Knowledge-Based Opportunistic Forwarding in Vehicular Wireless Ad Hoc Networks

Jason LeBrun*, Chen-Nee Chuah*, Dipak Ghosal†, Michael Zhang‡

*Dept. of Electrical and Computer Engineering, University of California, Davis

†Dept. of Computer Science and Engineering, University of California, Davis

‡Dept. of Civil Engineering, University of California, Davis

***Abstract*— When highly mobile nodes are interconnected via wireless links, the resulting network can be used as a transit network to connect other disjoint ad-hoc networks. In this paper, we compare five different opportunistic forwarding schemes, which vary in their overhead, their success rate, and the amount of knowledge about neighboring nodes that they require. In particular, we present the MOVE algorithm, which uses velocity information to make intelligent opportunistic forwarding decisions. Using auxiliary information to make forwarding decisions provides a reasonable trade-off between resource overhead and performance.**

I. Introduction

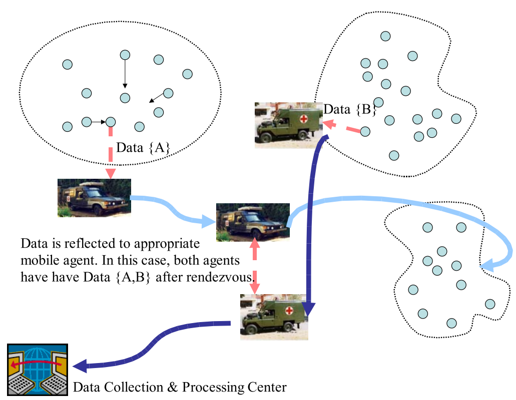

Fig. 1. Vehicles act as mobile routers (MRs) to connect disconnected sensor networks to a known destination.

The FCC has recently allocated the 5.85-5.925GHz portion of the spectrum for inter-vehicle communications and vehicle-to-roadside communications, known as Dedicated Short Range Communications (DSRC) [1]. One of the DSRC's key application areas is Intelligent Transportation Systems (ITS) which aims to improve driver safety by exchanging information among vehicles [2]. In this paper, we consider an emerging application scenario where vehicles are used as *mobile routers (MRs)* to collect and deliver data between static nodes (e.g., sensor networks and central server) that are otherwise disconnected [3], [4], as shown in Fig. 1. Such *transit* network comprising of highly mobile nodes fall into a newly defined class of networks known as *Delay Tolerant Networks* (DTNs) [5].

In DTNs, end-to-end connectivity is not guaranteed, and hence messages may need to be cached for an arbitrary amount of time at the mobile carrier or intermediate static nodes. Inter-vehicle communications can only occur when two MRs are within the transmission range of each other, which we refer to as *opportunistic forwarding*. One key design question is: *how do the MRs decide to (or not to) forward the data to adjacent MR when the global topology is unknown and changing rapidly*. The INFOSTATION project [6] studies opportunistic forwarding of information from mobile nodes to a static server. The Data Mule project [4] used mobile nodes to collect data from a source, and then deliver to a destination. But neither of them consider opportunistic forwarding between the *mobile nodes*. The message ferrying project [7] proposes a novel approach of controlling the physical paths of mobile routers, to optimize the delivery patterns in the network.

This paper considers a scenario where mobility of vehicles (MRs) cannot be controlled. We assume each MR knows its own position through Global Positioning System (GPS). The destination node is static and its position is known globally. We explore different knowledge-based opportunistic forwarding schemes that leverage different amounts of information exchanged between vehicles to make an intelligent forwarding decision. The design goal is to deliver the data successfully and with minimal delay to a destination. We use QualNet [8] to simulate and compare five algorithms: NoTalk, Broadcast, Location-Based, MOVE, and MOVE-Lookahead. The location-based algorithm makes use of relative position between vehicles and destinations to make a forwarding decision, while MOVE-vector and MOVE-lookahead take into account the relative velocities of the vehicles. We quantify the trade-offs between performance gain and overhead of these schemes.

