

PREDICTION OF NEAR LIKELY NODES IN DATA-CENTRIC MOBILE WIRELESS NETWORKS

Yingying Chen*, Hui(Wendy) Wang[†], Xiuyuan Zheng* and Jie Yang*

*Department of Electrical and Computer Engineering, [†]Department of Computer Science

Stevens Institute of Technology

Hoboken, NJ, 07030

Email: {yingying.chen, hwang4, xzheng1, jyang}@stevens.edu

ABSTRACT

As the increasing amount of data is collected in mobile wireless networks for emerging pervasive applications, data-centric storage provides energy-efficient data dissemination and organization. One of the approaches in data-centric storage is that the nodes that collected data will transfer their data to other neighboring nodes that store the similar type of data. However, when the nodes are mobile, type-based data distribution alone cannot provide robust data storage and retrieval, since the nodes that store similar types may move far away and can not be easily reachable in the future. In order to minimize the communication overhead and achieve efficient data retrieval in data-centric mobile environments, we propose a fully distributed neighborhood prediction scheme that utilizes past node trajectory information to determine the near likely node in the future as the best content distributee. We developed two methods that predict the future neighborhood based on the correlations of the past trajectories. Our extensive simulation results demonstrate that our prediction approaches can effectively and efficiently predict the future neighborhood with high accuracy.

I. INTRODUCTION

The development of data-centric storage has enabled efficient data dissemination of wireless networks. In data-centric storage, data is stored by attributes or types (e.g., geographic location and event type) at nodes within the network [1]–[3]. Queries for data with a particular attribute will be sent directly to the relevant node(s) instead of performing flooding throughout the network, thereby data-centric storage enables efficient data dissemination/access.

There has been work in data-centric storage of sensor networks. In these work, sensors that collect the data, called *collector nodes*, and store their data by attributes or types on other nodes, called *storage nodes* [3]. Most existing data-centric storage models can only deal with *static* wireless sensor networks. However, with the increasing deployment of wireless networks, there are emerging pervasive applications that rely on the mobility of wireless

devices. Two representative examples are: (1) the sensors are used for animal migration tracking, (2) the wireless devices are equipped with police officers to monitor their daily patrol routes, collect crime information by areas, and record corresponding law enforcement actions. In the first scenario, static centralized data storage may not be available, and in both two scenarios, efficient data retrieval can be achieved if the data-centric storage is enabled, i.e., the data is stored by the types of animals, by the activities performed by the animals, or by the tasks that are carried out by the police officers. The challenge is to design schemes that support data-centric storage in mobile environments. In this paper, we consider a fully distributed approach, in which there is no node playing the sole role as storage; each node can act as both the collector and the storage node. Further, *to reduce the communication overhead and achieve efficient data query*, the storage nodes are picked from the *neighborhood*, i.e., the nodes in the transmission range, of the collector node.

However, in mobile wireless networks, it is possible that both storage and collector nodes move in a broad area, which brings the possibility that the storage nodes store similar types may move far away and can not be easily reachable in the future. Therefore, it is desirable that the collector nodes migrate their data to the storage nodes that not only possess similar data types but also highly likely travel with them in the future. We define this kind of storage nodes as *near likely nodes*, which are the nodes that are in the neighborhood (i.e., *near*) and carry the same type of data that needs to be stored (i.e., *likely*).

In this paper, we propose mechanisms to predict near likely nodes for data-centric mobile wireless networks to achieve efficient data storage and retrieval. We first define a probability-based neighborhood prediction model. We then propose two approaches, namely *point-based* and *traced-based*, which predict the future neighborhood based on the correlations of the past trajectories. Further, we define a probability-based metric to measure the accuracy of prediction. Our approach of data transfer based on neighbor prediction helps to reduce communication overhead and consequently the overall energy consumption

during data retrieval.

To evaluate the effectiveness and efficiency of our scheme, we conducted experiments using mobile wireless networks simulated based on a city environment and its vicinity in Germany [4]. Our experimental results show high prediction accuracy and low computational time when using our proposed mechanisms, thereby providing strong evidence of the effectiveness of using data-centric approach through the prediction of near likely nodes in mobile wireless applications.

