JoWUA Review Results: Paper 307 entitled "Network Size Estimation in Opportunistic Mobile Networks: The Mark-Recapture Method"

Ilsun You <ilsunu@gmail.com>

Tue 11/24/2020 10:41 PM

To: Soelistijanto B <b.soelistijanto@usd.ac.id>;udevmanoah@gmail.com <udevmanoah@gmail.com>

Dear Authors,

We regret to inform you that
your paper 307 entitled "Network Size Estimation in Opportunistic Mobile Networks: The Mark-Recapture Method" cannot be published in Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications (JoWUA).

However, during the review process, we realized that after being revised it can be accepted for publication.
Therefore, we suggest you carefully revise the paper according to the review comments at the bottom of this email.

Please send us the response letter and the revised version whose changes should be colored in red
by **December 13, 2020**.

As soon as you receive this email, it would be highly appreciated if you would confirm the corrections.

All the best,

Prof. Dr. Ilsun YOU, FIET
EiC of JoWUA

==

REVIEW 1:

This paper proposes a novel method based on the Mark-Recapture technique to perform active node counting in opportunistic mobile networks. The proposed method achieves lower cost than flooding-based forwarding algorithms, i.e. epidemic routing. The authors have performed experiments with multiple scenarios including both random movement and real human mobility trace datasets and provided performance results in terms of convergence time and ability to capture the true number of nodes. The paper makes an important contribution to the literature on opportunistic networks, and the paper is quite well-written overall. However, some issues need to be addressed to further improve the quality of the paper:

- The authors could consider focusing more on the significance of the problem for different opportunistic network settings. How will the model be utilizable for different conditions involving different types of nodes and different mobility scenarios (e.g. military opportunistic networks etc)?

- The current convergence time of the algorithm could be a bit slow for some scenarios. The authors should comment on the utility of the algorithm for different settings. i.e. under what conditions would it make sense to use this algorithm. Also, it would be nice to include the convergence time performance of existing algorithms from the literature, so the reader can have a reference for comparison. E.g. Epidemic algorithms probably converge faster, but have higher cost. It would be nice to provide benefits/disadvantages of each.

- The paper has some minor grammatical errors that should be fixed with thorough proofreading.
- The term "popular" could mean different things in different settings. The authors should explain the term in more detail.
- The references could be updated to include more recent work in the domain.

==
REVIEW 2:

I think that the contribution is suitable for JoWUA. That is why research methodology is clear, and validation is detailed.
However, it should give a more state of the art and discussion of the novelty of the work over state-of-the-art contributions.
==

Response to the reviewer comments.

We are very pleased that both Reviewer 1 and 2 have recommended that the article can be accepted, and we are very grateful to both the reviewers for their comments; they have allowed us to refine and clarify the paper. Our specific responses to their comments are below.

Reviewer 1:

-"The authors could consider focusing more on the significance of the problem for different opportunistic network settings. How will the model be utilizable for different conditions involving different types of nodes and different mobility scenarios (e.g. military opportunistic networks etc)?"

Response: We have revised the manuscript to include some discussions about what conditions that the proposed algorithm can be utilized in various opportunistic network scenarios, namely:

- In Introduction (on P. 2), we list the most important questions we answer on the paper. One of these is "*how does node mobility impact the performance of the Mark-Recapture counting algorithm?*". We intend to emphasize the effect of node movement on the performance of our algorithm. From this, we can understand what kind of opportunistic network scenarios that are appropriate for our proposed algorithm.
- In Related Work (on P. 4), we give a thorough discussion of node mobility considerations on the existing works (Section 2.2). We show that most of the state-of-the-arts algorithms are developed and evaluated based on the assumption of random mobility.
- In Performance Evaluation (on P. 8-12), we realize that the proposed algorithm works ineffective in real human mobility cases, i.e. Haggle and Reality. Due to a non-random contact pattern, nodes with few contacts with others (we call them less popular nodes) suffer from a recapture issue, since they cannot recapture the (previously) marked nodes with the same probability. In addition, we also see that our benchmark algorithm, the gossip-based pairwise average, also has the same problem in these nodes.
- However, in P. 11 we comment that in some specific cases of human mobility, for instance the work of UrbanCount [9], human mobility for some extent can be considered as random movement. UrbanCount uses a City-Square model, which is actually an improvement of Random-Waypoint. Consequently, we believe that in specific cases of human mobility (e.g. disaster, military, or crowd) our proposed algorithm can possibly be applied.
- Finally, in P.12, we refer to mobile crowd sensing applications that use a client-server paradigm, where mobile devices sensing and reporting data to a central server (in a cloud) that process the data and distribute the result to nodes who need it. We suggest that the Mark-Recapture algorithm can also use this client-server paradigm, in that only a few (popular or most active) nodes perform a counting process and send the counting data to the server that will provide the final result to all the nodes.

- “The current convergence time of the algorithm could be a bit slow for some scenarios. The authors should comment on the utility of the algorithm for different settings. i.e. under what conditions would it make sense to use this algorithm. Also, it would be nice to include the convergence time performance of existing algorithms from the literature, so the reader can have a reference for comparison. E.g. Epidemic algorithms probably converge faster, but have higher cost. It would be nice to provide benefits/disadvantages of each.”

Response: We have included a gossip-based pairwise average as a benchmark to evaluate the performance of the proposed algorithm. We can list the findings as follows:

- In Performance Evaluation (P. 8-9), we can see that Mark-Recapture can outperform the benchmark scheme, in terms of convergence time, in the random scenario (Fig. 3).
- However, in the real human mobility scenario (Haggle and Reality) both the algorithms cannot work effectively, particularly in nodes having a few contacts with others (less popular nodes) (Fig. 5 and 7).
- Nevertheless, as we described above, in some works (e.g. UrbanCount) human movement for some extend can be considered a random process. We believe Mark-Recapture can work appropriately in these specific cases.
- On the other hand, referring to the case of mobile crowd sensing applications that use a client-server paradigm, we therefore suggest that Mark-Recapture can be incorporated with these client-server model, where only a small number of nodes (i.e. popular or most active nodes) perform node counting and report data to the server that finally process and provide the final result to all nodes. Indeed, the use of a client-server architecture can solve the issue of node counting in real human mobility cases (e.g., in our case Haggle and Reality).
- Lastly, our work in this paper as well as most of the existing works assume a closed system, where the number of nodes are constant during the experiment. An open system with churn, where nodes are allowed to enter and leave the area, is challenging and we take this as future work.

-“The paper has some minor grammatical errors that should be fixed with thorough proofreading.”

Response: We have checked and revised the grammatical errors in the paper.

-“The term “popular” could mean different things in different settings. The authors should explain the term in more detail.”

Response: We have clarified what “popular” mean in human-based opportunistic networks. In P. 9 we write: “*Typically, individuals move to places or meet other people to fulfill their social needs, and social (contact) graphs are commonly used to describe their social relationships. The authors of [25] investigated several real human contact datasets and confirmed that human mobility typically possesses a non-random contact pattern, where a few nodes (individuals) have contacts (or relations) with many others, but majority of nodes only have few ones. The nodes having a large number of contacts with others are therefore socially very popular in the networks (these popular nodes also called hub nodes in social network analysis, SNA).”*

In a brief, we assert that “popular” in this social context meaning that popular nodes are very active in the area (or network) and have a large number of contacts with other nodes, therefore these nodes are very popular among others (in SNA, these nodes also called high degree nodes).

-“The references could be updated to include more recent work in the domain.”

Response: We have added more recent works in the area of node counting in the references, such as :

- the work of UrbanCount, 2017 [9]
- Estimation on a large network and its communities using random sample, 2016 [8]
- The capture-recapture approach in computer networks, 2015 [15]
- A survey of distributed data aggregation algorithms, 2015 [17]
- Data collection and node counting by opportunistic communication, 2019 [19]
- Mobile crowdsensing, 2011 [27]
- Smartphone collaboration in data acquisition and distributed computing, 2014 [30]

Reviewer 2:

-“I think that the contribution is suitable for JoWUA. That is why research methodology is clear, and validation is detailed.
However, it should give a more state of the art and discussion of the novelty of the work over state-of-the-art contributions.”

Response: in Related work (P. 3-4), we gave more existing works and discussion to make our motivations and contributions clear. Subsequently, we revised the discussion in Related work, categorizing it into three categories:

- Distributed algorithms for node counting: we divide the existing algorithms into two classes: *aggregation* and *statistical sampling* algorithms. We discuss the advantages and disadvantages of the schemes in each class.
- Node mobility consideration: we discuss the assumptions and settings of node mobility in which the existing algorithms were developed and evaluated.
- Applications of node counting: we describe the current applications of node counting in different scenarios, e.g. P2P networks, MANETs, urban settings, routing in OMNs.

