Utility-Driven Information Dissemination in Vehicle Ad-Hoc Networks

Daniel Greene Juan Liu Marc Mosko Jim Reich [*]
Yukio Hirokawa Tatsuo Mikami Tomoyoshi Takebayashi [†]

Abstract

We present an approach to managing vehicle network resources for diverse data traffic types, such as safety, traffic, and commercial. Our approach uses application-specific utility descriptions, which are reduced to simpler and more compact "microutility" approximations. These microutilities travel with the data to guide critical in-transit resource decisions such as dropping data, access to infrastructure networks, and storing data for later forwarding. Our system dynamically adjusts the quality of service in the network to support diverse application needs efficiently and fairly. This paper describes the system, the algorithms used, and presents the results of several simulation studies.

INTRODUCTION

Vehicle Ad-Hoc Networks (VANETs) and similar networks for pedestrians and bicyclists are very different from traditional ad-hoc networks. Node mobility, extremes in network density and topology from urban gridlock to rural traffic, and the rapidly changing information needs of moving participants make VANETS harsh and demanding networks. In this paper, we consider the problem of information dissemination in VANETs, where (1) the network consists of a multitude of data sources and data users; each vehicle is potentially a data source and user at the same time; (2) diverse types of applications, such as situational awareness, traffic management, and commercial services share the same networking infrastructure; and (3) unlike in typical communication frameworks where individual data items are the focus, each application outputs a *continuous stream* of data into the network. The goal of information dissemination is to best utilize network resources to serve the data needs of all users.

Several wireless communication standards (e.g. 802.11p and ARIB T55) have been developed specifically for vehicles, and there are also several general standards which may be useful to vehicles, e.g. WiMax, 802.11n and GSM. These standards allow multiple applications to share networking resources for vehicle-to-vehicle and vehicle-to-infrastructure

---

[*] Palo Alto Research Center, [greene,jjliu,mmosko,jreich]@parc.com
[†] Fujitsu Limited, y.hirokawa@jp.fujitsu.com, nra24256@nifty.com, take@labs.fujitsu.com

communication. However, while there is functionality to implement priority schemes in these standards, there is currently no established practice on the use of these standards, nor are there standard protocols for multihop communication or communication combining multiple network media.

This paper considers the quality of service (QoS) problems of information dissemination that occur when multiple competing applications share constrained communication resources. We propose a novel utility-driven information dissemination approach in which each application is assigned an application-specific utility function, specifying its desired spatial and temporal data delivery characteristics. Some applications, such as situational awareness, favor short-range, frequent, data updates while others (e.g., parking garage information) favor long-range coverage with lower data frequency. The utility specification can proactively and reactively shape the bandwidth usage to deliver data samples to where they are most needed.

This could be a quite complex process, but we provide a key simplifying element: the "microutility". Stated loosely, the utility function specifies the value of a *continuous stream of data* is to an application, and microutility indicates how valuable a particular *data packet* is at a given time and place. Microutility encodes a small amount of information in each packet to efficiently approximate the utility, allowing the system to treat packets in a distributed manner with little overhead. Collectively, an ensemble of packets approximates a utility-maximizing behavior. Microutilities are evaluated dynamically in-transit, allowing congested nodes to prioritize packets. This is more far flexible than specifying a fixed priority at the source. We focus on push-based systems because push-style communication is well-suited for VANET's many-to-many communication patterns and time-sensitive data. However, in principle, pull-style mechanisms can also participate in this utility-driven approach.

RELATED WORK

Our design extends our earlier work on variable-resolution information dissemination (VRID) [1] and makes use of several economic concepts. Others have proposed to use economic mechanisms (e.g. markets and auctions) to allocate computational and networking resources (e.g. [2], [3]). We have not adopted such a full market approach because of the complexity it would impose on individual elements of the system. In our system, costs are set algorithmically, utility is supplied by application writers on behalf of all expected consumers of information, and then approximated across an ensemble of packets.

Utility was initially introduced in networking to help understand QoS mechanisms [4] and to analyze existing network protocols [5]. Little work to date has used utility to directly guide the operation of the system; it has mainly been used via direct optimization of a utility function (e.g. [6]). In this work, we approximately optimize it in a distributed fashion, evaluating microutilities dynamically as the packets propagate.

