

Least Squares Method Prediction-based Spray and Focus Routing Protocol in Opportunistic Networks

Zhuowei Shen

Key Lab of Computer Network and Information Integration
(Southeast University), MOE
School of Computer Science and Engineering
Southeast University
Nanjing, China, 025- 52090901-805
zwshen@seu.edu.cn

Ying Tu

Key Lab of Computer Network and Information Integration
(Southeast University), MOE
School of Computer Science and Engineering
Southeast University
Nanjing, China, 025- 52090901-805
tuying@seu.edu.cn

Abstract—Opportunistic Networks develops rapidly in recent years. With the popularity of GPS (Global Positioning System), velocity or position prediction plays an important role in opportunistic network routing. In this paper, a new opportunistic routing protocol named LSMPSF (Least Squares Method Prediction-based Spray and Focus) is proposed. With least squares method, LSMPSF predicts velocity by curve fitting. According to the predicted velocities, it estimates the neighbor nodes' delivery ratio and makes routing decision. Simulation results reveal that LSMPSF effectively increases the prediction accuracy and reduces resource consumption. Compared with SF and PROPHET, LSMPSF achieves better performance on delivery ratio and overhead ratio.

Keywords—Opportunistic Networks, Spray and Focus, Least squares method

I. INTRODUCTION

Opportunistic Networks [1, 2], unlike existing ad-hoc networks requiring global connectivity, use “stay-carry-forward” pattern to transmit messages. With the popularity of GPS technology, more and more nodes in opportunistic networks have the ability to obtain their position information, which provides potential capability to improve the existing routing mechanisms and enhances the performance of network.

Several routing protocols have been proposed [3–5]. Among them, SF (Spray and Focus) [6] is a very popular scheme. It controls the number of message copies. In spray phase, a source node or relay node distributes its message copies to other nodes until one message copy is left. Then SF turns into focus phase: relays predict neighbors' utility values and choose the neighbor with the highest utility value to forward messages. However, the utility value in SF is based on the time elapsed since the two nodes last encountered each other, without considering the positions and the relative velocity of the two nodes. Moreover, even if a message is delivered successfully, the other copies are still kept until their TTL (Time to live) expire, which waste lots of resources.

Lebrun et al. [7] proposed a MOVE algorithm. Wang and Amza [8] proposed motion-based routing protocols. With these protocols, nodes can get their velocity and position information,

while destination node is stationary and its geographical position is globally known. The former set the forwarding utility value as the closest distance to the destination. The latter uses different motion information, ranging from speed and direction, to the past or expected trajectory of the local node. The drawback of these schemes is the difficulty to get position and velocity information of the destination. PPSW [9] is another routing protocol proposed by Wang et al. With PPSW, a node records all the other nodes' history positions, which are used to predict a node's future position with Lagrange Interpolation method. A node will select the nearest neighbor node away from the destination as a relay to forward messages. However, this approach wastes storage, communication, computing resources. Moreover, high order polynomials in Lagrange interpolation method can be highly oscillatory.

In this paper, while the destination node is in motion and its geographical position is not globally known, we propose a new routing protocol named LSMPSF (Least Square Method Prediction-based Spray and Focus) based on SF, which uses least square method curve fitting to predict the velocity, position and calculate the encounter possibility. In addition, AMI (Ack Message IDs) [10] mechanism is used to delete the copies belonging to the successfully delivered messages. With LSMPSF, each node needs to record its past positions and corresponding time, which are used to predict the node's velocity vector by least square method cure fitting. When two nodes encounter, the only information need to be exchanged is the whole utility information table. Compared with other schemes, LSMPSF not only reduces the storage and communication consumption, but also needs only local calculation for utility value update.

The rest of the paper is organized as follows: Section II addresses the LSMPSF routing protocol and buffer management method. Section III evaluates the performance of LSMPSF by simulations. We conclude the paper in section IV.