The rest of the paper is organized as follows. We place our work in the context of the related research in Section II. In Section III, we provide an overview of our problem and formulate our probability-based neighborhood prediction model. We next discuss the likelihood of neighborhood by presenting our two prediction approaches in Section IV. Further, we present the new metric for measuring accuracy prediction and the experimental evaluation of our approach in Section V. Finally, we conclude our work in Section VI.

II. RELATED WORK

There has been active work on data-centric storage [1]–[3]. [1] achieved data-centric storage based on the GPSR routing algorithm and an efficient peer-to-peer lookup system. [2] developed schemes for resilient data-centric storage from the viewpoint of energy savings and scalability in sensor networks. Whereas the security and privacy concerns in data-centric storage are addressed in [3]. Most of these current work only deals with *static* wireless networks. In this paper, we study data-centric storage in *mobile* wireless networks.

To detect node mobility, [5] used received signal strength in wireless LAN to detect wireless device mobility. [6] determined mobility from GSM traces using different metrics. In [7] signal variance is used with Hidden Markov Model (HMM) to eliminate oscillations between the static and mobile states for mobility detection. Further, [8] proposed to use correlation coefficients on RSSI traces to detect wireless devices that are moving together.

The work that are most closely related to ours are [9], [10]. A user-centric approach was proposed in [9] for colocation prediction that is used for media sharing based on repeating similar journeys in the urban transportation environment. Unlike [9], our approach does not require repeated trajectory patterns, and thus is more generic and can be applied to a broad array of pervasive applications involving mobile sensors. [10] addressed the detection of nodes of similar mobility patterns in group caching in MANET. However, these work do not support fully distributed models. Further, their work focused on *current*

neighbors, not the prediction of *future* ones. Our work is novel in that we utilize the past sensor trajectories to predict the future co-movement of sensors for data-centric storage in mobile environments.

III. PROBLEM OVERVIEW

In this section, we first present our assumptions, and then define our probability model for neighborhood prediction.

A. Assumptions

When considering data-centric mobile wireless networks, we have the following assumptions:

- **Mobility** The wireless devices are moving, randomly or in some pattern, in a well-defined area. We assume that the nodes know their physical locations at all time points during moving. It is a reasonable assumption because in many cases the data is useful only if the location of its source is known. For example, knowing that a crime occurred, which requires a law enforcement action, but without knowing where it occurred is useless. Localization of the mobile nodes can be achieved through the use of GPS or some other approximate but less burdensome localization algorithms [11]. We assume that the nodes are not aware of their moving patterns, if there is any.
- **Neighborhood** Each node has a communication range and can communicate only with nodes within its transmission range. We call the nodes in the transmission range the *neighbors*. Mobility of nodes may result in the change of the neighborhood. However, we assume that for every node, it has a stable neighborhood within a period of time. For example, police officers who carry out the same tasks are kept neighborhood while they are on duty.
- **Data-centric storage** We assume that the storage is *data-centric*, i.e., the particular node that stores a given data object is determined by the object's type such as event type [1], [2]. Hence, all data with the same type will be stored at the same node (not necessarily the collector node), so that the subsequent data retrieval requests could be efficiently directed. In particular, we propose to transfer data of the same type to a node's near likely nodes. We assume there will be nodes storing the same type of data in the neighborhood. The subsequent data queries will reach a collector node first through routing protocols for mobile wireless networks [12] and will then be redirected to the corresponding storage nodes. Our future work will address the case when there are no nodes storing the same type of data available in the neighborhood.