Network Size Estimation in Opportunistic Mobile Networks: The Mark-Recapture Method

Bambang Soelistijanto* and Geraldev Manoah

Department of Informatics, Sanata Dharma University

b.soelistijanto@usd.ac.id, udevmanoah@gmail.com

Abstract

This article addresses the issues of counting the number of nodes in opportunistic mobile networks. The global knowledge of network size is commonly required to design optimal routing algorithms in OMNs. However, due to the inherent characteristic of long transfer delay, node counting in such intermittently-connected networks is a challenging task. In this paper, we propose the *Mark-Recapture* method to estimate the number of nodes in a network. In ecology, the statistical technique has been widely used to predict the population sizes of animals in open areas. The scheme initially samples nodes in the network, and an estimate of the network size is then calculated based on this partial knowledge of the network. Through extensive simulations driven by random movement and realistic mobility models, we show that the proposed method is able to produce a good estimate of network size within a relatively short duration of time. Finally, by tweaking Epidemic routing with the local estimate of network size, we can reduce the delivery cost of this flooding strategy without significantly degrading the overall network delivery performances.

Keywords: network size, node counting, the Mark-Recapture method, opportunistic mobile networks

1. Introduction

Nowadays, opportunistic mobile networks (OMNs) [1] have received much attention by industry and research community. These networks are an extension of mobile ad-hoc networks (MANETs) and are an instance of delay tolerant networks (DTNs). While MANETs require end-to-end paths between sources and destinations to enable message transfer, OMNs are capable of performing communication despite the absence of stable paths between any pair of nodes. In MANETs node movement is considered as a potential disruption, but in OMNs data transfer is performed by opportunistic communication, leading to a higher delay than that of MANETs. Data dissemination in OMNs is thus delay-tolerant in nature. Some realizations of OMNs exist, including emergency scenarios and natural disasters [2], military operations [3], and social-based networks [4]. The widely use of mobile wireless devices, such as smart phones, gadgets, and laptops, is the main factor in the proliferation of these systems.

In OMNs, searching for optimal paths between a pair of nodes is non-trivial task. Since the stable paths between any pair of nodes rarely exist at all the time, conventional routing algorithms proposed for MANETs would fail in this setting. This imposes a new model for routing in OMNs, the *store-carry-forward* paradigm [5]. This suggests that a message is stored and carried by relay nodes, and finally is forwarded when the destination is encountered. In this regard, choosing good relays for message transfers is indeed crucial in OMNs. A bulk of researches in OMNs have focused on developing effective routing protocols. To achieve this goal, the algorithms typically require complete information of the current network states. In practice, however, this global knowledge is commonly unavailable to all the network nodes. To improve the delivery performance, several algorithms opt to increase message redundancy in the network. Naïve approaches (e.g., Epidemic routing [6]) forward a message replica to each contacted node, so that the copies are quickly dispersed over the network. This oblivious forwarding assumes unlimited node resources, but this is hard to achieve in practice. On the other hand, some algorithms (e.g., adaptive Spray-Wait [7]) attempt to reduce the number of message replicas by

* Corresponding author: Department of Informatics, Sanata Dharma University, Yogyakarta, Indonesia, 55282, Tel.: +62-274-883037.

capping the message replication at a maximum value. To this aim, the protocol at each node needs to know the number of nodes in the network. However, estimating this global parameter in a decentralized manner is a non-trivial task in OMNs, due to the highly dynamic topology changes and long transfer delays.

In this work, we focus on the particular case of node counting in OMNs (the global statistic of the total number of nodes in a network is also referred to as *network size*). To date, distributed node counting has attracted interest from researchers, since a local estimate of network size is often very useful for building applications that are adaptive and robust. For example, the population algorithm in [8] uses a random sample to estimate the size of a large network and its communities; a crowd counting system in [9] estimates crowd sizes and densities for city administration and disaster management; a data dissemination protocol in [10] predicts the network size for limiting message redundancy. In the literatures, several distributed computing algorithms have been proposed in the area of global information collection and estimation in opportunistic networks. In addition, majority of them are modifications of data aggregation schemes proposed for well-connected networks (e.g., [11]). Even though Aggregation provides accurate estimates in the conventional networks, but it suffers from a number of difficulties in the context of OMNs as follows [12]: first, the delay time to converge to the actual network size is very long in such delay-tolerant networks; second, node failures will significantly degrade the performance of Aggregation. As an alternative to Aggregation, several distributed estimation algorithms for OMNs (e.g., [10,13,15]) are developed based on statistical sampling techniques.

In this paper, we propose the Mark-Recapture method [14], a statistical technique used to estimate the number of nodes in an OMN. This technique has been widely used in ecology to predict the population sizes of animals or fishes in forests or seas, respectively. In the area of communication networks, the method has been utilized to estimate the network size in peer-to-peer (P2P) networks as well as multicast networks [15]. To the best of our knowledge, however, this paper is the first work that applies Mark-Recapture to perform node counting in OMNs. In addition, most of the existing works in distributed node counting in OMNs only consider a simple random *i.i.d* model when designing and evaluating the algorithms. In fact, such model may not be realistic to describe real human mobility cases [16]. In this paper, we investigate the proposed algorithm under both random movement and realistic mobility scenarios. The underlying node mobility contributes to node mixing, and in turn to the spreading of data. Consequently, the most important questions we answer in this paper are:

- How does node mobility impact the performance of the Mark-Recapture counting algorithm in OMNs?
- Can Mark-Recapture outperform Aggregation in OMNs in terms of estimation accuracy and convergence time?
- Can a local estimate of network size improve the delivery performance of Epidemic routing [6] in OMNs?

The main contributions of this paper are:

- We present a distributed counting algorithm based on Mark-Recapture [14] to estimate the number of nodes in an OMN.
- We evaluate the proposed algorithm via extensive simulations driven by random movement and real-life mobility models.
- We identify the performance improvement of Mark-Recapture compared to Aggregation in terms of estimation accuracy and convergence time.
- Using local estimates of network size, we improve the delivery cost performance of Epidemic routing without significantly degrading the overall network delivery performances.

The remainder of the paper is organized as follows. In Section 2, we introduce the related works and position our work concerning the state-of-the-arts in the area of node counting in OMNs. The problem description and the proposed distributed counting algorithm based on Mark-Recapture are presented in Section 3. In Section 4, we evaluate the estimation accuracy and convergence time of the scheme in OMNs through simulations under random movement and realistic mobility scenarios. Subsequently, we compare the performance of Mark-Recapture with that of Aggregation in terms of estimation accuracy and convergence time. Finally, we investigate the delivery performance improvement of Epidemic routing with local estimates of network size. We conclude the paper and present directions for future work in the last section.

2. Related Work

In this section, we review some state-of-the-art node counting algorithms for OMNs to indicate our motivations and contributions. We discuss the existing works that are related to our work in the following three categories.

2.1. Distributed algorithms for node counting

We can broadly distinguish two classes of methods for node counting in OMNs. Techniques of the first type are based on *data aggregation algorithms*, while those of the second type are based on *statistical sampling algorithms*.

A. Aggregation algorithms

To date, Aggregation has played an important role in modern distributed systems [17]. It can perform the evaluation of global properties of the systems in a decentralized way. Moreover, network size is a typical system-wide property required by algorithms in many contexts. Jelasity *et al.* [11] proposed a distributed gossip-based aggregation algorithm for large dynamic networks. In this algorithm, each node periodically chooses one node among the neighbours, and afterwards the pair of nodes exchange and update their local estimates to assure quick convergence to the desired aggregate value. Since the scheme was developed under the assumption of stable links, it will not work properly in the context of opportunistic communication. In OMNs, links between nodes are created by sporadic contacts, occurring when they come in direct radio range. Consequently, the list neighbours is often not known in advance, and there is no neighbour sampling before links are established between the node and its neighbours.

Guerrieri *et al.* [12] introduce a set of node counting strategies based on Aggregation for OMNs, namely pairwise average and population protocols. The former is a class of gossip protocols. At the beginning, one node (called an initiator) stores a value equals to “1” and all the remaining nodes stores “0”. At every contact, the nodes exchange their current values and update the stored value as the average of its value and the peer’s. Eventually, the algorithm converges to $1/N$ and the number of the network nodes is achieved as the inverse of the estimate. In contrast, the population protocols use tokens to calculate network size. At the initial run, each node is allocated a single token. At each contact, two nodes toss a fair coin and the one winning the ballot collects the whole peer’s tokens. At the end, tokens gather on a node that has the accurate estimation of the network’s size. However, randomly choosing nodes to collect tokens during node contacts may result in suboptimal performance: it leads to long convergence time and low estimation accuracy. To cope with these issues, Ning *et al.* [18] therefore propose a new technique that incorporates effective contact probability into counting process. On the other hand, the works in [9,19] apply a different strategy based on Aggregation of node states. When two nodes come into contact, they exchange the state sets and each node then establishes a union set containing the elements of both its own set and the peer’s. In the end, all nodes converge to have a set including the ids of all nodes in the network, and the network size is determined by the cardinality of the set.

However, all the abovementioned Aggregation schemes suffer from common problems in OMNs, namely long convergence time and estimation accuracy sensitive to node failures. To deal with these issues, we propose a node counting algorithm based on a statistical sampling technique, i.e., Mark-Recapture [14]. In the following, we discuss the existing works that are developed based on statistical sampling techniques.

