicle will keep it. Hence, packet duplications are avoided. An ARQ scheme is used and hence we assume perfect channel and entirely reliable data transmission for the flow-level model. Broadcast is an interesting extension and has the potential to improve throughput. Similarly, network coding is a powerful technique that could improve throughput. In this paper, we restrict to unicast to simplify both analysis and algorithm design.

Consider a flow f. The source and destination (or sink) are located on edges s_f and d_f, respectively. Since the width of a road is usually shorter than typical Wi-Fi communication range (100 m), on a dual-way road where the source is located, both edge s_f and its reverse edge $\bar{s}_f$ are considered as targets for loading data from the source. Similar rule applies to edge d_f and reverse edge $\bar{d}_f$ when vehicles are unloading data to the destination.[3] For each flow f, define λ_i^f as its flow rate on edge i, and its throughput then equals the sum of the flow rates on the destination edges, i.e., $\lambda_{d_f}^f + \lambda_{\bar{d}_f}^f$. The four basic operations on data packets are explained in more detail next.

1. Loading from a source to a nearby vehicle: The source node of flow f transmits packets to vehicles traveling on road i at rate y_i^f for $i \in \{s_f, \bar{s}_f\}$ (if $\bar{s}_f$ exists). The source can select any vehicle among the vehicles in the wireless range. The total loading rate is upper bounded by θ_f, which is determined by the wireless scheme at the source.

2. Transportation through physical mobility: Vehicles store received packets in a buffer and carry it along when they move. When a packet is transmitted to another vehicle or destination, it is removed from the storage buffer. Each vehicle is assumed to have a large hard disk with storage capacity denoted by B bits. This leads to a "transportation capacity" upper bound given by $\sum_{f \in [1,M]} \lambda_i^f \le \nu_i B$ on each edge i, where ν_i is the rate of vehicles on edge i.

3. Wireless switching between vehicles: Vehicles can utilize wireless transmissions to switch packets from one vehicle to another. Define $\mathbf{X}^f \triangleq [x_{ij}^f]$ $(x_{ij}^f \ge 0)$ as the "wireless switching matrix" for each flow f, where x_{ij}^f indicates the flow rate switched from edge i to edge j through wireless transmission. The total transmission rate at each vertex $u \in \mathcal{V}$ cannot exceed the wireless capacity c_u at that vertex. The wireless capacity constraints are motivated from the following observations: Since a vertex is a road intersection in the real world, all vehicles near that intersection share the same wireless spectrum. A conservative assumption is that at each intersection there can only exist one data transmission for which the rate is limited by the DSRC rate c. Since intersections are far apart compared to typical wireless range, wireless transmissions at one intersection do not interfere with the wireless transmissions at another intersection. Thus, full spatial reuse is possible across intersections. Hence, we assume that $c_u = c$ at all intersections. These assumptions are made only for the flow-level model. In packet-level simulations, CSMA is used to determine successful

[2]Data flows should not be confused with transportation flows.

[3]For a dual-way road, our convention is to name the edge nearer to where the Wi-Fi access point is located as s_f or d_f.

parallel transmissions.

Limited by the communication range, wireless intersection switching can only involve neighboring edges. We have a further "non-greedy" restriction on the switching scheme.

Scheme 1: At each vertex u, wireless switching can only happen between two different outbound edges $i, j \in \mathcal{O}_u$. □

Without the above restriction, a vehicle could relay packets to another vehicle moving ahead on the same edge, or greedily relay the packets to an edge j which is "nearer" (under a certain metric) to the destination even before reaching the intersection. This "greedy relay" to some extent benefits the delay performance, but when wireless capacity is a bottleneck, especially when the number of flows grows, only switching packets from "wrong" paths to "right" paths is an approach which can better utilize capacity resource.

4. Unloading from a vehicle to a destination: When a vehicle is within the communication range of the destination of flow f, it will send packets at a certain transmission rate (usually the maximum possible rate $\bar{\theta}_f$) to the destination. The maximum rate is determined by the wireless scheme used.

