A Tracking Assisted Relaying Scheme for Decentralized Wireless Networks

Baohua Shao, Mark S. Leeson
School of Engineering
University of Warwick
Coventry CV4 7AL, UK
e-mail: b.shao@warwick.ac.uk; mark.leeson@warwick.ac.uk

***Abstract*— We present a computationally simple approach for mobile node motion recognition and prediction in decentralized wireless networks. The ability to observe and analyze movement data and infer targeted mobile subscriber motion strategies offers performance gains in Delay Tolerant Networks (DTNs). The method here offers new mechanisms for the Store-Carry and Forward Relaying Scheme to create opportunities for the system to increase its overall performance. The motion data are processed by the Kalman Filter (KF) algorithm, which may be viewed as a series of prediction, tracking and smoothing calculations of the movement of mobile subscribers. The outcomes illustrate that the algorithm performs well on prediction and tracking the motion of mobile users. A Java based routing protocol implementation of the KF algorithm offers a delivery probability of at least 0.7 and a mean hop count of between 2.1 and 3.7 for small and large networks, respectively. For modest node densities, between 60% and 80% of the messages will be delivered but after a substantial delay. Thus, the approach described will be of benefit in data collection as part of the Internet of Things and small sensor networks.**

Keywords*— *Store-Carry and Forward; Decentralized Wireless Networks; Kalman Filter

I. INTRODUCTION

One of the aspects introduced into networking by the use of wireless links is the potential mobility of the communicating nodes but in a classic infrastructure-based wireless network, a fixed access point provides onward routing into a wired network [1]. In recent years, the idea of mobile ad hoc networking without a pre-defined structure or designated central controller has gained in popularity with the use of multiple network hops providing the necessary communication range [2]. In such a decentralized wireless network, each of the mobile nodes is allowed to determine how to assist other network nodes since stable end-to-end paths rarely exist because of node mobility, node sparsity and node connection or disconnection [3]. The prevailing network conditions produce a Delay Tolerant Network (DTN), in which connectivity is provided between a pair of nodes despite the intermittent connectivity and long delays to provide more network flexibility and resilience [4]. The network uses the limited connectivity to forward segments of the payload, which can thus be randomly forwarded to any neighboring nodes. This has the potential to lead to unmanageable system efficiency, intolerable overall delay and considerable energy wastage. As a result, specialized routing paradigms have been developed for DTNs [5] that utilize the Store-Carry and Forward (SCF) relaying scheme [6] that is further discussed in Section II. There has also been substantial interest in the use of predictive information to assist in the SCF process, which will also be discussed in Section II. The mechanisms used to determine the forwarding in SCF will critically influence the wireless network performance. The key metrics include network efficiency, transmission delay, Quality of Service (QoS), energy efficiency and network load distribution. Wireless networks provide subscriber mobility and flexibility, allowing portable device vendors to implement more advanced features than in wired networks. The SCF relaying scheme utilizes this mobility to achieve relaying but the positional uncertainty of the wireless nodes can lead to uncontrollable DTN performance. Thus, subscriber motion prediction offers the prospect of well-managed and optimized relay routing.

The rest of the paper is organized as follows. In Section II, the context for the work is provided by a short overview of the literature. The methods of node movement prediction are introduced in the Section III and then the simulation methods are contained in Section IV, culminating with the results obtained. The final section presents the conclusions and suggestions for future work.

II. RELATED WORK

To deal with the motion of the nodes within a DTN, it has been common to form routing paths between nodes that are in each other's direct communication range [7]. Thus, the network needs to maintain an end-to-end structure whilst its intermediate structure varies with node movement. This is difficult because the variations in node positions constantly change the underlying communication graph and mean that nodes must quickly adapt to the new configurations. One of the methods for solving this problem is link reversal [8], which models the problem as a directed graph, reversing the link directions when needed as a result of motion induced connection loss. Unfortunately, as shown in [8], the time to produce a stable link for communication grows as the square of the number of nodes in the network, limiting the scalability of such algorithms. As a result, the SCF approach [6] was developed, in which intermediate mobile nodes store messages in their local memories if they do not encounter a suitable relay node. The messages are then carried whilst the nodes move until they find an appropriate node to which they can forward their data towards the destination.

