2011 Eighth IEEE International Conference on Mobile Ad-Hoc and Sensor Systems

The Trajectory Exposure Problem in Location-aware Mobile Networking

Li Ma* Jiangchuan Liu*
*School of Computing Science, Simon Fraser University
British Columbia, Canada
{lma8, jcliu, oba2}@cs.sfu.ca

Limin Sun† Ouldooz Baghban Karimi*
†Institute of Software, Chinese Academy of Sciences
Beijing 100190, China
sunlimin@is.iscas.ac.cn

***Abstract*—Location information improves the routing effectiveness and facilitates the development of diverse novel applications in mobile networking. While they can lead to better user experiences, given privacy concerns and hardware constraints, a mobile user often exposes a limited number of locations only. We are thus interested in the *Trajectory Exposure Problem* in this context, i.e., to what degree that the user's trajectory (i.e., its route) is exposed? Furthermore, can the user adaptively control the exposure of its trajectory and yet offer useful information for location-based services? In this paper, we explore Gaussian Process Regression, an effective tool to re-construct the trajectory of the mobile user with selected exposed locations. We examine how the re-constructed trajectory differs from the real trajectory, i.e., evaluating the *exposure rate*. We present an effective heuristic that adaptively controls the trajectory exposure rate by carefully choosing the exposed locations. We further demonstrate a practical routing protocol, MoRPTE, which, controlled by a single parameter, utilizes location information flexibly and adaptively in the spectrum from zero knowledge to full knowledge to fit the applications' demands.**

***Index Terms*—Location-aware Mobile Networking, Trajectory Privacy, wireless routing**

I. Introduction

With the great penetration of GPS and other tracking services (e.g., through WiFi or cellular based stations), the geographical location is now readily available in numerous mobile devices. Such information improves the routing effectiveness and facilitates the development of diverse novel applications, making mobile users better interact with their ambient environment [1][2].

While these new applications and routing designs can lead to better user experience, the exposure of personal location nevertheless involves privacy concerns as well. The concern is particularly severe given that the broadcast nature of a wireless channel. Recent survey [3] shows that 22% of the adult users use location based services at least once a week. However, there're still around 78% simply disable the location services in their iPhone devices or yet do not use such services often, based on privacy and other concerns. In some more critical cases, for example battle fields, exposed location would even lead to disastrous consequences [4].

Fortunately, most of the common people do not have such a stringent requirements as in battle fields [5] [6]. To benefit the improved performance and convenience, they often do not mind a few location points to be divulged, as long as their visited locations are not always exposed. In other words, the exposed data should not enable an accurate re-construction of the whole route. This is also true for commercial organizations where the daily routes are strategically valuable data, e.g., for Taxi companies. We have used the taxi location data from a major vendor in Beijing city in the performance evaluation of our work. The companies do not mind exposing some location information to improve service or vehicle-to-vehicle communications. Yet they do not expect the taxis' daily route to be exposed, which would greatly benefit their competitor's strategic plans. In fact, most of their taxi drivers are not comfortable if their daily routes are fully exposed, either.

Besides privacy, other concerns might involve the consideration of battery energy and the low sample rates that GPS can provide. Thus a user cannot expose all its locations continuously in location-based services and routings. Both location-oblivious and location-based services have been extensively studied in the literature. Yet services in between these two extremes have been largely unexplored. Considering the above concerns, we thus ask the following two critical questions:

If a mobile user exposes a limited number of locations only, what is the exposure rate of its trajectory (i.e., its route) ?

How can we adaptively control the exposure of a mobile user's trajectory (i.e., its route) and yet offer useful information for location-based services?

In this paper, we explore Gaussian Process Regression, an effective tool to re-construct the trajectory of the mobile user with selected exposed locations. We examine how the re-constructed trajectory differs from the real trajectory, i.e., evaluating the *exposure rate*. We present an effective heuristic that adaptively controls the trajectory exposure rate by carefully choosing the exposed locations. In state-of-art location-based routings, the location information is periodically updated and not real-time. We substitute it with predicted location which performs better than historical advertised location. We further demonstrate a practical routing protocol, MoRPTE, which, controlled by a single parameter, utilizes location information flexibly and adaptively in the spectrum from zero knowledge to full knowledge to fit the applications' demands. Our performance evaluation based on both the real Beijing Taxi data and synthetic data demonstrate the flexibility and

*This research was supported by a Canada NSERC Discovery Grant and Engage Grant.