II. LEAST SQUARES METHOD PREDICTION-BASED SPRAY AND FOCUS ROUTING PROTOCOL

A. Least Squares Method-based Trajectory Prediction

Curve fitting [11] is the process to make a function that has the best fit to a series of data points. We can predict the trajectory of the node in future using the history of positions via the curve fitting method. By minimizing the sum of squared errors, least squares method [12] can obtain the best linear function matching to data. In reality, nodes (people, vehicles, etc.) often move along urban roads. The trajectories are not completely random or chaotic. However, the high order polynomials are unstable. Considering the urban road and low polynomial is more suitable to fit. According to the average velocity during a period of time, the future position can be effectively predicted. We define linear function as follows:

$$\begin{cases} x = v_x t + a \\ y = v_y t + b \end{cases} \quad (1)$$

Where (x_i, y_i) denotes the node position, (V_x, V_y) is the velocity, and t is the time. We suppose that each node periodically updates its own position records every Δt seconds. According to the least squares method, we can solve these unknown coefficients as follows:

$$v_x = \frac{\sum_{i=1}^n t_i x_i - n \bar{t} \bar{x}}{\sum_{i=1}^n t_i^2 - n \bar{t}^2} \quad (2)$$

$$v_y = \frac{\sum_{i=1}^n t_i y_i - n \bar{t} \bar{y}}{\sum_{i=1}^n t_i^2 - n \bar{t}^2} \quad (3)$$

$$a = \bar{x} - v_x \bar{t} \quad (4)$$

$$b = \bar{y} - v_y \bar{t} \quad (5)$$

where $\bar{t} = \frac{1}{n} \sum_{i=1}^n t_i$, $\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i$, and $\bar{y} = \frac{1}{n} \sum_{i=1}^n y_i$,

are arithmetic average values.

B. Least Squares Method Prediction-based Spray and Focus Method

In LSMPSF, each node maintains a SV (Summary Vector) with IDs of messages in buffer, the whole utility information table, and IDs of the delivered messages. The utility information of a node includes the time, the position and the velocity last updated which were calculated via Equations (2~5). In addition, the whole utility information table contains the utility information of itself and all the other nodes which a node could get in the past. As SF, LSMPSF is divided into two phases: spray phase and focus phase.

1) Spray phase

Whenever nodes encounter, they exchange SVs to check the messages they have in common. LSMPSF limits the

number of message copies to L , and a node maintains forwarding token for every message in its buffer. Compared with flooding, it efficiently reduces the network overhead. If a node carrying a message copy with $n > 1$ forwarding token encounters a node with no copy of that message, it spawns and forwards a copy to the neighbor: it hands over $\lfloor n/2 \rfloor$ tokens and keeps $\lceil n/2 \rceil$ for itself. When there is only one token left, it will turn into the focus phase.

2) Focus phase

When two nodes encounter each other, they update the utility information of themselves and get the utility information of other nodes via interchanging SVs. With this information, a node can calculate its' neighbors forwarding utility values, which indicates the success possibilities that neighbors can relay messages to destination. Here we use the cosine value between the direction of motion and the distance vector to the destination as forwarding utility value. When the angle is smaller, the message will be more likely to be finally delivered.

As shown in Figure 1, suppose there is a message in node i need to be delivered to destination d . Node i encounters node j and they exchange SVs of themselves, then they update the whole utility information table in their buffer. The utility values of node i and j are: $u_i = \cos \lambda$, $u_j = \cos \phi$. The position of destination d is (x_d, y_d) , which is calculated by Equation (1). Meanwhile, the time, the velocity and the last position are coming from the utility information of the destination node in the table. The velocities and positions of node i and j are $V_i (v_{ix}, v_{iy})$, (x_i, y_i) , $V_j (v_{jx}, v_{jy})$, (x_j, y_j) . The distance vector from i to d is $(x_d - x_i, y_d - y_i)$, distance vector from j to d is $(x_d - x_j, y_d - y_j)$. Therefore, the utility values of node i and j can be calculated as follows:

$$u_i = \cos \lambda = \frac{(x_d - x_i, y_d - y_i) \cdot (v_{ix}, v_{iy})}{|(x_d - x_i, y_d - y_i)| \times |(v_{ix}, v_{iy})|} \quad (6)$$

$$u_j = \cos \phi = \frac{(x_d - x_j, y_d - y_j) \cdot (v_{jx}, v_{jy})}{|(x_d - x_j, y_d - y_j)| \times |(v_{jx}, v_{jy})|} \quad (7)$$