B. Statistical sampling algorithms

Statistical sampling methods produce a prediction of the system’s global properties based on the statistics attained from uniformly random samples. Sample-Collide [13] is proposed to calculate peer counting in overlay networks. The work in [10] applies a sampling technique based on Taxi-Problem (also known as Racing-Car Problem) to predict the number of active nodes in an OMN. In general, Taxi-Problem works as follows: one (also called initiator) wishes to estimate the number of taxis currently operating on the streets of a city. The taxis are numbered consecutively from one to some unknown number *N*. The initiator observes and records the ids (=serial numbers) of all taxis that have passed in a given time interval. In addition, this scheme assumes that each taxi is equally likely to pass the initiator at any given time. Using the sampling data, an unbiased minimum variance estimator (UMVE) is finally computed as the best estimate of the total taxis in the given city. As shown in [10], the counting algorithm based on Taxi-Problem can work properly in OMNs to give a good estimate of network size. Despite the elegance of this technique, however, the effectiveness of Taxi-Problem in OMNs strictly depends on two conditions as follows: first, all nodes are consecutively numbered from 1 to *N*; second, the probability of encounter between any pair of nodes is uniformly distributed in the network. As opposed to Taxi-Problem, our proposed algorithm is relaxed from these constraints: it does not require the nodes either to be successively numbered or to have a homogeneous contact pattern.

2.2. Node mobility considerations

In the literatures, majority of the node counting algorithms proposed for OMNs are developed under the assumption of a simple random *i.i.d* model. In the class of Aggregation, for example, the gossip-based pair-wise average method [12] suggests that in each pair-wise contact nodes exchange their current values and store the new value as the average of their present values. Given that all nodes have an equal opportunity to meet any other node in the network, the algorithms of all the nodes eventually converge to a single values of actual network size. Moreover, the work of crowd counting in [9] proposes a fully decentralized Aggregation to calculate an accurate estimate of the crowd size. During a node contact, two nodes exchange their state sets containing the identities of the nodes already seen before. By assuming that all nodes (individuals) in the crowd follow a random walk (RW) mobility, they finally converge to have a set that includes the ids of all nodes in the crowd and the crowd size is then determined by the cardinality of the set.

As similar to Aggregation, most of the existing works of node counting based on statistical sampling methods also rely on the assumption of random mobility. For instance, the work based on Taxi-Problem [10] strictly requires that the node contact pattern should be homogenous, so that the probability of any node encountering the initiator will be equal. In fact, however, real-life mobility deviates from the assumption of random *i.i.d.* mobility [16]. In [19], Li *et al.* study the effect of node mobility on data collection and node counting in OMNs. However, their investigation is still based on a homogeneous mobility pattern, where each node randomly selects the destination and speed: the destination follows a uniform distribution, but the speed follows a Gaussian distribution with the mean is constant, but the standard deviation varies during the experiment. Our proposed algorithm, however, is investigated under both random movement and realistic mobility scenarios. For the latter case, we use real human mobility models, which intrinsically possess a heterogeneous contact pattern [20], where a few nodes (called hub nodes) have many contacts with others, but majority of nodes only have few ones.

2.3. Applications of node counting

With the more powerful mobile wireless devices nowadays, it is not required to offload the processing to an edge server or a cloud computing service. In mobile computing, a computational task is executed independently in each node (mobile device), and by using communication all nodes share their individual outcomes and ultimately arrive at a convergence result. One of the typical tasks in distributed computing is calculating network size (i.e., the number of nodes in the network). This information can then be used as input by other applications or protocols. Some applications of node counting are as follows: network size is used for building and maintaining the distributed hash table in P2P networks [15]; in [21] the statistic is exploited in wireless mobile ad hoc networks (MANETs) to set up a quorum of a membership service; UrbanCount [9] applies a fully distributed crowd counting protocol to estimate crowd size during open-air events or rush hours for city administration; in [10] the knowledge of network size is required to optimize the performance of a routing algorithm in OMNs by minimizing the delivery cost. In this paper, we use local estimates of network size to improve the delivery cost performance of a flooding-based algorithm, i.e., Epidemic routing [6], by capping the message replicas to be a half of the network size.

3. The Mark-Recapture Distributed Estimation Scheme

In this section, we propose a novel strategy of distributed estimation based on the Mark-Recapture technique to predict the number of nodes in an OMN. We initially introduce the basic scheme of Mark-Recapture widely used in ecology. We then discuss the system model and problem description and finally propose the Mark-Recapture distributed estimation algorithm for OMNs.

3.1. The Basic Mark-Recapture Method

Wildlife managers commonly use the Mark-Recapture technique [14] (also called the Lincoln-Petersen method) to estimate the population size of animals or fishes in forests or seas before hunting or fishing seasons, respectively. The scheme comprises a single marking episode (also called a capture episode) and a single recapture episode. It initially starts with taking a sample of individuals in a natural population, marking and then sending them back to the original population and finally recapturing some of them as a basis for predicting the population size at the time of initial marking. The basic principle of the algorithm is that if a sample of the population is marked in some way, returned them to the original population, and after fully dispersed in the population a second sample (also called a recapture sample) is taken, the ratio of total marked individuals (*m*) to sample size (*C*) in the recapture sample will be equal to the ratio of total marked individuals in the initial sample (*M*) to the population size (*N*). That is,

$$\frac{m(total\_marked\_in\_recapture)}{C(recapture\_size)} = \frac{M(total\_marked\_initially)}{N(population\_size)} \qquad (1)$$

By rearrangement (1), we can calculate the estimate of the population's size at the time of initial marking, as

$$\hat{N} = \frac{MC}{m} \qquad (2)$$

However, the accuracy of Mark-Recapture relies on several assumptions as follows:

- The population size should be constant during the period between the initial marking episode and the recapture episode.
- The probability of all individuals being captured should be the same during both the episodes.
- There must be sufficient time between the capture and recapture periods to allow all the marked individuals to be randomly mixed all over the population.
- The marked individuals should not lose their marks between the two periods.

3.2. System Model and Problem Description

We consider an opportunistic mobile network, where the nodes move independently in a given area and communicate to the peers wirelessly. Communication occurs when nodes come into contact within their radio ranges. Our study is based on several assumptions as follows:

- There are *N* mobile nodes in the network.
- Nodes participate equally in the counting process.
- Nodes do not provide fake information to others.
- Nodes do not stop operations or abruptly leave the network all the time.
- Any node can initiate a counting process whenever it needs to know the network size.

initiator-id	total marks	seq. number	TTL

Fig. 1. The marking message structure

The purpose of this study is to make a prediction on the number of nodes in an OMN with high accuracy and a low delay. This particularly becomes a complex task in OMNs, since the node contacts are unpredictable and are limited in terms of time and bandwidth. Furthermore, this paper considers node counting in a closed system. In this setting, the number of nodes is fixed but unknown and needs to be predicted. Different scenarios may allow nodes to enter and leave the area (called an open system with node churn). However, as previously demonstrated in [9], node counting in an open system is more challenging, and providing an accurate count is not trivial. Therefore, we restrict the discussion in this paper to the case of closed systems, and all kinds of

Algorithm 1. The Mark-Recapture Distributed Algorithm

```
/*initial marking phase: */
initiator-id ← initiator's serial number
total-marks ← M
seq.number ← 0
TTL ← ttl
round ← 0

//initiator starts a new counting round by creating a new marking message s
if initiator then
        createMarkingMessage(s.initiator-id, s.total-marks, s.seq.number++, s.TTL);
        round++;
end if

//marking the encountered nodes with binary-marking until total-marks=1
if contacted_node.marked = false and s.total-marks>1 then
        sendMarkingMessage(s.total-marks=⌊M/2⌋);
        updateMarkingMessage(s.total-marks=⌈M/2⌉);
end if

/*recapture phase: */
recapture ← {}
marked-nodes ← 0
estimates ← {}

//when contact occurs with node B
if initiator and s.TTL>0 then
        if !(recapture.contains(B)) then recapture.add(B);
                if B.marked = true then marked-nodes++;
                end if
        end if
end if

/*the counting round terminates */
if initiator and s.TTL=0 then
        estimate=calculateEstimate();     // using (2)
        estimates(round).add(estimate);
end if

//calculate final estimate
final_estimate=avg(estimates())
```

opportunistic network applications that meet the requirements of Mark-Recapture mentioned above can use our proposed algorithm for estimating network size.

3.3. The Proposed Algorithm

We now discuss the proposed distributed estimation algorithm based on Mark-Recapture for OMNs. We divide the algorithm into two phases: *initial marking* and *recapture*. For node marking, we firstly define a marking message (Fig. 1) as a small (control) message containing a number of variables: *initiator-id*, *total-marks*, *sequence-number*, and *TTL*. Initiator-id represents the identity of a node that initiates counting; total-marks is the maximum number of nodes that can be marked during the marking session; sequence-number is the unique identity of a marking message, incrementing by one for each new counting initiation; finally, TTL is the time-to-life of a marking message which directly represents the duration of a single counting round.