E. Wireless "U-Turn" Switching inside Dual-Way Roads

Vehicles can switch packets using wireless connectivity even if it is not at an intersection. A useful scenario is that of switching packets between vehicles on different directions (forward and reverse) on a dual-way road. In particular, when a vehicle travels along a dual-way road and calculates that the reverse direction is preferable, it could transmit packets to vehicles travelling in the reverse direction. We call this *wireless "U-turn" switching*, of which the capacity has to be modeled differently due to the potential for spatial reuse. Specifically, we add a virtual vertex v_i between the two vertices ω_i and $\bar{\omega}_i$ of edge i (which are two real intersections). The edges are then relabeled, and for the new transition matrix $\mathbf{P}$, old inter-edge probabilistic transitions are kept and new deterministic intra-edge transitions are added (details omitted). The wireless capacity at each new virtual vertex v_i is denoted as c_{v_i}, which is usually larger than the wireless capacity at an intersection due to possible space reuse on the whole road. One approximation is to use a reuse factor that is proportional to the length of this road l_i normalized by the DSRC range r with exclusion regions at the two intersections, i.e.,

$$c_{v_i} = [l_i/r - 1]^+ \cdot c, \tag{1}$$

where c represents the DSRC rate.

III. Joint Transportation-Wireless Capacity Utilization

First, we characterize the joint transportation-wireless throughput region of VBN, and formulate a convex optimization problem for throughput maximization and delay reduction. Next, we develop a packet-level distributed routing/congestion control algorithm for fair resource allocation.

A. Throughput Characterization & Fair Resource Allocation

A VBN's throughput region can be characterized as follows:

$$\lambda_i^f = \sum_{\substack{j:\ i\in\mathcal{N}_j,\\ j\notin\{d_f,\bar{d}_f\}}} p_{ji}\lambda_j^f + \sum_{j:j\in\mathcal{O}_{\omega_i}} \left(x_{ji}^f - x_{ij}^f\right) + y_i^f, \quad \forall i \in \mathcal{E},\ f \in [1,M]; \tag{2}$$

$$\lambda_i^f \geq y_i^f, \qquad \forall i \in \{s_f,\ \bar{s}_f\},\ f \in [1,M]; \tag{3}$$

and

$$\sum_{f=1}^{M} \sum_{i,j\in\mathcal{O}_u} x_{ij}^f \leq c_u, \qquad \forall u \in \mathcal{V}; \tag{4}$$

$$\sum_{f=1}^{M} \lambda_i^f \leq \nu_i B, \qquad \forall i \in \mathcal{E}; \tag{5}$$

$$y_{s_f}^f + y_{\bar{s}_f}^f \leq \theta_f, \qquad \forall f \in [1,M]; \tag{6}$$

$$\lambda_{d_f}^f + \lambda_{\bar{d}_f}^f \leq \bar{\theta}_f, \qquad \forall f \in [1,M]; \tag{7}$$

$$y_i^f = 0,\ \forall i \notin \{s_f,\ \bar{s}_f\}, \qquad \forall f \in [1,M]; \tag{8}$$

$$x_{ij}^f = 0, \qquad \forall f \in [1,M],\ \forall i,\ j \in \mathcal{E}\ s.t.\ i = j \text{ or } \omega_i \neq \omega_j; \tag{9}$$

$$x_{ij}^f,\ \lambda_i^f,\ y_i^f \geq 0, \qquad \forall i \in \mathcal{E},\ f \in [1,M]. \tag{10}$$

Note that equality constraint (2) describes the flow conservation on each edge i for each individual flow f. Constraint (3) is imposed in order to push as much data as possible onto the right path during the data loading phase. Inequalities (4), (5), (6) and (7) are the constraints due to wireless capacity (for switching), hard disk storage, maximum loading rates at sources and maximum unloading rates at destinations, respectively, as described in Subsection II-D.

Now given this throughput region, the problem of fair resource allocation can be formulated as

$$\max_{\{\mathbf{X}^f,\mathbf{y}^f,\boldsymbol{\lambda}^f\}_{f=1}^M} \sum_{f=1}^{M} U_f\left(\lambda_{d_f}^f + \lambda_{\bar{d}_f}^f\right) \tag{11}$$

subject to (2)-(10). In the network utility maximization formulation above, $U_f(\cdot)$ is a non-decreasing concave utility function for flow f, which measures the "happiness" of end users. Maximizing the network utility results in some form of fair resource allocation among the end users. In particular, the utility functions can be chosen as the functions in the "α-fairness" family (e.g., log utility for proportional fairness).

