Simulation Results of DiffPrivacyOppNet

In this document, we present the additional simulation results about the routing performance of the Opportunistic Network in the bicycle sharing system. These simulation results are the supplements of the experiments in the paper[1].

I. The ONE Simulator and the Python Simulator

Fig. 1 and Fig. 2 present the routing performance in the Python simulator and the ONE simulator, respectively. There are two different points in the implementation details between the ONE and the Python simulator.

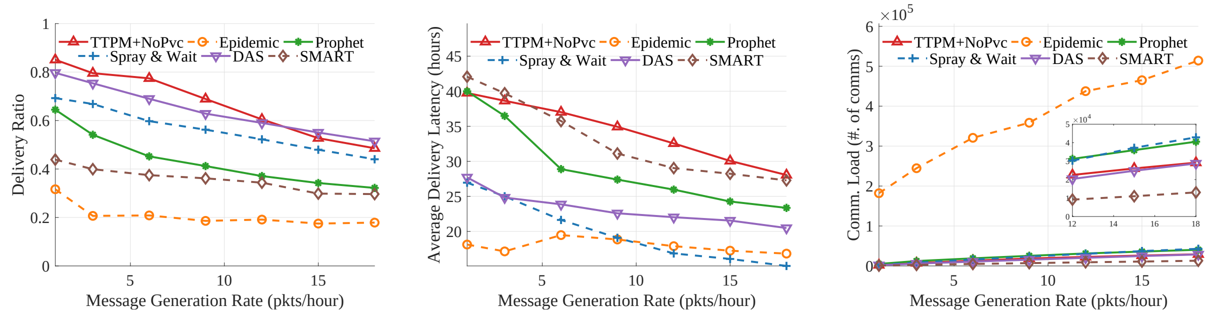
Based on the parameters provided by Python, we conducted the simulations in the ONE simulator based on the BSS dataset. The routing performances in the ONE and the Python simulator are shown in Fig. 1 and Fig. 2 of this response letter, respectively. There are two different points in the implementation details between the ONE and the Python simulator.

1) The message transmission in the ONE simulator is two-way when a node contact occurs. For example, when node i contacts node j, the messages in node i can be forwarded to node j and the messages in node j can also be forwarded to node i. However, in the BSS scenario, when the trip $i \rightarrow j$ occurs, the messages can only be forwarded from node i to node j via the contact $i \rightarrow j$, which is a one-way channel. Thus we added the codes in the ONE simulator to support the one-way transmission in the BSS-OppNet scenario. The corresponding codes are presented in the folder '05-the-one-master' of our repository[2].

2) In the ONE simulator, the whole simulation duration will be divided into a lot of equal update intervals according to the 'World.java' file. For example, the whole simulation duration in this paper is 14 days (July 1-14). The update interval, e.g., 0.1 seconds, can be set according to 'Scenario.updateInterval' in the setting file of the ONE simulator. Therefore, *the message forwarding decision and the message transmission in one update interval are independent to the other update interval.* For example, assume that a node contact $i \rightarrow j$ are available in [10s, 10.5s]

[1]The paper is under review.

[2]https://gitee.com/gaoyang23nj/DiffPrivacyOppNet



(a) Delivery ratio. (b) Average delivery latency. (c) Communication load.

Fig. 1: Routing performance of different routing methods with the Python Simulator.

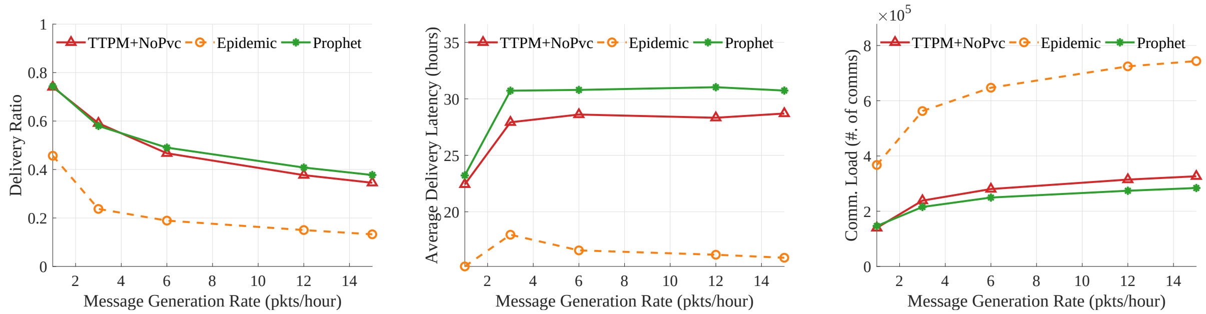
for the message forwarding. And there are 3 messages in node i's buffer can be forwarded to node j according to the routing algorithm. In the first update interval [10.0s, 10.1s], the routing decision about forwarding the message m_1 from i to j is made. Then in [10.1s, 10.2s], forwarding m_2 to j is decided. In [10.2s, 10.3s], forwarding m_3 to j is decided. But if node j exceeds its buffer size after j receives m_3, the messages (including m_1) may be discarded according to the First-In-First-Out buffer mechanism. Note that in the next interval [10.3s, 10.4s], m_1 will be transmitted to node j again. This situation will bring about unnecessary transmissions.