A key subproblem in our system is multi-hop data dissemination. We have used Geocast [7, 8] methods to disseminate data through defined areas, but we have coupled this with "gossip" methods [9], a form of epidemic algorithm [10], to reduce redundant transmissions. Typical geocast approaches flood a data stream to a fixed pre-designated region, but in our approach, the geocast regions are modified on a per-packet basis within a stream of packets to achieve much greater utility. Based on utility specifications, users at different distances to the data source may receive a different amount of data. Another key technique we use is a utility-aware form of "anti-entropy" [10] to synchronize stored information between moving vehicles.

There is recent work evaluating peer-to-peer wireless networking using both vehicle and wireless simulation (e.g. [11], [12], [13]). Rather than having to design and maintain a new integrated simulator consisting of lower-fidelity vehicle and network simulators, our evaluation methodology is to link together two of the most popular, most sophisticated simulators: the NS-2 network simulator and the Paramics traffic microsimulator.

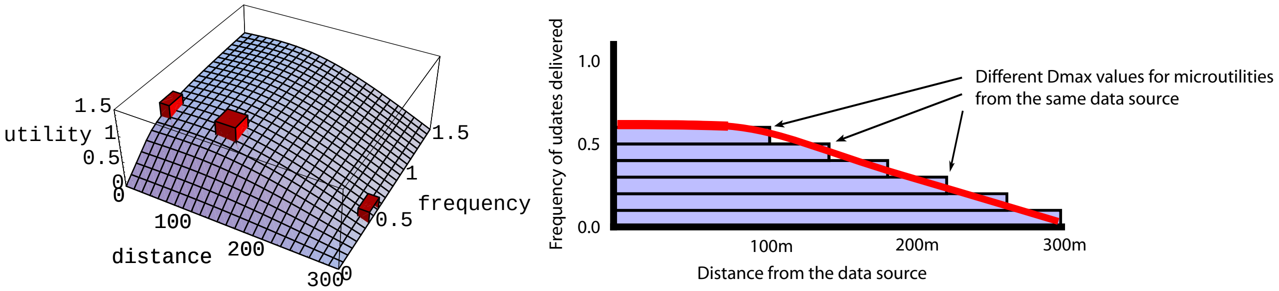
UTILITY

Our architecture bases its data delivery decisions on *utility functions*, application-defined functions describing how valuable data delivery is to recipients based on parameters such as the user's location and the timeliness of delivery. It is intuitively useful to think of utility as the amount that a recipient would pay for the data delivery (e.g. $10); this is not strictly correct, but we will follow the economist's convention and treat utility as a unitless, quantitative measure of satisfaction. Real-world utility functions can be complex and high-dimensional, impractical to capture and use in a system. Nevertheless, economists and system architects have found utility functions– even when they must be simplified somewhat — to be a useful way to describe economic behavior and prescribe useful system operation.

In this paper we will focus on applications in which multiple data sources (e.g. vehicles, traffic lights, gas stations, etc.) generate continuous streams of sampled, dynamically-varying data useful to multiple users; this many-to-many pattern common in many vehicle applications. Examples of applications having these kind of data traffic patterns include situational awareness applications, where vehicles continuously broadcast their location and velocity, traffic applications providing measurements of traffic density on urban streets, or safety applications where traffic signals broadcast their status.

We have chosen three salient features on which to base utility: time delay, distance from the source, and frequency of samples received, defining utility $U(\tau, d, f)$. The importance of time here is straightforward, since out-of-date information becomes increasingly inaccurate as time passes. Distance is important in many applications, since recipients closer to a source are more likely to be affected by that source and find the data relevant; however, this distance might not actually be euclidean distance, but rather some related metric, such as "time to arrival" at some point. The third dimension, frequency of samples received, is less obvious, but nonetheless important to many applications. For example, in a situational awareness, the frequency of data samples received will affect the accuracy of a Kalman filter's estimates of vehicle location.

Each application will supply its own utility function. For example, the situational awareness example would broadcast vehicle positions and velocities over a relatively small area, probably with high utility, while the garage data might be useful kilometers away. Our information dissemination system's primary function is to deliver packets to maximize the total utility across *all* applications.



