# Machine Learning Approach to Report Prioritization with an Application to Travel Time Dissemination

Piotr Szczurek Bo Xu Jie Lin Ouri Wolfson
University of Illinois at Chicago
{pszczu1,boxu, janelin ,wolfson}@uic.edu

## ABSTRACT
This paper looks at the problem of data prioritization, commonly found in mobile ad-hoc networks. The proposed general solution uses a machine learning approach in order to learn the relevance value of reports, which represent sensed data. The general solution is then applied to a travel time dissemination application. Through the use of offline learning, the paper analyzes the feasibility of the proposed approach and compares the accuracy performance of several common machine learning algorithms. The results show that not all machine learning algorithms may be used for prioritization and that the use of the logistic regression algorithm is particularly suited for the problem. The learned logistic regression model is then used in a simulated VANET environment. The results of the simulations show that it is better at prioritizing reports in terms of their usefulness in aiding vehicles to choose the shortest travel time paths.

## Categories and Subject Descriptors
H.4.3 [**Information Systems Applications**]: Communications Applications; C.2.4 [**Computer-Communication Networks**]: Distributed Systems; I.5.2 [**Pattern Recognition**]: Design Methodology – *Classifier design and evaluation*;

## General Terms
Algorithms

## Keywords
data dissemination, VANET, data prioritization, machine learning, traffic information systems

## 1. INTRODUCTION
The current expansion of computing devices with wireless communication capabilities helped to motivate creation of various information dissemination applications. Many of these applications are related to communicating traffic information. Examples include systems for disseminating parking availability [3], travel speeds [11, 15], or traffic video clips [7, 8]. By disseminating the information, drivers can make better choices regarding their routes or destinations. However, due to the limitations of communication devices, it might not be possible to send all of the information. As a result, a prioritization scheme has to be developed which ranks the usefulness of the information and allows for only the most useful to be disseminated. The method for finding such a prioritization scheme is the subject of this paper.

To find a prioritization scheme for a variety of applications, this paper proposes the use of a machine learning approach. In this method, the information is assumed to be contained in reports which are disseminated over time. The receivers of such reports then use the contained information to possibly alter their behavior. By examining the characteristics of the reports including the sender information and analyzing its impact on the recipient, it can be determined which reports should be considered the most useful. A useful report is one that has an impact on the decision making process of the receiver. For example, in a travel time dissemination application, a report will be useful when it changes the path of a vehicle.

To find the usefulness of reports, this work uses machine learning algorithms. The results show that the machine learning technique achieves good accuracy and hence can be used reliably for prioritization of the reports. Additionally, simulations of a travel time dissemination application show that the learned model provides better information to vehicles than common heuristics in terms of providing better routes with lower travel times.

In the next section, some relevant work on the topic of prioritization will be discussed. The following section will describe the model used for the general machine learning approach. The method itself will then be described in the subsequent section. Later, a specific application (travel time dissemination) of the general approach will be described. This will be followed by a presentation and discussion of the results and a conclusion.

## 2. RELEVANT WORK
Prioritizing reports for memory (cache) management and bandwidth management in mobile wireless networks has been studied in a number of works. In [10] the rank of a report is a weighted sum of its popularity, reliability, and size. The paper does not discuss how the weights are determined. In [14] reports are ranked such that the number of replicas of each report is proportional to the square root of its access frequency. According to [5], such a distribution of replicas has the optimal replication performance in minimizing the query cost. However, for the dissemination of real-time traffic information, the access frequency is not a suitable solution because the access frequency to a newly produced report is always small but the newly


*IWCTS `09* November 3, 2009, Seattle, WA, USA.

produced report is usually of most interest. Thus for traffic information we use machine learning to determine the report relevance. In [15], traffic reports are ranked using an ad-hoc formula in which the rank is in reverse proportion to the sum of the age and distance of a report. Finally, in [6][9][13] reports are ranked based on an abstract utility function which is to be defined by specific applications. Our ranking method can be viewed as an instantiation of the utility function.

