Prioritizing Travel Time Reports in Peer-to-Peer Traffic Dissemination

Piotr Szczurek*, Bo Xu*, Ouri Wolfson*, Jie Lin** and Naphtali Rishe***
* University of Illinois at Chicago, Department of Computer Science, Chicago, IL, U.S.A.
** University of Illinois at Chicago, Department of Civil and Materials Engineering, Chicago, IL, U.S.A.
*** Florida International University, School of Computing and Information Sciences, Miami, FL, U.S.A.
pszczu1@uic.edu, boxu@cs.uic.edu, wolfson@cs.uic.edu, janelin@uic.edu, rishen@cis.fiu.edu

Abstract— **Vehicular ad-hoc networks (VANETs) is a promising approach to the dissemination of spatio-temporal information such as the current traffic condition of a road segment or the availability of a parking space. Due to the constraint of the communication bandwidth, only a limited number of information items may be transmitted upon a vehicle-to-vehicle communication opportunity. Ranking becomes critical in this situation, by enabling the most important information to be transmitted under the bandwidth constraint. In this paper we propose a method for online learning of spatio-temporal information ranking for a travel time dissemination application within a VANET. In this method, vehicles judge the relevance of incoming information items and use them as training examples for Naive Bayesian learning. Additionally, a separate machine learning algorithm is used to estimate the probability of a duplicate item being transmitted. The method is used in place of commonly used heuristics, and is shown to be superior in the application of travel time dissemination.**

Keywords-machine learning, VANET, data prioritization, dissemination

I. INTRODUCTION

A. The Contribution

A vehicular ad-hoc network (VANET) is a set of vehicles that communicate with each other via unregulated, short-range wireless technologies such as WiFi or DSRC [1].Many of the applications in VANETs are related to communicating real-time traffic information. Examples include systems for disseminating parking availability [2], travel speeds [3, 4], or traffic video clips [5, 6]. By disseminating the information, drivers can make better choices regarding their routes or destinations. However, due to the limitations of the communication bandwidth, it might not be possible to send all of the information. As a result, a ranking 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 ranking scheme is the subject of this paper.

The focus of this paper is on travel time dissemination. To find a ranking scheme for this application, this paper proposes the use of a machine learning approach. We postulate that the usefulness of a piece of information can be estimated based on its attributes such as the age and distance. In our method, the VANET disseminates reports over time where each report represents the travel time on a road segment. The receivers of such reports then use this information to possibly improve their travel route. By examining the attributes of the reports 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 is useful if it changes the path of a vehicle. To find the usefulness of reports, this work uses machine learning algorithms. These machine learning algorithms learn the probability that a report is useful as a function of its attributes. It does so by first identifying the attributes that indicate the usefulness. Second, the Naïve Bayes learning method is used to find a mapping from the attribute values to the probability of a report being useful.

The advantage of the proposed machine learning method is the ability to combine several, known to be relevant, attributes into a single ranking value. Simulation results show that the machine learning technique achieved better performance than the use of individual attributes or an arbitrary combination of the attributes. In the simulated environment the vehicles were able to choose better routes and lower their travel times as a result of the use of the proposed technique for ranking reports.

B. Relevant Work

The feasibility of the machine learning approach has been studied in [7]. There, various learning methods were evaluated for accuracy. The learning was done offline. The logistic regression method was then selected and used for ranking in travel time dissemination. While some improvement was shown, the results were only marginally better. This paper uses the framework of [7] and develops an online learning version of the algorithm, which improves on the offline method by allowing for dynamic adaptation of the learned model to the environment and also uses a duplication avoidance model to help in ranking performance. Other work in prioritizing reports has been done for memory and bandwidth management in mobile wireless networks. In [8] the rank of a report is a weighted sum of its popularity, reliability, and size, but determination of weights is not discussed. In [9] reports are ranked such that the number of replicas of each report is proportional to the square root of its access frequency. According to [10], 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

This research was supported in part by the National Science Foundation IGERT program under Grant DGE-0549489 and also supported by NSF DGE-0549489, IIS-0957394, IIS-0847680, IIS-0837716, CNS-0821345, HRD-0833093, IIP-0829576 and IIP-0931517.

because the access frequency to a newly produced report is always small but the newly produced report is usually of most interest. Thus for traffic information we use machine learning to determine the report relevance. In [4], 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 [11][12][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.