(a) (b)

Figure 1: Illustrations of utility and VRID. (a) A utility template where the application writer has specified $d = 100$, $f = 0.5$, and $d = 300$. (b) VRID: disseminating information to different distances with variable delivery frequency.

In a push system, the utility must be defined at the data source. This is counterintuitive; recipients have their own utilities for data, and therefore might be expected to specify their own individual utility functions. However, in a push system, packets must be sent without knowledge of specific recipients, and must depend on a stochastic model of their most likely location and information needs. This greatly simplifies the operation of the system, which no longer must link senders directly with recipients. While developing utility functions may be complex, we have developed a library of templates, described in more detail in [14], to make the application writer's task simpler. Figure 1a shows the distance d and frequency f dimensions of one such template. In this illustration the application writer has specified that sending data samples at 0.5 samples per second a distance of 100 meters would meet the application needs well, and that utility drops to zero after 300 meters. The template makes assumptions to "fill in" the rest of the utility function. Using templates makes specifying utility only slightly more complicated than specifying a geocast region, but allows the full benefits of a sophisticated behavior specification to be realized.

Assuming there is only one application in the network with the prescribed utility function, a utility-maximizing system tends to deliver data more frequently to nearby distances and less frequently to far distances. We have coined the term "variable resolution information dissemination" (VRID) [1] for the data delivery pattern that users at different distances receive data at variable rates. It is easy to see qualitatively the connection between the utility function (Figure 1a) and the data delivery variation (Figure 1b). The utility function at a nearby distance saturates (arriving at a very small slope in the frequency direction) at a high data delivery frequency. In contrast, the utility function for further distances (e.g., 250m) saturates at a much lower frequency.

However, there still remains a question: which applications deserve the largest utility functions and consequently the largest allocation of network resources? We do not propose to solve that problem here; for now, we assume that this is handled as part of the utility function design, which must implement the requirements of some combination of policy processes and financial incentives.

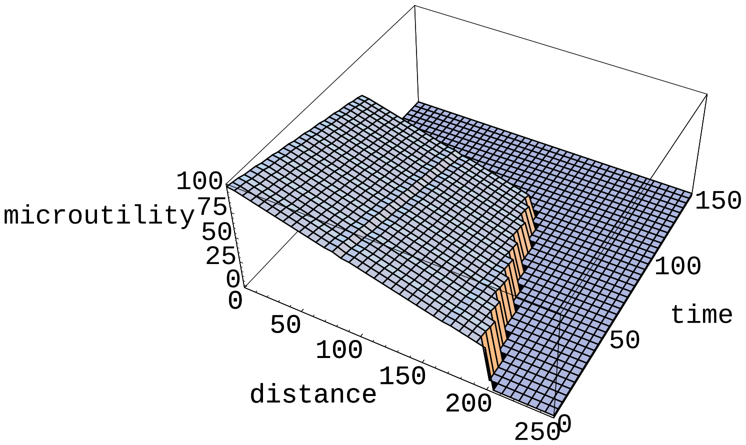


Figure 2: An example microutility: $T_{\text{max}} = 100$, $D_{\text{max}} = 200$, $D_{\text{last}} = 150$, $U_{\text{origin}} = 100$, $U_{\text{time}} = 65$, and $U_{\text{distance}} = 40$.

MICROUTILITY

While utility functions are good for specifying the patterns of information delivery, they are hard to use operationally to guide data dissemination; It is impractical to assume that every node in the network knows the utility function of every packet it forwards, or to include full utility functions in each data item. Even if the utility function were known to relay nodes, it would be difficult for them to make distributed, coordinated forwarding decisions to maximize utility in real time.

To allow efficient forwarding decisions, we introduce the *microutility.* Conceptually, microutility is an approximation of utility tagged to each packet for in-transit processing. Starting from a utility function for a data stream, the system breaks it down into microutilities for an ensemble of packets which collectively approximate the utility function. Microutility is a function of two dimensions: time since the data was generated, and the distance from the data source (frequency is coded only in the ensemble of microutilities). The microutility value indicates, if this packet were to be dropped now, how much utility would be lost. For simplicity, we have designed them as a linear approximation, so that any forwarding node, even those equipped with little memory or processing, can evaluate them. Figure 2, illustrates a microutility. In addition to this linear function, they also include a geometric region limiting their propagation to the most useful geographic regions. However to simplify the exposition we will assume isotropic dissemination. A different microutility is associated with each data sample in a stream of data, giving it a priority which changes dynamically while in transit. Each microutility represents a part of a *proactive* utility-maximizing “plan” by specifying a propagation distance for each packet in a stream and its relative priority so that *reactive* in-transit forwarding decisions can still occur in a utility-aware fashion.