# 3. MODEL AND PROBLEM DEFINITION

The system consists of a set of *mobile nodes*. A node is a physical entity capable of data computation, storage, and short range wireless communication. A node can also observe its environment through a *sensing device*. Examples of nodes include vehicles equipped with on-board computers and Wi-Fi. The sensing device may be a camera installed in the car, an odometer, or GPS.

At any point in time, a node may create a *report*, which contains the data derived from the sensing device. The data is formed as a fixed set of *attributes* and their values. An attribute identifies the type of the data value. An example of a report is a speed report, whose attributes are time and average speed. Other examples of reports include reports about traffic accidents or available parking spaces.

Every node carries a *reports database* of fixed size. The reports database contains reports the node has received or created over time. The reports stored in the database are communicated over time to a subset of other nodes in the network. The determination of when, how many, and which reports get communicated is controlled via a *communication protocol*. It is assumed that the communication capabilities are limited by the bandwidth and hence not all reports in the reports database may be transmitted. A reports prioritization mechanism is thus employed in order for the communication protocol to determine which reports to transmit.

Each node in the network has the ability to judge the *relevance of a report*. The relevance represents the utility a report holds when it would be sent to other nodes, given the sending node's current characteristics and the attribute values of the report. In other words, how useful would the report be to the recipient? This value is numeric and can be either Boolean or real valued. In cases where nodes can only judge whether the report was good or bad, the report's value is Boolean (0 for bad, 1 for good). For example, consider a report that represents the availability of a parking space. The report is judged by a node (vehicle) as "good" if the parking space remains available when the node reaches it. In cases where nodes assign numeric values, those will be assumed real valued in the range of 0 to 1.

The problem is thus to find a method for estimating the relevance of a report before it is sent, given the characteristics of the node holding the report and the attribute values of the report. By knowing the relevance, the communication protocol has greater information about which reports provide the most benefit to the nodes. Also, estimating the relevance can also be useful to determine which reports are kept and which are discarded by the node, given that its local database is of limited size.

In this paper we assume that the size is the same for all the reports. Thus the relevance of a report does not depend on its size. The extension to variable reports sizes is straightforward via a greedy solution to the Knapsack problem (see [1]).

# 4. METHOD DESCRIPTION

The general idea behind our method is to use the received reports as an input to a machine learning process. Given that nodes can make judgments regarding the relevance of a report, a supervised learning algorithm can be used with the judged relevance value as the given output. Over time, each node learns a model that can estimate the value of a report at any time.

To provide the necessary training data for the learning algorithm, each report is augmented with additional attributes related to the sender of the report. Although dependent on the actual application, the attributes in spatio-temporal environments would generally depend on time and space. By knowing these attributes, the receiving node can learn the mapping from the sender's and report's characteristics to the relevance value of a report. The receiving node, which would later resend the report, can then have a better estimate of the value to the next receiver.

There are generally two ways in which the machine learning method can be used: *online* or *offline*. In the online method, nodes continually learn the model by using the incoming reports as training examples. The communication protocol would then use the most current model to estimate the relevance value of reports. In the offline method, there are two stages. In the first stage, nodes only gather the training examples without modifying their models. Note that these training examples may be generated through simulations. After a period of time, the examples are gathered and fed to the machine learning process. After this, the learned model is used by all nodes in the network. The advantage of this method is that nodes would not have to incur the overhead of the machine learning algorithm. The disadvantage is that the learned model cannot adapt to changing situations. Nevertheless, since in the online case, nodes initially have no model, the offline method is useful for learning the model a priori, thereby providing a way of bootstrapping. Also, the offline method can be used to analyze which attributes should be used for learning and which machine learning algorithm is best for the given application. The focus of this paper is on the offline method and in particular, how it can be used in a specific, transportation related application.

The machine learning method for relevance value estimation can be applied to a variety of applications. The main constraint is that every node should be capable of evaluating the relevance of reports and that the value be based on a goal common to all nodes. This is frequently true for applications set in environments such as peer-to-peer networks, including mobile and vehicular ad-hoc networks (MANETs and VANETs). One application that has recently been studied by researchers is travel time information dissemination. The next section of the paper looks at how the machine learning approach can be used in this application.