978-0-7695-4469-4/11 $26.00 © 2011 IEEE
DOI 10.1109/MASS.2011.12

efficiency of our solution. It also facilitates the understanding of the fundamental utility and impact of location information exposed at various degrees.

II. MOTIVATION

We now discuss the motivation toward the *Trajectory Exposure Problem* in mobile routing in more details. Clearly, if a mobile user exposes every location it visits, the whole trajectory of its route can be accurately reconstructed. On the other hand, if no any location is exposed, then the trajectory basically is hidden as well. While existing systems implements one or the other, between the two extremes, the mobile user indeed may expose some of the locations only. Besides the obvious privacy concerns, users might have different concerns. Considering the limited battery energy, the user (or even the mobile devices itself) might not always turn on GPS devices. Even when the GPS is turned on, depending on its implementation, the location update rate is often low. The first ”real-time” tracking device New WorldTracker GPS can only update every 15 seconds.

Fig. 1 provides an example where three locations along a route are exposed by the mobile user. Given these incomplete *location samples*, an adversary can predict the trajectory of the user. With limited location samples, the predicted trajectory clearly will deviate from the real one. More importantly, the deviation largely depends on the quality of the exposed location samples, while not necessary their amount.

What we are interested in this paper, as motivated by the above example, is to understand the basic relations between the location exposure and the trajectory exposure. More explicitly, how can we evaluate the prediction deviations and how can the exposed locations be selected so that the risk of trajectory exposure be minimized? Beyond this, we look further into a practical protocol design that can utilize the location information flexibly and adaptively. With a single control parameter (the *exposure rate* of a trajectory), the protocol will work in the whole spectrum from zero location knowledge to full knowledge, so as to fit diverse applications' demands.

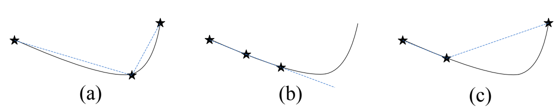

Fig. 1. Different trajectory predictions

III. TRAJECTORY RECONSTRUCTION AND EVALUATION

In this paper, we will focus on the trajectory prediction with *Gaussian Process Regression (GPR)* [7]. The Gaussian Progress Regression has been widely used in different fields. For trajectory prediction, it incurs much lower overhead and yet achieves excellent accuracy as compared with other state-of-the-art tools e.g., multi-layer neural networks. Our solution framework for the trajectory exposure problem however are generally applicable with other prediction tools, and we will also compare with other classical tools in later sections.

We now briefly review GPR; more details can be found at [7]. We will also derive the trajectory exposure rate under the GPR model in this section.

A. Gaussian Process Regression (GPR)

A Gaussian Process [7] is defined as a collection of random variables, any finite number of which have joint Gaussian distributions, and is fully specified by a mean function and a covariance function. The inference of continuous values with a Gaussian process prior is known as Gaussian process regression. The mean is often set to zero for notational simplicity, and hence the Gaussian Process Regression only includes determining covariance function. The covariance function can be defined by a Radial Basis Function (also called Squared Exponential) with some hyper-parameters inside the function. The GPR has a natural Bayesian interpretation and has various desirable properties, e.g., ease of obtaining and expressing uncertainty in predictions and the ability to capture a wide variety of behaviors through a simple parameterization [8].

Consider x as a general random variable, a Gaussian process can be written as:

$$f(x) \sim \mathscr{GP}(m(x), k(x, x')), \tag{1}$$

where $m(x)$ is the mean function of the distribution of random variable x, $k(x,x')$ is the general form of covariance function, and $f(x)$ is the target value of variable x. The latter specifies the covariance between pairs of random variables. When observations are noise-free, given two samples of variable x, namely x_p and x_q, we have

$$\mathrm{cov}(f(\mathrm{x}_p), f(\mathrm{x}_q)) = k(\mathrm{x}_p, \mathrm{x}_q) = \exp(-\tfrac{1}{2}|\mathrm{x}_p - \mathrm{x}_q|^2), \tag{2}$$

As mentioned, for notational simplicity, the mean function $m(x)$ is generally set to 0. However, considering noises, we assume additive independent identically distributed Gaussian noise ε, with variance σ_n^2, such that $y = f(x) + \varepsilon$, where y is the real target value with noise. And thus the prior is

$$cov(y) = K(X,X) + \sigma_n^2 I, \tag{3}$$

where X is the matrix of sampled data and $X = [\mathrm{x}_1, ..., \mathrm{x}_n]$.