Under normal operation, when data samples are spreading rapidly and effectively, the microutility plays a proactive role. A data sample will be geocast throughout the geometric domain defined in the microutility, with the distance from the origin limited by D_{max}. However, the geometric domain, particularly D_{max}, may change with every sample from the source, so unlike a typical geocast, the frequency of samples received can be shaped with distance. This is shown in Figure 1b, and is explained in more detail

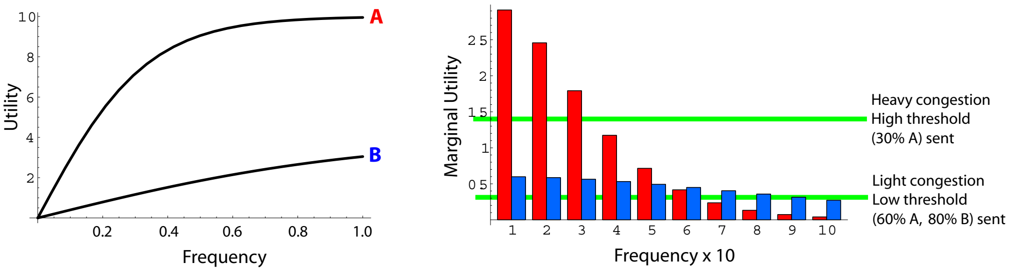


Figure 3: The utility functions and marginal utilities for two different data streams (A,B).

in [1].

There are also reactive uses of microutility, affecting in-transit decisions such as dropping data, sending data via alternate channels, or holding data for subsequent forwarding. Here, the linear function in the microutility assigns a dynamically-changing priority to the data sample, making these simple priority decisions. For the remainder of this section we will illustrate the use of microutilities for just one in-transit decision: dropping data when there is network congestion. Later sections will cover other uses of microutilities.

We consider an relay node C that is forwarding data from two data sources A and B. When C experiences congestion it establishes a threshold T, and at the time of packet reception, evaluates the microutility of each packet it receives, dropping those less than T. The threshold is set to drop enough data to relieve congestion, based on the current observed distribution of dynamic priorities and current congestion level in the neighborhood of C. More details can be found in [14].

This threshold-based dropping mechanism has three attractive properties. First, it is simple and efficient to implement and requires no state maintenance or application knowledge on the relay nodes. Secondly, threshold based dropping is a distributed well-coordinated mechanism; even if neighboring nodes see somewhat different pictures of the world, they will behave similarly. If this were not so, the geocast flooding process would find alternative paths, preventing congestion reduction. However, in the case of severe congestion, any neighbors that are not congested will set a lower T, providing a useful alternate path. Thirdly, threshold-based dropping implements congestion reduction that maximizes recipient's utility without directly using utility functions. We'll illustrate this for node C in the example above.

For illustration, consider the simplified utility functions for C's forwarding data from applications A and B. (This is a frequency projection of the sum of the utilities at larger distances.) These are the kind of utility curves that might occur when application A has higher priority, but needs fewer updates to function well. The right side of the figure shows the marginal utilities (slope of the left-side curve) for each application. The microutilities are assigned to data samples so that when they will approximate the same ensemble of values as these marginal utilities. Thus, dropping data according to a microutility threshold selects frequencies that equalize the marginal utilities of the different data streams in Figure 3, and therefore maximizes the overall utility. This example illustrates data traffic from A and B at position C, but since the linear function in the microutility of Figure 2 is fit to the marginal utilities, threshold dropping at various