II. TRAVEL TIME DISSEMINATION

This section will discuss the environment assumed for the travel time dissemination application.

A. Vehicles

The environment consists of a set of vehicles. A subset of these vehicles is equipped with GPS and devices capable of computation and short-range wireless communication. The set of vehicles that are equipped we call *participating vehicles*, otherwise they are called *non-participating vehicles*. Every vehicle traverses a road network to a particular destination. We assume that each vehicle has a predetermined destination which it reaches along the path with the shortest travel time, given the information it currently has in its local database. In the rest of this paper, when we say shortest path we refer to the shortest travel time path.

B. Travel Time Measurements and Reports Database

As each participating 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 and road segment identifiers, travel time,* and *timestamp*. The report identifier provides a unique number for each report and is used for duplicate detection. The road segment identifier is used to match the report to a particular road segment. The travel time is the time measured by the given vehicle's GPS. The timestamp is the time at which the report is produced.

When a report is created, it is stored in a reports database. Each vehicle carries its own reports database, which can hold at most *RRsize* reports. The reports are sorted in order, according to a value given by the ranking function. The *ranking function*, *Rf*, maps every possible report into a *rank* i.e. number between 0 and 1. It is assumed that higher ranks are given to more important reports for an arbitrary recipient. The machine learning method proposed in this paper instantiates 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.

C. Report Dissmination

Each vehicle *v* can transmit to and receive from other vehicles that are within *transmission range,* denoted *Tr*. These vehicles are called *neighbors* of *v*. Every *Bi* seconds each vehicle broadcasts *Bsize* reports to its neighbors. The time between broadcasts is called the *inter-broadcast interval* and the number of reports that are broadcast is called the *broadcast size*. The value of *Bsize* depends on the report size and the available bandwidth and can be computed using a bandwidth optimization method such as the one introduced in [14]. The reports with the highest ranking values are sent in each broadcast.

D. Digital Map

Each vehicle holds a *digital map* used for storing information about road segments and their travel times. The *digital map* of a vehicle *v* is a weighted graph G=<V,E> where V is the set of vertices (intersections) and E is the set of edges (road segments), with the weight of each edge *e* being the travel time estimate of *e* maintained by *v*. A number of properties are associated with each road segment. The properties of road segments that are of interest in this paper are: *road segment identifier, road type, travel time estimate*, *list of k most recent reports pertaining to the segment.* The road segment identifier uniquely determines the particular road segment in the digital map. Road type indicates the physical characteristics of the particular road segment. There are several types that are defined (e.g. highways, arterial roads) and each corresponds to a different free-flow travel time on that segment. We call any non-highway segment a *city street segment*. The travel time estimate is the estimated time required to traverse the road segment. This estimate is calculated as the average travel-time of the *k* reports the vehicle has received or generated with the most recent timestamps.

E. Travel Time Updates

The following travel time update policy was used for the purposes of this paper. For each road segment s, a vehicle keeps a sliding window of the *k* youngest reports (i.e., the reports with the greatest timestamps) it has received in the digital map. In the experiments of this paper *k* is set to 10. When a report *z* regarding *s* is received, *z* is applied to update the travel time of *s* as follows. If the timestamp of *z* is smaller than the least timestamp in the sliding window (i.e., *z* is older than the oldest report in the sliding window), then *z* is discarded. Otherwise, the report in the sliding window with the least timestamp is replaced by *z*; the travel time of *s* is updated to be the average of the reports in the new sliding window. After the travel time of *s* is updated, the shortest path is recalculated. Thus, the shortest path is recalculated for each received report.

III. RANKING METHOD DESCRIPTION

The general idea behind our method is to use the received reports as an input to a supervised machine learning algorithm. The algorithm uses attributes of the report as input and report relevance as the given output. In general, a report is relevant if it has an impact on the decisions of the recipient. For travel time dissemination applications, we define being relevant as follows: a report *r* received by vehicle *v* is *relevant* if it changes the shortest path from *v*'s current location to its destination. A report thus becomes a positive example if the report changes the shortest path of the recipient vehicle. Otherwise, it is a negative example. Over time, each vehicle learns a single model that can estimate the probability that a report is relevant to an arbitrary recipient, and the model can then be used as a ranking function.

