MONICA: One Simulator for Mobile Opportunistic Networks

Peiyan Yuan, Mingyang Song
{peiyan@htu.cn, hsdyjssmy@126.com}

School of Computer and Information Engineering, Henan Normal University
46# East Construction Road Xinxiang, Henan, China 453007

Abstract. Routing protocols are the focus in mobile opportunistic networks, and the simulation test is one of the effective ways to evaluate their performance. This paper presents a mobile opportunistic network simulator called MONICA. It provides a programmable interface to users, which is easy to learn and easy to use. It also integrates eight classic routing algorithms, six caching strategies and two types of mobility models, so that users can make a more comprehensive evaluation of the routing protocol. In addition, MONICA can trace abnormal behaviors with the network event function and observe the real simulation process by the visual module. All of these guarantee the efficiency of the simulator in evaluating the routing performance.

Keywords: Mobile opportunistic networks, simulator, routing protocol

1 Introduction

With the rapid development of short distance wireless communication and sensor technology, the current mobile devices, such as smart phones, tablet PCs and notebook computers have powerful capabilities in communication and data sensing. Mobile opportunistic networks (MONs) composed of these devices can be used to perform complex self-organizing communication tasks and support emerging applications ranging from disaster rescue service to mobile social networking service. Different from traditional wireless sensor networks or ad hoc networks with end-to-end data transmission paths [14] [3] [4] [10], in MONs, there is no reliable end-to-end path between the source and destination, the transmission of packets depends mainly on the mobility of nodes and the opportunistic contact between them. Therefore, how to transmit packets effectively is a big challenge in MONs.

The performance of MONs obviously relies on the mobility of nodes and many routing protocols have been proposed to deal with the intermittently connected scenarios. One consequent problem is how to evaluate the routing protocols? If we set up a test bed to evaluate them, the cost will increase exponentially with the increase of the number of mobile equipment (in fact, it needs a lot of people or vehicles carrying devices for packet transmission and reception). In addition, a single network scenario cannot reflect the true performance of routing protocols. Considering the above issues, it is of great significance to develop a scalable and powerful simulator for MONs.

In this paper, we propose a mobile opportunistic network simulator MONICA based on the Microsoft visual C++ platform, it integrates a variety of classic opportunistic protocols and mobility models. Our contributions are summarized as follows.

MOBIMEDIA 2018, June 21-22, Qingdao, People's Republic of China

DOI 10.4108/eai.21-6-2018.2276563

• MONICA provides a scalable simulation architecture which supports the basic functions such as generation of network events, modeling of network nodes, packet switching between nodes, buffer management and power consumption.

• MONICA integrates 2 kinds of mobility models, 8 routing algorithms, 6 caching strategies, one energy consumption model, as well as the interface defined by the extension function. In addition, a dynamic visualization window permits users to observe the real process.

The rest of this paper is organized as follows. In the second section, we review the related work of mobile opportunistic network. In the third section, we introduce the architecture and functions of MONICA. In the fourth section, we introduce how to use MONICA to set up the simulation scene. In the fifth section, we evaluate and analyze the performance of routing algorithms under different mobility models with the MONICA. finally, we summarize the paper.

2 RELATED WORK

The core of routing protocols in MONs is how to select the next node to store and forward packets [12]. From the point of view of whether additional information is needed to assist routing forwarding, the current routing protocol can be divided into two categories: zero information and auxiliary information [18]. The zero information routing algorithm [17] [11] [20] [16] does not use any information to make forwarding decisions. For example, the Epidemic [17] only floods packets to each node. The auxiliary information routing algorithm [8] [5] [9], on the other hand, needs some information to help the routing decision. For example, the contact probability in the PROPHET [8] is used as a reference for routing decision.

The performance of opportunistic routing protocols is mainly dependent on the mobility model of nodes. There are two main types: (1) real trace based model, (2) synthetic model. A number of projects [6] have dedicated to the collection of real human trajectory (including the location and time stamp) or connection data between users and wireless access point. However, the real trace is limited by its acquisition environment terrain, and the routing algorithms have different performance under different real traces. The synthesized model is a flexible and scalable option, such as random waypoint model [1], community mobility model [17]. These models can be used to match the moving patterns of pedestrians and vehicles by adjusting the moving speed, residence time and communication range etc.