The marking episode starts when a node (called an *initiator*) initiates a counting process by creating a new marking message. When the initiator encounters another node, it marks the contacted node by sending a copy of the message to the peer. Moreover, we assume that the marked nodes do not drop the marking message before the message TTL expires. For the marking process, we have two possible strategies: first, only the initiator itself can mark the encountered nodes; indeed, this strategy is simple but takes a long time to completely perform node marking in an OMN. To speed up the process, the second strategy, we call it *binary-marking*, allows the already marked nodes to help the initiator to perform node marking: the initiator initially starts with a marking message with total-marks is set to M marks; when any node A (either the initiator or the marked node) that has total-marks $m > 1$ encounters another node B that has not yet been marked, A then forwards the copy of the marking message to B with total-marks $\lfloor m/2 \rfloor$ and keeps $\lceil m/2 \rceil$ for itself; if the total-marks is left with only one mark, the node terminates marking other contacted nodes; particularly, when this case happens in the initiator, the algorithm subsequently switches the marking phase to a recapture phase (in this algorithm, we assume that only the initiator itself is able to perform the recapture process).

Before commencing a recapture episode, the initiator must wait for some time to allow all the marked nodes to be randomly dispersed over the network. During the recapture period, at each contact the initiator records the id of the encountered node and then categorizes it into a marked or unmarked node: if the contacted node has the marking message with the id-initiator matches with the id of the initiator, the initiator then increments the marked-node counter. When the message TTL expires, the recapture episode finishes, and in turn the current counting round completely ends. In future, the initiator can launch another counting round by initially creating a new marking message with a unique sequence number (that is, the algorithm increments the sequence number by one for each new marking message creation). At the end of each counting round, the initiator computes the total number of nodes in the network using (2). The algorithm eventually returns the final estimate of network size as an average of the estimates obtained from the all previous counting rounds. We depict the pseudo-code of the Mark-Recapture distributed algorithm in Alg. 1.

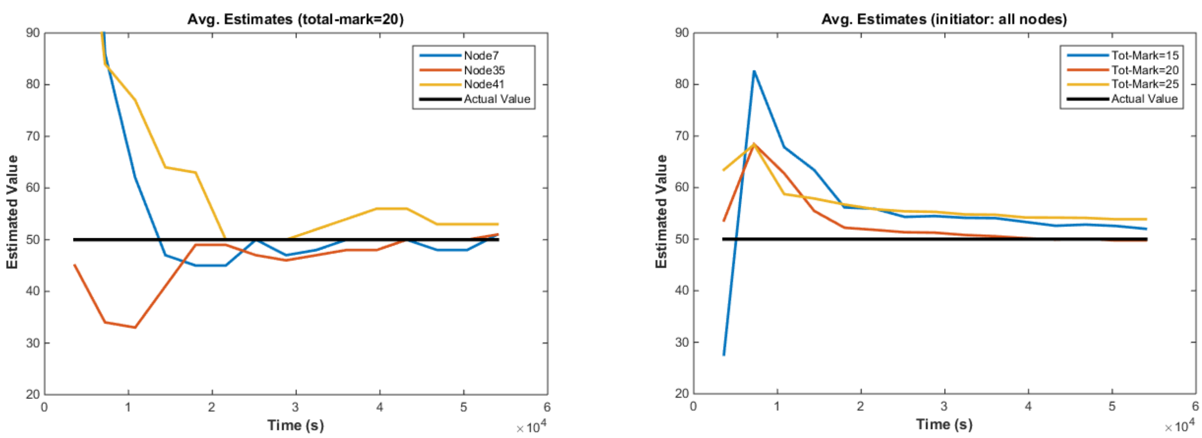


Fig. 2. Node counting in the random mobility scenario (N=50)

4. Performance Evaluation of the Proposed Algorithm

In this section, we evaluate the performance of the Mark-Recapture distributed algorithm in OMNs. Initially, we conduct extensive simulations to investigate the estimation accuracy and convergence time of the proposed algorithm. Subsequently, we examine the performance improvement of Epidemic routing with local estimates of network size. In this study, we use the ONE simulator [22], a discrete-event simulator for delay-tolerant networks. For simulation's mobility scenarios, we consider both random movement and real human mobility models. For the former case, we use the Random-Walk (RW) model packaged along with the ONE simulator. For the latter one, we consider two real human contact data traces, namely Haggle [23] and Reality [24], which represent the short-term and long-term human mobility traces, respectively. The Haggle trace captured the activities of 41 participants during the 2005 Infocomm conference lasted for 3 days in Miami, USA. However, Reality logged the activities of 97 students and staffs at the MIT campus during one academic year. The study was actually performed around 10 months.

4.1. The Estimation Accuracy and Convergence Time of the Mark-Recapture Distributed Algorithm

In this section, we discuss the accuracy and convergence time of Mark-Recapture in estimating the network size of an OMN. We initially consider the random movement scenario. In Fig. 2, we show the simulation results of Mark-Recapture that estimates the number of nodes in an OMN in the random case. In this setting, the total nodes in the network (N) is 50 nodes, the node mobility speed (v) is 1.5-2.5 m/s, and the simulation area is 5000 X 5000 m^2. We randomly choose nodes in the network as initiators (e.g., node ids 7, 35 and 41) and subsequently depict the counting results of these nodes with respect to simulation time in Fig. 2 (left) for total-marks=20. We see that the average estimates of all the given nodes eventually converge to the actual network size (N=50 nodes) at nearly the same time (≈ 18,000 sec or 5 hours). In Random-Walk, the probability of node contact is identically, independently distributed (*i.i.d*) in the network. All nodes therefore have the same probability of being captured in both the initial marking and recapture periods. As a result, as shown in Fig. 2 (left) the counting algorithms of all the nodes show similar counting performances, in terms of accuracy and convergence time. Afterwards, in Fig. 2 (right) we describe the effect of total-mark values on the algorithm's performance when all the network nodes simultaneously perform node counting. We notice that for network with N=50, total-marks=20 gives the best performance among the others in terms of both accuracy and convergence time. Nevertheless, the performance differences among the given total-marks are insignificant in this random scenario, and all of them are eventually able to converge to the actual network size at a slightly different time. Finally, in Fig. 3 we compare the performance of Mark-Recapture with that of an Aggregation scheme, i.e. the Pair-Wise Average method [12] (hereafter, we call it *PW-Avg* for short), in the random scenario (the brief discussion of how PW-Avg works is given in Section 2.1.A). We again randomly select nodes in the network as initiators and then run

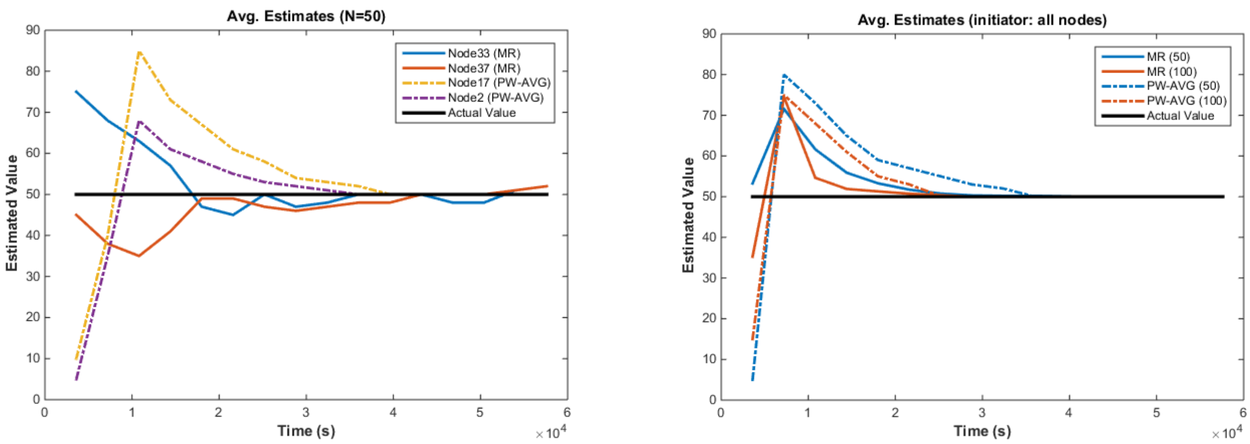


Fig. 3. Mark-Recapture (MR) vs. Pair-Wise Average (PW-AVG) in the random mobility scenario