The specification of the covariance function implies a distribution over functions. By choosing a number of points that form a matrix of estimate data set X_*, we can generate a random Gaussian vector f_* for target value with this covariance matrix into a Gaussian Distribution,

$$\mathrm{f}_* \sim \mathscr{N}(\mathbf{0}, K(X_*, X_*)). \tag{4}$$

If the vector f on sampled data set, where $\mathrm{f} = [f_1, ..., f_n]$, contains sampled data outputs of function f and to get the predicted target value of f_* on estimate points X_*, considering noise, the joint distribution according to the prior will be

$$\begin{bmatrix} \mathrm{y} \\ \mathrm{f}_* \end{bmatrix} \sim \mathscr{N}\left(\mathbf{0}, \begin{bmatrix} K(X,X)+\sigma_n^2 I & K(X,X_*) \\ K(X_*,X) & K(X_*,X_*) \end{bmatrix}\right). \tag{5}$$

The posterior distribution over functions can be obtained by restricting the joint prior distribution to include which agrees with the sampled data

$$\mathrm{f}_*|X,\mathrm{y},X_* \sim \mathcal{N}(\overline{f_*}, cov(f_*)), \text{where} \tag{6}$$

$$\begin{aligned}\overline{f_*} &\triangleq \mathbb{E}[f_*|X,\mathrm{y},X_*] \\ &= K(X_*,X)[K(X,X)+\sigma_n^2 I]^{-1}\mathrm{y}\end{aligned} \tag{7}$$

$$cov(f_*) = K(X_*,X_*)[K(X,X)+\sigma_n^2 I]^{-1}K(X,X_*). \tag{8}$$

Thus f_* which corresponds to X_* can be sampled from the joint posterior distribution by evaluating the mean and covariance matrix from Equation 6, and it gives the predicted mean value $\overline{f}(X_*)$ for estimated target value on estimate data set and an prediction uncertainty related variance $\mathbb{V}[\mathrm{f}_*]$ [7][9].

B. Trajectory Reconstruction

We then focus on how an estimated trajectory can be constructed, using the Gaussian Process Regression with a selection of sampled location points.

A trajectory is the path a moving object follows through space as a function of time. Specifically, we have sampled location points from x_1 to x_n, referring as the *location samples* in Section II. The location points are represented by two dimensional points $\mathrm{x}_i = (x_i, y_i)$, and X is the sampled data set for all (x_i, y_i)s. Trajectories will be represented by generated Gaussian Process Regression functions which is determined by a covariance function and a mean function. Therefore we hope to learn f such that it can be mapped to the changes in location: $\Delta\mathrm{x}_{i+1} = \mathrm{x}_{i+1} - \mathrm{x}_i$.

We adopt Gaussian process Regression reviewed in the previous Section III-A to estimate trajectory. By deriving the conditional distribution in Equation 6, if we let $K = K(X,X)$, $K_* = K(X,X_*)$, denote the vector of covariance between the estimate points and the n location samples, and $k(x_*) = k_*$ in case there's only one estimate point x_*. Therefore, compactly we can get the scenario of one estimate point case, where Equation 7 and 8 can be shorten as

$$\overline{f_*} = K_*^\top (K+\sigma_n^2 I)^{-1}\mathrm{y} \tag{9}$$

$$\mathbb{V}[f_*] = k(x_*,x_*) - k_*^\top (K+\sigma_n^2 I)^{-1} k_*. \tag{10}$$

On obtaining Equation 9 and 10, we can predict the changes in two trajectories at future times. The mean of the prior Gaussian process is normally determined to be zero. And for covariance function, we can use bias term and noises added version of original, as shown in Equation 3. After combining them, we get the following Algorithm 1 for prediction. Here X is the data set, y is the target value, k is covariance and σ_n^2 is the noise, while x_* is the data for estimation. And $\overline{f_*}$ is the mean value, $\mathbb{V}[f_*]$ is variance, while $logp(y|X)$ is marginal likelihood. In the algorithm, $cholesky(K+\sigma_n^2 I)$ is the Cholesky decomposition on the matrix of $K+\sigma_n^2 I$.
A More detailed explanation can be referred to [7].