 ISBN: 978-1-61208-602-6

Early DTN protocols such as epidemic routing [9] operated without network information to aid their decisions. The target in such an approach is to spread packets rapidly throughout the network without a node selection criterion (that would need extra information). Packets are copied at all node encounters and persist in the network until they reach their destination or exceed a chosen lifetime. Protocol performance drops with increasing load because of the growing demands for storage space and low probability that useful forwarding nodes will be encountered rapidly. Limiting the number of copies permitted was introduced by protocols such as Spray-and-Wait (SnW) [10]. In this method, once the maximum number of copies is reached, the carrying node keeps the packet until it reaches the destination, storage limits are exceeded or the packet times out. To overcome the limitations of the random approach above, many protocols have been developed that collect network information to select relay nodes enhancing delivery probability despite limited storage and energy resources [5]. A well-known example of a protocol that predicts contacts among DTN nodes is PROPHET [11]. This produces a node metric via the number of meetings between nodes; the link weightings between nodes are increased when they meet along with the weightings of other nodes that they have met. The adoption of this method produces an increased delivery ratio but at the price of an increased average packet delay. The information gleaned from node interactions may also be used to detect what can be described as *social* relationships between the network nodes [12]. These formalize the concept that to be considered part of the same community, nodes should be in frequent, regular and long-lasting contact that will suggest promising forwarding paths. For brevity, the summary above naturally leaves out many variations on the themes presented, so the interested reader is referred to [13] for further details and references.

With particular reference to uncertainty in wireless subscriber movement prediction, it is known that given knowledge of a large population, accuracies approaching 90% can be achieved [14]. However, here we need real-time estimation based on limited information. Sometimes, the DTN in question will have movement restrictions such as that considered by Ahmed and Kanhere [15]. They considered operation where public transport networks or street patterns reduced the range of subscriber movement choices to simplify the prediction work. In general, we need to allow the networks nodes more freedom and the approach taken can be reactive or proactive [16]. In the former, nodes report their location to a central network authority such as a base station. However, in the latter, prediction is used and this has the potential to reduce the inevitable latency whilst waiting for location updates. The uncertainty arises from the mobility model extending into the future based on known mobility history data. The success of a mobility model depends on how well it can learn and predict future node locations based on the available scenario history [16]. User movements are to a large degree predictable [17] so the problem becomes one of designing an efficient location prediction algorithm using past data.

Similarly, the idea of using prior probability and Bayesian inference to properly drive a search process in ad hoc delay tolerant networks has been exploited [18]. This use of a generic computable inference mechanism to increase the performance of DTNs has gained popularity in the last few years, culminating in a recent study employing a weighted feature Bayesian predictor that outperforms a naïve Bayesian approach [19]. However, there is no comprehensive and systematic research study on the entire system to improve the network performance by using rigorous prediction and analysis methods. Although Kalman filtering has been used to update connection probabilities [20], the work in [18] was the first adoption of Bayesian inference, in the context of DTN routing. However, the main focus of the paper is on gradient routing in which the message tends to follow a gradient of increasing utility function values towards the destination. Another paradigm has been employed by Talipov et al. [21], who utilize a hidden Markov model to predict to predict the future location of individuals. The inspiration for the scheme is the same as ours and based on the observations of Gonzalez et al. [22] that human trajectories show a high degree of temporal and spatial regularity, and in social environments individuals move subject to a deterministic schedule with only a few random deviations.

III. PREDICTION METHOD

The movement prediction of mobile terminals comprises a series of estimations of moving targets. This can use techniques from problems in different areas such as tracking flying objects using radar. Tracking is a special case of estimation, as the inference of mobile subscriber movement will be represented as a set of complex state space estimation elements [23], each of which records a certain mobile subscriber's position, instantaneous velocity and instantaneous acceleration (or deceleration). For each particular moment or interval, every individual mobile node has its own state data set indicating its state space information forming a state space identification vector. A series of these vectors record the trajectory of a mobile subscriber or preset mobile user group within the network.

All the mobile nodes have the ability to move freely around the radio frequency coverage area, with their random motion forming a random walk stochastic system. The unknown state of the targeted wireless subscriber (denoted by X) is computable based on the observation or measurement (denoted by Y) of mobile subscriber behavior. Prediction is then possible with inference using historical measurements [24].