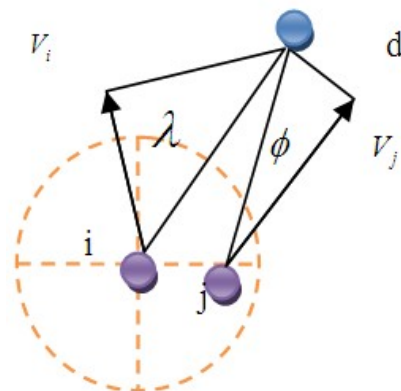


Figure 1. Parameters for LSMPSF Method

If $u_j > u_i$, it shows that the angle of ϕ is smaller than λ , which means that the node j is moving closer to destination d than node i . Therefore, the message should be forwarded to node j . Moreover, for increasing efficiency in focus phase, we set the threshold Δu . Node i forward the message to node j only if $u_j - u_i > \Delta u$. And messages will be sorted by utility value so that the message with higher utility value is sent earlier to improve the performance when the bandwidth is limited. The pseudo code of the focus phase is shown in Figure 2.

```

//process of focus phase in LSMPSF
for each node i
  update its own position every  $\Delta t$ ;
  for each encounter node j
    send messages with the destination of j;
    delete them from its buffer B(i);
    update its own utility information;
    exchange SV with j;
    update the whole utility information table of i;
    for each message in B(i) but not in B(j)
      if  $U(j) - U(i) > \Delta u$ 
        add into the forwarding list;
      else
        keep the message;
    end
  end
  sort the messages in the forwarding list;
  forward those messages in the forwarding list;
end
end

```

Figure 2. Pseudo code of the focus phase in LSMPSF

C. Buffer management mechanism

With original SF routing protocol, a node keeps a message copy and forward it until its TTL expired or was deleted for buffer shortage, even if other copy of that message has been successfully delivered to the destination. That results in a waste of bandwidth, buffer and computing resources. To address this problem, LSMPSF applies the AMI (Ack Message IDs) mechanism. Each node maintains an AMI vector, which includes the IDs of those successfully delivered messages. The AMI is updated while interchanging SVs in spray and focus phases. According to the AMI, all the message copies with the same IDs in AMI can be removed immediately. The pseudo code of buffer management mechanism in LSMPSF is shown in Figure 3.

```

//process of buffer management mechanism
for each node i
  delete message whose TTL expires or whose ID in AMI every  $\Delta t$ ;
  if a new message produced or receive a message but buffer B(i) runs out
    delete the message received at the earliest;
  end
  for each message
    if its TTL expires
      delete its ID from AMI;
    else
      keep the ID;
    end
    if its destination is i
      add its ID into AMI;
    else
      do nothing;
    end
  end
end
end

```

Figure 3. Pseudo code of the buffer management mechanism

III. SIMULATION RESULTS

We used the ONE (Opportunity Networking Environment) [13, 14], a discrete event simulator developed by the Helsinki University, to simulate and analyze opportunity network routing protocols. Suppose all nodes are cooperative and homogeneous. They move along urban road based on Shortest Path Map Based Movement model. We compare the performance of LSMPSF against the other three routing protocols: SF, PROPHET [15] and LSF. LSF is the same as LSMPSF, except that the velocity prediction is not made by least squares method, but by Lagrange Polynomial Interpolation [9]. Because the existence of divergence known as Runge's phenomenon Lagrange Polynomial Interpolation may yield a polynomial oscillating above and below the true function. And this behavior tends to grow with the number of points. There we set the number of points 3. In our simulation, number of the copies of a message is set to 10% of the number of nodes, and Δu is set to 0.2. The main simulation parameters are shown in Table 1.

TABLE I. SIMULATION PARAMETERS

Simulation area Size[m ²]	4500 × 3400
Simulation time[h]	12
Transmission speed[Kbps]	250
Number of nodes	50~250 Default:100
Moving speed[m/s]	0.5~7.5 Default:1.5~2.5
Message size	500KB~1MB
Message TTL[h]	1
Message creation interval[s]	25~35
Buffer size[MB]	1~25 Default:5
Transmission range[m]	10~50 Default:30

We use delivery ratio and overhead ratio as performance metrics, which are described as follows:

$$delivery_ratio = \frac{delivered_msg}{created_msg} \quad (6)$$

$$overhead_ratio = \frac{relayed_msg - delivered_msg}{delivered_msg} \quad (7)$$

where $delivered_msg$ is the number of the messages which were delivered successfully to the destination nodes, $created_msg$ is the number of the messages which were created by those source nodes, and $relayed_msg$ is the number of the message copies which were relayed. Delivery ratio and overhead ratio reflect the performance and average consumption of the networks.

```

//process of buffer management mechanism
for each node i
  delete message whose TTL expires or whose ID in AMI every  $\Delta t$ ;
  if a new message produced or receive a message but buffer B(i) runs out
    delete the message received at the earliest;
  end
  for each message
    if its TTL expires
      delete its ID from AMI;
    else
      keep the ID;
    end
    if its destination is i
      add its ID into AMI;
    else
      do nothing;
    end
  end
end
end