Without loss of generality, we next assume the utility function is linear, i.e., $U_f(x) = x$ for all f, and derive a centralized congestion control and routing algorithm. For other utility functions, the algorithm can be derived similarly. With the linear utility functions, the objective of the network utility maximization problem is to maximize the aggregate data throughput. We have noticed from the simulations that an undesirable consequence of maximizing the aggregated throughput is that the packets are "flooded" unnecessarily onto too many paths. To resolve this issue, a certain form of "network cost" must be added as a penalty function in the optimization objective. A natural cost we come up with for each edge i is the "bits on-the-fly" which equals the product of the data rate and a vehicle's traveling time on that edge. The new network utility maximization problem is:

[OPT]

$$\max_{\{\mathbf{X}^f,\mathbf{y}^f,\boldsymbol{\lambda}^f\}_{f=1}^M} \sum_{f=1}^{M} \left(\lambda_{d_f}^f + \lambda_{\bar{d}_f}^f\right) - w \cdot \sum_{i\in\mathcal{E}} \frac{l_i}{v_i} \sum_{f=1}^{M} \lambda_i^f \tag{12}$$

subject to (2)-(10). □

B. Distributed Routing/Congestion Control

We consider CSMA/CA with flow arrivals/departures that happen at a slower time scale compared to the time scale for physical mobility between sources and destinations.[4] The challenges in developing a distributed routing/congestion control are two-fold: (*i*) Flow-level optimization has to be translated to an algorithm running on each vehicle, and (*ii*) the algorithm has to be distributed. In order to simplify the algorithm design, we assume that the mobility pattern $\{p_{ij}\}$ and data flows (locations and maximum loading rate) are known a priori. This assumption can be realized in practice due to the following: The statistics change slowly and vehicles can obtain updated statistics from a centralized entity (e.g., cloud) using low-overhead wireless transmission, either through *roadside units (RSUs)* or cellular connectivity. Note that the map information can be pre-loaded or obtained in a similar fashion.

Next, we describe our algorithm.

Distributed Routing/Congestion Control Algorithm

The wireless switching matrices $\{\mathbf{X}^f\}$ for all flows $f \in [1, M]$ and the data loading vectors $\{\mathbf{y}^f\}$ can be computed by solving OPT. We use slotted time t. RSUs at each edge i (or a cloud) maintain some information, i.e., variables which track the average capacity usage of successful wireless switching events from this edge i to each edge $j \in \mathcal{O}_{\omega_i} \backslash \{i\}$ for each flow f up to time slot t, expressed as $\bar{x}_{ij}^f(t) \triangleq \frac{1}{t}\sum_{\tau=1}^{t} \hat{x}_{ij}^f(\tau)$, where $\hat{x}_{ij}^f(\tau)$ represents the actual effective rate for successful transmissions during time slot t. Maintaining these variables has the following small control overhead:

- At the beginning of time slot t, the information about $\{\bar{x}_{ij}^f(t),\ \forall j \in \mathcal{O}_{\omega_i} \setminus \{i\},\ f \in [1, M]\}$ is broadcast to vehicles on edge i to help them make routing decisions.
- At the end of time slot t, a nonzero $\hat{x}_{ij^*}^f(t)$ is fed back from the vehicle (on the same edge i) which has made a successful transmission to edge j^*.

Based on the above pre-computation and information dissemination actions assisted by RSUs or a cloud, below are two separate descriptions of algorithm operations respectively at vehicles and sources.

- Operations at Each Vehicle

A vehicle at edge i performs the following at time slot t:

1. *Give higher priority to "unloading" operations:* Before competing for the channel, the vehicle checks whether there exists a destination in the DSRC range for any flow f the vehicle carries. If yes, perform CSMA backoff with higher priority compared to regular wireless switching. After getting channel access, go to Step 4, i.e., unload data to the targeted destination.
2. *Routing decision:* The next-hop candidate set is defined as $\mathcal{C}_i \triangleq \{(j, f) : x_{ij}^f > 0$, and at least one vehicle within the DSRC range is on edge $j\}$. If $\mathcal{C}_i = \emptyset$, stay idle; otherwise, find the lowest "achieved rate fraction" against the benchmark (OPT's solution), i.e., $(j^*, f^*) \in \operatorname{argmin}_{(j,f)\in\mathcal{C}} \left\{\bar{x}_{ij}^f(t)/x_{ij}^f\right\}$ (solving ties uniformly at random). Then, if $\bar{x}_{ij^*}^{f^*}(t)/x_{ij^*}^{f^*} \geq 1$, stay idle.
3. *CSMA priority adjustment:* Unless staying idle in the previous step, the value $\bar{x}_{ij^*}^{f^*}(t)/x_{ij^*}^{f^*}$ is used to adjust the CSMA backoff priority (details depend on how backoff is implemented) when competing for the channel, where a smaller value indicates a higher priority.
4. *Packet transmission:* Once getting access to the channel, pick the geographically nearest vehicle on edge j^* (or the destination if nearby) and broadcast packets of flow f^* with that receiver's ID (e.g., MAC address) included in the packet header. In the DSRC range, only the node with that ID will keep received packets.