The evaluation of the performance of opportunistic routing protocols should be carried out in a network environment with certain conditions. However, in reality, the use of a large number of mobile devices for opportunistic routing protocol testing will cost a lot. Therefore, it is a more economical and effective way to evaluate the performance of opportunistic routing using simulator. At present, NS-2 is one of the classic simulators. It is an object-oriented network simulator with a discrete event generator, and can be used to simulate a variety of different types of networks. NS-2 uses C++ and OTcl to add functionalities. Compared to NS-2, NS-3 adds some new features, such as multi card processing, the use of IP addressing strategy, more detailed 802.11 modules, etc. NS-3 use C++ or Python to implement functions, the user can freely and flexibly choose one of them to write script code. Both NS-2 and NS-3 can implement the simulation for wireless networks, but they do not support any of opportunistic routing protocols. In addition, it needs to learn OTcl language and modify many complex modules before adding opportunistic routing protocols into NS-2 or NS-3, which is difficult for beginners to get a start. The simulator ONE [7] is developed based on Java, which is suitable for the simulation of routing protocol in intermittent network environment, it has the characteristics of object-oriented, discrete event driven and visual simulation of real MONs environment. Note that before running the ONE simulator, it first needs to install the jdk with proper version numbers and guarantee that the environmental variables are configured correctly. Otherwise, it cannot be opened. In the official site of ONE, there is a column related to questions and answers.

3 MOBILE OPPORTUNISTIC NETWORKS SIMULATOR

MONICA is a simulator based on microsoft visual C++ platform. It was designed especially for the mobile opportunistic network environment, and has the characteristics of object oriented, discrete event driven and realistic network environment. Compared with other network simulators, it is easy to use and is convenient for users to add extend function modules. MONICA can simulate and analyze multiple opportunistic routing protocols and caching strategies. The simulator engine is based on discrete event driven, in which the event represents the change of network state. When the event changes, the simulator can captures this change and executes different tasks. The simulator also includes functional modules such

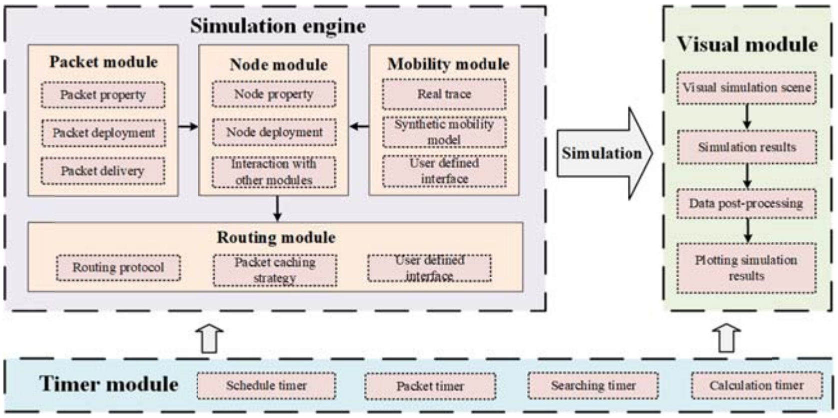


Fig. 1. Function module diagram.

as the packet module, node module, mobility module, routing module, visual module and event scheduling module as shown in Figure 1. The packet module is responsible for the generation and transmission of data packets between nodes. The node module is responsible for the deployment of nodes in the network and interaction with other modules. The mobility module contains a variety of node mobility models (synthetic and real traces), which is used to control the motion of nodes in the network and decides the communication distance between nodes and the scanning interval of neighbor relation. The routing module includes 8 kinds of classical routing algorithms and integrates a variety of caching strategies to further evaluate the performance of routing algorithms. In addition, users can add custom routing policies. The visual module outputs the simulation results, and the users can use the post-processing tool (such as the matlab) for graphical display. The graphical interface can be used to display the information of nodes such as the speed, duration, pause time and movement angel. Event scheduling module is responsible for the schedule of calculation timer and neighbor searching timer etc.

The interaction procedure between models is as follows: the node module deploys the node in the simulation area, and the packet module deploys the packet to the buffer of the node. Then the mobile module selects the mobility model for all nodes and the routing module selects the routing algorithm for these nodes. After the simulation, the experimental results can be plotted by the visual module.

