Hindawi
Wireless Communications and Mobile Computing
Volume 2019, Article ID 6359806, 15 pages
https://doi.org/10.1155/2019/6359806

Research Article

Routing in Mobile Opportunistic Social Networks with Selfish Nodes

Annalisa Socievole,[1] Antonio Caputo ,[2] Floriano De Rango ,[2] and Peppino Fazio [2]

[1] *National Research Council of Italy (CNR), Institute for High Performance Computing and Networking (ICAR), Via Pietro Bucci, 7-11C, 87036 Arcavacata di Rende (CS), Italy*
[2] *Department of Informatics, Modeling, Electronics and Systems Engineering (DIMES), University of Calabria, 87036 Arcavacata di Rende (CS), Italy*

Correspondence should be addressed to Floriano De Rango; derango@dimes.unical.it

Received 27 July 2018; Accepted 29 November 2018; Published 3 February 2019

Guest Editor: Enrique Hernández-Orallo

When the connection to Internet is not available during networking activities, an opportunistic approach exploits the encounters between mobile human-carried devices for exchanging information. When users encounter each other, their handheld devices can communicate in a cooperative way, using the encounter opportunities for forwarding their messages, in a wireless manner. But, analyzing real behaviors, most of the nodes exhibit selfish behaviors, mostly to preserve the limited resources (data buffers and residual energy). That is the reason why node selfishness should be taken into account when describing networking activities: in this paper, we first evaluate the effects of node selfishness in opportunistic networks. Then, we propose a routing mechanism for managing node selfishness in opportunistic communications, namely, SORSI (Social-based Opportunistic Routing with Selfishness detection and Incentive mechanisms). SORSI exploits the social-based nature of node mobility and other social features of nodes to optimize message dissemination together with a selfishness detection mechanism, aiming at discouraging selfish behaviors and boosting data forwarding. Simulating several percentages of selfish nodes, our results on real-world mobility traces show that SORSI is able to outperform the social-based schemes Bubble Rap and SPRINT-SELF, employing also selfishness management in terms of message delivery ratio, overhead cost, and end-to-end average latency. Moreover, SORSI achieves delivery ratios and average latencies comparable to Epidemic Routing while having a significant lower overhead cost.

1. Introduction

Even if the Internet with its ubiquity has revolutionized the way in which we communicate, there are still some scenarios where this network infrastructure is not available. Examples of these scenarios are disaster/recovery situations, big sports events, and music festivals, or more in general, areas where the Internet is expensive, not available, or overloaded. In such situations, an alternative communication medium is necessary. Delay Tolerant Networks (DTNs) [1–3] and Opportunistic Networks [4] have gained a lot of interests in these last years thanks to their capability to face the uncertainty of a fixed network infrastructure providing connectivity between mobile devices. In future Internet there are some scenarios where it is not possible to guarantee an any-time end-to-end connectivity and it is not always known a priori the topology or network infrastructure. Examples of these scenarios are disaster/recovery situations, big sports events, and music festivals, or more in general, areas where the Internet is expensive, not available, or overloaded. In such situations, an alternative communication medium is mandatory and, at this purpose, DTNs and ONs can become a promising communication paradigm [5–7]. Since an end-to-end path between mobile phone, hand-on devices, or other terminals is not always available, opportunistic routing algorithms need to be designed with different features in comparison with routing for traditional Mobile Ad Hoc Networks (MANETs). Because nodes are not always part of a precomputed path, it is necessary for mobile nodes to store messages waiting to meet during their movement some nodes with good characteristics (on the basis of the metric) where to forward the message. Because it is not known how long time a packet can be buffered, novel forwarding strategies and a specific bundle layer need to be designed to guarantee a hop-by-hop data delivery. Moreover, a stronger store-carry-forward approach needs to be applied for each node and all nodes form a DTN domain due to their capability to tolerate data delivery delay. In this context, opportunistic networking and forwarding can be a useful and mandatory approach to offer to each node an opportunity to send data packet freeing space in its buffer [8–11]. The opportunistic approach assumes that nodes that want to transmit can use some specific metric to select the next hop where forwarding the data, and the receiving node is assumed that want to receive the data packet sent by node encountered. However, in these last years, the second assumption is not always considered because it has been observed as in the real life nodes can be also selfish instead of cooperative. Because some resources are limited such as buffer size, energy budget, or other constraints, mobile nodes cannot see benefits in participating in a collaborative way to the communication. This selfish behavior can also be evident not at the beginning of the communication but during the network dynamic on the basis of the constraints and budgets that can be exhausted. In this case, if some countermeasures are not adopted, there is the risk that after some period, some nodes stop to forward data packet increasing the data delay, reducing the data packet delivery ratio, and losing the advantage of a flexible paradigm such as DTNs.

This paper proposes a routing mechanism for opportunistic networks able to mitigate the effect of node selfishness. Exploiting the social-based nature of human encounters, we propose a selfishness-aware opportunistic routing scheme able to maximize message delivery. As a matter of fact, message dissemination in opportunistic networks can be efficiently performed using social-based routing metrics. Several works such as [6, 7, 12–17] demonstrate that the nodes with a highly social behavior are good carriers and are thus able to improve message delivery. However, how to extract node sociality and define efficient routing metrics is still challenging. First, opportunistic networks are highly dynamic and it is not easy to reconstruct a node's social behavior. Second, when representing the network as a social graph, several metrics are able to define the sociality and the social position of a node (see, for example, the social metrics analyzed in [18]). Each of these metrics is not always easily computable in a distributed way on a dynamic graph. Third, each node may interact in different social contexts like real world when moving, online world using social networking websites, and so on. As such, each node has a sociality that can be defined "multilayer" [18].