Algorithm 1 Predictions(X, y, k, σ_n^2, x_*):

$L = cholesky(K+\sigma_n^2 I)$
$\alpha = L^\top \backslash (L\backslash y)$
$\overline{f_*} = k_*^\top \alpha$
$v = L\backslash k_*$
$\mathbb{V}[f_*] = k(x_*,x_*) - v^\top v$
$logp(y|X) = -\frac{1}{2}y^\top\alpha - \Sigma_i logL_{ii} - \frac{n}{2}log2\pi$
$return(\overline{f_*}, \mathbb{V}[f_*], logp(y|X)$

C. Exposure Rate Evaluation

To calculate the exposure rate of a trajectory, we compare how much corresponds the estimated trajectory is, in which we calculate the deviation $(\overline{f_*} - y)$ at each location point, and summarize along the whole path. We introduce a measurement Exposure Rate (ER), which ensures the overall consistency limited within r%, as in Equation 11, where the exposure rate gets 1 when $\overline{f_*}$ has value the same with y, and gets a minimum of 0 when the deviation of $\overline{f_*}$ is $|y|$ towards y. When the deviation is too large and exceeds threshold, the exposure rate will be counted as 0, although it is indeed negative.

$$ExposureRate(ER) = \frac{1}{n}\sum\left(1 - \left|\frac{\overline{f_*}-y}{y}\right|\right) * 100\%. \tag{11}$$

We then can ensure location trajectory exposure to be maintained within a percentage $r\%$ using the above equation. If the computed result is less than r%, then the differences between the original trajectory and the predicted trajectory is great enough, so that the exposure is safe and trajectory exposure is maintained under a Exposure Rate ER. Otherwise the exposure is beyond the limitation and it is over exposed.

D. Controlling Exposure Rate

We now address the problem of how to control the trajectory exposure rate $r\%$ through a careful selection of exposed locations. As we have illustrated in Fig. 1, exposing different location samples, even with the same quantity, can lead to different trajectory reconstructions, and hence different exposure rates. A natural question therefore is: *given a threshold r%, which locations can be exposed by a mobile user, such that the exposure rate of trajectory does not exceed r%?*

Clearly, an optimal solution of this problem for general trajectories involves the enumeration of different subsets of location samples. It is known that state-of-the-art GPS devices updates the location information at a rate of $15s/update$. That is, for a constantly moving user, 4 samples can be generated in one minute, and the total number quickly accumulates over time. As such, an optimal solution is computational infeasible, particularly for computation-power- and memory-storage-limited mobile devices [10].

We now present an effective heuristic that captures the key features of a trajectory. We define *critical value* of each location sample as the criticalness when the location is exposed. The higher the value is, the more critical it should be towards the trajectory. The points representing features mostly occur at direction turnings in the trajectory, as the middle point in Fig. 1 and Fig. 2(a). Thus the middle point in Fig. 2(b) is not critical. We define the location critical value by comparing a location sample with its neighbor points, to determine if it represents some feature of trajectory, in Equation 12:

$$CriticalValue = \left|\frac{\Delta y_{left} + \Delta y_{right}}{2}\right|. \tag{12}$$

The heuristic solution works as the following: First, the critical value of each location sample of the whole trajectory is evaluated. Then the location samples are sorted in ascending order by critical values. Every iteration we choose a point with minimum critical values, so on and so forth, and apply our proposed mechanism to do prediction and see if $r\%$ is maintained. Thus, given the threshold of $r\%$, we can determine the maximum number of location samples that can be exposed and which these points are, and all possible ways of different location points' combinations.

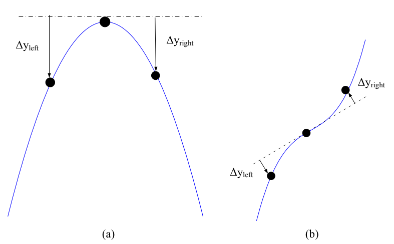

Fig. 2. Different cases in determining location critical value

IV. Mobile Routing with Partial Trajectory Exposure (MoRPTE) — A Case Study

We now proceed with a case study using our trajectory exposure control tool for mobile routing. A conventional wisdom for location-based mobile routing is to decouple the location advertisement and path determination [1][2]. First, a location advertisement module enables each node to broadcast its locations to others with which they can use in routing. Such advertisement can be done by periodically broadcasting its location information to others that spreads out the whole network or to some server which keeps records of all nodes' up-to-date location data. Such information will then be used by the path determination module to locate the best route.