```

Figure 4. Delivery ratio under different buffer size

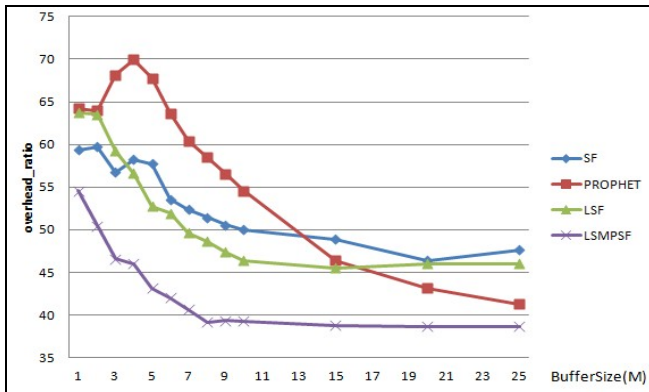


Figure 5. Overhead ratio under different buffer size

In the first simulation experiment (Figure 4, 5), we vary buffer size of nodes from 1MB to 25MB. If buffer size is small, an effective message is more likely to be deleted due to the occupancy of buffer, which results in low delivery ratio and high overhead ratio. As shown in Figure 4 and 5, with more accurate prediction and deleting invalid message through AMI, LSMPSF shows much better performance at delivery ratio and overhead ratio compared with SF, LSF and PROPHET.

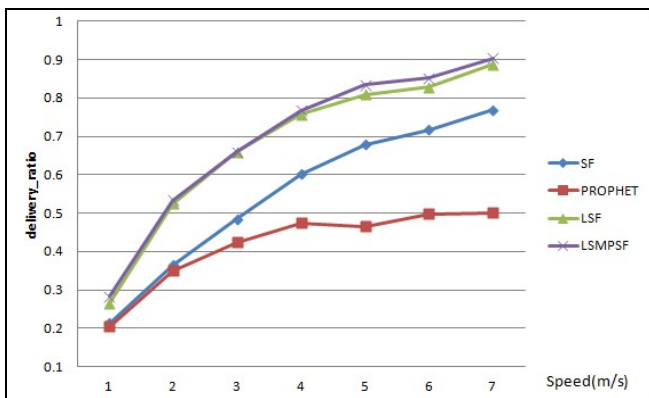


Figure 6. Delivery ratio under different node speed

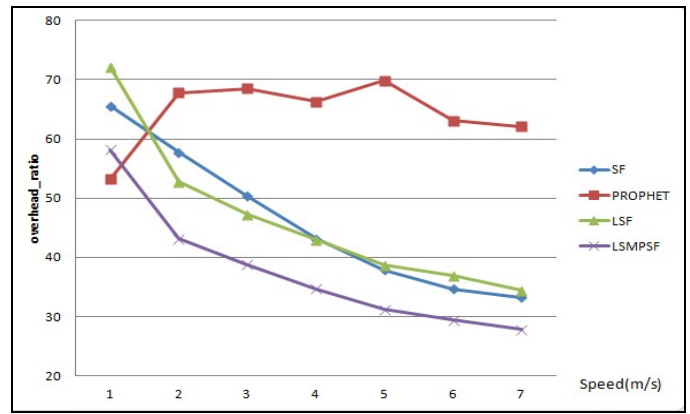


Figure 7. Overhead ratio under different node speed

In the second simulation experiment, we vary the node speed from 1m/s to 7m/s. When speed is increasing, the time for communicating between neighbor nodes becomes shorter. On the other hand, increasing speed also brings about more meeting chances. The overhead ratio is decreasing because the utility information can be updated more frequently and the prediction can be more accurate. As Shown in Figure 6 and 7, LSMPSF gains the highest delivery ratio and the lowest overhead ratio.