# 5. APPLICATION – TRAVEL TIME DISSEMINATION

This section will discuss how the general machine learning method can be used in a particular application: dissemination of vehicle travel times on a road network. The dissemination is done by vehicles carrying computational devices with communication capability. The communication protocol, which those vehicles will use, is based on the TrafficInfo algorithm [15]. In this algorithm, a simple heuristic is used to evaluate which reports should be periodically broadcasted. In this section, the machine learning approach is applied in order to replace the heuristic with a learned model that is better at providing the relevance value of the reports. In the following sections, we define the model in which the TrafficInfo algorithm is applied and which will be used for the machine learning method.

## 5.1 Vehicles

The model consists of a set of vehicles. A subset of these vehicles is equipped with GPS and devices capable of computation and fixed-sized storage. Additionally, we assume the devices provide communication capabilities (i.e. 802-11b), and have transmission range of 250 meters. We will assume that each vehicle has a predetermined destination which it reaches via a shortest travel time path, according to the information it currently has in its database. When a vehicle reaches its destination, it chooses another one immediately.

## 5.2 Digital Map

Each vehicle holds a *digital map* used for storing information about travel times. The digital map is made up of a set of road segments. The properties of each road segment that are of interest in this paper are:

- Road segment identifier
- Coordinates of the segment endpoints
- Road type
- Travel time estimate
- List of reports used for the estimate
- Time period number

The road segment identifier uniquely determines the particular road segment in the digital map. The endpoint coordinates provide connectivity information used for shortest path calculations. Road type indicates the physical characteristics of the particular road segment. The road type could be identified as either a highway or a city street segment. The travel time estimate is the estimated time required to traverse the road segment in the given time period. This estimate is calculated as the average of travel times contained in the list of reports. The time period is a 5 minute interval between subsequent times that time in minutes is evenly divisible by 5. The time period number identifies the given period by a unique number. Time period numbers start at 0 during initial system startup and increase by 1 for each subsequent period. For example, if the first interval is 12:00pm-12:05pm; the time period number at time 12:11pm is 3 and represents the time period 12:10-12:15pm. The initial value of the time period number, when no reports have been received is -1. For the travel time estimation, the initial value is equal to the free-flow travel time.

## 5.3 Travel Time Measurements and Reports Database

As each equipped vehicle fully traverses a particular road segment, it uses its GPS to record the travel time. This information is then saved in a *travel time report*. This report stores the following fields:

- Report identifier
- Road segment identifier
- Travel time
- Time period number

The report identifier is unique to every created report and allows for duplicate detection. As in the digital map, the road segment identifier is used to uniquely match the report to a particular road segment. The travel time for reports is the time measured by the given vehicle's GPS. The time period number identifies the interval in which the time measurement was taken.

When a report is created, it is stored in a *reports database*. Each vehicle carries its own reports database, which can hold at most 250 reports. The reports are sorted in order, according to a value given by the *ranking function*. When it is the case that the reports database is full, upon insertion and re-ranking, the lowest ranked reports will be discarded until all reports can be stored within the given capacity.

## 5.4 Reports Dissemination

Vehicles exchange reports according to a peer-to-peer dissemination protocol. We will assume a particular dissemination algorithm called TrafficInfo is used by the vehicles. TrafficInfo uses a combination of flooding and periodic broadcasting to disseminate reports. Flooding is used for newly created reports, while periodic broadcasting is done for reports stored in the local database.. When a broadcast is initiated, all reports in the vehicle's reports database are ranked according to the ranking function. All top ranked reports that can fit inside the transmission message size are then broadcast. The goal is thus to find the relevance value of a report before the transmission in order to achieve the best ranking.

## 5.5 Travel Time Updates

Initially, all vehicles have the free-flow travel time as the estimate for every road segment. After every 5 minute interval ends, all reports in the database that have been received in that interval are used for updating the travel time. Before updating, the reports are first analyzed based on their period number. For each road segment, the period number of all reports for the segment is recorded and only the reports pertaining to the maximum period number are kept. All other reports are discarded.