- Operations at Each Source

Every time slot, the source of flow f loads data to vehicles on edge i, where $i = s_f$ with probability y_{s_f}/θ_f and $i = \bar{s}_f$ with probability $y_{\bar{s}_f}/\theta_f$. If $1 - (y_{s_f} + y_{\bar{s}_f})/\theta_f > 0$ (i.e., whenever flow control is needed), with this remaining probability, the source will keep idle. Loading operations get higher priority over regular wireless switching operations.

Remark 2: In the above algorithm, every vehicle assumes knowledge of the IDs and locations of the other vehicles within its DSRC range. This information is collected based on a "neighbor discovery" mechanism. □

Since vehicles are not really "flows," the following issues are present: First, a vehicle may not find any other vehicles on the "optimal" edges calculated from the optimal algorithm. Second, channel sharing among neighboring edges are no longer uniform as implicitly assumed in the flow-level model, due to different densities of vehicles on different edges, and different durations of link connectivity (determined by the DSRC range) caused by the fact that each pair of neighboring edges forms a different angle. These issues motivate us to use a CSMA priority adjustment scheme in the algorithm.

The benchmark rates provided by the optimal solution are long-term averages. Hence, there might exist bursty flows that flood out of the "controlled area" defined by the optimal solution (the mathematical definition of an "uncontrolled area" is a vertex set $\{u \in \mathcal{V} :\ \lambda_i^f = 0,\ \forall i \in \mathcal{O}_u\}$ and switching variables x_{ij}^f are not assigned in this area, while its complement is named "controlled area"). To solve this issue, we apply shortest-path routing in the uncontrolled area to switch those "leaked" flows back to the controlled area.

IV. Simulation Results

In this section, we evaluate the performance of our algorithms using flow-level and packet-level MATLAB simulations. Unless mentioned explicitly, the following default parameters are used in the simulations. The hard disk storage size at each vehicle is 1 TB. The constraint (5) (with $B = 8 \times 10^{12}$ bits) is not a bottleneck except for the "national highway" scenario in Subsection IV-C. For each vehicle's speed, we use a constant value of $v = 11.2$ m/s (25 mph) for the two urban-area maps (Subsections IV-A and IV-B), and $v = 29.1$ m/s (65 mph) for the national highway map (Subsection IV-C). We set both the maximum loading rate and the maximum unloading rate to $\theta = 400$ Mbps, and the DSRC wireless switching rate to $c = 200$ Mbps. The DSRC and Wi-Fi (for loading and unloading) communication ranges are both set as 200 m.

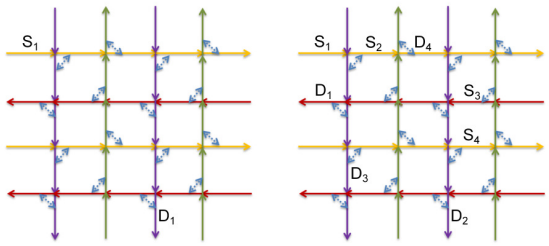
[4] In practice, a small portion of the wireless capacity can be reserved to serve short-lived flows.

Flow ID	Throughput (Mbps)		
	c = 0	c = 100	c = 200
1	34.8	160.3	275.9
2	32.2	180.9	283.3
3	42.9	146.6	216.3
4	1.3	101.7	222.3
Total	111.2	589.5	997.8

Table I: Throughput affected by wireless capacity in the multi-flow scenario (Manhattan map)

A. Manhattan Map

We numerically solve the optimization problem OPT and demonstrate flow-level performance gains. As shown in Figure 3, we use a 4×4 grid road network with one-way streets in alternating directions. Since this is a simplified model of the Manhattan area in New York City, we call it *"Manhattan map"*. At each intersection, a vehicle goes straight with probability 0.75 and makes a turn with probability 0.25.



(a) Single Flow (b) Multiple Flows

Figure 3: Manhattan map: 4×4 grid road network

First, we consider the single-flow scenario shown in Figure 3(a). The weight affiliated with cost minimization is chosen as $w = 0$. Throughput as a function of the wireless capacity at intersections is given in Figure 2 (Section I). This clearly shows that joint utilization of transportation and wireless networks, i.e., OPT algorithm, is better than utilizing one network alone. If wireless capacity alone is used (for example, multi-hop relaying), wireless capacity ($y = x$ line) will be an upper bound. If transportation capacity alone is used, only 6.75% of θ is achieved. However, with joint utilization, wireless capacity equal to 10% of θ improves the throughput to 40% of θ (6.4x gain), and wireless capacity equal to 37% of θ is sufficient to achieve the full data loading rate at the source (16x gain). Note that the "transportation" part of capacity will decrease when the number of hops between the source and the destination increases, and the "wireless" part of capacity will decrease when the number of flows grows.