However, such routing schemes assume nodes' location advertisements real-time, which is not in real world. A user may have different concerns as mentioned in Section II. Realistically, a user cannot expose its trajectory at a exposure rate of 100%. In the state of art location-based routings, all locations of nodes are assumed to be up-to-dated. But the location of a user's neighbor might be updated a period of time ago, leaving its current location unknown. We target SOGR [1] and present our MoRPTE (Mobile Routing with Partial Trajectory Exposure) which utilizes predicted location in substitute of the previously advertised locations, to make routings more efficient, while also considering trajectory privacy.

In the original SOGR protocol, an intermediate node makes packet forwarding decisions based on its knowledge of the neighbors' positions and the destination's position. It uses a geographic-based greedy forwarding that forwards data

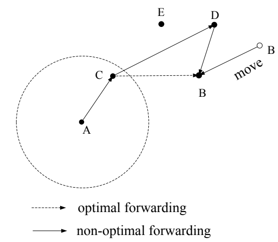

Fig. 3. Less precise prediction occurs

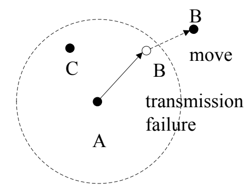

Fig. 4. Handling inaccurate destination location problem

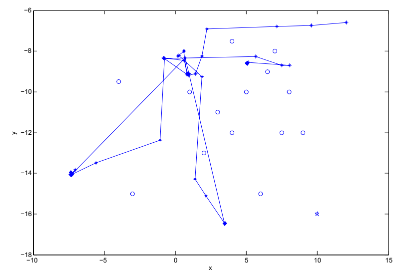

Fig. 5. Taxi A73198 trajectory path sample

greedily to a neighbor that progresses the most towards the destination. However, the locations of intermediate nodes are advertised periodically and not up-to-date, though it is assumed to be real-timed. We adopt a predicted location in substitute of original location, so as to deliver the packets efficiently, while the rest of SOGR routing protocol is remained the same, including back-off replies and route optimizations.

The advertisement of locations is thus more flexible, adopting the proposed GPR prediction in Section III-B. Predicted locations are now used for calculation to determine next-hops. Thus, when a node doesn't advertise its location for some time, an estimated location will behave better than last-updated location. SOGR offers a simple estimation during the phase of validity estimation, in which simply last two recorded locations are used to form a simple line. Our prediction methods behaves more realistic, considering the randomness and realities of node movements.

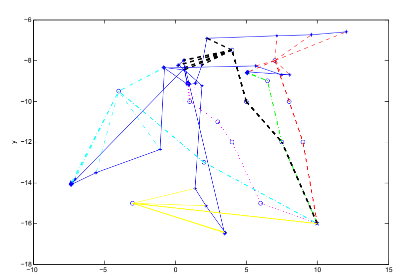

Fig. 6. Routing Paths when r%=10%

Although it seems that locations are simply substituted by predicted ones, there exists a certain problem, that prediction might sometimes bring non-optimal paths. Different nodes might obtain different historical data of each other, which will bring different predicted future locations of intermediate nodes. Thus during path calculation, a prediction not precise enough might be generated and a non-optimal next-hop might be selected. As the Fig. 3 shows, A wants to deliver a packet to the B based on the predictions of current locations of its neighbors and the location of B, it selects C as its next-hop. However, C might not be able to predict the real-time current location of B, and will select D as its optimal next-hop. D predicts that B will move to the current new location, it will send the packet to B and thus brings a longer path than directly from C to B. We solve the problem with the help of other overhearing nodes. During the forwarding, if a node E which possesses more information about the movement of B and can predict its current location, E will send a correction message to C, notifying C to change its next-hop as B, when $dis_{(C,B)} < dis_{(C,D)} + dis_{(D,B)}$. Our mechanism can also handle the inaccurate destination location problem. As shown in Fig. 4, if B moves out of the transmission range of A but A doesn't predict that, the transmission will fail. However another node C can predict B's movement and correct A's next-hop so as to fit the current location of B.

Therefore, locations for calculation are predicted and fit the real-time locations more, rather than the ones advertised before. It brings more efficient and precise data forwardings, also guarantees the exposure rate during location advertisements.