The model used to estimate the probability that the report will be relevant consists of two parts: duplication

model and conditional relevance model. The duplication model is used to find the probability that a given report is not a duplicate to a neighboring vehicle. The conditional relevance model estimates the probability that a given report is relevant to the recipient, assuming the report is new to the recipient. The rank value is then the multiplication of the estimates from both models:

$$\text{rank}(r) = \text{Prob}(r \text{ is new to a neighboring vehicle } v) \times \text{Prob}(r \text{ is relevant to a neighboring vehicle } v, \text{ provided it is new to } v).$$

Note that the separation into duplication and conditional relevance models is not necessary, because a duplicate report is automatically not relevant. However, our experimental testing revealed that using separate models achieves much higher performance. Therefore, performance figures in this paper show the results for using only the separate models.

A. Duplication Model

In order to learn the probability that a sent report would be a duplicate, we use a technique based on the MALENA algorithm [15]. This technique works as follows. For each report r stored at a vehicle v, v maintains a *duplication indicator vector* (DIV) for r. The DIV consists of two attributes of r: *fin* and *aro*. Fin is the number of times r has been received by v. Aro is the arrival order of r and is defined as the number of reports in v's current database that v received before r. When r is transmitted, its DIV is attached to r. A receiver n of r checks whether or not r is a duplicate, and the respective DIV becomes a training example. Specifically, if r is a duplicate, i.e. if r already exists in n's local database, then *neg*, the number of negatives for the respective DIV is increased by one. Otherwise, *pos*, number of positives is increased by one. Initially, both *neg* and *pos* start at zero for all DIVs. Given the DIV, the probability that report will not be a duplicate can be calculated simply by dividing the number of positives by the sum of positives and negatives.

The previous work using the MALENA technique with the fin and aro indicators was shown to perform well in estimating the probability of duplication. However, we have discovered that by replacing the aro indicator with the *broadcast age*, the performance improves considerably[1]. The broadcast age of report r for vehicle v is the number of broadcasts that have been sent by v since r was last broadcast by v. To calculate the broadcast age of r, v remembers the time of the last broadcast that includes r. The broadcast age of r is then calculated by dividing the time passed since the last broadcast of r by the length of the inter-broadcast interval. For newly created reports that have never been broadcast, the broadcast age is infinite. For reports that have been received but not yet broadcast, the broadcast age is defined as zero.

B. Conditional Relevance Model

This model assumes that the report that is received has never previously been received. Then the model estimates the probability that the report is relevant to the recipient, which we call the conditional relevance. To provide the necessary training data for learning the relevance model, each report is augmented with additional attributes related to the sender of the report.

Due to the spatio-temporal nature of the travel time reports, the most obvious attributes to include relate to time and space. To capture the temporal aspect of the reports we will define the *age* of a report. The *age* of a report r is the difference, in seconds, between the current time and the time at which r was created. To capture the spatial aspect, we introduce a distance measure that will be defined as follows: the *distance* of a report r contained by vehicle v with digital map *DM* about road segment *rs* is the shortest travel time, in seconds, from v's current location to the mid-point of the *rs* when the weight is the free-flow travel time for every road segment in DM. By knowing these attributes, the receiving node can learn the mapping from the sender's and report's characteristics to the relevance of a report. The receiving vehicle, which would later resend the report, can then estimate the relevance to a future receiver. For the purpose of learning the conditional relevance model, the well known Naïve Bayes method is used. In [7] the Naïve Bayesian algorithm was shown to be performing similarly to the logistic regression model for relevance probability estimation. Additionally, simple and efficient online versions of the Naïve Bayesian algorithm exist. This means the vehicles would not have to incur a high computational cost for maintaining the relevance model. The overall computational overhead for using the machine learning technique will thus be negligible in a VANET environment.

C. Learning Procedure

Training examples for both the duplication and conditional relevance models are created using the RelevanceTrain algorithm as given by the following pseudocode.