Next, we consider the multi-flow scenario shown in Figure 3(b). Table I also demonstrates that adding a small amount of wireless capacity results in significant throughput gain under OPT algorithm over the "transportation-only" scheme (5.3x gain with $c = 100$ Mbps and 9.0x gain with $c = 200$ Mbps).

We investigate the congestion control ability of OPT algorithm by comparing it with shortest-path routing, which is "congestion-unaware." Specifically, we compare the flow-level results between OPT and shortest-path routing for the four-flow scenario in Figure 3(b), using $c = 100$ Mbps. In the two cases in Table II, we respectively let flow 2 and flow 4 use a lower loading rate than the others. With shortest-path routing, the low-source-rate flow almost gets starved, while under our OPT scheme, all flows are quite fairly treated. In case (a), Jain's fairness index [20] improves from 0.8224 to 0.9332, and in case (b) the improvement is even higher: from 0.7370 to 0.9402. The aggregate throughput also increases under OPT.

(a) Flow 2 uses a lower loading rate			
Flow ID	θ_f (Mbps)	Throughput (Mbps)	
		Shortest-Path	OPT
1	400	171.21	167.15
2	400	124.41	184.64
3	400	124.02	147.86
4	100	28.20	81.91
Total	1300	447.83	581.5

(b) Flow 4 uses a lower loading rate			
Flow ID	θ_f (Mbps)	Throughput (Mbps)	
		Shortest-Path	OPT
1	400	174.37	178.14
2	100	40.28	100.00
3	400	120.28	146.14
4	400	42.39	100.00
Total	1300	377.33	524.29

Table II: OPT vs shortest-path (Manhattan map)

B. Boston Map

We provide flow-level and packet-level simulation results for the Cambridge urban area in Boston, MA (called *"Boston map"*) with real-world traffic statistics data[5] for each major road segment within this area (see Figure 4), reported online by Massachusetts Highway Department in the United States [21]. With these data, we calibrate vehicles' Markovian mobility model which further matches a common real-world phenomenon that most of the traffic tends to go in a straight line and drivers try their best to avoid U-turns, based on an optimization framework detailed in Appendix A.

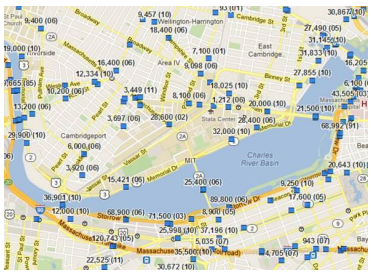


Figure 4: The Cambridge area of Boston: map and daily traffic volume data from [21]

A single-flow scenario is shown in Figure 5. The yellow triangle and rectangle respectively represent the source and destination. Each value shown near one vertex of a road indicates the data rate in Mbps on the edge of which this vertex is the end vertex. Figure 5 demonstrates the effect of cost penalty

[5]The data is recorded for annual average daily traffic (ADDT).

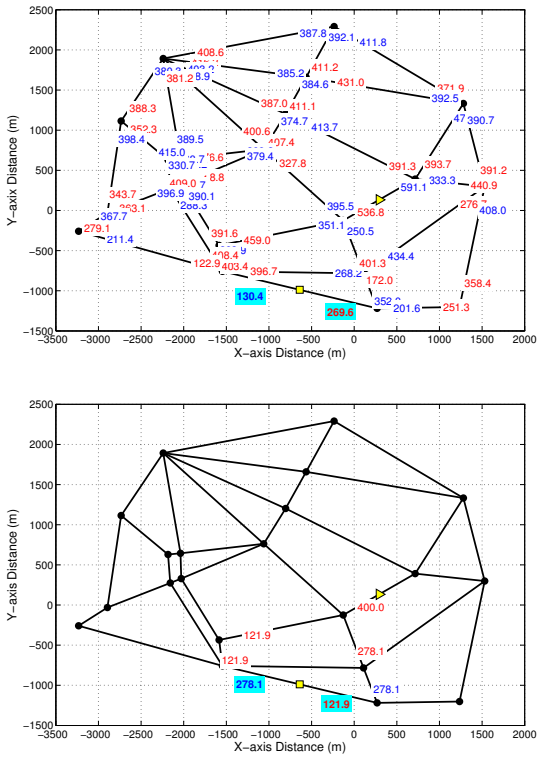


Figure 5: Cost minimization with $w = 0$ for top and $w = 5 \times 10^{-5}$ for bottom (Boston map, flow-level)