3.1 Packet Module: Generation and Delivery

The properties of packets include the source ID, the destination ID, TTL, hops, packet size, etc.. Users can also add new properties to the packet module according to their own needs. The packet module has two aspects: how to generate packets and how to transmit packets. At the beginning of the simulation, by default, the system generates packets to source nodes randomly. Both the number of packets and the size of packets are randomly set. Users can change the above parameters by modifying the packet deployment interface.

3.2 Node Module

The node is the basic simulation unit, and it generally represents a person or a vehicle carrying a mobile communication device. One node in MONICA includes the ID, wireless interface, routing related information, buffer and power level. Besides the aforementioned configuration parameters, the complex features such as the mobility of nodes and the routing decision are realized by other specific functional modules.

Nodes can interact with other functional modules and share the simulation parameters and current states. For example, the mobility module can change the behavior of a node by modifying specific parameters (such as position, motion path, movement angle, etc.), and then manipulate the movement of nodes. The routing module can check and call the node's context information, buffer status, energy consumption and other information to make routing decisions, and then determines the packet forwarding. Since we mainly focus on the evaluation performance of routing protocols, we do not consider the underlying physical mechanism, such as communication signal attenuation, physical media congestion, and so on.

3.3 Mobility Module

The movement trajectory and path formed by the motion of nodes are the basis for evaluating the performance of routing algorithms. Therefore, MONICA integrates two types of mobile models: synthetic mobile model (random model, community model), real traces (KAIST, NCSU, et al.).

Random waypoint model is a classical mobility model for mobile networks, where nodes can select any position in the simulation area from the current position in a random direction and speed, the new speed and the moving angle are limited in a certain range, and the user can adjust them. At the same time, they also include some disadvantages such as the edge effect [2]. After analyzing some real data sets, we found that the human motion has self-similarity, using this clustering behavior can provide an effective heuristic knowledge for routing algorithm, therefore, the simulator integrates the community mobility model to reflect this self-similarity. In this model, the whole simulation area is divided into several regions, one of which is the gathering area, and the remaining area is the common community. Each node belongs to a subordinate home community, which is more likely to be visited than other communities. Each community has a fixed node as the gateway of that community. Nodes usually randomly selected and move to a destination for a certain period of time, and stay there for a while and then select another destination to move, when a node is in the home community, it has a greater chance to visit the gathering point, and nodes is not in their home communities, they return to their home communities with a big probability.

The simulator also includes some real data sets such as the KAIST and NCSU [13]. KAIST data set is from South Korea's KAIST, where 34 school volunteers handheld GPS devices to record their daily moving traces in a 10-second interval over a year, and total 92 operations were collected. NCSU data set is from the State University of North Carolina, where each week 2 to 3 students were randomly selected with mobile devices to record their daily activities, and total of 35 mobile traces were collected. The advantage of this kind of data set is that it can reflect the performance of the routing algorithm in the real situation. However, the performance of routing protocols reflected by different data sets may be slightly different, due to the constraints of the location.

3.4 Protocol Module: Routing Algorithm and Buffer Management Strategy

The simulation of opportunistic routing algorithm is the core function of this simulator. The routing algorithm mainly determines the packet transmission between nodes, so that the basic function of routing module is general, which includes the buffer queue, packet transmission, etc.. The calculation of routing information and the processing of forwarding decisions are made by routing information calculation function, therefore, when users add a custom routing algorithm, they only need to add a new calculation function.

MONICA contains 8 classic Routing protocols, i.e., Epidemic [17], DirectContact [11], Two-hop [20], SprayWait [16], Prophet [8], Simbet [5], PeopleRank [9], Hotent [19]. For Epidemic routing, when a node encounters another node, it will forward a copy of the packet it carries to the latter. In direct delivery, the node will always carry the packet until it reaches its destination. In the Two-hop routing, sources send packets to the encountered nodes. In the spray waiting, when two nodes meet, part of packets are copied to the other side. When the number of packets in the node is one, the direct delivery strategy is adopted. The Prophet calculates the probability of a node to the destination, and copy packets to nodes with a higher probability. Simbet used the centrality and similarity to select the relay node. Based on the classic web algorithm pagerank [15], PeopleRank calculates the centrality of nodes in a distributed way, and makes the high social nodes as relay nodes. The Hotent calculates the centrality of nodes using the relative entropy of the public hotspots and the private hotspots, and then calculates the personality of the node on the private hotspot area, at last, the authors use the law of gravity to assess the forwarding metric.