II. Background

Our work in opportunistic forwarding is largely motivated by the VMesh Demand-Response [3] project. This project proposes a solution for exchanging pricing info from the utility companies and usage information from homes. Vehicles are used as mobile routers, to move data between a power plant gateway and aggregation points for residential areas. In such an application, the mobile routers may experience intermittent connectivity—that is, there may not be an instantaneous end-to-end path between the source and the destination at a given point in time. As a result, mobile routers will need to hold on to data, and then forward it at a later time, after physically moving. The original VMesh paper considered only the scheme in which no nodes communicate (NOTALK), and the scheme in which nodes always communicate (BROADCAST).

The NOTALK scheme is adapted from the communication method used in the Data Mule [4] project. In this project, mobile collection agents called "mules" move about a scenario, collecting data from data sources, to deliver them eventually to a central gateway.

The key question that this paper would like to answer is: when two nodes participating in an opportunistic forwarding system come in contact with each other, what is the best way to decide whether to hold or forward data? This problem is very similar to work that is being done in Delay-Tolerant and Disruption-Tolerant networking. Studies of DTN also consider cases in which the edges of the network graph are time-varying.One of the most interesting results of the DTN work is the extension of Dijkstra's shortest path algorithm [5] into a modified shortest-path algorithm that can take into account time-varying edge weights. Such an algorithm has the potential for providing an upper bound in performance, since the assumption is that all connectivity information for all times of interest is known at the start.

One of the methods we consider for the opportunistic forwarding decision making process uses location information of nodes to make a decision. It should be noted that numerous *location- based* routing algorithms have been proposed, where location information derived from GPS or location service is used to determine the routes to forward data packets. However, these routing algorithms consider a more standard end-to-end type routing, rather than a situation in which physical movement of the nodes is necessary for data to eventually reach the destination. The most basic location-based routing algorithms are the three variations of the *greedy algorithm* [9]: (a) Most Forwarding Progress within radius R (MFR), (b) Nearest with Forwarding Progress (NFP), and (c) compass routing. MFR tries to minimize the number of hops by minimizing the remaining distance toward the destination. NFP chooses the closest node that can still provide forwarding progress as the next hop. Compass routing tries to maintains on the straight line toward the destination so it selects the node that has the minimum perpendicular distance to the straight line connecting source and destination. *Greedy Perimeter Stateless Routing (GPSR)* [10] is a more sophisticated algorithm that uses MFR greedy forwarding, but switch to perimeter routing (i.e., route around the perimeter of faces according to the right hand rule) when MFR causes local optimizations.

We were not able to find any literature that has studied the use of location information in an opportunistic forwarding setting. Regardless of whether or not end-to-end connectivity exists, however, the idea is the similar: we would like to forward data to a node that is physically closer to the destination.

III. Opportunistic Forwarding Strategies

This section describes five opportunistic forwarding strategies, including two basic methods for baseline comparisons, and three knowledge-based schemes that leverage location and mobility information exchanged between nodes.

All of these methods use a HELLO-RESPONSE technique for detecting approaching MRs. MRs carrying data send out periodic HELLO beacon. If a neighboring node hears a HELLO message, it will send a RESPONSE message to announce its presence. This message will also contain mobility information about the responding node.

The five strategies differ in how an MR decides to which neighbor to forward the data , as described in the following subsections.

A. NOTALK

NOTALK is similar to the strategy used in the Data Mule project [4]. An MR will accept data from a data source and cache it in its buffer. The MR will continue to carry this data until it receives a RESPONSE from the destination, at which point it will deliver the data and remove it from its cache. In this method, MRs do not communicate with each other. NoTalk uses a minimum amount of system-wide buffer space, since only one copy of each packet propagates throughout the system. However, the delay for this case is maximal, since packet delivery time is only as fast as the movement of the MR nodes. Since the simulation time is bounded, the packet delivery success rate is very small.