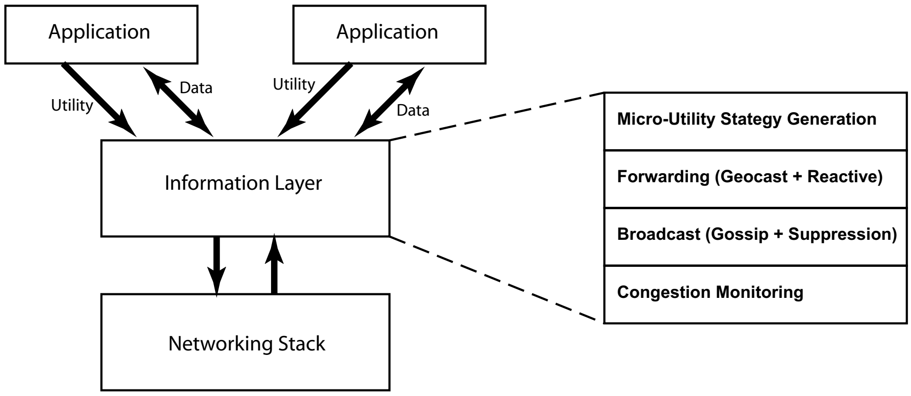


Figure 4: Protocol stack for the information layer.

distances will be optimal.

A particularly attractive aspect of this approach is illustrated by the two different thresholds on the right side of Figure 3. Depending on the congestion, different allocations are selected between the traffic types. Both of these are optimal for their circumstances. This is beyond the capabilities of a fixed priority approach that would strictly favor one traffic type over the other.

ARCHITECTURE

To allow diverse applications to share the same information dissemination machinery, we have architected an “information layer”, shown in Figure 4. This information layer sits between VANET applications and a traditional network stack, but its proximity to the applications allows them to submit detailed utility functions to drive networking decisions. Figure 4 shows the breakdown of the information layer into four sublayers.

At the time of utility function submission or major changes in network statistics, the topmost *strategy sublayer* calculates an ensemble of potential microutilities (this does not need to occur in real time) which it later assigns to data samples as they are sent. The *forwarding sublayer* then acts to send the data over multiple hops to reach the geometric region specified in the microutility. This sublayer is the central decision-maker. It receives data samples, evaluates their microutilities, and makes decisions on handling both new data samples and relayed data. It is responsible for managing the threshold to reducing traffic load. The *broadcast sublayer* is responsible for sending the data. It implements the gossip algorithm, queueing data briefly and listening for redundant broadcasts of the same data to reduce the number of duplicate transmissions. Finally the *congestion sublayer* monitors the local level of congestion by observing the queue length in the protocol stack below. This information is fed back to the strategy and forwarding sublayers so they can determine how much data to send, and what threshold to use to drop data.

EXPERIMENTAL EVALUATION

To evaluate our approach we used Paramics for vehicle traffic simulation and NS-2 for ad-hoc network traffic simulation. Paramics is a commercial traffic microsimulator with realistic mobility models. We used the Paramics demonstration model of Petuluma, California, which included single and multi-lane roads, traffic lights, traffic circles, and parking garages. This model included a variety of vehicle types and driving behaviors, and demonstrated both the rapid connectivity changes and wide-ranging spatiotemporal density variations. NS-2 is a well-accepted standard tool for simulating ad-hoc networks. We used an existing NS-2 implementation of 802.11b because it resembles 802.11p, however our approach should work equally well on other DSRC standards such as ARIB T55. In our experiments, we varied the available bandwidth to simulate the effects of additional applications which would need to share the channel.

UTILITY-DRIVEN INFORMATION DISSEMINATION

Our first simulation investigated the value of using utility functions to guide information dissemination. We modeled three applications: vehicles sending their current locations and velocities, parking garages announcing their current free spaces, and traffic lights sending their current state. These applications span a range of important characteristics, e.g., short vs. long range and safety-related vs. more commercial priorities. There are about 300 vehicles on the road, 9 parking garages, and 5 traffic lights. We set radio power to produce an average fixed radio range of 30m (a compromise value which we hope to address more carefully in future work), and set the gossip algorithm to allow up to 4 transmissions of each packet.

In this experiment, we specified appropriate utility functions for each application and compared our utility-driven system with a “baseline” system consisting of geocast flooding. This baseline used the identical forwarding and gossip mechanisms as our system, but sent data to a fixed radius around the data source, with the radius selected experimentally on a per-application basis to achieve a high utility value.