In the Python simulator[3], the messages, which can be forwarded from i to j, are chosen before the transmissions begin. In one contact $i \rightarrow j$, a message can be forwarded *once* at most from i to j. Thus the Python simulator can skip the update intervals, in which neither the message generation event nor the contact event occurs.

In the BSS scenario, the messages are carried by the bike at once, when $i \rightarrow j$ occurs. It means that all the related message decisions can be made before any message transmissions in this contact begin. Therefore, the Python simulator is more suitable for this BSS-OppNet scenario.

According to the above analysis, higher communication load will be brought about by the ONE simulator, which is corresponding to Fig. 2c. The whole tendency of the routing performance on the Python simulator does not deviate much from that on the ONE simulator, when the message generation rate increases from 1 packets per hour to 18 packets per hour.

[3]The routing algorithms are implemented in the '/01-code/Main/Routing' folder of our repository.



(a) Delivery ratio. (b) Average delivery latency. (c) Communication load.

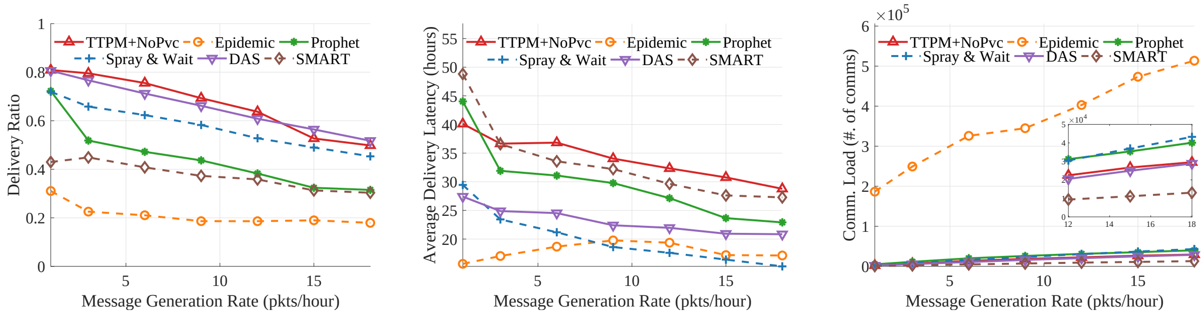
Fig. 2: Routing performance of different routing methods with the ONE Simulator.

II. Qiaobei Dataset and Pukou Dataset

As a main part of a State-Level New Area in China, Qiaobei has been accelerating the informatization process. A great amount of data (i.e., the amount of bike trips) exists in the bicycle sharing system in Qiaobei, Nanjing. We can get the sufficient data conveniently in this area. Moreover, there are the metro stations, the residential areas, the office buildings and the shopping malls in Qiaobei. Thus the bike trips in Qiaobei can represent the pattern of the urban commutes. Based on the collected data of this area, there are about $588,000$ bike trips among 99 stations in 2017. Thus we conduct the simulations with the Qiaobei dataset in the paper.