These potentially computationally burdensome tasks, such as algorithm computation and historical data storage, need to be performed by each individual subscriber's mobile device, such as smartphones, tablets, e-book readers, portable handsets and laptops. The outcomes need to be propagated wirelessly. Each mobile device will have its own limitations on processing capacity, embedded storage memory and particularly wireless bandwidth. Thus, the computerized algorithms need to be simplified and utilized on a minimized scale, which is within mobile device capabilities including the available wireless link bandwidth. Each mobile node only needs to track and predict nodes that can establish direct bi-directional radio connections between the two adjacent nodes. The prediction information is only exchanged among these neighboring mobile nodes. Thus, for prediction, each mobile node needs to obtain its neighbor node state information nodes by movement tracking.

Here, established Bayesian statistical methods are used to accomplish the moving object motion prediction operation [25]. According to the overall behavior of mobile subscribers, the nodes will be classified into different categories by utilizing different criteria, for instance, non-maneuvering objects and maneuvering objects. If the objects are maintaining a constant velocity so that they may be classified as non-maneuvering objects, then the system is Linear Quadratic Gaussian (LQG) [24] and may be solved using linear system models. The motion of maneuvering objects is normally more dynamic with different accelerations and the trajectory is non-linear so the solution will be more difficult and perhaps only sub-optimal solutions can be obtained [26].

IV. TRACKING STRATEGIES

The tracking problem is actually to estimate the state of moving mobile subscribers (targets) based on the observation data via statistical algorithms. The state of the targets can thus be seen as in a dynamical system [27] with time independent states, forming an autonomous system. The trajectories of targeted mobile subscribers are normally continuous, but observations are made at fixed time intervals and so are taken in a discrete mode. This mathematical statistics status is called the continuous – discrete filtering mode [23], with the discrete observations forming the state space information input.

The classical Bayesian approach provides us with a method to deduce the further states of observed moving objects. Bayes' theorem [25] implies that the mobile node states can be predicted from the observation data, which is the joint probability of the state of event *x* and the observation of event *y* divided by the unconditional probability of the observation of event *y*, which is the normalization factor.

The movement of a mobile subscriber is a random walk [24] obeying the Markov property [28], so the stochastic motion of each mobile node can be treated as a series of Markov process individually. A first order Markov chain can be used for predicting the state space identification of each mobile subscriber step by step. The recursive Bayesian solution is [28]:

$$p(\boldsymbol{x}^k|\boldsymbol{y}^k) = \frac{p(\boldsymbol{y}_k|\boldsymbol{x}_k)}{p(\boldsymbol{y}_k|\boldsymbol{y}^{k-1})} p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}) p(\boldsymbol{x}^{k-1}|\boldsymbol{y}^{k-1}) \quad (1)$$

Leading to a state conditional density:

$$p(\boldsymbol{x}_k|\boldsymbol{y}^k) = \int_{\boldsymbol{x}_{k-1}} p(\boldsymbol{x}^k|\boldsymbol{y}^k) d\boldsymbol{x}_{k-1} \quad (2)$$

In these equations, the superscripts refer to vectors of all *x* or *y* values from one to *k* or *k*-1 whereas the subscripts denote single instances of *x* or *y*.

A. Simulation Model

The targeted system and observation methods are based on linear system models with quadratic system optimization. The wireless system and observation are subject to Gaussian noise so they obey the basic LQG regulator [24]. Hence, the object tracking and movement prediction problem can be solved by a Kalman Filter (KF) [24]. Equation (2) is the recursive estimation of the state conditional density function and the term $p(\boldsymbol{x}^{k-1}|\boldsymbol{y}^{k-1})$gives the prior probability density function. In the Bayesian recursive solution, $p(\boldsymbol{x}_k|\boldsymbol{y}^k)$ is a conditional density of the targeted mobile subscriber state $\boldsymbol{x}_k = (x_{k1}, x_{k2}, \dots, x_{kn}) \in \mathbb{R}^{\mathrm{n}}$ at the moment k given all the observed data $\boldsymbol{y}^k = (\boldsymbol{y}_1, \boldsymbol{y}_2, \dots, \boldsymbol{y}_k)$ with $\boldsymbol{y}_k = (y_{k1}, y_{k2}, \dots, y_{km}) \in \mathbb{R}^{\mathrm{m}}$.