Usually, after a certain period of time, the node will receive more packets, the congestion will emerge and it affects the routing efficiency. Therefore, a good routing algorithm should take into account the buffer congestion strategy. MONICA integrates five buffer management strategies[14]: Drop Head (DH), Drop Tail (DT), Drop Random (DR), Drop Most Forwarded (MOFO) and Drop Oldest(DO). DH deletes packet in the head of the queue and inserts the new packet to the end of the buffer queue. DT deletes the packet at the end of the buffer queue. DR randomly deletes the packets in the buffer and inserts the new packet. MOFO drops the most forwarding packets and inserts the new packet in the queue according to the number of forwarding times. DO drops the packet with the smallest TTL. BetCentrality deletes packets with minimum centrality.

The simulator supports users to add a new routing module to customize the routing protocol. The routing module has the basic functions such as the actions of the two nodes when they enter the communication range (i.e., traversing and sending the packets, storing and managing buffers, etc.). For different routing algorithms, the decision function is different. Users can change these rules to make a new routing algorithm, and they can also construct a new buffer management strategy by the buffer storage and congestion management methods.

4 VISUAL SIMULATION SCENE

To create a simulation scenario, it is necessary to set the system parameters, mobile models and routing algorithms. We here show an example of the basic steps to create simulation scenario with MONICA.

Example: we need to simulate the Prophet protocol with the community mobility model, and to output the simulation results. The simulation steps are as follows:

First of all, select the system parameter item from the menu bar, and the system parameter setting interface is shown in Fig.2. The layout of this interface includes three parts: the simulation area, the data package and the node. In the simulation area, the user can set the width and length of the simulation area, as well as the simulation time and the packet generating rate. The maximum and minimum values of the packet size can be set in the package setting area. In the node setting area, users can set the number of nodes, the maximum communication distance between nodes, the period of nodes searching for neighbors (within the communication distance), the node buffer size. Since this example is only used to describe the configuration steps of the simulation scene, the specific experimental parameters of the sample are shown in the figure without specific elaboration.

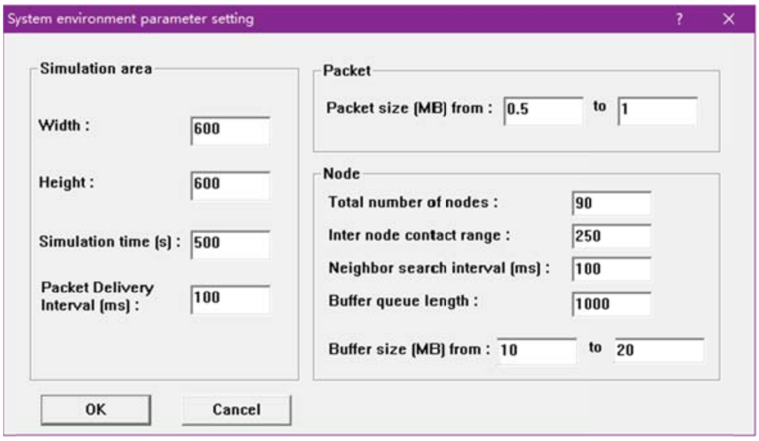


Fig. 2. Parameter settings interface.

After setting the system environment parameters, we can open the mobility model interface as shown in Fig.3. The interface is divided into the synthetic model area and the real model area. In the part of the synthetic area, the user can choose the random mobility model (including the random waypoint model and the random group model) and the community mobility model. Users can control the number of communities in the community mobility model by setting the number of rows and columns in the community. The setting of the node moving speed and dwell time can be used for all of the above synthetic mobility models. KAIST and NCSU can selected from the real model area.

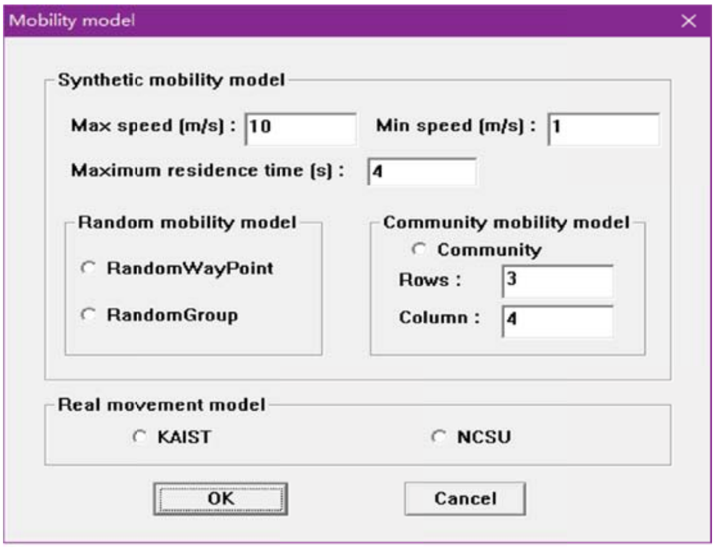


Fig. 3. Mobility Model settings interface.