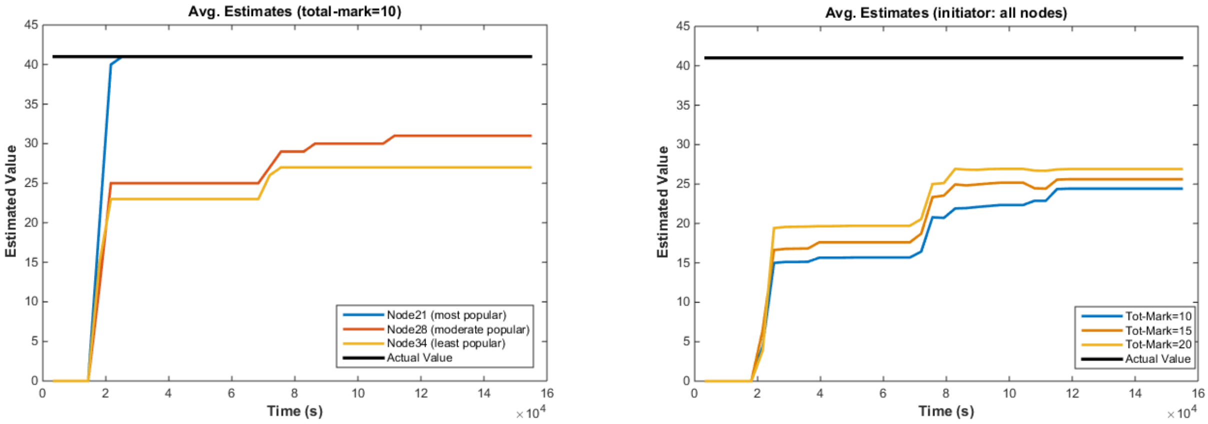


Fig. 4. Node counting in the Haggle mobility scenario (N=41)

the simulations by successively applying both the algorithms on the nodes. In Fig. 3 (left), we depict the counting performance of Mark-Recapture (total-marks=20) compared with that of PW-Avg in the random scenario for N=50. Even though the estimates of both the algorithms in the given nodes can eventually reach the actual network size, the estimates of Mark-Recapture nodes converge at a shorter delay time. The use of sampling strategy in Mark-Recapture effectively produces a good estimate of network size within a relatively short duration of time, while PW-Avg requires more time to enable the initiator to meet more nodes before having a proper result. Subsequently, in Fig. 3 (right) we show the effect of node density (e.g., N=50 and 100) on both the algorithms' performances when all the network nodes perform node counting simultaneously. It is obvious that the increase of node density can reduce the convergence times of both the algorithms. Moreover, within the same node density, Mark-Recapture again outperforms PW-Avg in terms of convergence time.

We now discuss the performance evaluation of the Mark-Recapture algorithm in real human mobility scenarios. Typically, individuals move to places or meet other people to fulfill their social needs, and social (contact) graphs are commonly used to describe their social relationships. The authors of [25] investigated several real human contact datasets and confirmed that human mobility typically possesses a non-random contact pattern, where a few nodes (individuals) have contacts (or relations) with many others, but majority of nodes only have few ones. The nodes having a large number of contacts with others are therefore socially very popular in the networks (these popular nodes also called hub nodes in social network analysis, SNA). Firstly, we consider the short-term contact traces, the Haggle dataset [23]. In this scenario, we deliberately choose 3 nodes as counting initiators, namely node ids 21, 28, and 34, which represent the most-popular node, moderate-popular node, and the least-popular node, respectively, in Haggle. We then depict the Mark-Recapture performances on these nodes in Fig. 4 (left) for total-marks=10. We notice that node 21 (the most popular node) can accurately estimate the network size in a relatively short time. In contrast, the less popular nodes (node 28 and 34) fail to predict the network size (i.e., the estimates of these nodes never converge to the real network size throughout the simulation time). As opposed to random movement, human mobility possesses a heterogeneous contact pattern. Consequently, the most popular node can perform the recapture process properly (i.e., it can meet the marked nodes with same probability), leading to accurately estimate the network size. In the less popular nodes, however, when the marking process can be assisted by other nodes (since our scheme uses the binary marking scheme), the nodes cannot (re)capture the marked nodes with the same probability; in turn, this results in inaccurate estimates of network size. Furthermore, in Fig. 4 (right) we depict the average estimates of network size for several total-marks when all the network nodes simultaneously initiate counting processes. We notice that the average estimates of all the network nodes are significantly below the actual network size. Due to the inherent characteristic of non-random contact, only a few popular nodes (as initiators) can produce a good estimate of network size, while most of the network nodes (i.e., less popular nodes) fail to do this. Consequently, as shown in Fig. 4 (right), the whole counting processes initiated by all the Haggle nodes result in the average

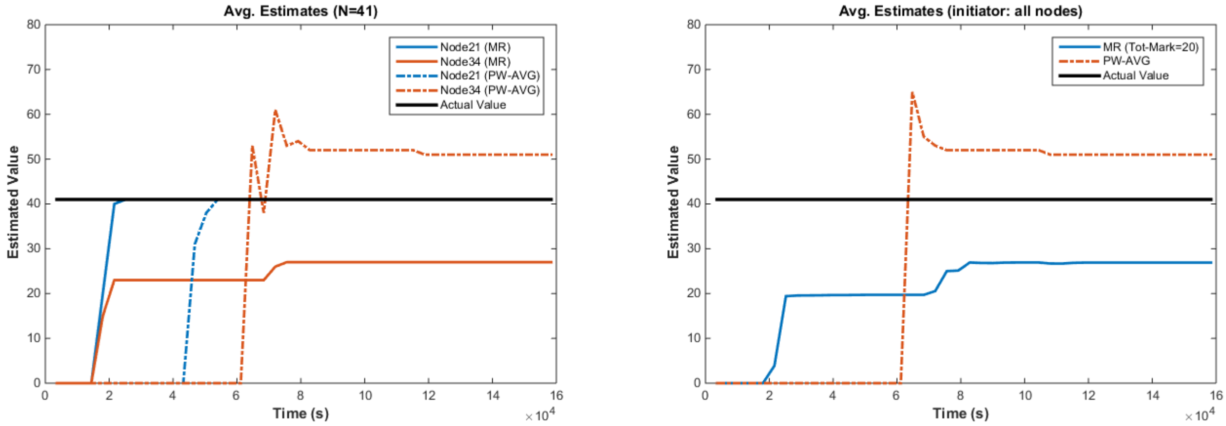


Fig. 5. Mark-Recapture (MR) vs. Pair-Wise Average (PW-AVG) in the Haggle mobility scenario (*N*=41)

estimates below the actual network size. Lastly, we compare the performance of Mark-Recapture with that of PW-Avg in the Haggle scenario. We again choose two nodes in Haggle as before, namely node 21 and 34, that represent the most popular node and the least popular node, respectively, and then apply both the algorithms on the nodes successively. In Fig. 5 (left), we show the counting performance of Mark-Recapture (total-marks=20) compared with that of PW-Avg in the two nodes in Haggle (*N*=41). It is obvious that both the algorithms in the most popular node (node 21) can work properly (i.e. the estimates of the node eventually converge to the actual network size). Moreover, Mark-Recapture can converge in a shorter time compared to PW-Avg in this popular node. In contrast, both the algorithms in the least popular node fail to produce a good estimate of network size. As described above, in Mark-Recapture the least popular node suffers from a recapture issue, as it cannot recapture the marked nodes with the same probability. Similarly, in PW-Avg the least popular node has only few contacts with others, therefore it cannot update the counting value properly, resulting in an incorrect estimate of network size. Furthermore, in Fig. 5 (right) we describe the counting performance of Mark-Recapture (total-marks=20) compared with that of PW-Avg when all the network nodes initiate counting processes simultaneously. Since majority of nodes in Haggle are less popular nodes (due to the inherent characteristic of heterogeneous contact in real human mobility), we then see in Fig. 5 (right) that both Mark-Recapture and PW-Avg are unsuccessful to predict the network size. Indeed, only a few nodes (i.e. popular nodes) in Haggle are able to effectively estimate the network size.

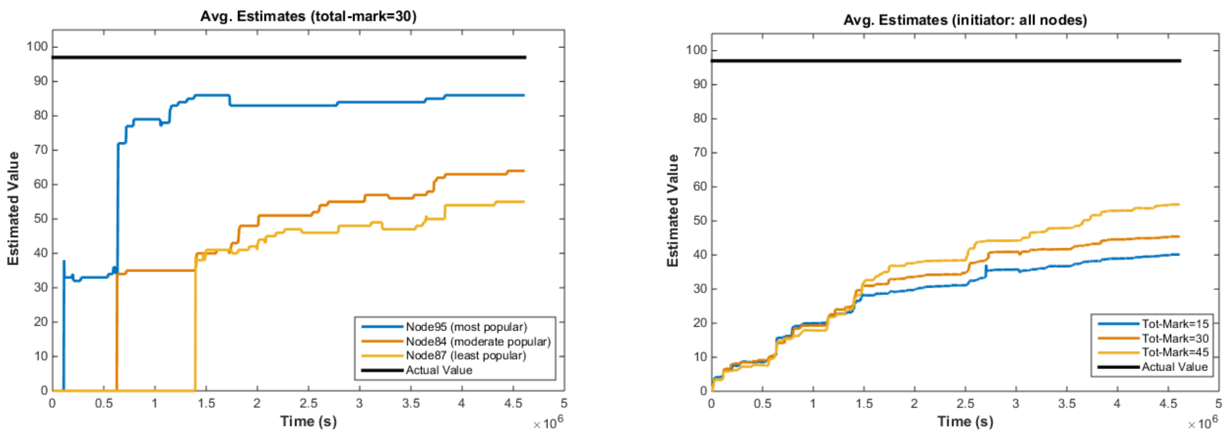
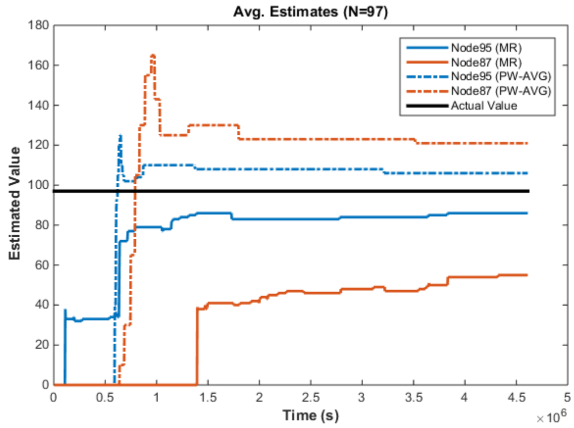