The moving object tracking algorithm with noise is:

$$\boldsymbol{x}_k = \mathrm{f}(\boldsymbol{x}_{k-1}) + \boldsymbol{v}_k \quad (3)$$

where $\mathrm{f}(\boldsymbol{x})$ is some function of $\boldsymbol{x}$ and $\boldsymbol{v}_k$ is a vector of Gaussian noise.

In practice, the movement of mobile users cannot remain at a constant velocity or absolute steady state but relatively small perturbations occur that can be regarded as Gaussian noise. Given that only a small portion of wireless users will exhibit high mobility [22], such a model is of some utility.

In the decentralized wireless networks designed to date, each mobile node has to observe the movement of other nearby nodes and try to estimate the state to implement the SCF relaying scheme. Here, this state is restricted to the position and velocity of the mobile subscriber wireless nodes. The observation cannot be ideal, and there is always some noise that enters the system. Generally, the KF algorithm is able to deal with two kinds of noise, namely measurement or sensor noise and transition or process noise [29]. Both types of noise are zero mean Gaussian in nature, and the dynamic and observation models are linear Gaussian. The filtering model presented above obeys the basic LQG regulator as mentioned before, so the filtering equation can be expressed as [30]:

$$\boldsymbol{x}_k = \mathbf{A}\boldsymbol{x}_{k-1} + \boldsymbol{q}_{k-1} \quad (4)$$

$$\boldsymbol{y}_k = \mathbf{H}\boldsymbol{x}_{k-1} + \boldsymbol{r}_k \quad (5)$$

where $\boldsymbol{x}_k$ is the hidden state vector and $\boldsymbol{y}_k$ is the observation vector at time *k*, respectively; $\boldsymbol{q}_{k-1} \sim N(0, Q)$ is the transition noise; $\boldsymbol{r}_k \sim N(0, R)$ is the sensor noise.

The movement of the mobile subscriber is described by two-dimensional Cartesian coordinates, so the hidden state vector has four dimensions $\boldsymbol{x}_k = (x_{k1}, x_{k2}, x_{k3}, x_{k4})$. The first two elements capture the position of the mobile node and the second two represent its corresponding velocity. The observation vector is $\boldsymbol{y}_k = (y_{k1}, y_{k2})$.

The matrices within the dynamic model are:

$$\mathbf{A} = \begin{pmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\mathbf{Q} = \begin{pmatrix} 0.1 & 0 & 0 & 0 \\ 0 & 0.1 & 0 & 0 \\ 0 & 0 & 0.1 & 0 \\ 0 & 0 & 0 & 0.1 \end{pmatrix}$$

where Δt is one second in the simulations and $\boldsymbol{Q}(i, j)$ is the transition covariance [14].

The matrices in the observation model are:

$$\mathbf{H} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}$$

$$\mathbf{R} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

where $\boldsymbol{R}(i,j)$ is the observation covariance [14].

Here, the KF equations can be described as two steps [30]:

(i) prediction:

$$\boldsymbol{m}_k^- = \mathbf{A}_{k-1}\boldsymbol{m}_{k-1} \quad (6)$$

$$\boldsymbol{P}_k^- = \mathbf{A}_{k-1}\boldsymbol{P}_{k-1}\mathbf{A}_{k-1}^T + \boldsymbol{Q}_{k-1} \quad (7)$$

(ii) update:

$$\mathbf{S}_k = \mathbf{H}.\mathbf{P}_k^-.\mathbf{H}^T + \mathbf{R} \quad (8)$$

$$\mathbf{K}_k = \mathbf{P}_k^-.\mathbf{H}^T.\mathbf{S}_k^{-1} \quad (9)$$

$$\mathbf{m}_k = \mathbf{m}_k^- + \mathbf{K}_k.\{\boldsymbol{y}_k - \mathbf{H}.\mathbf{m}_k^-\} \quad (10)$$

$$\mathbf{P}_k = \mathbf{P}_k^- - \mathbf{K}_k.\mathbf{S}_k.\mathbf{K}_k^T \quad (11)$$

In which

$\boldsymbol{y}_k$ is the measurement at the time step *k*;

$\mathbf{P}_k$ is the covariance of a Kalman/Gaussian filter at the time step *k*;

$\mathbf{P}_k^-$ is the predicted covariance of a Kalman/Gaussian filter at the time step *k* just before the measurement $\boldsymbol{y}_k$;

$\boldsymbol{S}_k$ is the innovation covariance of a Kalman/Gaussian filter at step *k*;

$\mathbf{K}_k$ is the gain matrix of a Kalman/Gaussian filter;

$\mathbf{m}_k$ is the mean of a Kalman/Gaussian filter at the time step *k*;

$\mathbf{m}_k^-$ is the predicted mean of a Kalman/Gaussian filter at the time step *k* just before the measurement $\boldsymbol{y}_k$.