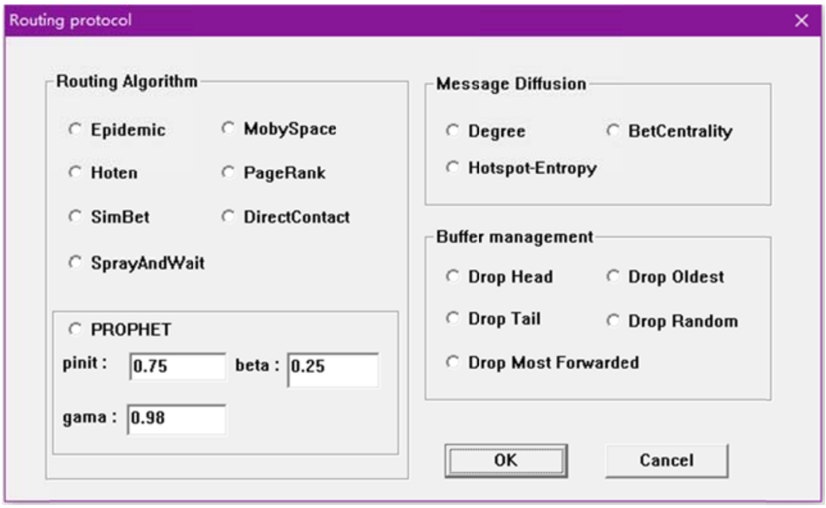


Fig. 4. Routing Protocol settings interface.

Finally, select the routing algorithm from the routing protocol setting interface as shown in Fig. 4. The algorithm interface includes the routing algorithm area and the buffer management policy area. The routing algorithm area integrates eight kinds of opportunistic routing protocols, where the Prophet algorithm can adjust the parameters of α, β and γ. The buffer management strategy area integrates six buffer management strategies. Users can choose any of them to simulate.

After the above three steps, click the menu bar to start the simulation. The simulator can show the simulation process and results in a visual way. The mobile node, the node information (including location, ID number, path, the connection between nodes) and the overview of the simulation area can be displayed in real time through the graphical user interface (GUI). Fig. 5 describes the simulation process of Prophet routing. It can be seen from the graph that the whole simulation area is divided into several communities with red lines. The nodes follows the movement rules described in the above section, the relevant information, including the node ID, moving angle and speed etc., is also marked. The blue line between nodes represents the communication between two nodes.