B. BROADCAST

BROADCAST is the other extreme compared to *NoTalk*, where an MR unconditionally exchange data with every other MR it meets—whenever a MR hears a RESPONSE from another MR, it will forward all of the data in its buffer to the neighboring MR The redundancy provided by such a strategy helps to maximize the packet delivery success rate and to decrease the overall delay. However, *BROADCAST* incurs very high overhead in terms of number of messages exchanged and buffer space.

C. Location-based

Location-based is a form of greedy, geographical-based routing [11]. An MR forwards data to a responding neighbor only if the neighbor is closer to the destination than its own current position.

D. MoVe

The *Motion Vector (MoVe)* scheme leverages the knowledge of relative velocities of an MR and its neighboring nodes to predict the *closest distance* that they are predicted to get to the destination, following their current trajectories (straight-line paths). This information is piggy-backed on the HELLO-RESPONSE messages. The *closest distance* between a node C (the current node) and destination D is denoted as d_C and is determined as follows:

1) Let $\vec{v_C}$ be the motion-vector of a node, pointing in the direction that the node is moving.
2) A second vector is drawn from the node to the destination, and is denoted by $\vec{CD}$.

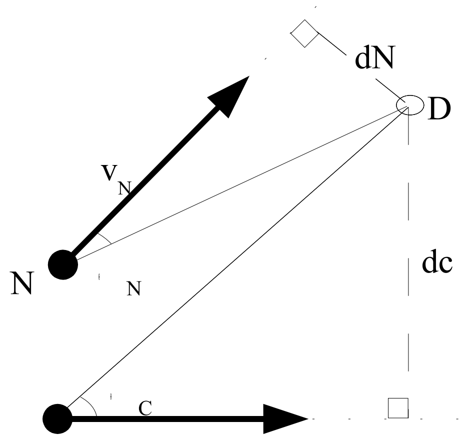

Fig. 2. Parameters for the MpVe algorithm.

3) The angle between these two vectors is given by:

$$\theta = cos^{-1}(\frac{\vec{CD} \cdot \vec{v_C}}{|\vec{CD}||\vec{v_C}|}) \quad (1)$$

.

4) The predicted closest distance of the node to the destination is $sin(\theta) * \vec{CD}$

Furthermore, if $\theta > 90$, the node is heading away. If $\theta < 90$, the node is moving towards the destination. The node may also be standing still. With N as the neighbor node, Table I summarizes the rules for forwarding messages to the neighbor.

TABLE I
RULES FOR FORWARDING DECISIONS FOR THE *MOVE* ALGORITHM

Current	Neighbor	Forward
Still	Still	No
Still	Away	No
Still	Towards	Yes
Away	Still	Yes
Away	Away	if $\vec{ND} < \vec{CD}$
Away	Towards	Yes
Towards	Still	No
Towards	Away	No
Towards	Towards	if $d_N < d_C$

E. MOVE-Lookahead

The *MOVE-Lookahead* method uses the basic rules as in MOVE, with one modification. Now, each MR “looks ahead” for its next *waypoint*, where its trajectory changes. If a node C change its directions before it reaches the point at which it will be closest to the destination, the distance between the waypoint and the destination is used instead as an estimate of d_C in the forwarding decision phase.

IV. PERFORMANCE EVALUATION

A. Simulation Setup

We consider a terrain size of 4000m x 4000m with one source and one destination located at opposite corner, as shown in Figure 3. There is an additional rim of 1000m as a border zone. Nodes move in piece-wise linear fashion, following city street structures. Nodes move between 5-10 m/s, pausing for 10 seconds at each waypoint (point at which it changes course, e.g., intersections). A packet of 512 bytes is sent every 10 seconds, for 1000 seconds. We run the simulation for 5000 seconds, to allow packets to propagate through the system after the data sending phase is complete. Nodes with cached data broadcast HELLO beacons at 10-second intervals. We vary the total number of nodes and message forwarding schemes in the simulations.