Before the filtering process starts, both the state vector **initial_state** (which is a column vector) and the state covariance vector **initial_V** have to be initialized thus:

$$\textbf{initial\_state} = \begin{pmatrix} 10 \\ 10 \\ 0 \\ 0 \end{pmatrix}$$

$$\textbf{initial\_V} = \begin{pmatrix} 10 & 0 & 0 & 0 \\ 0 & 10 & 0 & 0 \\ 0 & 0 & 10 & 0 \\ 0 & 0 & 0 & 10 \end{pmatrix}$$

B. *Algorithm Simulation*

To simulate the scenario studied, the true mobile user locations are generated by MATLAB, producing a stochastic linear dynamical system, which is a type of hidden state [29]. This is because the mobile node states cannot be directly measured by neighboring mobile subscribers and KF algorithms are used for estimation. Figure 1 illustrates the results of simulated KF algorithms using 50 individual states in each time step. These are the true states that simulate the real locations of the mobile subscriber during a continuous period of time, and that are represented by the black squares. The trajectory shown by the black line linking the black squares is the ‘real path’ of the motion of a certain mobile node. The blue stars indicate the observed location of the mobile device which simulates the measurements from another neighboring mobile terminal. The red crosses show the KF outcomes, processed by the neighboring mobile smart device with the estimated path represented by the red dotted line.

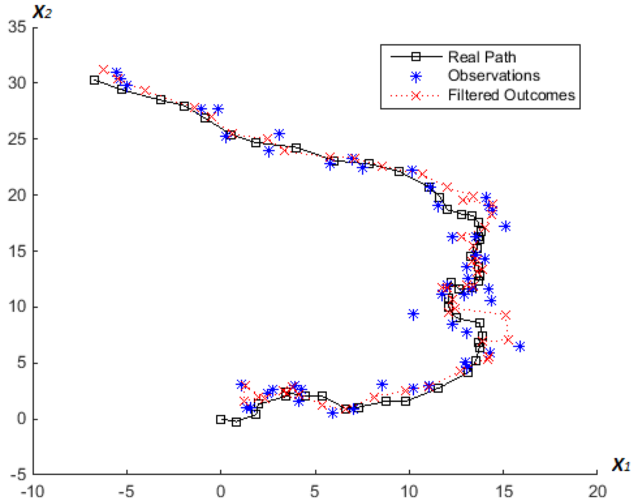


Fig. 1. Results of the prediction simulation for the filtering model.

It may be seen in Figure 1 that for most of the time, the filtered trace represents the true path well. Only when the mobile user’s movement is more dynamic (close to the maneuvering model), particularly the right hand side of Figure 1, does the algorithm have difficulty following the true path. Nevertheless, when the motion of the object exhibits behavior that is close to the non-maneuvering scenario, the outcomes still reflect the real motion of the target very well as in the top and bottom parts of the trajectory, and the mismatched portion is relevant small.

C. *Protocol Simulation*

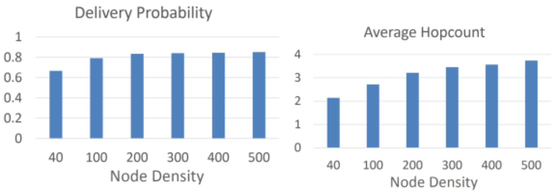
The simulation testbed for this part used the Opportunistic Network Environment (ONE) simulator and a JAVA based protocol for the KF routing scheme was developed. For testing the performance, resilience and tolerance of designed protocol, the sample dataset that comes with the ONE simulator package was utilized to simulate a complex wireless network condition, which is the data collected from the downtown Helsinki area. Parameters for the simulation configurations are specified in Table 1. These are chosen to be of the same order as the parameters in [5] with the buffer size large enough that it does not impact performance [5].