B. Probability Model

Generally, with mobile devices, the neighborhood may change over time. Some nodes may move into or move out of the transmission range periodically. To predict the future neighborhood, we utilize the *trajectory* of the mobile nodes. We assume the position of the nodes at each time point is in a 2-dimensional space. We note that our results can be easily extended to more than two dimensions. We denote the location of mobile node s at time t as $s_t(x, y)$, where x and y denote the x - and y - coordinates of s at time point t . Then given a time window $W(t_1, \dots, t_m)$ that consists of m time points, the trajectory of node s of W is denoted as $T(s_{t_1}(x, y), \dots, s_{t_m}(x, y))$. Given a set of trajectories $\mathcal{T}\{T_1, \dots, T_n\}$ of n nodes, our goal is to use \mathcal{T} to predict the neighborhood of s at a future time point $t_i > t_m$. To achieve this goal, we define a probability model of prediction that quantifies the likelihood of the future neighborhood. Since we assume that for every node, it has a stable neighborhood within a period of time, our prediction is based on the principle that *the nodes that are not only the neighbors in the past but also moving in the same direction are highly likely to be neighbors in the near future*. Based on this, we define two probability parameters.

- 1) *Neighbor probability* Pr_n : it is used to reflect the belief from the trajectories \mathcal{T} that a node s' is in the same neighborhood of the node s .
- 2) *Direction probability* Pr_d : it is used to measure the likelihood from the trajectories \mathcal{T} that two wireless nodes s' and s are moving in the same direction.

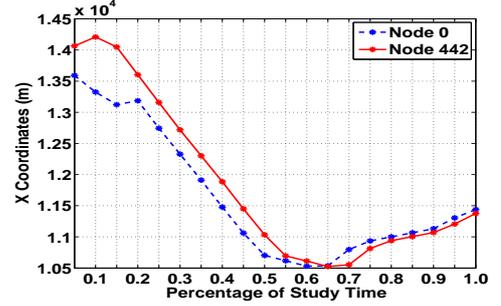
We further define the belief probability that node s' is in the neighborhood of s in the future as Pr_{dt} expressed by:

$$Pr_{dt} = Pr_n * Pr_d. \quad (1)$$

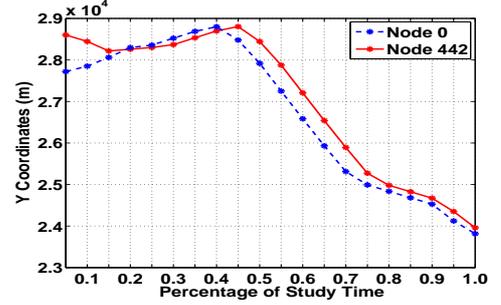
Given the time window W , a collector node s and its neighbor nodes, if s needs to store its data on its near likely node, then from its neighbor nodes that have available storage and store the same type of data that needs to be transferred from s , s picks the node that is of the maximum Pr_{dt} . Our model can be easily extended to choosing k nodes that are of top- k Pr_{dt} .

IV. NEIGHBORHOOD PREDICTION

In this section, we first explain how to compute the neighbor probability Pr_n . We then propose two approaches, namely *trace-based* and *point-based* approaches, to calculate the direction probability Pr_d .



(a) X dimension, PCC = 0.96



(b) Y dimension, PCC = 0.96

Fig. 1. Illustration of the x and y coordinates vs. time series for nodes 0 and 442 when they are moving together.

A. Neighbor probability

Given a node s within a time window $W(t_1, \dots, t_m)$, for any node s' , let $N(s') = \{t_i \mid 1 \leq i \leq m, s \text{ and } s' \text{ are neighbors at time point } t_i\}$. Then the neighbor probability is

$$Pr_n(s, s') = |N(s')| / m. \quad (2)$$

Intuitively at more time points that s' is in the neighborhood of s in the past, it will be more likely that s' remains as the neighbor of s in the future.

B. Direction probability

If two wireless devices are moving in the same direction, they should have similar trajectories and their x - and y - coordinates must follow the similar traces, and consequently may result in a strong correlation between their x - and y - coordinates respectively, and vice versa. Figure 1 shows an example of the coordinates versus time series when two nodes move together. We observed that these two nodes have highly-correlated traces in both X and Y dimensions.