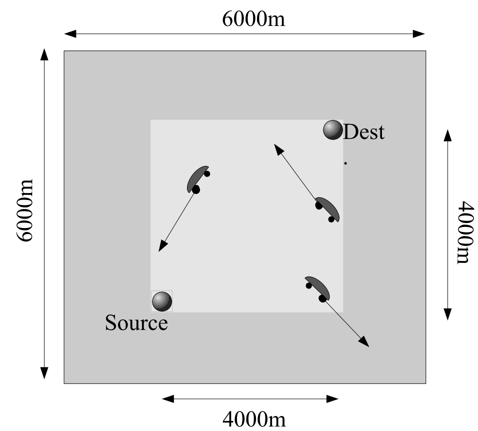

Fig. 3. Simulation setup.

B. Results

The simulation results compare the packet success rate of the system, the number of control overhead packets (HELLO/REPONSE), and the average buffer space used per node. Figure 4 shows that data success rate for each of the five schemes tested, as a function of the number of nodes in the network. Since geographical size of the area is fixed, this also represents a density of nodes in the network. As is expected, increasing the density will improve the success rate of all schemes that involve communications between routers. Comparing the different schemes, we can see that as the number of nodes reaches an asymptotic performance point, the MOVE scheme performs consistently better than the scheme that uses only location information. However, it still does not match the success of the aggressive broadcasting scheme. Furthermore, we can see that the performance is essentially the same between the normal MOVE scheme, and the MOVE-lookahead scheme. So, our preliminary results show that knowing only the next waypoint does not provide any significant improvement in the algorithm, and so this informationis unnecessary overhead in the control packets.

In Figures 5 and 6 we compare the overhead of different algorithms in terms of two different resources: buffer space and communication bandwidth. As expected, the overhead for the BROADCAST method is high both in terms of buffer usage as well as in the number of control packets generated. The control packet overhead is most likely slightly lower for the MOVE cases that for the LOCATION case because as more packets are delivered, there are less nodes that need to send HELLO beacons out to announce that they are holding data.

Finally, in Figure 7, we compare the end-to-end packet delivery time for the various schemes. In these simulation scenarios, there is no statistically discernible difference in end-to-end delivery time for the various router communications schemes.

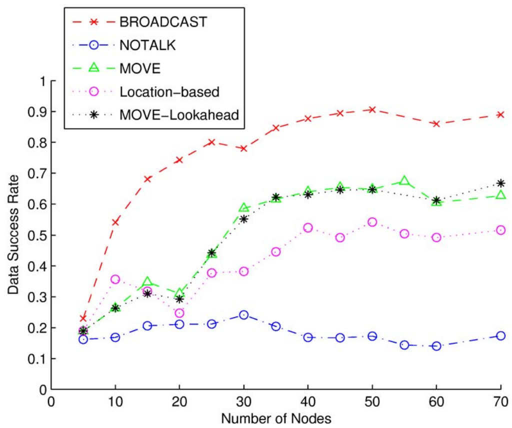

Fig. 4. Data success rate as a function of the number of nodes.

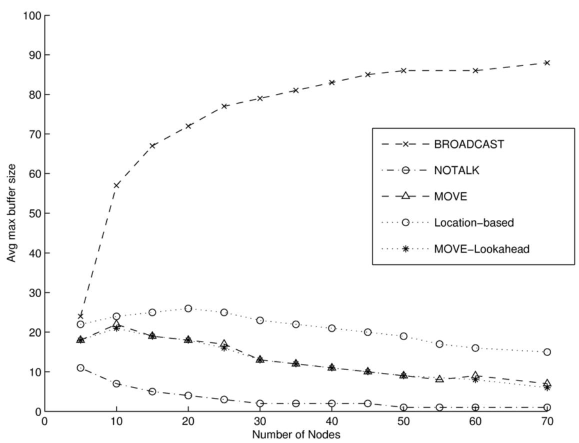

Fig. 5. System-wide buffer usage as a function of the number of nodes.