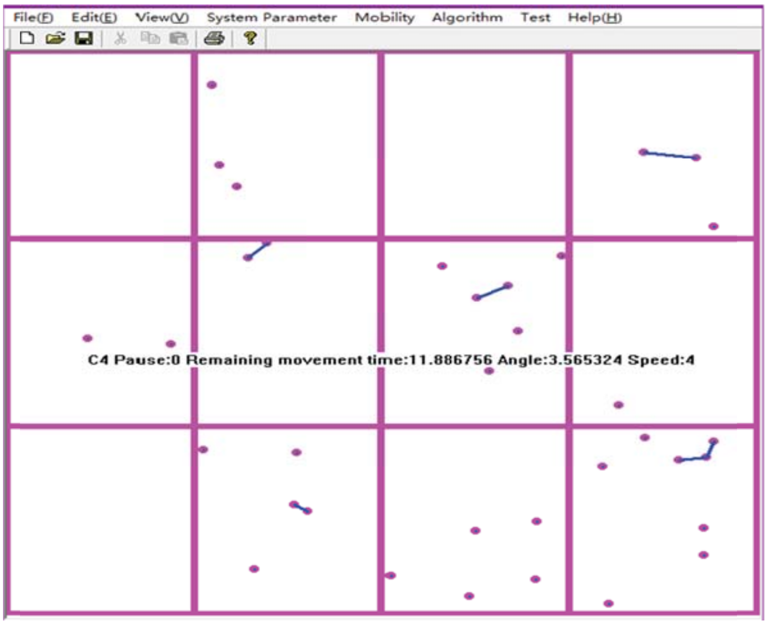
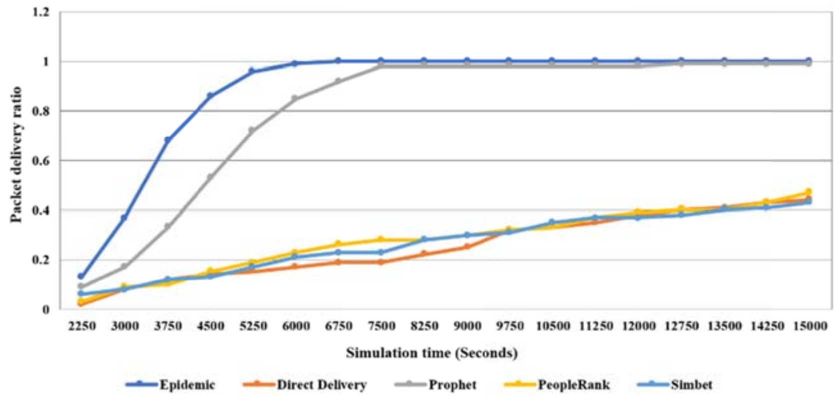


Fig. 5. Visual simulation process.

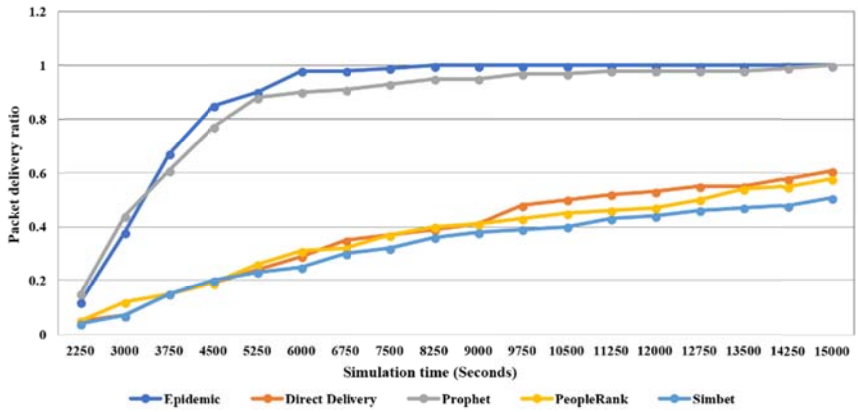
When the simulation is finished, the simulation results (routing performance information) are placed in the program root directory. The simulator provides the basic performance evaluation index of the routing algorithm, including the delivery ratio, delay, cost and hops. The delivery ratio is the ratio of the number of packets arriving at the destination over the total number of packets (excluding the copy of the package). The delay is the average time required for all packets from birth to reach destination. The cost is the average number of copies required to deliver a package. Hops represent the average number of relayed required for the packet to reach its destination node. The above indicators are presented in an accumulated manner. We plot the results and evaluate the performance of protocols in the following section.

5 EVALUATION AND ANALYSIS

In this section, we simulate five routing protocols with the synthetic mobile model (stochastic mobile model, community mobility model) and real mobile model (KAIST, NCSU). The simulation scenario is basically as follows: simulation time is 15000 seconds, the simulation area is 600m×600m. A total of 90 nodes participated in the simulation, the maximum communication distance between nodes is 10m, and the neighbor node is searched every 100 milliseconds. The buffer capacity of the node is 20mb. The way to deploy packets is to randomly select several nodes and deploy packets to their buffers. The packet size is evenly distributed from 500 kb to 1mb. The routing algorithms are Epidemic, DirectContact, Prophet, PeopleRank, Simbet. The difference in the performance of routing algorithm under different mobile models is analyzed in terms of packet delivery ratio.