Thus, to measure whether two nodes are moving in the same direction, we can use the *Pearson correlation coefficient* [13]. In general, the Pearson correlation coefficient is a statistical method that measures the strength and direction of a linear relationship between two given random variables. More specifically, given two random variables $P = \{p_1, \dots, p_n\}$ and $Q = \{q_1, \dots, q_n\}$, the

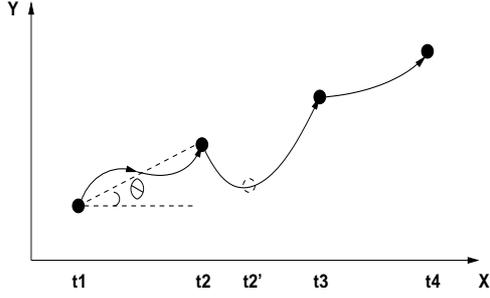


Fig. 2. An example of direction measurement

Pearson correlation coefficient PCC is defined as:

$$PCC = \frac{1}{n} \sum_{i=1}^n \left(\frac{p_i - \bar{P}}{\sigma_P} \right) \left(\frac{q_i - \bar{Q}}{\sigma_Q} \right), \quad (3)$$

where \bar{P} (\bar{Q} , resp.) and σ_P (σ_Q , resp.) are the mean and standard deviation of P and Q . The PCC value ranges from -1 to +1. Correlation +1/-1 means that there is a perfect positive/negative linear relationship between P and Q . In Figure 1, the high PCC value 0.96 for both the X dimension and the Y dimension shows high correlation between the coordinates of two nodes that are moving together.

Further, to measure the direction probability, we develop two schemes, namely *point-based* and *trace-based* schemes, based on the Pearson correlation coefficient. These two schemes consider both spatial and temporal changes of nodes in mobile environments.

1) *Point-based Scheme*: This approach utilizes the moving direction of the node s and s' at each time point t_i within a time window W to determine whether two nodes are moving together. The key idea is that the collector node computes the moving directions of the neighbor nodes at all time points in the time window W and measures the Pearson correlation coefficients of the moving directions.

Given the node s and its trajectory $T(s_{t_1}(x, y), \dots, s_{t_m}(x, y))$, at each time point t_i ($1 < i \leq m$), we define the *gradient* θ_i to measure the moving direction at the time point t_i .

$$\theta_i = \frac{s_{t_i}.y - s_{t_{i-1}}.y}{s_{t_i}.x - s_{t_{i-1}}.x}. \quad (4)$$

As defined, the gradient quantifies the direction that the node moves from the time point t_{i-1} to t_i . Figure 2 illustrates an example. Although the gradient θ may not be accurate when the trajectory between the time points t_{i-1} and t_i is not linear, we argue that we can always reduce the error by adding more time points on the non-linear trajectories, so that the sub-trajectories are close to linear format. For example, as shown in Figure 2 we can split the non-linear trajectories between t_2 and t_3 into smaller units by adding a time point t'_2 between t_2 and t_3 , as a

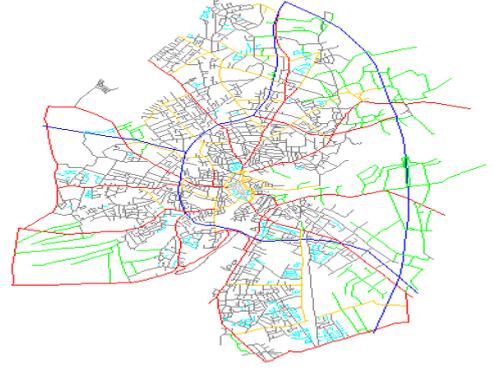


Fig. 3. The experimental data sets are generated based on the city and its vicinity in Germany.

result the trajectories between t_2 and t'_2 as well as between t'_2 and t_3 are close to linear.