The way the report is used for updating depends on the relation of its period number to the period number contained in the vehicle's digital map for the given segment. There are three cases:

1. Report's period is smaller than the digital map's. The report is thus discarded since it contains old information.
2. Report's period is greater than the digital map's. The report then replaces all previously received reports for the segment. The period in the digital map becomes the report's period.
3. The periods are equal. In this case, we will first make sure the received report is not a duplicate. If it is, it will be discarded. Otherwise, it will be added to the report list for the given road segment and the travel time estimate will be recalculated by averaging all reports' travel times in the list.

This updating is done for every report received in the last 5 minutes. At the end, the vehicle will recalculate the shortest travel time path to its destination.

## 5.6 Offline Learning

As described earlier, the first step in the method is to augment the reports with additional attributes. Since the data is of spatio-temporal type, the most obvious attributes to include relate to time and space. For time, the age of the report will be used, which is calculated as the difference between the current time period and the time period in which the travel time measurement was taken. To reflect the distance, the attribute used will be calculated as the free-flow travel time along the shortest path to the mid-point of the road segment specified by the report. Additionally the type of road will also be included as an attribute. The type can refer to either a highway or city street road segment.

Since the value of the road type attribute is the same for the sender and receiver, there will only be two additional attributes added to the travel time report. These attributes are:

- Age of the report at time of creation or sending
- Sender's (or creator's) free-flow travel time from vehicle's current position to the mid-point of the road segment

The age and free-flow travel time to the segment are initially filled in at the time of report's creation and updated when the report is broadcast. That way, the receiver of the report knows the condition of the sender when the report was broadcast. These report attributes will serve as the input to the machine learning process. The relevance value of a report, which will serve as the output, is determined by the receiving vehicle as either 1 or 0, depending on whether the report changed the vehicle's travel path. Additionally, any discarded report will also be labeled negatively (i.e., 0).

Although offline learning can be performed using real vehicles, the Scalable Wireless Ad hoc Network Simulator (SWANS) and STreet RAndom Waypoint (STRAW) [2,4] vehicle mobility model were used for the purposes of this paper. This simulator combines a mobile ad-hoc network communication simulator with a vehicular mobility simulator based on vehicles choosing random waypoints in the road network and using car-following theory to model individual vehicular movements.

In order to collect the necessary input/output pairs, the learning was done using *epochs*. Each epoch consisted of a single road network and a group of vehicles, each randomly placed and having a random destination. Every vehicle initially started with their digital map containing free-flow travel times for every road segment. In each epoch, a set of travel time reports is created about a single, randomly chosen, road segment. Each report in the set has its travel time set to a random number, chosen uniformly from 0 to the free-flow time for that segment. The time period of each report varies from 0 to 100. The number of reports is thus 101, with the first report having time period of 0, the next report 1, etc. The current time period is then set 100. Therefore, the ages of the created reports range from 100 to 0.

After the reports are created, each vehicle chooses a random number between 0 and 100 to serve as the period number for the given road segment in its digital map. While the travel time for that segment will still be the free-flow time, the use of the random period number will allow learning of the effects of age since the travel time updates are dependent on the relation of the report's period to the current period. Once the period is chosen for the vehicle, the 101 reports are then used to independently update the digital map of that vehicle. This means that after each update is performed, the digital map is returned to its original state.

During each update, it will be determined whether the shortest path of the vehicle would have changed as a result of the update. If so, an example labeled as 1 will be created, otherwise an example will be labeled as 0. The offline learning procedure is outlined formally below.