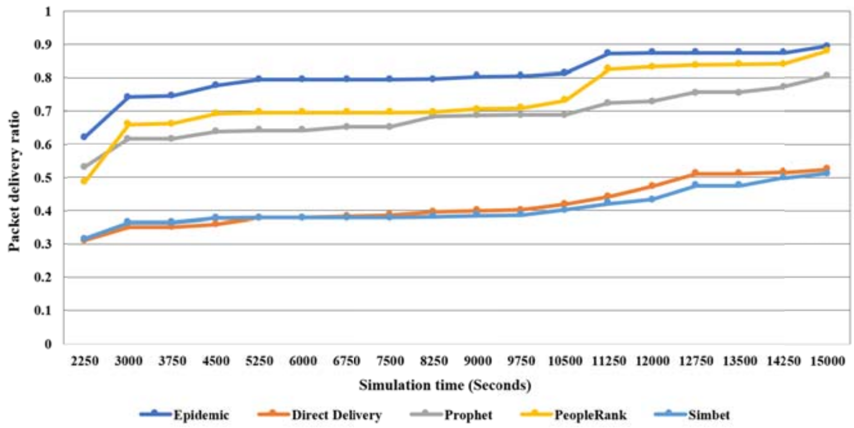
(a) Random waypoint model



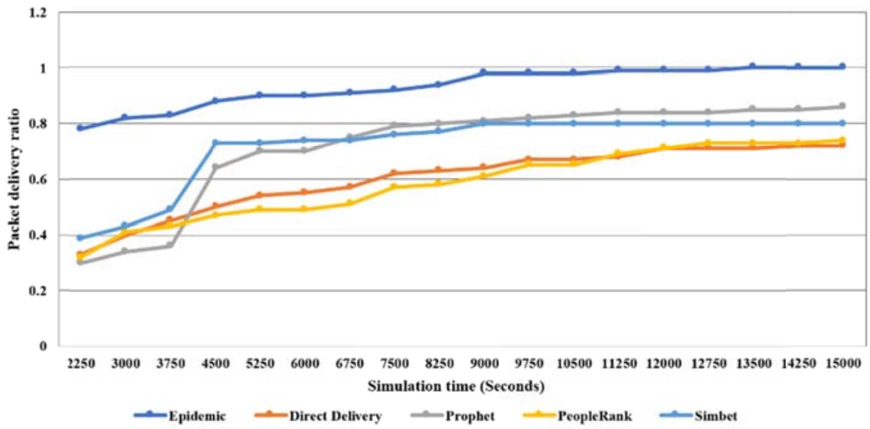
(b) Community mobility model

Fig. 6. Packet delivery ratio (synthetic mobile model)

Fig. 6 shows the cumulative delivery ratio of the five routing algorithms in two synthetic mobile models. Where the upper subfigure represents the simulation results under the random waypoint model, and the lower subfigure shows those with the community mobility model. It can be seen from the figure that the performance of the routing algorithm in the community mobility model overweighs those in the random waypoint model. This is mainly because that most of the nodes in the community mobility model tend to move toward the gathering area, resulting in a larger density of nodes in that area, which can promote the delivery efficiency. Especially for the two social-based routing algorithms, that is, the Prophet and SimBet, both of them are more affected.



(a) NCSU



(b) KAIST

Fig. 7. Packet delivery ratio(real traces).

Fig. 7 shows the cumulative delivery ratio of the five routing algorithms in the two real traces during the simulation time. Where the upper subfigure represents the simulation results under the NCSU, and the lower subfigure represents the simulation results under the KAIST. It can be seen from the figure that routing performance in KAIST is higher than that in NCSU. This is mainly because in the former, there are more nodes so the density of nodes is bigger than the latter. As a result, the contact ratio is bigger, which has a positive influence on the routing performance. This phenomenon also illustrates that the simulation environment is a key factor when we judge the performance of routing protocols. Different routing protocols may have different performance trends in different scenarios. That is to say, if a routing protocol has a better performance than another in one scenario, it cannot guarantee that it still keeps the superiority in other scenarios.

6 CONCLUSION

In this paper, we propose a mobile opportunistic network simulator MONICA, which is based on the C++ platform. It provides a powerful feature that allows users to simulate opportunistic routing protocols in a comprehensive scenario. Visualization of the interface can facilitate users to observe the changes in real-time conditions and enhance the user experience. The simulator integrates eight classic routing protocols and six buffer management strategies to help users to evaluate the performance of routing algorithms. Two types of mobility models are embedded to imitate the movement of nodes. In addition, a variety of custom interface modules helps users add modify some needed functions. In the future, we will develop and open more interfaces to the user.