Figure 5 compares our approach with the baseline. The vertical axis is the total amount of utility delivered, summed over all applications. The horizontal axis is the total number of bytes sent by all nodes in the network as an indication of the communication cost. Our utility-driven approach significantly outperformed the baseline in all cases, delivering far more utility with the same communication cost. For example, with 5×10^8 bytes sent to the network (the vertical line in Figure 5), our approach delivers a utility sum of around 2.3×10^6, while the baseline delivers only around 1×10^6. A different way to look at this is shown by the horizontal lines in Figure 5: to deliver the same utility (e.g., 1×10^6 or 2×10^6, how much communication cost would have to incur in both approaches? It is clear that at utility$= 1 \times 10^6$, our utility-driven approach only spends around 1/4 of the communication cost when compared to the baseline. The baseline never reaches 2×10^6 utility, no matter how much network capacity is available. This is due to the effect of diminishing returns: delivering data too frequently in the geocast region has little additional value. In contrast, our approach does a better job in utilizing bandwidth, delivering small, but useful, amounts of data at larger distances. The utility curve does not saturate, even with very generous bandwidth settings. We have seen far higher performance gains of 10 times improvement and utility/bandwidth ratios for denser urban grid scenarios, but these have not been validated against the realistic vehicle traffic models of Paramics.

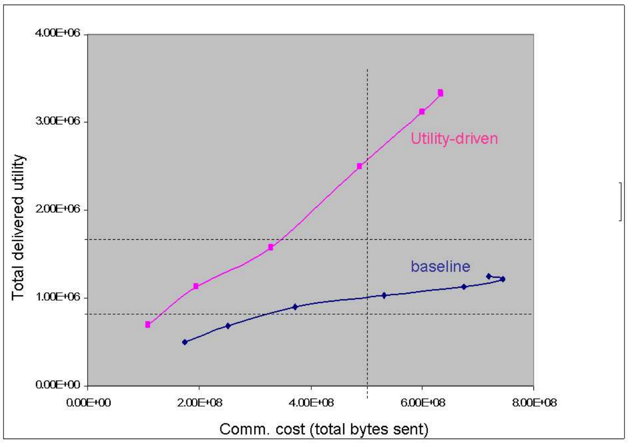


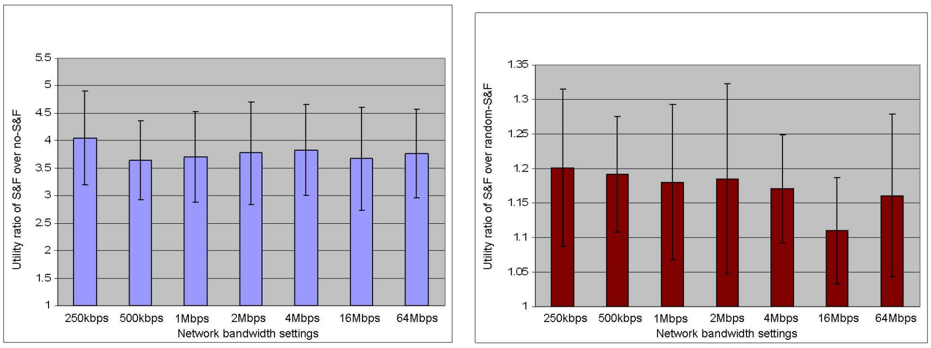
Figure 5: Tradeoff between communication cost and delivered utility.

STORE-AND-FORWARD PROPAGATION GUIDED BY MICROUTILITY

Dense urban areas require congestion reduction and careful selection of which data to forward, but the system also must support situations with *sparse* traffic, such as rural scenarios and urban scenarios where traffic is clustered at a traffic light with a large gap in between. In these cases, gaps between vehicles preclude peer-to-peer multihop communication. For this problem we use store-and-forward propagation—vehicles cache data and then forward it later when they are again in communication range with neighboring vehicles.

Obvious approaches to store-and-forward, for example, having vehicles periodically beacon the information they are storing are often inefficient or ineffective. If the periodic retransmissions are too frequent, then when vehicles are close together, the retransmissions cause severe congestion, but if the periodic beacons are too infrequent, then in sparse areas vehicles can drive past each other without exchanging valuable data. To address this, we have adopted an approach in which vehicles first send a *synopsis* of the data they are storing. This is a very compact representation of the stored data that is piggybacked on other data transmissions. Vehicles receiving a synopsis compare it with the data they are storing, and then send just the data that is missing.