TABLE I. PARAMETERS OF SIMULATION CONFIGURATIONS

Simulation Time (s)	86400
Buffer Size (MB)	50
Packet Lifetime	100 minutes
Message Interval (s)	3, 5, 10, 20, 30, 60
Message Size (kB)	500
Number of Nodes	40, 100, 200, 300, 400, 500

The message interval simulated the information rate of the sender. The parameters for this category tested the circumstances from a low packet generation rate of 1 packet per minute (67 kbps) to a high packet generation rate of 20 packets per minute (1.33 Mbps). The number of nodes varied the density of the wireless system from a low-density (40 nodes) mobile network to an extremely high-density (500 nodes) system.

In this work, there are four key factors of wireless system that are addressed to evaluate the overall performance of proposed mobile routing strategy, which are: Delivery Probability, Overhead Ratio, Average Latency and Average number of hops.

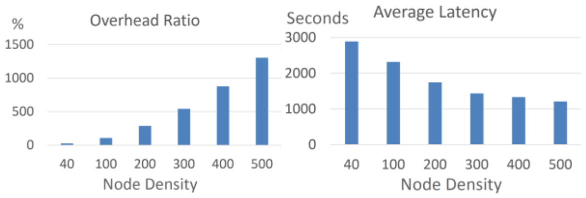


(a) (b)

Fig. 2. (a) Delivery probability; (b) average hop count for different network densities.

Figure 2 shows the performance of the proposed protocol at the maximum bit rate considered. It provides good resilience for different network densities and maintains a delivery probability in excess of 0.7 for all circumstances. Moreover, as the algorithm is able to predict the movement of portable nodes, the protocol delivers an average hop count of between 2.1 and 3.7, leading to the involvement of fewer intermediate nodes in the relaying path saving retransmission energy and improving efficiently.

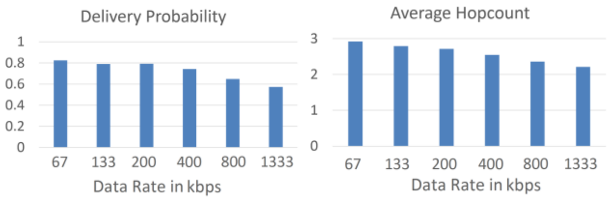
Figure 3 shows that the overhead ratio rises sharply with the number of nodes since there are more possible packet relay candidates. However, there is a corresponding decrease in the average latency as there are more nodes that can complete delivery. The balance of these two factors makes the protocol able to maintain useful performance when the network setup changes.



(a) (b)

Fig. 3. (a) overhead ratio; (b) average latency for different network densities.

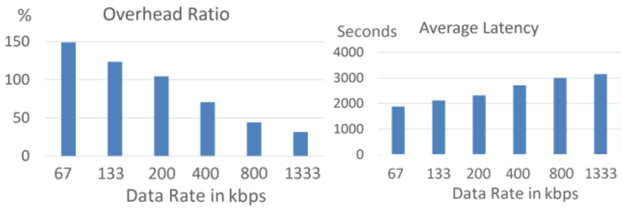
To test the capability of the protocol to deal with various traffic volumes, the packet generation rate in a network comprising 40 nodes was varied. Figure 4 illustrates the variation in delivery probability and hop count as the data rate increases. The former drops with increasing traffic volumes but the KF protocol still maintains a probability of approximately 0.6 whilst the hop count falls from almost three to a little over two with increasing bit rate.



(a) (b)

Fig. 4. (a) Delivery probability; (b) average hop count as a funcion of data rate for low node density.

Figure 5 shows that the overhead ratio decreases from 148% to 31% as the bit rate increases but this is accompanied by an increase in average Latency from 1875 seconds to 3153 seconds.



(a) (b)

Fig. 5. (a) Overhead ratio; (b) average hop count as a funcion of data rate for low node density.

The KF relaying scheme exhibits a good overall performance which benefits from the portable device movement predication ability allow more packets to arrive successfully at the receiver or be relayed to the correct intermediate nodes. This feature maintains the delivery probability at a high value whilst and keeping the average hop count at a low level.