Given two nodes s_1 and s_2 , let T_1 and T_2 be the trajectories of s_1 and s_2 of the time window W . At each time point t_i ($1 < i \leq |W|$) on both T_1 and T_2 , the node s_1 calculates the gradient θ_i by using Equation 4. All θ_i s are put into two vectors Θ_1 and Θ_2 , with θ_i from trajectory T_1 in Θ_1 and from T_2 in Θ_2 . It is straightforward that with m time points in W , there are $m - 1$ θ s in Θ_1 and Θ_2 . Finally, we measure the Pearson correlation coefficient (Formula (3)) of Θ_1 and Θ_2 . If the coefficient is positive, we take it as the direction probability Pr_d of node s_1 and s_2 . Otherwise, we value Pr_d as 0.

$$Pr_d = \begin{cases} PCC(\Theta_1, \Theta_2) & \text{if } PCC(\Theta_1, \Theta_2) > 0; \\ 0 & \text{otherwise.} \end{cases} \quad (5)$$

2) *Trace-based Scheme*: In this approach, opposite to the point-based approach, the collector node does not calculate the moving direction at each time point. Instead, it measures the Pearson correlation coefficients of two trajectories. To be more specific, given two trajectories T_1 and T_2 of two nodes s_1 and s_2 , first, the collector node s computes the Pearson correlation coefficient between the x-coordinates of T_1 and those of T_2 and collects the positive coefficients c_x . Similarly, it calculates the Pearson correlation coefficient of the y-coordinates of T_1 and that of T_2 . Let the set of positive coefficients be c_y .

$$c_x = \begin{cases} PCC(T_1.X, T_2.X) & \text{if } PCC(T_1.X, T_2.X) > 0; \\ 0 & \text{otherwise.} \end{cases}$$

$$c_y = \begin{cases} PCC(T_1.Y, T_2.Y) & \text{if } PCC(T_1.Y, T_2.Y) > 0; \\ 0 & \text{otherwise.} \end{cases} \quad (6)$$

As illustrated in Figure 1, when two nodes are moving together, the values of correlation coefficients are high in both X and Y dimensions. Since the correlation coefficients on X and Y dimensions are independent, we

multiply c_x and c_y as the direction probability:

$$Pr_d = c_x * c_y. \quad (7)$$

Moreover, Pr_d is normalized as needed.

V. EXPERIMENTAL EVALUATION

In this section, we describe our experimental methodology and evaluation metrics. We then present the results that evaluate the effectiveness of our approaches.

A. Methodology

We would like to evaluate the feasibility of our approach in an environment close to real applications (e.g. status monitoring of patrol officers) using mobile wireless networks, we conducted experiments based on mobile nodes generated from a city environment and its vicinity in Germany [4] as shown in Figure 3. These mobile nodes are the simulation of wireless devices carried by people. The size of the area is $25000m \times 25000m$. For our study of number of neighbors, two data sets, we call *small* and *large*, are obtained using this simulation environment with 1000 and 5000 nodes generated respectively and placed randomly inside the city. The nodes are moving along the roads in the city with walking speed around 4 feet per second. For the duration of our study, some new nodes may move into the city environment and some existing nodes may move out the city environment. There are no pre-defined trajectories for each node. However, group of nodes may travel together to common destinations (e.g., the city center). For instance, we observed that in the small data set the average number of neighbors increases from a few nodes to around 14 nodes as the study time moves along, indicating that groups of nodes are gradually formed and traveling together to the similar destinations. Thus, this environment is suitable for our neighborhood prediction study.

B. Metrics

We develop the following performance metrics to evaluate the effectiveness of proposed mechanism in terms of prediction of near likely nodes:

Prediction Accuracy: To measure the prediction accuracy, we split the time points into two time windows, namely past W_p and future W_f . The window of past W_p is used as the “training set” to predict the near likely nodes, whereas the window of future W_f , is used as the “test set” to verify the accuracy of the prediction. We choose n nodes, denoted as \mathcal{S} , as the “test participants”. Our accuracy measurement consists of two steps:

- **Step 1: Training.** For each node $s_i (1 \leq i \leq n)$ in \mathcal{S} , we find its near likely node s'_i that is of the

maximum Pr_{dt} (Formula (1)) in the time window W_p . For n nodes, we collect n such neighbor nodes and put their Pr_{dt} into a vector P . Thus P consists of n probability values.

- **Step 2: Testing.** For each near likely neighbor $s'_i (1 \leq i \leq n)$ from Step 1, we calculate its Pr_{dt} of the window W_f and store Pr_{dt} in a vector Q , which is also a set of n probability values. Our measurement of accuracy is based on the distance of P and Q . The smaller the distance is, the more accurate the prediction result will be.