w appended to the throughput maximization objective. When $w > 0$, the algorithm reduces "data flooding" by using fewer and shorter paths. If w is below an appropriate level, maximum throughput can be achieved. However, once w becomes too large, the loading rates at sources have to be lowered to avoid cost, and as a result, the throughput will be negatively impacted. In this example, both cases achieve the full capacity, but with $w = 5 \times 10^{-5}$ (bottom) data flows are limited to two main paths. Note that this Boston map is a "closed world," so even relying on the transportation network only will result in full capacity (but with large delay). In a real-world network, the throughput improvement seen in Manhattan map and delay improvement seen in Boston map can be achieved using OPT algorithm.

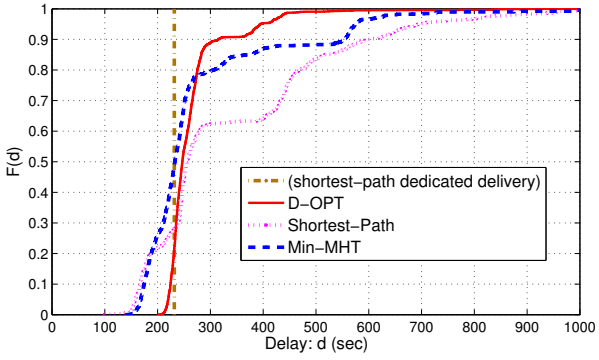


Figure 6: CDF of the packet delivery delay (Boston map)

Next, we perform packet-level simulations to evaluate the performance of our distributed routing/congestion control algorithm, called D-OPT for short. In this simulation, road intersections are placed on Cartesian coordinates according to their latitudes and longitudes and connected by roads. The number of vehicles placed on each edge i before a simulation starts is calculated as part of the mobility model calibration process, and totally 1636 vehicles are placed on the map. We run the simulation for 2000 sec with each time slot equal to 0.5 sec. The single-flow source-destination pair is the same as in Figure 5. We use the penalty weight $w = 5 \times 10^{-5}$ to solve OPT a priori for our distributed algorithm. By comparing the achieved data rates on all edges with the corresponding flow-level result, we find that D-OPT approximates OPT's solution quite well. The figure is omitted here due to space limitation.

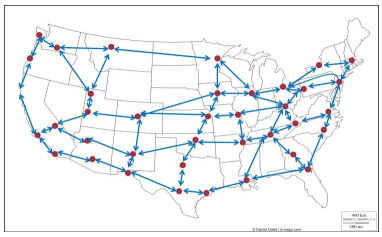


Figure 7: National highway map

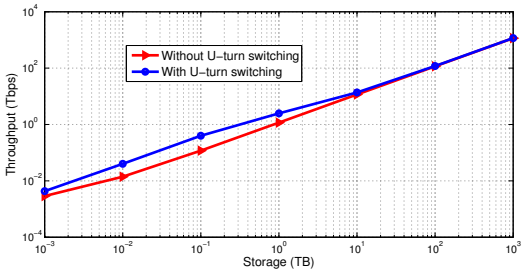


Figure 8: National highway map: throughput vs. storage, and the advantage of wireless U-turn switching

We compare three distributed algorithms: shortest-path, Min-MHT and D-OPT. Min-MHT ("minimum mean hitting time") routes packets to the neighboring edge with minimum expected delay (counted in hops). Specifically, for each flow f, we change d_f and $\bar{d}_f$ to two absorbing states in the transition matrix $\mathbf{P}$ and solve a system of linear equations to obtain the expected hitting time on either absorbing state from every other edge. Hence, this algorithm requires knowledge of the mobility model and pre-computation of expected delay values (similar to the one in [12]). D-OPT requires some more system information to do pre-calculation and further requires limited infrastructure support in online decision-making. In Figure 6, CDFs of end-to-end packet delivery delay are plotted for the three distributed algorithms. The dotted brown line represents the time T_{sp} for a dedicated vehicle (e.g., UPS truck) to physically deliver a hard disk which stores infinite data through the shortest path (and assuming continuous data flow), used as a benchmark time. D-OPT algorithm operates more conservatively within the "low-delay" region in this figure, since it is not as greedy or "delay-oriented" as the other two. However, it limits the delay of 90% of the packets under $1.26 \cdot T_{sp}$, which is much better than Min-MHT and shortest-path routing.