**Algorithm**: Offline learning for single region

**Input**: *R*, road network within region w/ free-flow travel times
*n*, number of vehicles
*e*, number of epochs

**Output**: *L*, List of labeled learning examples

For each epoch 1..e:
1. Randomly place *n* vehicles on road network *R*
2. Randomly choose a segment *s* in *R*
3. Randomly choose a travel time *t* between 0 and free-flow travel time on segment *s*
4. For each vehicle 1..*n*:
   a. Choose a random period number *p* between 0 and 100
   b. Create a travel time report about segment *s* with a free-flow travel time and period *p* and use it to update vehicle's digital map
   c. For period number *Pi:* 0..100:
      i. Create a travel time report about segment *s* with travel time *t* and period *Pi* and use it to update vehicle's digital map
      ii. Recalculate the vehicle's path to destination given current state of digital map. If path has changed, set *label* to 1, otherwise 0.
      iii. Create learning example consisting of age=100-*p*, distance=free-flow travel time from vehicle's location to segment *s*, roadType=(roadType of *s*), *label*
      iv. Restore previous digital map state.

# 6. EVALUTION

The purpose of the evaluation is to establish the feasibility of the offline learning method. The next section describes the tests done to determine the model accuracy obtained by various machine learning algorithms. The following section uses one of the learned models to evaluate how it performs in terms of vehicle route choice.

## 6.1 Machine Learning Accuracy

The Weka learning toolkit [12] was used for evaluation of various machine learning algorithms. The training data consisted of a set of learning examples gathered using the offline learning algorithms for two regions from the city of Chicago.

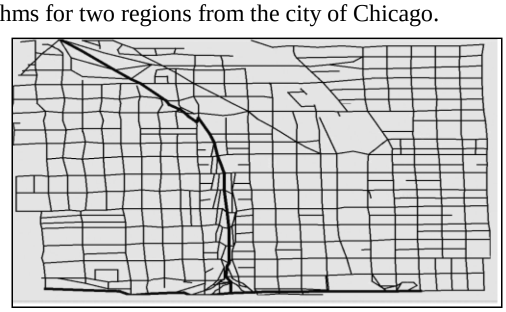

**Figure 1. Region 1 road network.**

**Figure 2. Region 2 road network.**

The road networks of these two regions are shown in figures 1 and 2, respectively. Region 1 is approximately 2.76 sq. mi. of northwest Chicago, while region 2 is about 6.75 sq. mi. from Chicago's south side.

For each region, the offline learning algorithm was used with 25 epochs. 100 vehicles were used for region 1 and 250 for region 2. The examples output from both regions were combined into a single data set. Since the number of negative examples far outnumbered the positive ones, the SpreadSubsample Weka routine was used to downsample the negative examples. The result was a data set with 7677 positive and 7677 negative examples. This data set was then input to the following machine learning algorithms:

- Naïve Bayesian (NaiveBayes Weka implementation)
- Logistic Regression (using Logistic Weka implementation)
- Support Vector Machines (using SMO Weka implementation, w/ buildLogisticModels enabled)
- Artificial Neural Network (using Multilayer Perceptron Weka implementation)
- Decision Tree (using J48 Weka implementation)

Each of the algorithms was used with the default parameters used by Weka, with exceptions as stated above. The testing of learning algorithms was performed using a 10-fold cross validation method.

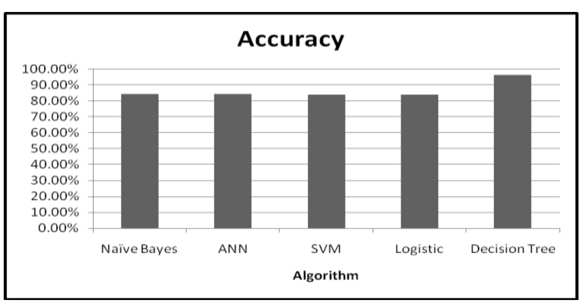


**Figure 3. Accuracy of various machine learning algorithms.**

Figure 3 shows the accuracy that resulted from using each of the 5 algorithms. All algorithms, with exception of decision trees, performed similarly with an accuracy of approximiatelly 83-84%. The decision trees had the best accuracy of 96.22%. It should be noted though, that while on one hand, the decision trees achieved extremely high accuracy, the resulting tree is very complex and not useful for report prioritization. The reason is that most of the leafs in the tree contained only one class of examples, which meant that almost every report would have a revelance value of 0 or 1. This is counterintuitive, given that two reports with different values of age, distance, or road type should be given a different probability of changing a vehicle's path. On the other hand, the other machine learning algorithms performed much worse than decision trees, but were able to capture the intuition behind the attribute relationships to the probability of changing a vehicle's path. The most understandable of these algorithms was the logistic regression model. It assumes that the relevance value of a report is a linear combination of the attribute values and finds the weights of the attributes that best fits the data. The model resulting from the used dataset was the following:

$$U = -0.0322*age - 0.02*distance + 0.3885*[road=highway] - 0.3885*[road=city\ street] + 4.9053$$