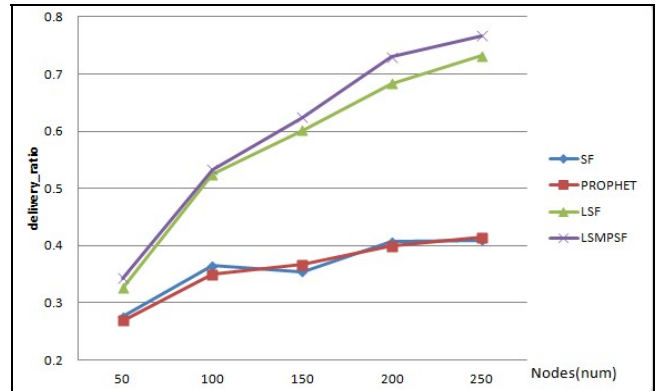


Figure 8. Delivery ratio under different nodes number

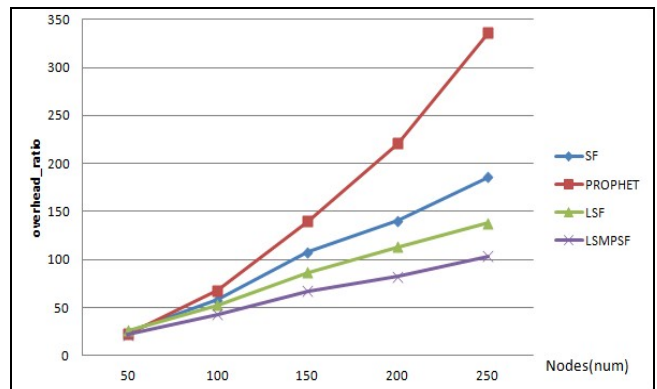


Figure 9. Overhead ratio under different nodes number

In the third simulation experiment, we increase the number of nodes from 50 to 250. With nodes number increasing, one node will meet more other nodes in a period of time and the connectivity of the network will be enhanced. Therefore, the

delivery ratios of all the protocols increase. However, more nodes and the changing networks lead to more possibility to forward messages in vain, which makes overhead ratios of these four protocols increase. As shown in Figure 8 and 9, LSMPSF, which has better prediction than LSF, avoids too much invalid forwarding in focus phase, shows the best performance at delivery ratio and overhead ratio.

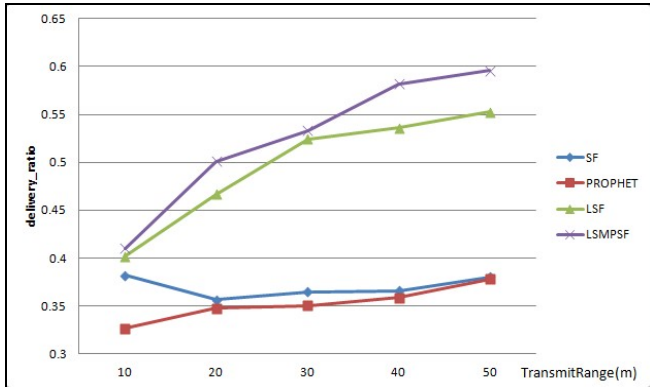


Figure 10. Delivery ratio under different transmit range

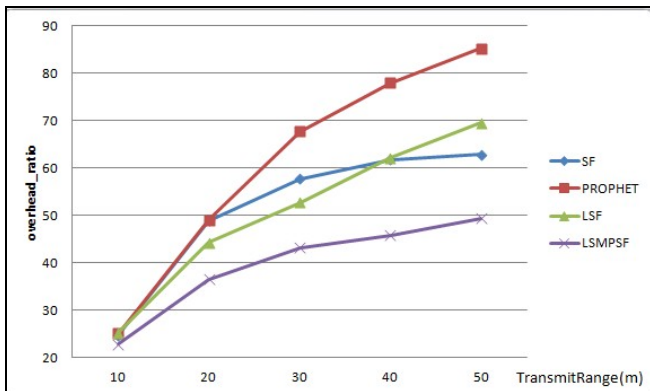


Figure 11. Overhead ratio under different transmit range

In the last simulation experiment, we change the transmit range of the nodes from 10m to 50m. Increasing transmit range prolongs communication time between neighbor nodes, and expands a node's communication coverage, which covers more neighbors. Therefore, it leads to a better delivery ratio. However, more chance to encounter a neighbor whose utility is much greater than this node leads to worse overhead ratio. As shown in Figure 10 and 11, LSMPSF has the best delivery ratio and overhead ratio among all these four protocols.