C. National Highway Map

Last, we perform a flow-level simulation on the "national highway map" in Figure 7. The real-world traffic data used for mobility model calibration are collected from [22]. An interstate highway is much longer than an urban road, so equation (1) indicates that a rich wireless capacity resource is available for U-turn switching and will support much higher throughput than only relying on intersection switching. Among all the highways, the median of the wireless capacity is around 250 Gbps (with a 200 Mbps DSRC rate). We measure the VBN capacity from New York City to San Diego (using a very large source loading rate). As claimed in Section I, VBN provides an "information pipe" with a capacity of the order of Tbps. For example, with a 1 TB hard disk installed in each vehicle, a capacity of 2.47 Tbps can be provided by the VBN between these two cities (with wireless U-turn enabled). Figure 8 shows the capacity scaling with the hard disk storage size (at least 1 GB). Wireless U-turns can achieve a gain as high as 3.4 (with about 100 GB storage size) in this example. When storage decreases, the need of U-turn switching is reduced since the data rates on wrong paths might not be that high. When storage increases, the improvement due to U-turn switching can be ignored compared to the huge transportation capacity.

V. Conclusion

In this paper, we build a model and design the architecture for a new type of backbone network composed of vehicles with uncontrolled mobility, which can jointly utilizes both transportation and wireless capacity to provide high-throughput data delivery service. We characterize its network capacity and apply convex optimization to seek a joint routing/congestion control scheme that achieves both throughput maximization and delay reduction. A distributed algorithm is also developed to approximate the optimal resource allocation. Simulations based on real-world maps and traffic statistics demonstrate significant gains over existing routing algorithms in different aspects, such as throughput, fairness and delay performance.

Appendix

A. Mobility Model Calibration Based on Real Traffic Data

Let $\boldsymbol{\pi} = [\pi_i]$ denote the steady-state distribution of the Markov chain. Statistically, π_i equals the fraction of vehicles on edge i. To get π_i, we calculate the steady-state amount of vehicles $n_i = \nu_i l_i / v_i$ by Little's Law [23], obtain $\pi_i = n_i / \sum_{j\in\mathcal{E}} n_j$, and formulate the following linear program:

$$\max_{\{\mathbf{P},\ \alpha\in[0,1],\ \beta\in[0,1]\}} \quad \alpha - \beta \tag{13}$$

$$s.t. \quad \boldsymbol{\pi} = \boldsymbol{\pi}\mathbf{P}; \tag{14}$$

$$\sum_{j\in\mathcal{N}_i} p_{ij} = 1,\ \forall i \in \mathcal{E};\quad p_{ij} \geq 0,\ \forall i,j \in \mathcal{E}; \tag{15}$$

$$\sum_{j\in\mathcal{N}_i^s} p_{ij} \geq \alpha,\ \forall i\ s.t.\ \mathcal{N}_i^s \neq \emptyset; \tag{16}$$

$$\sum_{j\in\mathcal{N}_i^r} p_{ij} \geq \alpha,\ \forall i\ s.t.\ \mathcal{N}_i^s = \emptyset,\ \mathcal{N}_i^r \neq \emptyset; \tag{17}$$

$$\sum_{j\in\mathcal{N}_i^u} p_{ij} \leq \beta,\ \forall i\ s.t.\ \mathcal{N}_i^u \neq \emptyset,\ \mathcal{N}_i^u \subset \mathcal{N}_i; \tag{18}$$

Given $\boldsymbol{\pi}$, only solving the system of linear equations (14) usually results in an infinite number of solutions for $\mathbf{P}$. Constraints (15) will limit $\mathbf{P}$ to the given graph structure. The motivation of adding the other constraints and choosing (13) as the objective function is to jointly maximize the probability of a straight-line move α and minimize the probability of a U-turn β. Whether a road switching from edge i to edge j is regarded as a straight-line move, regular turn or U-turn is determined by the angle of switching ϑ_{ij} for a vehicle to make. For each i, $\mathcal{N}_i^s$, $\mathcal{N}_i^r$ and $\mathcal{N}_i^u$ represent the sets of edges, to which a road switching respectively correspond to the above three switching types. More specifically, $\mathcal{N}_i^s \triangleq \{j : |\vartheta_{ij}| \leq \vartheta^s\}$, $\mathcal{N}_i^u \triangleq \{j : |\vartheta_{ij}| \geq \vartheta^u\}$ and $\mathcal{N}_i^r \triangleq \mathcal{N}_i - \mathcal{N}_i^s - \mathcal{N}_i^u$. In our simulations, we pick the boundary angle values as $\vartheta^s = 45^\circ$ and $\vartheta^u = 170^\circ$.