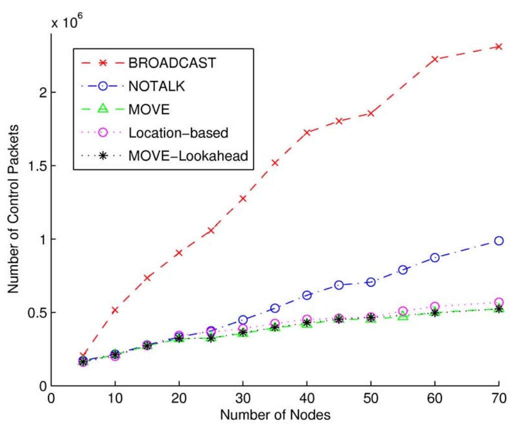

Fig. 6. Control-packet overhead as a function of the number of nodes.

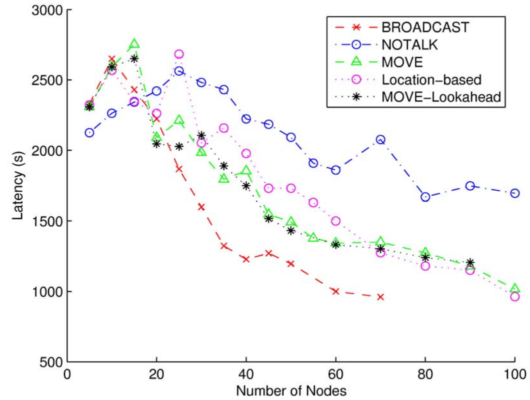

Fig. 7. End-to-end delay as a function of the number of nodes.

C. Impact of Node Mobility Models

The results based on randomized linear piecewise motion give us some preliminary insights into how different MR communications schemes perform in an asymptotic case, where a priori knowledge of the node trajectory is limited. Our simulation studies show that by leveraging the current velocity information of the mobile node, MOVE algorithm can achieve reasonably high success rate and low end-to-end latency with substantially less buffer space and communication overhead, compared to a BROADCAST approach. Even with this study, however, we are cautious about making generalization about how the different algorithms perform under different mobility models. For example, while our existing results apply in cases where node mobility is random such as ZebraNet Wildlife Tracker [12], they may not hold in other extreme scenarios where the routes of mobile routers are fixed (e.g., buses with fixed routes and schedules) or can be controlled (e.g., Message Ferrying Project [7]).

In order to accurately assess the utility of opportunistic forwarding algorithm, and to identify issues that would arise in a real-world implementation, it is crucial to be able to generate simulations that can accurately model different real-world situations. In our on-going and future work, we will study the impact of different node mobility models on the design of MOVE and its variations. To achieve this, we have acquired traces of actual buses in the public transit systems of various cities. The data consists of the GPS location of buses, reported once ever 90 seconds. This can give us a realistic trace of vehicles that will show instances of how potential mobile routers might move when constrained by city streets, and the presence of traffic.

In our preliminary study, we experimented with a realistic scenario consisting of a topology of about 12 kilometers by 12 kilometers. The same type of data transmission scenario is used: a source and a destination are fixed at a significant distance from each other, but both are in locations that are frequented by buses. Data packets of 512 bytes are generated for 1000 seconds, one every 10 seconds. The nodes move based on GPS data collected from actual buses in the San

Francisco MUNI system, as shown in Figure 8, through the NextBus project [13]. The data provided gives enough granularity and accuracy to create a simulation that will have mobility characteristics very similar to a real-world scenario.

Fig. 8. A map of the area corresponding to the realistic mobility data.

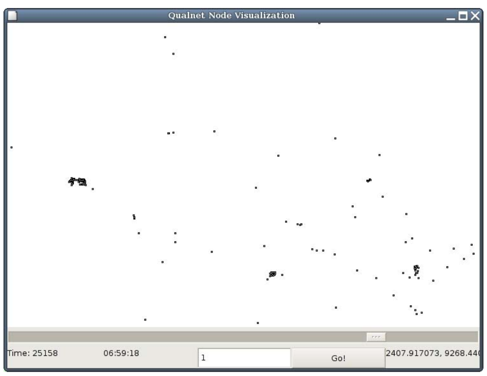

Fig. 9. Visualization of the mobile routers.