Fig. 6. Node counting in the Reality mobility scenario (*N*=97)



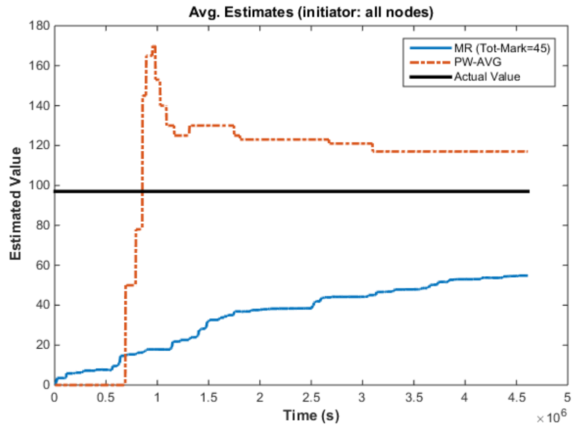


Fig. 7. Mark-Recapture (MR) vs. Pair-Wise Average (PW-AVG) in the Reality mobility scenario (N=97)

The final mobility scenario we consider is the Reality dataset [24], which is captured the long-term human mobility traces. We purposely select 3 nodes as initiators, namely node ids 95, 84, and 87, representing the most-popular node, moderate-popular node, and the least-popular node, respectively, and next apply Mark-Recapture on these nodes. In Fig. 6 (left), we illustrate the counting performances of Mark-Recapture in these nodes for total-marks=30. As similar to Haggle, we again see that the counting algorithm in the most popular node (node 95) outperforms those of the less popular nodes (node 84 and 87). The estimate of the most popular node is able to nearly approach the actual network size in a relatively short time. In contrast, the algorithms in the less popular nodes are ineffective to estimate the network size (i.e. the average estimates of these nodes never converge to the actual network size throughout the simulation time). As in Haggle, less popular nodes in Reality suffer from the recapture issue, since they cannot (re)capture the marked nodes with the same probability. Furthermore, in Fig. 6 (right) we depict the average estimates of network size when all nodes in Reality simultaneously initiate counting processes for several total-mark values. Since majority of the Reality nodes are less popular nodes (due to the heterogeneous contact pattern) and the less popular nodes are typically not able to perform node counting properly (due to the recapture issue), as shown in Fig. 6 (right) the average estimates produced by all the Reality nodes therefore are far below the actual network size for all the given total-mark values. Finally, we compare the performance of Mark-Recapture with that of PW-Avg in Reality. We again choose node 95 and 87 representing the most popular and the least popular nodes, respectively, and then apply both the algorithms on these nodes consecutively. In Fig. 7 (left), we illustrate the counting performances of Mark-Recapture (total-marks=45) and PW-Avg on both the nodes. It is clear that both the algorithms can provide a good estimate when they are applied on the most popular node (node 95). In the least popular node, however, both the algorithms fail to accurately predict the network size and their estimates never converge to the real network size during all the simulation time. Furthermore, in Fig. 7 (right) we show the counting performance of Mark-Recapture (total-marks=45) compared with that of PW-Average when all the network nodes initiate counting processes concurrently. As similar to Haggle, we again see that majority of the Reality nodes (i.e., less popular nodes) fail to produce an accurate estimate of network size in both the algorithms, resulting in the average estimates of all the nodes are far from the actual network size. Actually, only a small number of (popular) nodes can contribute a correct result in the average estimates in Fig.7 (right).

To sum up, the counting performance of the Mark-Recapture distributed algorithm in OMNs is optimal, in terms of accuracy and convergence time, when all nodes move in a random manner in the area. In this case, all nodes are able to perform node counting properly, leading to accurately estimate the network size. Furthermore, Mark-Recapture can outperform an Aggregation scheme, i.e. PW-Avg, in terms of convergence time in the random scenario. However, both the algorithms suffer from a common problem in the real human mobility case, where only a few (popular) nodes are able to carry out node counting appropriately, while majority of nodes work ineffectively. Meanwhile, in some specific cases of real human mobility, such as in disaster, military and crowd scenarios, human movement is typically modelled as a random process. For instance, the work of UrbanCount [9] used a City-Square model [26] to describe the movement of people in a city square. This movement model is actually an improvement of Random-Waypoint. Based on this study, we therefore believe that Mark-Recapture still can be utilized in these specific cases of human mobility. On the other hand, mobile crowd sensing applications [27] use a client-server paradigm, where mobile devices sensing and sending data to a server (in the cloud) that further processes the data and distributes the result to users who need the result. Using this client-server architecture, we suggest that Mark-Recapture can be utilized for counting the number of nodes in human-based opportunistic networks, where only a small number of (popular or most active) nodes sampling and reporting data to a central server, and the server eventually provide the final result to nodes requesting the information.

4.2. The Performance Improvement of Epidemic Routing with Local Estimate of Network Size

In this section, we discuss the application of node counting in data dissemination in OMNs. We exploit a local estimate of network size obtained from the Mark-Recapture algorithm to improve the delivery cost performance of Epidemic routing [6]. In Epidemic routing, a node forwards message copies to all the neighbours within the radio range so that the copies are quickly disseminated all over the network. This oblivious forwarding achieves near-optimal in terms of delivery latency when the node resources are assumed to be unlimited. In practice, however, Epidemic routing tends to quickly deplete the node resources, such as power and resources, and eventually greatly reduces the network delivery performance. We therefore improve Epidemic routing by tweaking it based on the observation in [28] as follows: *only the estimate of the number of nodes in the network* (N) *is required to tune the number of copies* (L), *and Epidemic routing with* $L = N/2$ *can achieve an optimal delivery delay with minimum resource overhead*. In order to incorporate the estimate of network size discussed so far in Epidemic routing, we associate a variable with each message, namely *total-copies (L)* denoting the total number of message copies that can be forwarded by the source node and other nodes receiving a copy to L distinct relay nodes. When L copies have been spread, Epidemic routing stops to forward and lets each relay carrying a copy to perform direct transmission to the destination. Furthermore, in this experiment we set L to be a half of the local estimate of network size ($\widehat{N}$). We eventually compare Epidemic routing with a local estimate of network size (hereafter, we call it *Epidemic-LE*) to conventional Epidemic routing (hereafter, we just call it *Epidemic*) for three performance evaluations, namely delivery latency, overhead ratio, and total message dropped. We do not show the delivery ratio results since Epidemic-LE is able to achieve the delivery ratio as high as that of Epidemic in all scenarios.

We firstly discuss the performance of Epidemic-LE compared with that of Epidemic in the random scenario. In Fig. 8 we show the performance comparison of these routing schemes in terms of the given evaluation measures for different network sizes. In this random case, all nodes independently initiate a counting process to attain a local estimate of network size ($\widehat{N}$), and subsequently create a new message with total copies $L = \widehat{N}/2$ and it is sent to a randomly chosen destination. In the simulation, we set the message generation interval to be 5-10 minutes with the simulation time is 12 hours. For other simulation settings, we use the same settings used in the earlier experiment for the random case. As shown in Fig. 8, Epidemic-LE outperforms Epidemic in terms of overhead ratio and total message dropped for all the given network sizes. With the delivery ratio performance

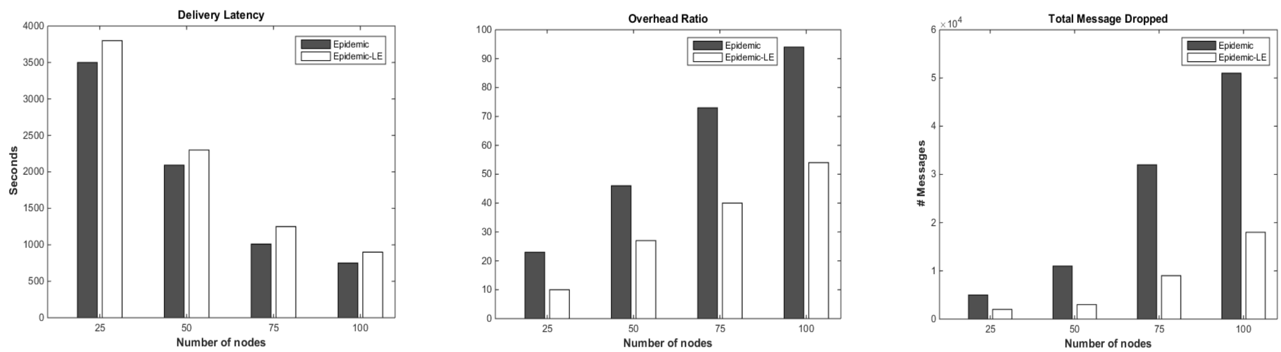


Fig. 8. Performance comparison of Epidemic and Epidemic-LE in the random scenario