V. Performance Evaluation

We have conducted a series of experiments, both with real traces and synthetic data. The traced data we use is the real trajectory correspondent data of taxis in Beijing. The research is partly supported by the State Key Program of National Natural Science Foundation of China(Grant No. 60933011). Such data is administrated by Beijing government and is hard to obtain. The data reflects real situations of taxis' movements throughout the whole city, and one single of all data files contains 10,050 taxis' trajectories with size over 100MB for one day. The data is collected periodically, with each taxi's identity, time, GPS location, velocity, direction information. The capacity of data is too huge so we picked a number of typical ones for performance evaluation. "Typical" means the taxi is mostly moving rather than staying at a same location.

A. Performance of MoRPTE Routing when Having Different Trajectory Exposure Rates

For instance, the trajectory of taxi A73198 from $20:38$ to $23:56$ is shown in Fig. 5, where the coordinate has been normalized. There are 100 nodes randomly deployed in the area, which represent intermediate nodes. For clarity, we only show 15 of them, which are used in the following comparisons. By MoRPTE routing, the following figures show the routing paths when setting the trajectory exposure rate parameter r to 10%, 50%, 100%, as in Fig. 6, Fig. 7 and Fig. 8, where different colored lines denotes different routing paths.

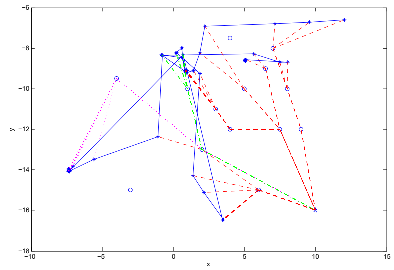

Fig. 7. Routing Paths when r%=50%

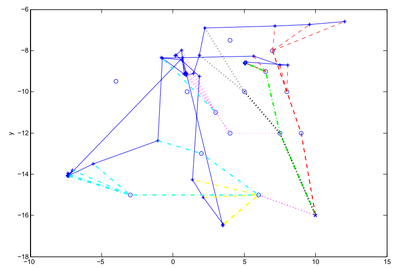

Fig. 8. Routing Paths when r%=100%

We can find in the graph that if the parameter is set 100% as shown in Fig. 8, the source node does not care about exposure of its trajectory information; therefore the routing paths are the same as when using MFR. Fig. 9 shows the trajectory prediction progress of one intermediate node.

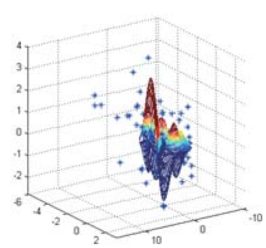

Fig. 9. The prediction at one intermediate node

We can see from the figure that, at some nodes the prediction consistency is low comparing others, which is due to data latency and loss. Therefore, using integration does not help to determine the consistency of prediction. Because of discontinued data sampling, we achieve the judging by Equation 11 instead. The previous case study shows the performance of MoRPTE routing, which dynamically controls users' trajectory exposure adaptively in the spectrum from zero knowledge to full knowledge, limiting the exposure rate within r%.

B. Performance on Prediction-based Data Forwarding

We simulated our mechanism MoRPTE on 1000 random nodes to compare with SOGR-HR which use last-advertised

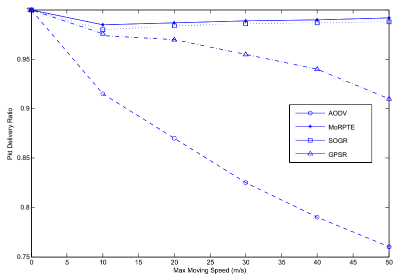

Fig. 10. packet delivery ratio

position as node's current location; AODV which do not use location information and GPSR [11]. We set SOGR parameters $Ref_{backoff}$, $Inc_{backoff}$, $Dis_{timeout}$ and $[Min_{timeout}, Max_{timeout}]$ to be 10ms, 2ms, 300m and [10s, 30s], with the Max_{hops} as 2. The movement models follows random way point model, setting the maximum speed to be 0m/s to 50m/s. Also, an advertisement option is added randomly, for each node to choose whether or not to advertise its current location. We study the same metrics as in [1]: packet delivery ratio, which is the ratio of packets delivered; control overhead, that the total number of control messages sending over each hop divided by the total number of packets received; and the total number of data packet forwarding accumulated from each hop over the total number of data packets received.