Algorithm - *RelevanceTrain*
Input – R: Set of received reports by vehicle v
Outputs – E_d: Set of training examples for duplication model
E_r: Set of training examples for conditional relevance model

1. Select and remove report r from R in any order.
2. IF r has been previously received by v, THEN: discard r, create negative example and add to E_d, GOTO step 1
   ELSE: Create positive example and add to E_d, GOTO step 1
3. Save the current digital map state and use report r to update the digital map of v using the previously describe travel time update policy.
4. Recompute the shortest travel-time path from the current location of v to the destination. IF shortest path changes, THEN create positive example and add to E_r
   ELSE create negative example and add to E_r
5. Restore the digital map state to the previous state (before the travel time update).
6. IF R is not empty, GOTO step 1

The algorithm works by first checking for duplicates. In order to detect the duplicate reception, v remembers the id's of all the reports it has ever received. At that point either a positive or negative example will be created for the duplication model. Then, if it is determined that the report is not a duplicate, relevance of the report is determined based on whether it would change the path of the vehicle. Based on this, a positive or negative report is created for the conditional relevance model. After all the reports in R are processed, any reports that were identified

[1] Tests were done within the later described travel time dissemination application in terms of percentage of duplicate received reports. While other combinations were tested, the fin and broadcast age combination performed the best.

as non-duplicate in step 1 are applied to update the digital map of the vehicle. Note that while the example creation procedure does temporarily update the digital map, the update is rolled back after the examples are created. This is done so that the order of examining reports does not affect whether the report will create a positive or negative example. Once examples are created, they are used as input to the machine learning for the duplication and conditional relevance models.

IV. EVALUATION

A. Simulation method

In order to evaluate the usefulness of the learned model for ranking reports, the STreet RAndom Waypoint (STRAW) simulator was used to generate a scenario in which vehicles disseminate travel time reports periodically. The road network is a 6 km by 4 km region of downtown Chicago taken from the digital map published by the Geographic Data Technology Inc. (see Fig. 1).

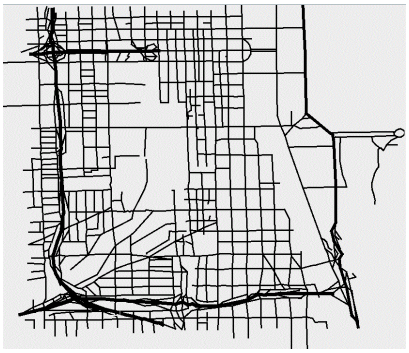

Figure 1. Simulated road network: portion of downtown Chicago.

100 vehicles were deployed in the road network. An open mobility model is assumed. In this model, vehicles are assumed to pass through the region, rather than travel within it. Thus each vehicle *v* is placed at a random location on the boundary of the road network, and another random boundary location on the road network is selected to be the destination. Vehicle *v* then moves from the origin to the destination, along the shortest travel time path given its current digital map state. When the destination is reached, the vehicle is assumed to have left the boundary. Another destination is randomly selected and the vehicle is assumed to be new[2]. Its digital map is thus reset to contain no travel time reports and its report database is emptied. The learned conditional relevance and duplication models are preserved. The justification for this is that a newly entering vehicle will also have a learned model, although it might have been from a different region. The average travel time is used as the metric for evaluation.

Out of the 100 vehicles, 10 are participating, meaning they broadcast and generate reports. The other 90 are non-participating and thus always follow the shortest free-flow travel time path. The simulation procedure is continued until 100 trips are made by each participating vehicle. We assume that during the time of simulation, non-recurring traffic conditions exist, such as in case of accidents on the road. To simulate these conditions, 40 slow-downs are initially introduced at randomly selected highway segments. For the slow-down segments, the maximum speed is set to be 3 km/h. Each slow-down lasts for a time period that follows an exponential distribution with the mean of 20 minutes. When a slow-down recovers, another slow-down is introduced at a randomly selected highway road segment. Thus at any point in time the number of slow-downs in the road network is fixed.

For the experiments a transmission range of 250 meters was used. Vehicles broadcast their reports every 5 seconds and stored up to 100 reports in their reports database. The number of broadcast reports was varied between 1, 10, 20, and 40 reports.

Since initially the online learning models do not contain any examples, their performance will be much lower than it potentially could be over time. Therefore, we chose to bootstrap the models using offline learning. This was done by running the simulation with default parameters for a total simulation time of 72 hours. After this time, the learned duplication and conditional relevance models are saved and used as the initial models for all tests.