Our initial simulation results show that behavior of the five algorithms is in fact different using the realistic data from NextBus. *Location-based routing algorithm* out-performs other algorithms in this scenario, where the buses follow known, fixed routes. This, in itself, is an important conclusion, and the next step is to investigate how we can leverage the a priori knowledge of the lay-out of buses and their schedules to optimize the opportunistic forwarding performance.

We are currently in the process of designing a visualization tool to help tailor the simulations accurately. Ultimately, this tool will help us to correlate potential data sinks and sources with bus stops. We also hope to be able to visualize actual data flow patterns through the opportunistic-forwarding system, in order to get a more intuitive understanding of the way that the different algorithms work.

V. Conclusion

Intelligent forwarding algorithm is a crucial component of a vehicular-based ad hoc networks that is used to relay data between a known source and destination pair. We have presented and evaluated five different opportunistic forwarding schemes, in the context of the VMesh Demand-Response project. In this scenario, information needs to travel from a fixed source to a fixed destination, via vehicular-based mobile routers. We have shown that exchanging mobility information can help achieve a trade-off between performance (success rate) and overhead (control messages and buffer space). Our preliminary study using realistic mobility data has revealed that for realistic situations, in which non-moving buses may still attempt to route data, or buses may become stuck somewhere for a long time, modifications should be made to the algorithm in order to avoid failures. Our next step in this work will be to do a more in-depth study of the relationship between opportunistic routing and realistic vehicular movement. Furthermore, we will consider modifications that involve caching and redundancy in the system. We believe that improvements might be made by duplicating packets in certain cases, instead of handing them off. Finally, we intend to look at ways that determinism in mobile router movement can be leveraged to improve opportunistic decision-making. For example, when the mobile routers are buses following a fixed route and schedule.

The use of intelligent decision-making algorithms for opportunistic forwarding will allow less connected mobile networks to have a higher degree of success for data delivery.

Acknowledgements

We are grateful to Tito Vandermeyden and George Mount from NextBus for sharing the San Francisco MUNI location traces. This work was supported in part by the GAANN Fellowship and UC Micro program.

References

[1] Berkeley PATH Project. `http://path.berkeley.edu/dsrc/`.
[2] M. Bechler, W. Franz, and L. Wolf. Mobile Internet Access in FleetNet. `http://www.fleetnet.de/`.
[3] C. Chuah, D. Ghosal, H. Chang, J. Anda, and M. Zhang. Enabling energy demand response with vehicular mesh networks. Conference on Mobile and Wireless Communication Networks. IEEE, 2004.
[4] R. Shah, S. Roy, S. Jain, and W. Brunette. Data mules: Modeling a three-tier architecture for sparse sensor networks. Sensor Network Protocols and Applications. IEEE, 2003.
[5] K. Fall, R. Patra, and S. Jain. Routing in a delay tolerant network. SIGCOMM. IEEE, 2004.
[6] D. Goodman, J. Borrs, N. Mandayam, and R. Yates. INFOSTATIONS and : A New System Model for Data Messaging Services. Vehicular Technology Conference. IEEE, May 1997.
[7] E. Zegura, M. Ammar, and W. Zhao. A message ferrying approach for data delivery in sparse mobile ad hoc networks. MobiHoc. IEEE, 2004.
[8] QualNet. `http://www.scalable-networks.com/`.
[9] X. Lin and I. Stojmenovic. GPS Based Distributed Routing Algorithms forWireless Networks. Unpublished manuscript, 2000.
[10] Brad Karp and HT Kung. GPSR: Greedy Perimeter Stateless Routing for Wireless Networks. Proc. of 6th ACM/IEEE MobiCom, pages 243–254, 2000.
[11] H. Takagi and L. Kleinrock. Optimal transmission ranges for randomly distributed packet radio terminals. *IEEE Transaction on Communications*, 32(3):246–257, 1984.
[12] ZebraNet. http://www.princeton.edu/ mrm/zebranet.html.
[13] NextBus. http://www.nextbus.com/.