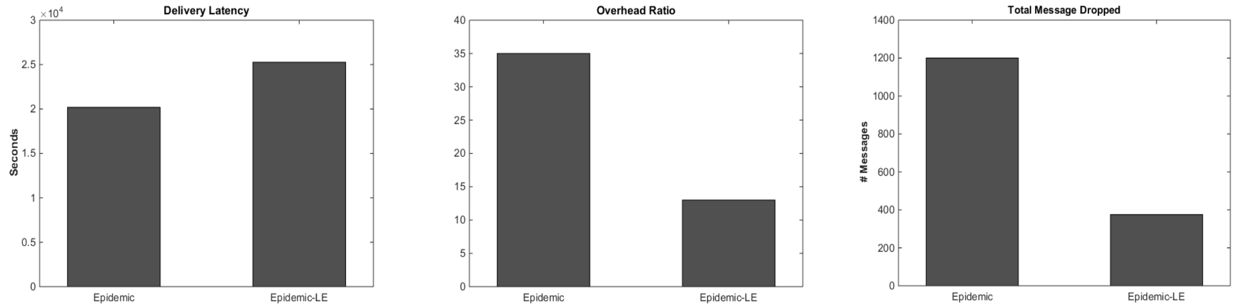


Fig. 9. Performance comparison of Epidemic and Epidemic-LE in the Haggle scenario

is almost the same between the two routing schemes, Epidemic-LE is able to reduce the copy redundancy in the network (indicated by the lower overhead ratio), leading to efficiently use the node resources, e.g. buffer or storage (showed by the significant decrease of total message dropped of Epidemic-LE compared to Epidemic's). Nevertheless, this reduced resource overhead of Epidemic-LE comes at a price, as the delivery latency slightly increases beyond that of Epidemic for all the given total number nodes in the network.

We next discuss the performance improvement of Epidemic-LE in the real human mobility scenarios, namely Haggle and Reality. In contrast to the random case, in these real-life cases only popular nodes can initiate a counting process (as we have described previously, the counting algorithms of less popular nodes fail to produce a good estimate of network size). In consequence, we assume that less popular nodes have ways to learn about the network size from the popular nodes (for example, using a simple flooding data dissemination algorithm or using a client-server architecture as in [27]). Subsequently, all the network nodes randomly create a new message with total copies L is set to a half of local estimate of network size, and the message is then sent to a randomly chosen destination. In Fig. 9 and 10, we show the simulation results of Epidemic and Epidemic-LE in Haggle and Reality, respectively, for the given performance metrics. In the simulations, we set the message generation interval to be 5-10 minutes with the simulation time is 3 days for Haggle, and the message generation interval to be 20-30 minutes with the simulation time is 3 months for Reality. From both the figures, we notice that by capping the total message copies distributed in the network at maximum $L = \widehat{N}/2$ replicas, Epidemic-LE can significantly reduce both the overhead ratio and total message dropped below those of Epidemic, while keeping the delivery ratio as high as Epidemic in both Haggle and Reality. However, as in the random case, we again see a trade-off between resource efficiency and delivery latency performance: the efficient use of node resources of Epidemic-LE increases the delivery delay beyond that of Epidemic in both the real human mobility scenarios. In addition, the increase of delivery delay is more obvious in Reality. Given that OMNs are a class of delay-tolerant networks (DTNs), this increase in delivery latency is not regarded substantial; instead, the reduction of node resource consumption, reflected in the improved overhead ratio and total message dropped, represents a significant improvement in the network's performance.

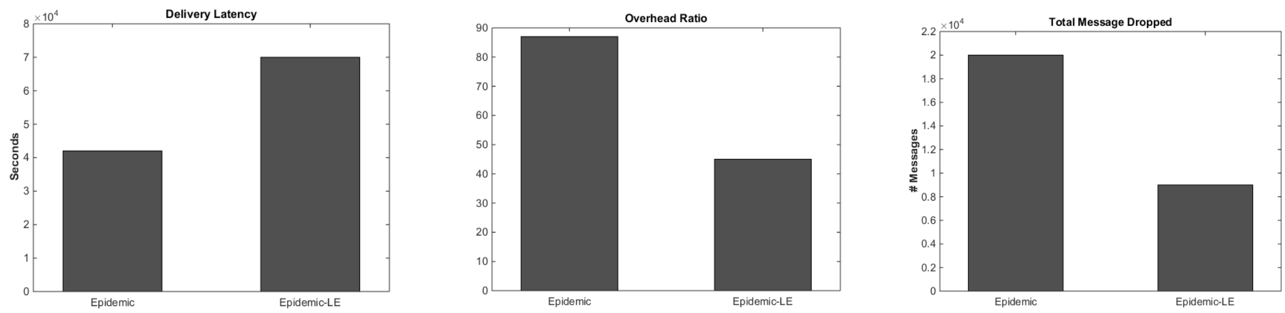


Fig. 10. Performance comparison of Epidemic and Epidemic-LE in the Reality scenario

5. Conclusions

We have presented the Mark-Recapture distributed algorithm, a novel node counting technique targeted at accurately estimate the total number of active nodes in an OMN with a low delay. We have demonstrated that the algorithm achieves high accuracy and a low convergence time in estimating network size in the random *i.i.d.* movement case. In addition, Mark-Recapture can outperform a gossip-based Pair-Wise Average scheme in terms of convergence time in this random scenario. However, in the real human mobility scenarios, only the algorithm in popular nodes (both Mark-Recapture and PW-Avg) can produce an accurate estimate of network size in a relatively short delay time, while majority of nodes (i.e. less popular nodes) are ineffective to perform node counting.

After this, we improved Epidemic routing by incorporating a local estimate of network size to the routing scheme to reduce message redundancy in the network. We showed that Epidemic with LE (local estimates) can achieve delivery ratio as high as conventional Epidemic routing, but at a lower overhead ratio and total message dropped in both the random and real-life scenarios. Nevertheless, this efficient delivery of Epidemic-LE slightly increases the delivery latency beyond that of (conventional) Epidemic routing.

Finally, for future works we can identify two points. First, we have shown that Mark-Recapture cannot work appropriately in less popular nodes in the real human mobility scenarios. Consequently, these nodes should rely on popular nodes to learn an accurate information of network size. In future, we therefore need to study a method to efficiently distribute the counting results of the popular nodes to all nodes in the network, such as a publish-subscribe scheme [29] or a client-server model [30]. Second, even though this paper only considers a closed system, we also need to take into account a more realistic scenario, i.e. an open system with churn, where nodes are allowed to enter and leave the area during the experiment. We believe that our proposed algorithm should be improved to accommodate this complex system.

Acknowledgements

This work has been partially supported by the Sanata Dharma University's research grant 2019.

References

[1] J. Hu, L. Yang, H. V. Poor, and L. Hanzo, "Bridging the Social and Wireless Networking Divide: Information Dissemination in Integrated Cellular and Opportunistic Networks", in *IEEE Access*, vol. 3, pp. 1809-1848, 2015.

[2] Z. Lu, G. Cao, and T. La Porta, "TeamPhone: Networking SmartPhones for Disaster Recovery", in *IEEE Trans. on Mobile Computing*, vol. 16, no. 12, pp. 3554-3567, 2017.

[3] Y. He, X. Tang, R. Zhang, X. Du, D. Zhou, and M. Guizani, "A Course-Aware Opportunistic Routing Protocol for FANETs", in *IEEE Access*, vol. 7, pp. 144303-144312, 2019.

[4] E. Wang, Y. Yang, J. Wu, and W. Liu, "Phone-to-Phone Communication Utilizing WiFi Hotspot in Energy-Constrained Pocket Switched Networks", in *IEEE Trans. on Vehicular Technology*, vol. 65, no. 10, pp. 8578-8590, 2016.

[5] C. Y. Aung, I. W. Ho, and P. H. J. Chong, "Store-Carry-Cooperative Forward Routing With Information Epidemics Control for Data Delivery in Opportunistic Networks", in *IEEE Access*, vol. 5, pp. 6608-6625, 2017.

[6] A. Vahdat and D. Becker, "Epidemic routing for partially connected ad hoc networks". *Technical Report CS-200006*, Duke University, 2000.

[7] J. Cui, S. Cao, Y. Chang, L. Wu, D. Liu, and Y. Yang, "An Adaptive Spray and Wait Routing Algorithm Based on Quality of Node in Delay Tolerant Network", in *IEEE Access*, vol. 7, pp. 35274-35286, 2019.

[8] L. Chen, A. Karbasi, and F.W. Crawford, "Estimating the Size of a Large Network and its Communities from a Random Sample", in *Adv. Neural Inf. Process Syst.*, vol. 29, pp. 3072-3080, 2016.

[9] P. Danielis, S.T. Kouyoumdjieva, and G. Karlsson, "UrbanCount: Mobile Crowd Counting in Urban Environments", in *Proc. of IEEE Annual Inf. Tech., Electronics, and Mobile Comm. Conf.*, Vancouver, BC, pp. 640-648, 2017.

[10] S. Batabyal and P. Bhaumik, "Estimators for global information in mobile opportunistic network", in *Proc. of the 2013 IEEE Intl. Conf. on Adv. Nets. and Telecom. Systems (ANTS)*, Kattankulathur, pp. 1-6, 2013.