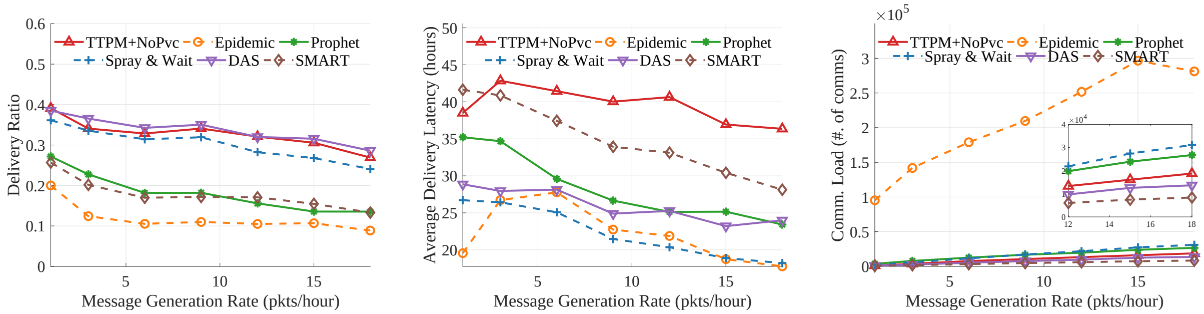
We also collected the data in Pukou, which is a old town of Nanjing in 2017. There are about $432,000$ bike trips among 127 stations in Pukou. Based on the collected data, we can find that the amount of the contacts (i.e., the trips among stations) in the Qiaobei dataset is much denser than the amount of the contacts in the Pukou dataset. Thus the chances of message forwarding in Qiaobei occur more frequently than Pukou.

Then we conducted the experiments based on the Qiaobei dataset and the Pukou dataset. The simulation results on the Qiaobei dataset and the Pukou dataset are shown in Fig. 3 and Fig. 4 of this response letter, respectively. We can find that the delivery ratios of all the routing methods (including DAS, Smart and TTPM) in the Qiaobei dataset are also higher than those in the Pukou dataset. Meanwhile, the relative sparsity of the node contacts also leads to the increase of the average delivery latency. Therefore, the dataset that contains the denser contacts (or the modern urban scenario) is more suitable for the deployment of the OppNet.



(a) Delivery ratio. (b) Average delivery latency. (c) Communication load.

Fig. 3: Routing performance of different routing methods on Qiaobei Dataset (July 1-14).



(a) Delivery ratio. (b) Average delivery latency. (c) Communication load.

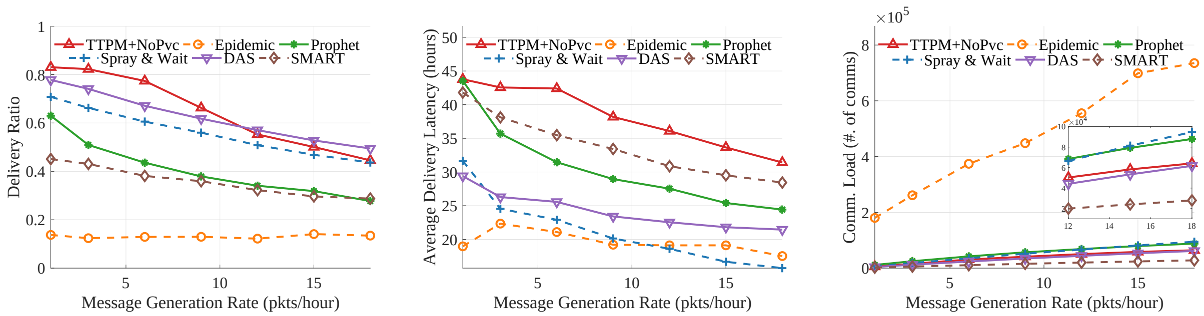
Fig. 4: Routing performance of different routing methods on Pukou Dataset (July 1-14).

III. Simulations with the Data in 31 Days

In the experiment of this paper, we conducted the simulations based on the bike trips from July 1 to 14. The reasons are as follow:

1) Because of the climate effects, lots of people in Nanjing ride bikes in July. So the dataset (or the bicycle records) in July can represent the characteristics of the urban commutes among metro stations, residential areas, shopping malls and so on.

2) More data adopted in the simulations will not change the performance much. To reveal this effect, we conducted the simulations based on the data from July 1 to 31. The corresponding performance results are shown in Fig. 5 of this response letter. We can find that the delivery ratios and the average latencies do not change much if more dataset is utilized in the simulations. The communication load (i.e., the total number of communications) increases because more trips and more messages are generated.



(a) Delivery ratio. (b) Average delivery latency. (c) Communication load.

Fig. 5: Routing performance of different routing methods on the dataset in July 1-31.