To measure the distance of two probability distribution P and Q , our metric of Prediction Accuracy is based on KL-divergence. KL-divergence is a non-commutative measure of the difference between two probability distributions in probability theory and information theory [14]. Specifically, for probability distributions P and Q , the KL-divergence of Q from P is defined as

$$D_{KL}(Q, P) = \sum_i Q_i \log \frac{Q_i}{P_i}. \quad (8)$$

The smaller the value of D_{KL} is, the more Q is similar to P , which consequently indicates that our prediction of future near likely node is more accurate.

In the following discussion, we use the percentage of study time as the measurement of window size. We investigate the impact of different window sizes of the past as well as the future on the prediction accuracy using both point-based and trace-based schemes.

Time Performance: By measuring the time that each scheme needs to provide the prediction results, we evaluate the feasibility of applying these schemes to wireless devices that usually have constraints on computational power and memory. The Time Performance metric helps to benchmark our approaches in the simulation environment and further indicates the possibility to implement them in real wireless devices.

C. Results

KL-divergence: We first study the neighborhood prediction accuracy in our proposed mechanism. Figure 4 presents values of KL-divergence under various situations for both point-based as well as trace-based schemes when the average number of neighbors is 5. We observed small KL-divergence values that are always less than 0.5. This is encouraging as the smaller KL-divergence values indicate that the distribution of the belief probability in the future is close to the distribution of that in the past. Further, as shown in Figure 4 (a) and (b), when fixing the time window of the future, as the size of the past time window increases, the KL-divergence value presents an overall

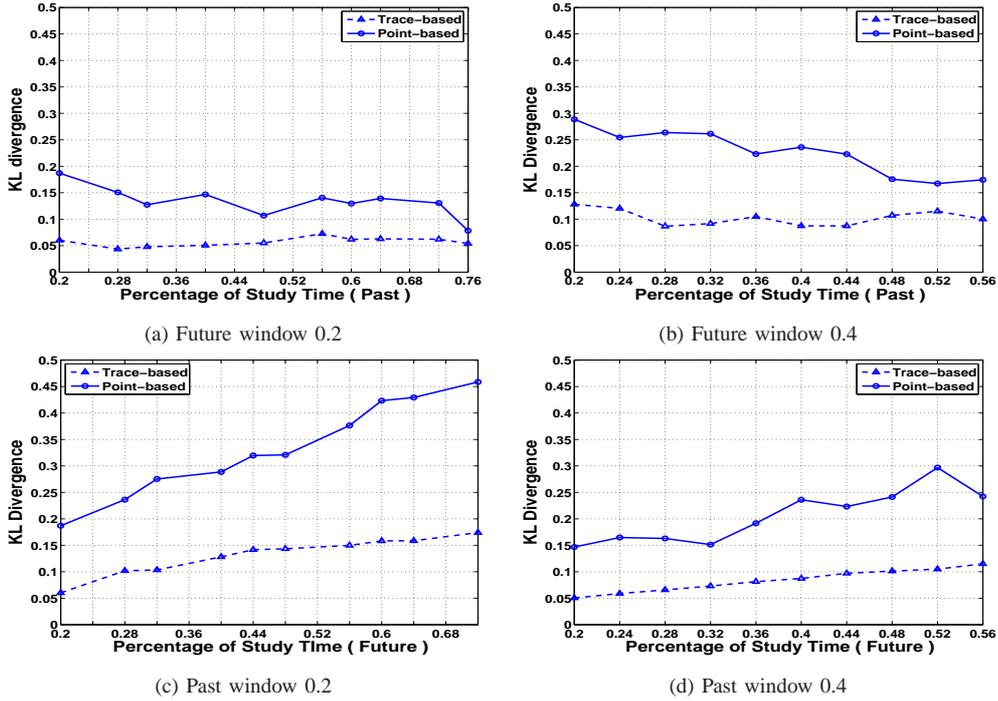


Fig. 4. KL-divergence when the average number of neighbors is 5: (a) and (b) are fixed past window size, the future window size is set to 0.2 and 0.4 of the total study time respectively; (c) and (d) are fixed future window size, the past window size is set to 0.2 and 0.4 of the total study time respectively.

decreasing trend for the point-based scheme. This means that by using the point-based scheme, the larger the past window size, the more accurate the prediction of near likely node can become. However, for the trace-based scheme, we observed that the KL-divergence value fluctuates. This is interesting since it shows that for the trace-based scheme, simply increasing the past window size does not increase the accuracy, which indicates that we need both expansion and shrinkage for adaptive adjustment of window sizes.