Acknowledgments

This work was supported by the National Natural Science Foundation of China under Grant No. U1404602, the Young Scholar Program of Henan Province under Grant No.2015GGJS-086, the Science and Technology Foundation of Henan Province under Grant No.172102210341, the Dr. Startup Project of Henan Normal University under Grant No.qd14136, and the Young Scholar Program of Henan Normal University with No. 15018.

References

[1] Boudec J Y L, Vojnovic M. Perfect simulation and stationarity of a class of mobility models[C]// INFOCOM 2005. Joint Conference of the IEEE Computer and Communications Societies. IEEE, 2004:2743-2754.

[2] Camp T, Boleng J, Davies V. A survey of mobility models for ad hoc network research[J]. Wireless Communications & Mobile Computing, 2010, 2(5):483-502.

[3] Jacquet P, Muhlethaler P, Clausen T, et al. Optimized link state routing protocol for ad hoc networks[C]// IEEE INMIC 2001. IEEE, 2001: 62-68.

[4] Park V. Temporally-Ordered Routing Algorithm (TORA) Version 1 Functional Specification[S]. Internet Draft, draft-ietf-manet-tora-spec-04.txt, 2001, 5949(3):106-116.

[5] Daly E M, Haahr M. Social network analysis for routing in disconnected delay-tolerant MANETs[C]// Acm Interational Symposium on Mobile Ad Hoc Networking & Computing. 2007:32-40.

[6] Hui P, Chaintreau A, Scott J, et al. Pocket switched networks and human mobility in conference environments[C]// ACM SIGCOMM Workshop on Delay-Tolerant NETWORKING. ACM, 2005:244-251.

[7] Keränen, Ari, Ott, Jörg, Kärkkäinen, Teemu. The ONE simulator for DTN protocol evaluation[C]// International ICST Conference on Simulation TOOLS and Techniques. 2009:55.

[8] Lindgren A, Doria A, Schelén O. Probabilistic routing in intermittently connected networks[J]. Acm Sigmobile Mobile Computing & Communications Review, 2004, 7(3):19-20.
[9] Mtibaa A, May M, Diot C, et al. Peoplerank: social opportunistic forwarding[C]// Conference on Information Communications. IEEE Press, 2010:111-115.
[10] C. Perkins, E. Belding-Royer, and S. Das. 2003. Request for Comments: Ad hoc on-demand distance vector (AODV) routing. Experimental Internet Society 6, 7 (2003), 90.
[11] Shah R C, Roy S, Jain S, et al. Data MULEs: modeling and analysis of a three-tier architecture for sparse sensor networks[J]. Ad Hoc Networks, 2003, 1(2–3):215-233.
[12] Rashid S, Ayub Q, Zahid M S M, et al. Impact of Mobility Models on DLA (Drop Largest) Optimized DTN Epidemic routing protocol[J]. International Journal of Computer Applications, 2011, 18(5):35-39.
[13] Rhee I, Shin M, Hong S, et al. On the levy-walk nature of human mobility[J]. IEEE/ACM Transactions on Networking, 2011, 19(3):630-643.
[14] Pan D, Ruan Z, Zhou N, et al. A comprehensive-integrated buffer management strategy for opportunistic networks[J]. Eurasip Journal on Wireless Communications & Networking, 2013, 2013(1):103.
[15] Brin S, Page L. The anatomy of a large-scale hypertextual Web search engine [C]// International Conference on World Wide Web. Elsevier Science Publishers B. V. 1998:107-117.
[16] Spyropoulos T, Psounis K, Raghavendra C S. Efficient routing in intermittently connected mobile networks:the multiple-copy case[J]. IEEE/ACM Transactions on Networking, 2008, 16(1):77-90.
[17] Vahdat A, Becker D. Epidemic Routing for Partially-Connected Ad Hoc Networks. Master Thesis, 2000.
[18] Yuan P, Fan L, Liu P, et al. Recent progress in routing protocols of mobile opportunistic networks[J]. Journal of Network & Computer Applications, 2016, 62(C):163-170.
[19] Yuan P, Ma H, Fu H. Hotspot-entropy based data forwarding in opportunistic social networks[J]. Pervasive & Mobile Computing, 2015, 16(PA):136-154.
[20] Zhao W, Ammar M, Zegura E. A message ferrying approach for data delivery in sparse mobile ad hoc networks[J]. Proc of Acm Mobihoc, 2004:187-198.