As can be seen, this model predicts that an increase in age will result in a decrease in the probability of changing path. This follows the intuition, since the travel time update policy discards many old reports. Similarly, the distance also varies inversely with relevance value. As could have been expected, the model also predicts more changes with highway segments than city streets. The approximately 80% accuracy achieved by the logistic regression method, coupled with the easy understanding of the resulting model, makes it the most promising in the use for prioritization.

Since the makeup of the road network might change the weights found in logistic regression, separate models for the two regions were also build to determine how the model would be affected. The results showed that both age and distance attributes did not vary much from the combined model. The age weight was determined to be 0.0313 for region 1 and 0.0336 for region 2. The distance weights were 0.0186 for region 1 and 0.0222 for region 2. The road type weight did significantly change between the two regions, with 0.7213 for region 1 and 0.0391 for region 2. This indicates a strong relationship between the road network and the road type of the segment on the probability of a report changing a vehicle's path. The constant factor between the two regions was relatively the same with -4.7806 for region 1 and -5.1552 for region 2.

## 6.2 Use of Learned Model in Prioritization

In order to evaluate the usefulness of the learned model, the SWANS/STRAW simulator was used to generate several scenarios in which vehicles disseminate travel time reports using the TrafficInfo algorithm. 100 vehicles were used for each scenario. The vehicles traveled along the road network constrained by region 1. In each scenario, vehicles were randomly placed and traveled to random destinations for 1 hour. After reaching a destination, the vehicle would choose another one immediately. The path to each destination was recomputed every 5 minute period according to the shortest travel time given the state of the vehicle's current digital map.

The highlight the differences in the various prioritization schemes, the TrafficInfo protocol was modified to disseminated only top 10 ranked reports. Additionally, the speed limits on most highway segments were lowered to 3 km/h in order to simulate a highly congested highway.

Two metrics were used for evaluation. The first is the average trip time, which measured the time it took to reach a destination averaged across all vehicles and destinations. This metric shows how it the prioritization scheme affects the travel time for an average vehicle. The second is the total travel time difference.

This metric was calculated by summing the absolute values of the differences between the travel time along the shortest path according to vehicle's current information and the path according to full information. Full information is defined as a digital map which was updated with every report ever created, as soon as it was created. The summation was performed every time a vehicle recalculated its path. This metric shows the level of information fidelity the prioritization scheme is able to achieve.