V. Conclusions and Future Work

The KF is an *optimal recursive data processing algorithm* [24] that provides the online estimation solution to solve the object tracking problem. We have presented a detailed evaluation of the performance of a protocol that utilizes KF algorithm models. This has shown that such an approach enables smart devices to predict and track the motion of a targeted mobile node and assist it to find the next hop as a better or best option for a relaying route. The subsequent routing protocol simulation results proved the theoretical idea. The strengths of the KF are that the algorithm is rather small and simple and thus the majority of mobile devices and sensors are able to process the program. Moreover, the algorithm does not require substantial memory resources to store the movement history of targeted mobile node.

The results indicate that the KF algorithm will face challenges when significant numbers of wireless users fall into the category where a maneuvering model is needed. When the user is moving unsteadily, both the direction and the velocity could be changing at all times; in the case, acceleration or deceleration will be included in future as another dimension of the state vector to indicate the state of the mobile subscriber. However, as the dimension of the inputs becomes high, the calculation volume will substantially increase exponentially. Thus, to let the algorithm still be available for individual smart devices, the computation time should be taken into account. Nevertheless, the results indicate that at modest node densities, the protocol will deliver between 60% and 80% of the messages but after a substantial delay. Thus, real-time applications will not be well served by the simple approach taken here but it will be useful, for example in data collection as part of the Internet of Things.

Our ongoing work shows that in sparse networks, the KF algorithm exhibits similar delivery probabilities, latency and hops counts to established protocols such as Spray and Wait. Although some of the advantage is lost with dense and complex networks, the protocol's simplicity offers utility to, for example, small sensor networks. Its modest bandwidth requirements also offer advantages in constrained communication environments.

As a classical optimal prediction and tracking algorithm, the KF is suitable for many scenarios, since only small portion of wireless users will exhibit high mobility [17]. The introduction of users who move rapidly according to a random walker model as described by Shang [31] would lead to significant prediction errors. Hence, to broaden the application of this smart relaying scheme to include such very mobile users, other algorithms that can improve the prediction and tracking performance for the maneuvering model, such as the Extended KF (EKF), Unscented KF (UKF), Particle Filter and other potential filtering schemes [28] will be examined in the future.

REFERENCES

[1] J. Loo, S. Khan and A. N. Al-Khwildi, "Mobile Ad Hoc Network" in Mobile Ad Hoc Networks: Current Status and Future Trends, J. Loo, J. Lloret Maur and J. Hamilton Ortiz, Eds. Boco Raton: CRC, pp. 3-17, 2012.

[2] C. M. Conti and S. Giordano, "Mobile Ad Hoc Networking: Milestones, Challenges, and New Research Directions", IEEE Communications Magazine , vol 52, no. 1, pp. 85-96, 2014.

[3] C. S. Jain, K. Fall and R. Patra, "Routing in a delay tolerant network", ACM SIGCOMM Computer Communication Review, vol. 34, no. 4, pp. 145–158, 2004.

[4] A. M. Vegni and T. D. C. Little, "VANETs as an Opportunistic Mobile Social Network", in Opportunistic Mobile Social Networks, J. Wu and Y. Wang, Eds. Boca Raton: CRC, pp. 438-465, 2015.

[5] T. Abdelkader, K. Naik, A. Nayak, N. Goel and V. Srivastava, "A Performance Comparison of Delay-Tolerant Network Routing Protocols", IEEE Network, vol. 30, no. 2, pp. 46 – 53, 2016.

[6] C. C. Sobin, V. Raychoudhury, G. Marfi and A. Singla, "A survey of routing and data dissemination in Delay Tolerant Networks", Journal of Network and Computer Applications, vol. 67,pp. 128-146, May 2016.

[7] I. Chatzigiannakis, "Communication in Ad Hoc Mobile Networks Using Random Walks", Encyclopedia of Algorithms, New York: Springer, pp. 357-363, 2003.

[8] C. Busch and S. Tirthapura, "Analysis of link reversal routing algorithms. SIAM Journal of Computing, vol. 35, no. 2, pp. 305-326, 2005.

[9] A. Vahdat and D. Becker, "Epidemic routing for partially connected ad hoc networks", Technical Report CS-2000-06, Department of Computer Science, Duke University, 2000.

[10] T. Spyropoulos, K. Psounis and C. S. Raghavendra, "Spray and Wait: An Effcient Routing Scheme for Intermittently Connected Mobile Networks," ACM SIGCOMM Workshop on Delay-Tolerant Networking (WDTN), Aug. 2005, pp. 252-259. ISBN:1-59593-026-4.