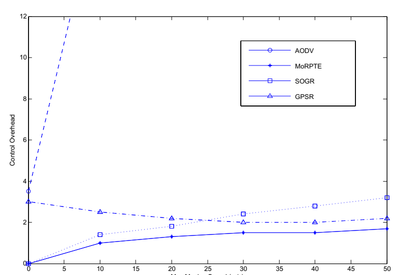

Fig. 11. control overhead

The scalability of AODV is limited, which is because of limited network range and restricted flooding. Therefore, when the networks are dynamic, the routes can be easily broken and packets are dropped. On the contrary, location-based routings such as MoRPTE are more scalable and robust. The following Fig. 10 shows the packet delivery ratios of different routing protocols with different maximum moving speeds. The location-oblivious routings have lower delivery ratios than location-based routings. MoRPTE behaves stable even when the nodes move fast, with the help of GPR location prediction of intermediate nodes. SOGR only use advertised locations and thus when nodes do not advertise its location for a period of time, the performance will be affected. Also the performance of SOGR decreases than the experiments conducted in [1], which is due to the random advertising and de-advertising of locations by intermediate nodes.

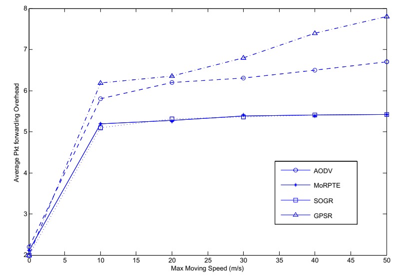

Fig. 12. average number of data packet forwarding

In Fig. 11, we find that MoRPTE generates least control messages. SOGR generate reasonably more messages as mobility increases. GPSR brings unnecessary overhead when the movements are slow. Fig. 12 shows that GPSR has the maximum overhead, due to non-optimal routing when topology is changed dynamically. SOGR's optimization process helps with the route changing and thus performs better. MoRPTE handles the situations too when in-precise prediction and detour occurs.

VI. Conclusion and Future Works

In this paper, we proposed a mechanism to adaptively control the exposure of a mobile user's trajectory while offer useful information for location-based services. We further discussed the controlling of exposure rate, specifying which locations to expose. We also proposed MoRPTE that utilizes location information flexibly in the spectrum from zero to full knowledge, assuring no trajectory exposure rate would exceed $r\%$. We may later introduce and examine more flexible controls, for instance, user specific or heterogeneous demands.

References

[1] X. Xiang, Z. Zhou, and X. Wang, "Self-adaptive on demand geographic routing protocols for mobile ad-hoc networks," in *IEEE INFOCOM*, 2007.

[2] C. Liu and J. Wu, "Efficient geometric routing in three dimensional ad hoc networks," in *IEEE INFOCOM*, 2009.

[3] Mobile consumer briefing. The Mobile Marketing Association, http://goo.gl/BTYv4.

[4] X. Lin, R. Lu, P. Ho, X. Shen, and Z. Cao, "Asrpake: An anonymous secure routing protocol with authenticated key exchange for wireless ad hoc networks," in *IEEE ICC*, 2007.

[5] X. Li, M. R. Lyu, and J. Liu, "A trust model based routing protocol for secure ad hoc networks," in *Aerospace Conference*, 2004.

[6] Y. P. Chen, A. L. Liestman, and J. Liu, "Clustering algorithms for ad hoc wireless networks," in *Ad Hoc and Sensor Networks. Nova Science Publishers*, 2004.

[7] C. Rasmussen and C. Williams, "Gaussian Processes for Machine Learning," in *The MIT Press*, 2006.

[8] W. Azmy, N. Gayar, A. Atiya, and H. El-Shishiny, "Mlp, gaussian processes and negative correlation learning for time series prediction," in *Springer-Verlag MCS*, 2009.

[9] D. Ellis, E. Sommerlade, and I. Reid, "Modelling pedestrian trajectories with gaussian processes," in *IEEE International Workshop on Visual Surveillance*, 2009.

[10] E. C. H. Ngai and Y. Zhou, "Reliable reporting of delay-sensitive events in wireless sensor-actuator networks," in *In Proc. of the 3rd IEEE MASS*, 2006.

[11] B. Karp and H. Kung, "Gpsr: Greedy perimeter stateless routing for wireless networks," *ACM MobiCom*, 2000.