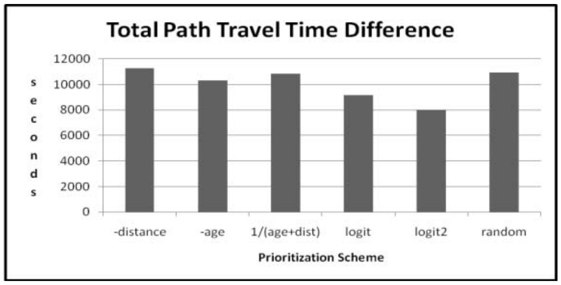


**Figure 4. Total path travel time difference achieved by various prioritization schemes.**

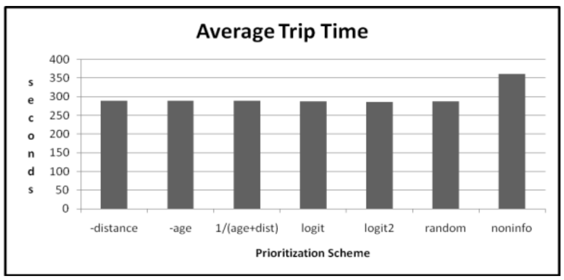


**Figure 5. Average trip time using various prioritization schemes.**

Figures 4 and 5 show the results of the simulations. The comparison was done using several heuristic methods and the learned models. The –distance and –age heuristics rank the reports inversely proportional to distance and age, respectively. 1/(age+dist) was the original heuristic used by TrafficInfo. The logit is the model learned using logistic regression using two regions. Logit2 is the model learned using only region 1. Lastly, the results were also compared to that of a randomized prioritization and a case, were no dissemination is performed (noninfo). The results show that the logit models outperformed all other methods in both metrics. In the total path travel time difference metric, the logit model outperformed original TrafficInfo heuristic by over 15%. The impact on the average trip time was small, with only 2 seconds difference. It should also be noted that the model (logit2) learned by using examples from the same region as was used in the simulations yielded much better performance in both metrics than the model learned from the combined data set (logit). This once again indicates a strong relationship of the road network to report's priority.

## 7. CONCLUSION

This paper proposed a machine learning approach to report prioritization for use in peer-to-peer environments. The method uses incoming reports in order to provide input to supervised machine learning algorithms. The learned model can then be used by all nodes in order to rank the reports to be disseminated. Through simulations, the paper was able to show the feasibility of using the machine learning method in a travel time dissemination application by achieving an accurate prediction model for the reports. Additional simulations showed that the learned model outperformed heuristics in terms of disseminating the information most likely to affect the vehicle's path.

## 8. ACKNOWLEDGMENTS

This research was supported in part by the National Science Foundation IGERT program under Grant DGE-0549489. It was also supported by National Science Foundation grants IIS-0847680, 0513736, 0326284, and 0957394.

## 9. REFERENCES

[1] C. Aggarwal, J. Wolf, P. Yu. Caching on the World Wide Web. TKDE, 11(1), pp. 94-107, 1999.

[2] R. Barr. An efficient, unifying approach to simulation using virtual machines. PhD thesis, Cornell University, 2004.

[3] M. Caliskan, D. Graupner, and M. Mauve. Decentralized discovery of free parking places. *VANET*, 2006.

[4] D. R. Choffnes and F. E. Bustamante. An Integrated Mobility and Traffic Model for Vehicular Wireless Networks. *VANET*, 2005.

[5] E. Cohen and S. Shenker. Replication strategies in unstructured peer-to-peer networks. *SIGCOMM*, 2002.

[6] A. Datta, et. al. Autonomous Gossiping: A self-organizing epidemic alg. for selective information dissemination in wireless mobile ad-hoc networks. ICSNW'04.

[7] M. Guo, M. Ammar, E. Zegura. V3: A vehicle-to-vehicle live video streaming architecture. *PerCom* 2005.

[8] U. Lee, et al. Dissemination and Harvesting of Urban Data Using Vehicular Sensing Platforms. IEEE Transactions on Vehicular Technology, (58)2, 2009, pp. 882-901.

[9] F. Perich, et al. On Data Management in Pervasive Computing Environments. TKDE, 16(5), 2004.

[10] F. Sailhan and V. Issarny. Energy-aware web caching for mobile terminals. *ICDCSW'02*.

[11] L. Wischhof, A. Ebner, H. Rohling, M. Lott, and R. Halfmann. SOTIS – a self-organizing traffic information system. *IEEE Vehicular Technology Conference*, 2003.

[12] I. H. Witten and E. Frank. Data Mining: Practical machine learning tools and techniques, 2nd Edition, Morgan Kaufmann, San Francisco, 2005.

[13] Y. Zhang, et al. ICEDB: Intermittently-Connected Continuous Query Processing. ICDE 2007.

[14] Y. Zhang, J. Zhao and G. Cao. Roadcast: A Popularity Aware Content Sharing Scheme in VANETs. *ICDCS*, 2009..

[15] T. Zhong, B. Xu, P. Szczurek, O. Wolfson. Trafficinfo: An Algorithm for VANET Dissemination of Real-Time Traffic Information. 5th World congress on ITS, 2008.