On the other hand, when fixing the time window of the past, as presented in Figure 4 (c) and (d), we observed an increasing trend of the KL-divergence value for both schemes as the window size of future is increasing when the average number of neighbors is 5, indicating that the near likely node may gradually moving away from the collector node when the future is long enough. In general, we found that the KL-divergence values of trace-based scheme is smaller than those using point-based scheme. Moreover, we observed similar results when the average number of neighbors increases to 15 and 45. Due to space limitation, the results are omitted. Therefore, *the trace-based scheme has better prediction accuracy than the point-based scheme.*

Additionally, we compared the values of KL-divergence between the small and the large data sets. We observed similar behavior for both large data set and small data set

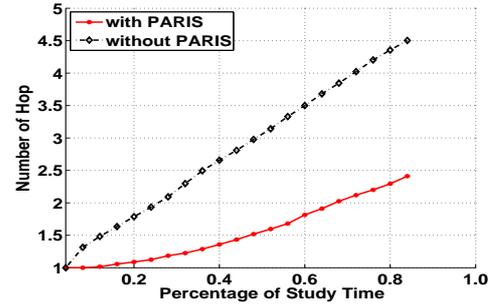


Fig. 5. Comparison of hop counts during data retrieval with and without our proposed scheme.

as the KL-divergence value presents an obvious decreasing trend when increasing the past window size and decreasing the future window size simultaneously. Furthermore, the KL-divergence values are smaller for the large data set. This is because there are more nodes in the large dataset, which form larger neighborhood and thereby provides better prediction result. Therefore, due to the space limitation, we only present the results obtained from the small data set in Figure 4.

Communication Overhead: We next study the communication overhead incurred in terms of hop counts during data retrieval. Figure 5 presents the number of hops traveled with and without using the scheme we proposed over the study time. We found that under our proposed scheme the number of hops traveled for data retrieval is less than half of that without using it, indicating that using

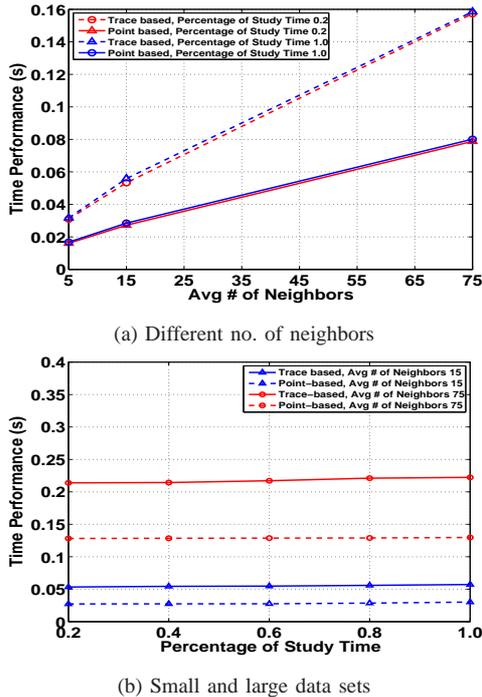


Fig. 6. Comparison of time measurements between point-based and trace-based schemes under different conditions: (a) different average number of neighbors using the small data set; (b) small and large data sets.

our scheme can significantly reduce the communication overhead and thus reduce the overall energy consumption of wireless devices. We will quantify the savings of energy consumption in our future work.