IV. CONCLUSIONS

In this paper, an opportunistic routing protocol named LSMPSF is proposed. Based on SF, it uses the least squares method to predict a node's velocity, and then calculate forwarding utility value, which is the index to make forwarding decision. Meanwhile, it uses AMI mechanism to delete the copies belonging to the successfully delivered messages, so as to save buffer, bandwidth, and computing resources. Compared with those existing algorithms based on position prediction, LSMPSF does not need to maintain the history of all the nodes' positions in a node. Instead, a node

needs only to record its history positions and calculate the velocity locally at a very low computational complexity. Moreover, when two nodes encounters, the information exchanged is very small. In urban roads scenario, LSMPSF can predict node movement accurately, which makes it achieve better performance. Simulation results validate this: compared with SF, PROPHET, and LSF, LSMPSF has better delivery ratio and overhead ratio in different simulation scenarios.

REFERENCES

- [1] Luciana Pelusi, Andrea Passarella, and Marco Conti, "IIT-CNR. Opportunistic Networking: Data Forwarding in Disconnected Mobile Ad Hoc Networks," *IEEE Communications Magazine*, pp. 134-141, 2006, 44(11).
- [2] C.M. Huang, K.C. Lan, and C.Z. Tsai. "A survey of opportunistic networks," *22nd International Conference on Advanced Information Networking and Applications (IEEE)*, pp. 1672-1677, 2008.
- [3] Xiong Yongping, Sun Limin, Niu Jianwei, and Liu Yan. "Opportunistic Networks," *Journal of Software*, pp. 124-137, 2009, 20(1).
- [4] Ren Zhi, Huang Yong, and Chen Qianbin. "Opportunity network routing protocol," *Journal of Computer Applications*, 2010.
- [5] Nguyen, Hoang Anh, and Silvia Giordano. "Routing in Opportunistic Networks," *International Journal of Ambient Computing and Intelligence (IJACI)*, pp. 19-38, 2009.
- [6] Spyropoulos T, Psounis IL, and Raghavendra CS. "Spray and focus: Efficient mobility-assisted routing for heterogeneous and correlated mobility," In: *Proc. of the IEEE PerCom Workshop on Intermittently Connected Mobile Ad Hoc Networks*, 2007.
- [7] Jason Lebrun, Chen-nee Chuah, Dipak Ghosal, and Michael Zhang. "Knowledge-based opportunistic forwarding in vehicular wireless ad hoc networks," *IEEE 61st Vehicular Technology Conference VTC 2005 Spring*, 29th May-1st June, Stockholm, Sweden, 2005.
- [8] Weihang Wang, and Cristiana Amza. "Motion-based routing for opportunistic ad-hoc networks," In *proceeding of: Proceedings of the 14th International Symposium on Modeling Analysis and Simulation of Wireless and Mobile Systems, MSWiM 2011, Miami, Florida, USA, October 31 - November 4, 2011*.
- [9] Wei Huang, Sihai Zhang, and Wuyang Zhou. "Spray and wait routing based on position prediction in opportunistic networks," In *proceeding of: Computer Research and Development (ICCRD), 2011 3rd International Conference on*, vol. 2, 2011.
- [10] Yong-Pyo Kim, Ja-Il Koo, Euihyun Jung, Keisuke Nakano, Mazakazu Sengoku, and Yong-Jin Park. "Composite Methods for Improving Spray and Wait Routing Protocol in Delay Tolerant Networks," *Communications and Information Technologies (ISCIT), 2010 International Symposium on*, pp. 1229-1234, 2010.
- [11] Richard L. Burden, and J. Douglas Faires. *Numerical Analysis*. Brooks Cole, December 10, 2004.
- [12] J. Wolberg. *Data Analysis Using the Method of Least Squares: Extracting the Most Information from Experiments*. Springer, 2005.
- [13] Ari Keränen, Jörg Ott, and Teemu Kärkkäinen. "The ONE Simulator for DTN Protocol Evaluation," *Proceedings of the 2nd International Conference on Simulation Tools and Techniques*, 2009.
- [14] The Opportunistic Network Environment simulator. <http://www.netlab.tkk.fi/tutkimus/dtn/theone/>.
- [15] Lindgren A, Doria A, and Schelén O. "Probabilistic routing in intermittently connected networks," *ACM SIGMOBILE Mobile Computing and Communications Renew*, pp. 19-20, 2003, 7(3).