The traditional approach to constructing a synopsis is to use a Bloom filter [15]. These are reasonably compact and efficient, but they indiscriminately identify all differences between data sets. Instead, we identify the most useful data to exchange based on the microutility approximation of the utility function. Our synopses are microutility-sorted lists of hashes, compared using Levenshtein distance computations. First we identify candidates for store-and-forward: updates that have propagated long enough in the ad-hoc network that they have largely completed local peer-to-peer propagation, but may still have utility for unreached nodes. These candidates are sorted in microutility order (with microutility evaluated over a predetermined quantized grid of location and time to give consistent ordering). This comparison can then be performed efficiently because it is only necessary to identify the first few, highest microutility, differences between the data



(a) (b)

Figure 6: Performance comparison: (a) the utility ratio S&F/no-S&F; (b) the utility ratio S&F/random-S&F.

sets being compared. (See [16] for a complete discussion of the synopsis design issues.)

To evaluate our store-and-forward approach in sparse traffic conditions, we lowered the density in Paramics to around 40-80 vehicles in the Petaluma model. We compare three approaches: (i) no store-and-forward, referred to as "no-S&F" in later text, (ii) store-and-forward guided by microutility, as described above, referred to as "S&F", and (iii) a benchmarking random store-and-forward approach, where synopses are exchanged, but the decision of which updates to forward is randomized among the set of all missing updates. (This randomization occurs only once per update at the source, so it is not completely random, it favors a proportional far sharing of the S&F resources.) We refer to this benchmark as "random-S&F". All proactive microutility machinery in S&F are used in random-S&F. Figure 6a shows the ratio between two delivered utilities, via the S&F approach and the no-S&F approach respectively, for a number of network bandwidth settings. The error bars are also shown. This ratio is between 3.5 and 4. This indicates that store-and-forward can deliver far more utility in sparse VANETs. Figure 6b shows the ratio between S&F and random-S&F. The ratio is consistently above 1, meaning that the S&F always outperforms the random-S&F approach. This shows the advantage of using the microutility in forwarding decisions. On average, using microutility brings about a 15% gain in delivered utility.

Another advantage of our S&F approach, not shown in figures due to space limitations, is the improved fairness across applications. For example, in simulation, the parking garages tend to deliver very low utility without store-and-forward. This is because (1) in sparse settings, not many cars get close to the parking garage, and (2) the network is overwhelmed by the vehicle status packets, since there are far more vehicles than parking garages. Our S&F performs much better — it delivers a factor of 241 more utility for the parking garage application. Random-S&F performs noticeably worse (by a factor of 2). This is also expected, since random-S&F makes decisions about which packet to forward randomly. This random selection mechanism tends to favor the vehicle status packets, since they are the majority. Our micro-utility guided approach does not penalize minority as the random-S&F approach does, but makes forwarding decisions based on micro-utility.

LONG RANGE PROPAGATION GUIDED BY MICROUTILITY

Multi-hop vehicle-to-vehicle propagation does well for spreading information over short to medium distances, but a more practical system would take advantage of alternative channels for long-range communication, such as long-range wireless (e.g. WiMax), fiber optics or cellular.

We have performed simulation experiments in which traffic lights and parking garages also maintain a longer-range WiMax network. We were able to make use of the our microutility mechanism to control access to the WiMax network. Because a WiMax channel covers a larger area, its spatial bandwidth is a scarcer resource. Here again we approximate complex utility decisions by establishing a second threshold and giving WiMax access to packets with microutilities above the threshold. The packets also continue to propagate vehicle-to-vehicle, with any potential duplicates suppressed by gossip. Space does not permit describing this experiment in detail, except to state that it provided large benefits in utility delivery, providing long-range data delivery in sparse scenarios without the long time delays of store-and-forward.

CONCLUSION

Vehicle-to-vehicle and vehicle-to-infrastructure communication promises to increase driving safety, traffic efficiency, improve the entertainment available in vehicles and support many applications yet to be envisioned. However the need of these diverse applications to coexist requires a mechanism to provide very different kinds of QoS to each application. This mechanism must be highly efficient and function correctly in dynamic and widely-varying network topologies and across multiple network protocols.