B. Using Machine Learning for Combining Attributes

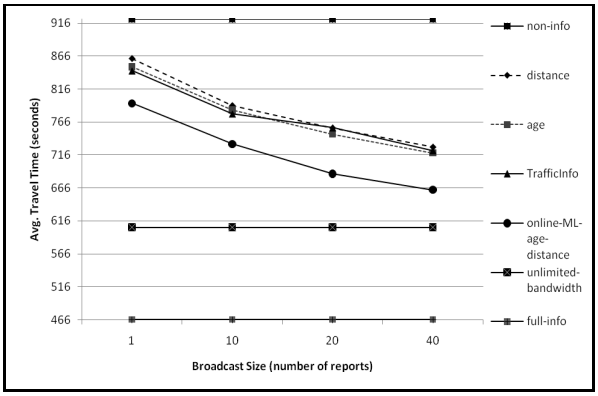

Figure 2. Impact of broadcast size on average travel time.

In this section we show how the machine learning method can be used to combine two attributes which are known to be relevant, in order to improve performance. Fig. 2 shows the performance of the different ranking methods when varying the broadcast size. We compared four ranking methods. The first method (age) ranks solely based on age. The second method (distance) ranks solely based on distance. The third method (TrafficInfo) uses a heuristic which combines the age and distance attributes. For this purpose, we used the formula devised for the TrafficInfo algorithm [4]. The fourth method (online-ML-age-distance) is our proposed method using age and distance as attributes. In addition, three baselines were also tested: *full-info*, *non-info*, and *unlimited-bandwidth*. The baselines reflect theoretical upper and lower bound performance. Full-info is an ideal case where vehicles receive all the reports as soon as they are created and thus have the full available information; no bandwidth, memory, or transmission range limitations are considered. Non-info is a case when vehicles do not exchange any information. Unlimited-bandwidth is a case when the report database size is unlimited and the vehicles

[2] For purpose of calculating vehicle trips, each vehicle is labeled at the start of the simulation. This label is persistent throughout the simulation even though the vehicle is assumed to be a new vehicle once it reaches its destination.

broadcast every report they have ever received or generated. The unlimited-bandwidth baseline shows the best achievable performance that can be expected in the mobile peer-to-peer environment when transmission range and connectedness are the only limitations. The difference between the unlimited-bandwidth and the full-info baselines is that in unlimited-bandwidth, the reports are disseminated periodically instead of instantly and that the dissemination is done within the transmission range of the disseminating vehicle.

As would be expected, the performance of all ranking methods generally improves as broadcast size increases. The machine learning method maintains a lead in performance across all broadcast size values. In comparison, the TrafficInfo method of combining the two attributes does not significantly improve performance over individual attributes, although it does offer a marginal improvement at small values of broadcast size. The TrafficInfo result shows that arbitrary combinations of age and distance might not result in good performance.

C. Improving Performance Through Additional Attributes

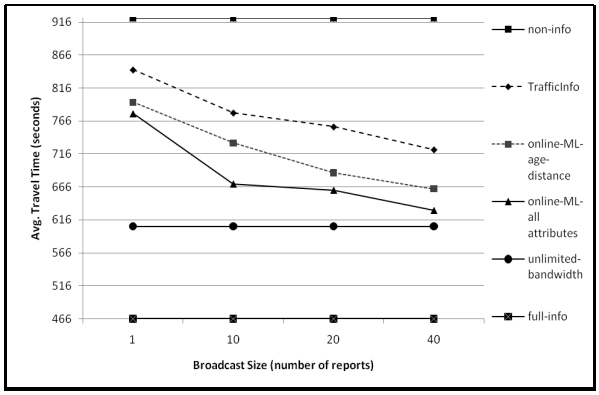

Figure 3. Comparison between attributes.

Aside from age and distance, there are also additional attributes that may be used to capture the relevance of a report. We used two additional attributes for the machine learning: road type and percentage of shortest paths. The *road type* can be either a highway or city-street segment, depending on its free-flow travel speed. Given a road network, *RN*, a vertex[3] *s* in RN and a road segment *rs*, the *percentage of shortest paths* (%SP(RN,s,rs)) is the number of shortest paths starting from *s* to all possible vertices in *RN* that pass through *rs*, divided by the total number of all possible vertices. Fig. 3 shows the performance when all four attributes are used for the conditional relevance model. As can be seen, the use of additional attributes provides a significant improvement for average travel time. Additionally, for broadcast size of 40, the performance comes close to the unlimited-bandwidth benchmark. This means that the online machine learning method provides near optimal ranking at that point.