[11] M. Jelasity, A. Montresor, and O. Babaoglu,"Gossip-based Aggregation in Large Dynamic Networks", in *Proc. of the ACM Transactions Computer System*, vol. 23, no. 1, pp. 219-252, 2005.

[12] A. Guerrieri, I. Carreras, F. De Pellegrini, A. Montresor and D. Miorandi, "Distributed Estimation of Global Parameters in Delay-Tolerant Networks", in *Proc. of the 2009 IEEE Intl. Symp. on a World of Wireless, Mobile and Multimedia Nets. & Workshops*, Kos, pp. 1-7, 2009.

[13] L. Massoulie, E. L. Merrer, A.N. Kermarrec, and A. Ganesh,"Peer Counting and Sampling in Overlay Networks: Random Walk Methods", in *Proc. of the 25th Annual ACM Symp. on Principles of Distributed Computing*, pp. 123–132, July 2006.

[14] J. N. Darroch,"The Multiple-Recapture Census: I. Estimation of a Closed Population", in *Biometrika*, vol. 45, no. 3/4, pp. 343-359, 1958.

[15] N. Accettura, G. Neglia, and L.A. Grieco, "The Capture-Recapture Approach for Population Estimation in Computer Networks", in *Computer Networks*, vol. 89, pp. 107-122, 2015.

[16] H. Barbosa, *et. al.*, "Human Mobility: Models and Applications", in *Physics Report*, vol. 734, no. 6, pp. 1-74, 2018.

[17] P. Jesus, C. Baquero, and P.S. Almeida, "A Survey of Distributed Data Aggregation Algorithms", in *IEEE Comm. Surveys and Tutorials*, vol. 17, no. 1, pp. 381-404, 2015.

[18] T. Ning, Z. Yang and H. Wu, "Counting in Delay-Tolerant Mobile Networks", in *Proc. of the 2010 IEEE Intl. Conf. on Comm.*, Cape Town, pp. 1-5, 2010.

[19] T. Li, S.T. Kouyoumdjieva, G. Karlsson, and P. Hui, "Data Collection and Node Counting by Opportunistic Communication", in *Proc. of IFIP Networking Conf.*, Warsawa, Poland, pp. 1-9, 2019.

[20] E.M. Volz, J.C. Miller, A. Galvani, L.A. Meyers, "Effects of Heterogeneous and Clustered Contact Pattern on Infectious Disease Dynamics", in *PLoS Comput. Biol*, vol. 6, no. 6, e 1002042, 2011.

[21] S. Manaseer and I. Alhabash, "Number of Nodes Estimation in Mobile Ad Hoc Networks", in *Intl. Journal of Interactive Mobile Technologies*, vol. 11, no. 6, pp. 65-72, 2017.

[22] A. Keranen and J. Ott, "The ONE Simulator for DTN Protocol Evaluation", in *Proc. of the 2nd Intl. Conf. on Simulation Tools and Techniques*, pp. 1-10, 2009.

[23] J. Scott, R. Gass, J, Crowcroft, P. Hui, C. Diot, and A. Chaintreau, *CRAWDAD Dataset Cambridge/Haggle/Infocom*, May 2009, (v. 2009-05-29), available: http://crawdad.cs.dartmouth.edu/cambridge/haggle/ imote/infocom.

[24] N. Eagle and A. Pentland, "Reality Mining: Sensing Complex Social Systems", in *Personal and Ubiquitous Computing*, vol. 10, no. 4, pp. 255-268, 2006.

[25] E. Yoneki, P. Hui, and J. Crowcroft, "Distinct Types of Hubs in Human Dynamic Networks", in *Proc. of the 1st Workshop on Social Network Systems*, pp. 7-12, 2008.

[26] M.S. Desta, E. Hyytia, J. Ott, and J. Kangasharju, "Characterizing Content Sharing Properties for Mobile Users in Open City Squares", in *Proc. of Wireless On-Demand Nets. Systs. And Services (WONS)*, pp. 147-154, 2013.

[27] R.K. Ganti, F. Ye, and H. Lei, "Mobile Crowdsensing: Current State and Future Challenges", in *IEEE Communication Magazine*, vol. 49, no. 11, pp. 32-39, 2011.

[28] X. Zhang, G. Neglia, J. Kurose, and D. Towsley, "Performance Modelling of Epidemic Routing in Computer Networks", in *Intl. Journal of Comp. and Telecoms. Net.*, vol. 51, no. 10, pp. 2867-2891, July 2007.

[29] C. Fang, H. Yao, Z. Wang, W. Wu, X. Jin and F. R. Yu, "A Survey of Mobile Information-Centric Networking: Research Issues and Challenges", in *IEEE Comm. Surveys & Tutorials*, vol. 20, no. 3, pp. 2353-2371, 2018.

[30] L. Duan, *et. al.*, "Motivating Smartphone Collaboration in Data Acquisition and Distributed Computing", in *IEEE Trans. on Mobile Computing*, vol. 13, no. 10, pp. 2320-2333, 2014.

JoWUA Final Acceptance - "Network Size Estimation in Opportunistic Mobile Networks: The Mark-Recapture Method"

Ilsun You <ilsunu@gmail.com>

Sat 8/14/2021 6:23 PM

To: Soelistijanto B <b.soelistijanto@usd.ac.id>;udevmanoah@gmail.com <udevmanoah@gmail.com>

2 attachments (86 KB)

JoWUA-copyright.rtf; jowua-guideline.zip;

Dear Bambang Soelistijanto and Geraldev Manoah,

We sincerely congratulate the paper titled "Network Size Estimation in Opportunistic Mobile Networks: The Mark-Recapture Method" for the final acceptance to Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications (http://jowua.com/)

As the next step, we would like to give the guideline for the camera ready version as follows:

- In our plan, your paper will be published in Vol. 12, No. 3 or Vol. 12, No. 4.
- To prepare for the camera ready version, please carefully read the file "edition-guideline-jisis.txt" included in "jowua-guideline.zip".
- Please send us the easychair latex source files and the authors' short bios. (txt file) with photos (jpg).
- Please use *.bib file for the reference.
- ESPECIALLY, PLEASE KEEP THE JOURNAL REFERENCE STYLE shown at the bottom of this email.
- Please give us the keywords of your paper (3~4).
- Please sign and send us the COPYRIGHT AND CONSENT FORM on behalf of the authors. (the file is attached)
- Please inform us of the corresponding author, and let us know his or her detail contact information such as postal address, phone number, email, and website.
- The deadline for the final copy, the short bios. and the signed COPYRIGHT AND CONSENT FORM is ****August 31, 2021****.
- For submission, please directly send the files to me via this email.

Please kindly confirm this email as soon as you receive it.

Best Regards,

Prof. Dr. Ilsun You, FIET
JoWUA EiC

++

Especially, please follow the below reference guideline when making bib file:

(1) in the case of the IEEE conference

```
// IEEE conference
@inproceedings{Janakiraman03,
author = "Chenfeng Vincent Zhou and Christopher Leckie and Shanika Karunasekera",
title = "Indra: a Peer-to-peer Approach to Network Intrusion Detection and Prevention",
booktitle= "Proc. of the 12th IEEE International Workshops on Enabling Technologies (WETICE'03), Linz, Austria",
<- Format = "Proc. of the ##th conference full name (conference short name), conference venue"
month= "June", <- please put month, year, pages exactly
year= "2003",
pages=" 226--231",
publisher = {IEEE}
}
```

(2) in the case of the ACM conference

```
// ACM conference
@inproceedings{Huang_2003,
author = {Chenfeng Vincent Zhou and Christopher Leckie and Shanika Karunasekera},
title = {A Cooperative Intrusion Detection System for Ad hoc Networks},
booktitle = {Proc. of the 1st ACM workshop on Security of Ad hoc and Sensor Networks (SASN'03), Fairfax, Virginia, USA},
<- Format = "Proc. of the ##th conference full name (conference short name), conference venue"

year = {2003}, <- please put month, year, pages exactly
month = {October},
pages = {135--147},
publisher = {ACM},
}
```

(3) in the case of the LNCS proceedings

```
// LNCS
@INPROCEEDINGS{dagon2004honeystat,
title={Honeystat: Local Worm Detection Using Honeypots},
```

```
author={Chenfeng Vincent Zhou and Christopher Leckie and Shanika Karunasekera},
booktitle={Proc. of the 7th International Symposium on Recent Advances in Intrusion Detection (RAID'04), French Riviera, France},
<- Format = "Proc. of the ##th conference full name (conference short name), conference venue, LNCS"
volume={2782}, <- please put volume number
series = {Lecture Notes in Computer Science}
pages={39--58}, <- please put month, year, pages exactly
year={2004},
month={September-October},
publisher={Springer-Verlag}
}

// Journal
@article{CVZHOU:JNW09,
author = {Chenfeng Vincent Zhou and Christopher Leckie and Shanika Karunasekera},
title = {Collaborative Detection of Fast Flux Phishing Domains},
journal = {Journal of Networks}, <- journal full name
pages = {75--84}, <- please put pages, volume, number, month, and year exactly.
volume = {4},
number = {1},
month = {February},
year = {2009}
}
```