This paper proposes a general-purpose utility-based information dissemination mechanism to meet these challenges. We show our utility mechanism is general enough to guide network behavior in three very diverse applications and that the identical mechanism used for multi-hop dissemination in congested scenarios can support store-and-forward for sparse scenarios and access control to take advantage of long-haul networks. The system supports both proactive shaping of traffic to optimize utility over space, and supports utility-optimizing in-transit responses to local congestion, temporary gaps in the network, and availability of additional long-haul networks.

We also introduce the microutility approximation—simple linear functions included in each packet allowing simple in-transit decisionmaking to achieve approximately optimal results, even for complex utility functions. In simulations, we show large gains in utility, robustness to gaps in the network and greatly increased fairness between applications.

Acknowledgements

We would like to thank Fujitsu Limited for sponsoring this research, and in particular Takashi Aoki, Yoshihiro Aragane, Kazuo xAsakawa and Teresa Lunt for their foresight and direction in formulating this joint project. We would also like to thank Ying Zhang, Qingfeng Huang, and Maurice Chu for their ideas and efforts in implementing and evaluating this work.

References

[1] Ghosh, A., Greene, D., Huang, Q., and Liu, J., "Variable-resolution information dissemination", in *Second Annual IEEE Communications Society Conference on Senso and Ad Hoc Communications and Networks (SECON 2005)*, IEEE, 2005.

[2] Miller, M. and Drexler, K., "Markets and computation: agoric open systems", in Huberman, B. (ed.), *The Ecology of Computation*, Elsevier, Amsterdam, The Netherlands, 1998.

[3] Varian, H. R., "Economic mechanism design for computerized agents", in *Proc. of Usenix Workshop on Electronic Commerce*, July 1995.

[4] Shenker, S., "Fundamental design issues for the future internet", *IEEE Journal on Selected Areas in Communication*, vol. 13(7), September 1995.

[5] Kelly, F., Maulloo, A., and Tan, D., "Rate control in communication networks: shadow prices, proportional fairness and stability", in *Journal of the Operational Research Society*, vol. 49, 1998.

[6] He, G., Zheng, R., Gupta, I., and Sha, L., "A framework for time indexing in sensor networks", *ACM Trans. Sen. Netw.*, vol. 1(1):pp. 101–133, 2005.

[7] Navas, J. C. and Imielinski, T., "GeoCast – geographic addressing and routing", in *Mobile Computing and Networking*, pp. 66–76, 1997.

[8] Ko, Y. and Vaidya, N., "GeoTORA: A protocol for geocasting in mobile ad hoc networks", Tech. Rep. 00-010, Dept. of Computer Science, Texas A&M University, Mar. 2000.

[9] Tseng, Y.-C., Ni, S.-Y., Chen, Y.-S., and Sheu, J.-P., "The broadcast storm problem in a mobile ad hoc networks", *Wireless Networks*, vol. 1(2/3):pp. 153–167, 2002.

[10] Demers, A., Greene, D., Hauser, C., Irish, W., Larson, J., Shenker, S., Sturgis, H., Swinehart, D., and Terry, D., "Epidemic algorithms for replicated database maintenance", in *PODC '87: Proceedings of the sixth annual ACM Symposium on Principles of distributed computing*, pp. 1–12, ACM Press, New York, NY, USA, 1987.

[11] Yin, J., ElBatt, T., and Habermas, S., "Performance evaluation of safety applications over dsrc vehicular ad hoc networks", in *Proceedings of The First ACM International Workshop on Vehicular Ad Hoc Networks*, pp. 1–9, ACM, 2004.

[12] Xu, Q., Mak, T., and Sengupta, R., "Vehicle-to-vehicle safety messaging in dsrc", in *Proceedings of The First ACM International Workshop on Vehicular Ad Hoc Networks*, pp. 19–28, ACM, 2004.

[13] Mangharam, R., Weller, D. S., Stancil, D. D., and Rajkumar, R., "Groovesim: A topography-accurate simulator for geographic routing in vehicular networks", in *Proceedings of The Second ACM International Workshop on Vehicular Ad Hoc Networks*, ACM, 2005.

[14] Greene D. et al, "Using utility and microutility to manage network resources", In preparation.

[15] Broder, A. and Mitzenmacher, M., "Network applications of bloom filters: A survey", 2002.

[16] Greene D. et al, "Store and forward of prioritized data", In preparation.