References

[1] "IBM: What is big data? - Bringing big data to the enterprise," http://www-01.ibm.com/software/data/bigdata.

[2] X. Hong, K. Xu, and M. Gerla, "Scalable routing protocols for mobile ad hoc networks," *IEEE Network*, vol. 16, no. 4, pp. 11–21, 2002.

[3] S. Ali, J. Qadir, and A. Baig, "Routing protocols in delay tolerant networks - a survey," in *Proc. of the 6th International Conference on Emerging Technologies (ICET)*, 2010, pp. 70–75.

[4] F. Li and Y. Wang, "Routing in vehicular ad hoc networks: A survey," *IEEE Vehicular Technology Magazine*, vol. 2, no. 2, pp. 12–22, 2007.

[5] M. Grossglauser and D. Tse, "Mobility increases the capacity of ad hoc wireless networks," *IEEE/ACM Trans. on Networking*, vol. 10, no. 4, pp. 477–486, 2002.

[6] J. Jose, A. Abdel-Hadi, P. Gupta, and S. Vishwanath, "On the impact of mobility on multicast capacity of wireless networks," in *Proc. of IEEE INFOCOM*, 2010, pp. 141–145.

[7] Z. J. Haas, J. Y. Halpern, and L. Li, "Gossip-based ad hoc routing," *IEEE/ACM Trans. on Networking*, vol. 14, no. 3, pp. 479–491, 2006.

[8] J. Luo, P. T. Eugster, and J.-P. Hubaux, "Route driven gossip: Probabilistic reliable multicast in ad hoc networks," in *Proc. of INFOCOM*, vol. 3, 2003, pp. 2229–2239.

[9] N. Wisitpongphan, O. K. Tonguz, J. Parikh, P. Mudalige, F. Bai, and V. Sadekar, "Broadcast storm mitigation techniques in vehicular ad hoc networks," *IEEE Trans. on Wireless Communications*, vol. 14, no. 6, pp. 84–94, 2007.

[10] A. Shokrollahi and M. G. Luby, *Raptor codes*. Now Publishers, 2011.

[11] H. Wu, R. Fujimoto, R. Guensler, and M. Hunter, "MDDV: a mobility-centric data dissemination algorithm for vehicular networks," in *Proc. of ACM VANET*, 2004, pp. 47–56.

[12] J. Zhao and G. Cao, "VADD: vehicle-assisted data delivery in vehicular ad hoc networks," *IEEE Trans. on Vehicular Technology*, vol. 57, no. 3, pp. 1910–1922, 2008.

[13] F. Malandrino, C. Casetti, C.-F. Chiasserini, and M. Fiore, "Content downloading in vehicular networks: What really matters," in *Proc. of IEEE INFOCOM*, 2011, pp. 426–430.

[14] V. Ramaiyan, E. Altman, and A. Kumar, "Delay optimal scheduling in a two-hop vehicular relay network," *Mobile Networks and Applications*, vol. 15, no. 1, pp. 97–111, 2010.

[15] D. Goodman, J. Borras, N. B. Mandayam, and R. D. Yates, "Infostations: A new system model for data and messaging services," in *IEEE Vehicular Technology Conference*, vol. 2, 1997, pp. 969–973.

[16] A. Pentland, R. Fletcher, and A. Hasson, "Daknet: Rethinking connectivity in developing nations," *Computer*, vol. 37, no. 1, pp. 78–83, 2004.

[17] Y. Ding and L. Xiao, "SADV: Static-node-assisted adaptive data dissemination in vehicular networks," *IEEE Trans. on Vehicular Technology*, vol. 59, no. 5, pp. 2445–2455, 2010.

[18] R. Shirani and F. Hendessi, "A Markov chain model for evaluating performance of store-carry-forward procedure in VANETs," in *IEEE International Conference on Communication Systems*, 2008.

[19] P. N. Michelini and E. J. Coyle, "Mobility models based on correlated random walks," in *Proc. the ACM International Conference on Mobile Technology, Applications, and Systems*, 2008, pp. 86:1–86:8.

[20] R. Jain, D.-M. Chiu, and W. R. Hawe, "A quantitative measure of fairness and discrimination for resource allocation in shared computer system," *DEC Research Report TR-301*, 1984.

[21] "The Massachusetts Highway Department annual traffic data report (interactive map)," http://mhd.ms2soft.com/tcds/tsearch.asp?loc=Mhd.

[22] "The AA Roads Interstate-Guide," http://www.interstate-guide.com.

[23] L. Kleinrock, *Queueing Systems, Vol. I: Theory*. Wiley Interscience, 1975.