[11] A. Lindgren, A. Doria and O. Schelén, "Probabilistic routing in intermittently connected networks", ACM SIGMOBILE Mobile Computing and Communications Review, vol. 7, no. 3, pp. 19-20, 2003.

[12] E. Bulut and B. Szymanski, "Exploiting Friendship Relations for EfficientRouting in Mobile Social Networks", IEEE Transactions on Parallel and Distributed Systems, vol. 23, no. 12, pp. 2254-65, 2012.

[13] P. K. Verma et al., "Machine-to-Machine (M2M) communications: A survey", Journal of Network and Computer Applications, vol. 66, pp 83-105, March 2016.

[14] X. Lu, E. Wetter, N. Bharti, A.J. Tatem and L. Bengtsson, "Approaching the Limit of Predictability in Human Mobility", Scientific Reports 3, art. 2923, 2013.

[15] S. Ahmed and S. S. Kanhere, "Characterization of a Large-scale Delay Tolerant Network," IEEE Conference on Local Computer Networks (LCN), Oct. 2010, pp. 56 – 63.

[16] N. Dimokas, D. Katsaros, P. Bozanis, and Y. Manolopoulos, "Predictive Location Tracking in Cellular and in Ad Hoc Wireless Networks," Mobile Intelligence, L. T. Yang, A. B. Waluyo, J. Ma, L. Tan and B.Srinivasan Eds., New York: Wiley, pp. 163–190, 2010.

[17] I. Burbey and T. L. Martin, "A survey on predicting personal mobility", International J. of Pervasive Computing and Communications, vol. 8, no. 1, pp.5-22, 2012.

[18] S. Ahmed, S.S. Kanhere,"A Bayesian Routing Framework for Delay Tolerant Networks", Wireless Communications and Networking Conference (WCNC), Apr. 2010, pp. 1-6.

[19] K. Jahromi, M. Zignani, S. Gaito and G. P. Rossi, "Predicting encounter and colocation events", Ad Hoc Networks, vol. 62, pp. 11-21, Jul. 2017.

[20] M. Musolesi, S. Hailes, and C. Mascolo. Adaptive routing for intermittently connected mobile ad hoc networks, IEEE WoWMoM, pp. 183–189, Jun. 2005.

[21] E. Talipov, Y. Chon and H. Cha, "Content sharing over smartphone-based delay-tolerant networks", IEEE Transactions on Mobile Computing, vol. 12, no. 3, pp. 581-595, 2013.

[22] M.C. Gonzalez, C.A. Hidalgo, and A.-L. Barabasi, "Understanding Individual Human Mobility Patterns," Nature, vol. 453, pp. 779-782, Jun. 2008.

[23] Y. Bar-Shalom, X. R. Li and T. Kirubarajan, Estimation with applications to tracking and navigation. New York: Wiley, 2001.

[24] P. S. Maybeck, Stochastic Models, Estimation and Control. New York: Academic Press, 1982.

[25] A. Gelman, J. B. Carlin, H. S. Stern, D. B. Dunson, A. Vehtari and D. B. Rubin, Bayesian data analysis. Boca Raton: Chapman & Hall/CRC, 2014.

[26] N. Ahmed, M. Rutten, T. Bessell, S. S. Kanhere, N. Gordon and S. Jha, "Detection and Tracking Using Particle-Filter-Based Wireless Sensor Networks," IEEE Transactions on Mobile Computing, vol. 9, no. 9, pp. 1332 – 1345, 2010.

[27] E. R. Scheinerman, Invitation to Dynamical Systems. New York: Prentice Hall, 1996.

[28] S. Challa, M. R. Morelande, D. Musicki and R. J. Evans, Fundamentals of Object Tracking. Cambridge, New York: Cambridge University Press, 2011.

[29] S. J. Russell and P. Norvig, Artificial intelligence: a modern approach. 3rd ed. Boston: Pearson, 2010.

[30] S. Särkkä, Bayesian filtering and smoothing. Cambridge: C.ambridge University Press, 2013.

[31] Y Shang, "Consensus in averager-copier-voter networks of moving dynamical agents", Chaos, vol. 27, art. 023116, [9 pages], 2017.