Time Performance: Figure 6 presents the comparison of time measurements under various conditions including different average number of neighbors and various window sizes for both small and large data sets. We found that the time to perform neighborhood prediction is in the order of milliseconds for both schemes. We observed that the point-based scheme runs at about two times faster than the trace-based scheme constantly under different average number of neighbors and various window sizes. This is because the trace-based scheme needs to calculate correlation coefficients for both X and Y dimensions, whereas the point-based scheme only calculates the correlation coefficient for gradient. Further, the time measurements of the large data set are also in the order of milliseconds as shown in Figure 6 (b). This indicates that even when a node has large number of neighbors, our schemes can efficiently predict the near likely nodes.

VI. CONCLUSION

In data-centric mobile wireless networks, to minimize the communication overhead and achieve efficient data retrieval we proposed a distributed neighborhood prediction model, which utilizes the past node trajectory information to predict the near likely node that stores the same type of

data and will most likely to remain in the neighborhood in the near future. These near likely nodes are chosen as the content distributee so that the later data retrieval is only needed in the neighborhood and thus more efficient in terms of communication overhead and energy consumption. We developed two schemes to predict the future neighborhood, point-based and trace-based schemes. We further derived a probability-based metric to measure the accuracy of prediction. Our results using simulation data generated from a city environment show that our prediction schemes of near likely nodes can both effectively and efficiently perform future neighborhood prediction, and consequently achieving efficient data retrieval.

REFERENCES

- [1] S. Shenker, S. Ratnasamy, B. Karp, R. Govindan, and D. Estrin, "Data-centric storage in sensor networks," *ACM SIGCOMM Computer Communication Review archive*, vol. 33, 2003.
- [2] A. Ghose, J. Grossklags, and J. Chuang, "Resilient data-centric storage in wireless ad-hoc sensor networks," in *Proceedings of the 4th International Conference on Mobile Data Management*, 2003, pp. 45–62.
- [3] M. Shao, S. Zhu, W. Zhang, and G. Cao, "pDCS: Security and privacy support for data-centric sensor networks," in *Proceedings of the IEEE International Conference on Computer Communications (INFOCOM)*, 2007.
- [4] T. Brinkhoff, "A framework for generating network-based moving objects," *GeoInformatica*, vol. 6, no. 2, pp. 153–180, 2002.
- [5] K. Muthukrishnan, M. Lijding, N. Meratnia, and P. Havinga, "Sensing motion using spectral and spatial analysis of wlan rssi," in *EuroSSC*, October 2007.
- [6] T. Sohn, A. Varshavsky, A. LaMarca, M. Y. Chen, T. Choudhury, I. Smith, S. Consolvo, J. Hightower, W. G. Griswold, and E. de Lara, "Mobility detection using everyday gsm traces," in *UbiComp*, September 2006, pp. 212–224.
- [7] J. Krumm and E. Horvitz, "Locadio: inferring motion and location from wi-fi signal strengths," in *MOBIQUITOUS*, Aug 2004, pp. 4–13.
- [8] G. Chandrasekaran, M. Ergin, M. Gruteser, R. Martin, J. Yang, and Y. Chen, "Decode: Detecting co-moving wireless devices," in *Proceedings of the Fifth IEEE International Conference on Mobile Ad Hoc and Sensor Systems (MASS)*, 2008.
- [9] L. McNamara, C. Mascolo, and L. Capra, "Media sharing based on colocation prediction in urban transport," in *Proceedings of the ACM International Conference on Mobile Computing and Networking (MobiCom)*, September 2008.
- [10] J. Huang and M. Chen, "On the effect of group mobility to data replication in ad hoc networks," *IEEE Transactions on Mobile Computing*, vol. 5, pp. 492–507.
- [11] K. Langendoen and N. Reijers, "Distributed localization in wireless sensor networks: a quantitative comparison," *Comput. Networks*, vol. 43, no. 4, pp. 499–518, 2003.
- [12] K. Akkaya and M. Younis, "A survey on routing protocols for wireless sensor networks," *Ad Hoc Networks*, vol. 3, pp. 525–349.
- [13] G. Casella and R. L. Berger, *Statistical Inference*. Belmont, California: Duxbury Press, 1990.
- [14] S. Kullback and R. A. Leibler, "On information and sufficiency," in *Annals of Mathematical Statistics* 22:79-86, 1951.