V. CONCLUSION

This paper proposed a machine learning approach to report prioritization for use in a travel time dissemination application. The method uses incoming reports in order to provide input to supervised machine learning algorithms. The learned model can then be used by vehicles in order to rank the reports to be disseminated. The method was evaluated using the STRAW simulator and the results showed that the method outperforms heuristic methods under different communication protocol parameters. The main advantage of using the proposed technique is that several, known to be useful attributes can be easily combined in a way that improves the dissemination performance.

[3] For this paper, we use the midpoints of road segments as possible vertices.

REFERENCES

[1] http://www.standards.its.dot.gov/Documents/advisories/dsrc_advisory.htm. Accessed on Nov. 15, 2009.

[2] M. Caliskan, D. Graupner, and M. Mauve, "Decentralized Discovery of Free Parking Places", in *Proc. of the 3rd international Workshop on Vehicular Ad Hoc Networks,* 2006, pp. 30-39.

[3] L. Wischhof, A. Ebner, H. Rohling, M. Lott, and R. Halfmann, "SOTIS – a Self-Organizing Traffic Information System," in *Proc. of the 57th IEEE Vehicular Technology Conference*, 2003, pp. 2442–2446.

[4] T. Zhong, B. Xu, P. Szczurek, and O. Wolfson, "Trafficinfo: An Algorithm for VANET Dissemination of Real-Time Traffic Information," in *Proc. of the 15th World Congress on Intelligent Transport Systems,* 2008.

[5] M. Guo, M. Ammar, and E. Zegura, "V3: A Vehicle-to-Vehicle Live Video Streaming Architecture," in *Proc. of the Third IEEE international Conference on Pervasive Comp. and Comm. (PerCom 2005)*, 2005, pp. 171-180.

[6] U. Lee, E. Magistretti, M. Gerla, P. Bellavista, and A. Corradi, "Dissemination and Harvesting of Urban Data Using Vehicular Sensing Platforms," *IEEE Trans. on Veh. Tech.*, vol. 58, no. 2, pp. 882-901, Feb. 2009.

[7] P. Szczurek, B. Xu, J. Lin, and O. Wolfson, "Machine Learning Approach to Report Prioritization with an Application to Travel Time Dissemination," in *Proc. of The Second International Workshop on Computational Transportation Science*, 2009, pp.31-36.

[8] F. Sailhan and V. Issarny, "Energy-Aware Web Caching for Mobile Terminals," in *Proc. of the 22nd international Conference on Distributed Computing Systems,* 2002, pp. 820-825.

[9] Y. Zhang, J. Zhao and G. Cao, "Roadcast: A Popularity Aware Content Sharing Scheme in VANETs," in *Proc. of the 29th IEEE International Conference on Distributed Computing Systems*, 2009, pp. 223-230.

[10] E. Cohen and S. Shenker, "Replication Strategies in Unstructured Peer-to-Peer Networks," SIGCOMM Comput. Commun., vol. 32, no. 4, pp. 177–190, 2002.

[11] A. Datta, S. Quarteroni, and K. Aberer, "Autonomous Gossiping: A Self-Organizing Epidemic Algorithm for Selective Information Dissemination in Wireless Mobile Ad-Hoc Networks," in *Proc. of the First International Conference on Semantics of a Networked World,* 2004, pp. 126-143.

[12] F. Perich, A. Joshi, T. Finin, and Y. Yesha, "On Data Management in Pervasive Computing Environments," *IEEE Trans. On Knowl. And Data Eng.*, vol. 16, no. 5, pp. 621-634, May 2004.

[13] Y. Zhang, B. Hull, H. Balakrishnan, and S. Madden, "ICEDB: Intermittently-Connected Continuous Query Processing," in *Proc. of the 23rd International Conference on Data Engineering (ICDE 2007)*, 2007, pp. 166-175.

[14] O. Wolfson, B. Xu, H. Yin and H. Cao, "Search-and-Discover in Mobile P2P Network Databases, " in *Proc. of the 26th International Conference on Distributed Computing Systems,* 2006, pp. 1-9.

[15] B. Xu, O. Wolfson and C. Naiman, "Machine Learning in Disruption-Tolerant MANETs," *ACM Trans. Auton. Adapt. Syst.*, vol. 4, no. 4, pp. 1-36, Nov. 2009.