Starting from the above considerations, to select an effective forwarding node, the routing scheme we propose, named SORSI (Social-based Opportunistic Routing with Selfishness detection and Incentive mechanisms), measures the forwarding capability of a node when compared to an encountered node in terms of node centrality (i.e., the importance/position of a node within a social graph), tie strength, and link prediction. Each of these routing metrics is computed in a distributed way on a particular social network layer that describes a social dimension. Since each mobile user may have social relationships on several social dimensions/layers (e.g., real life, Facebook, Twitter, etc.), this protocol exploits the user multilayer sociality for computing forwarding paths and improving message dissemination. Moreover, it adds a selfishness detection mechanism based on the history of message exchanges for incentivizing the forwarding of packets. Starting from the work in [19] where the effect of selfish nodes in the network performance has been presented, in this paper we propose a mechanism to detect selfish nodes and to discourage their behavior. This mechanism has been designed and implemented in a multilayer social routing in order to test its effectiveness and it has been compared with other well-known data dissemination techniques for opportunistic networks. So, the contributions of this paper can be summarized as follows:

(i) We characterize the situations in which an opportunistic network node may act selfishly.

(ii) We analyze the effect of node selfishness in opportunistic networks through simulations.

(iii) We propose a multilayer social-based routing scheme with selfishness management and compare it to three reference opportunistic routing protocols over two experimental datasets of human mobility with different connectivity patterns.

(iv) We show that, in general, as the proportion of nodes acting selfishly in the network increases, the message delivery ratio decreases and the average end-to-end latency increases.

(v) We show that the use of multilayer social network information together with selfishness detection is able to achieve delivery ratios and average latencies comparable to epidemic delivery with a much lower overhead cost.

The paper has been organized as follows. Section 2 provides background information. Section 3 discusses the selfish node problem in opportunistic networks and analyzes its effect on routing through simulations. Section 4 describes our proposal SORSI. In Section 5, we detail the simulation setup for SORSI evaluation and discuss the results. Finally, Section 6 concludes the paper summarizing the results obtained and discussing the future research directions.

2. Related Works

Many works in literature, in these last years, are focusing on the selfish behavior and on its effect on network performance. However, a few contributions till now have been related to DTNs and ONs because for these networks it is more challenging to detect malicious behavior and it is not easy to propose effective incentive mechanism that can tradeoff between network performance and resource draining.

2.1. Selfishness Management in MANET. In MANET and distributed wireless networks, some monitoring techniques to detect malicious behavior or selfish nodes are applied. One of the most adopted techniques is the watchdog technique such as that presented in [20]. According to this approach, nodes monitor each other's communications to ensure if the considered node behaves as expected or not. On the basis of the higher or lower adherence to an expected behavior, it is assigned a reputation value. The computed reputation value is then compared with some thresholds to decide if the node is reliable, selfish, or to be monitored. The main limitation of this approach, however, is that within an opportunistic network many encounters might not be observed by a third party. In [21] authors use the currency, a method based on purchasing credits to assign to mobile nodes for their forwarding service. A general model for electronic coupons where a central system spreads electronic coupons among interested users via access points installed in shops is proposed. In [22] authors analyze the effect in the introduction of a trust management scheme applied in MANET to detect selfish nodes and to isolate them. Also if trust management can be useful to avoid malicious nodes, it can be expensive in terms of energy if it is not applied carefully on a distributed network.

2.2. Selfishness Management in DTNs and ONs. The absence of a predetermined path and a priori nodes belonging to a path can present some risks in the data forwarding if some mechanism to incentive the cooperation is not applied. In [11] authors propose a mechanism where all nodes can store in their memory a list of topics/interests. When they meet some other node that can be interested to one or more of these topics, they can spread the topic if the other node behaves in the same way. This means that noncooperative nodes will not share interesting topics in the network and this penalizes the egoistic behavior. In [23] authors propose collaborative contact-based watchdog (CoCoWa) as a collaborative approach based on the diffusion of local selfish nodes awareness when a contact occurs, so that information about selfish nodes is quickly propagated. This collaborative approach reduces the time and increases the precision when detecting selfish nodes. In [18], authors propose an incentive mechanism called IRONMAN. It uses social network information to bootstrap the detection and discouragement of selfishness through an asynchronous bilateral trading. In this work authors try to give a rank to nodes that behave cooperatively and forward data. In particular, nodes that forward data on behalf of other node can increase their rank whereas other nodes that discard packets or behave selfishly decrease their rank and they are not involved in the data forwarding and sharing. The social network such as Facebook is considered in the proposal to establish the initial rank value on the online social network among involved nodes. In [24], Ciobanu et al. propose SPRINT-SELF, a routing mechanism for ONs that use preexisting online social information for detecting communities and the prediction of future encounters for routing data. This routing strategy allows node to keep info about past data transfers and battery level so that through these values they are able to compute an altruism score that is used to select or not the next hop forwarding the data. Only if this score related to the altruism is within certain range, the message is sent.

3. The Selfish Node Problem in Opportunistic Networks

From the viewpoint of an individual, selfishness is commonly defined as a set of attitudes and behaviors aimed uniquely to achieve the proper personal interests. A selfish individual pursues his goals even at the cost of damaging the interests of others. How this behavior is translated in the context of opportunistic networks? Many researchers such as Urpi et al. [25] consider the tradeoff between the cost in terms of energy consumption and the benefits in terms of network throughput as a key aspect to define an eventual selfish behavior of a node involved in a routing operation. An opportunistic network node may thus act in a selfish way when induced by *energy constraints*. Another valid reason for a node to act selfishly is related to *message buffer status*. If a node has its buffer full, it may act selfishly deciding to first drop the other nodes' packets. Moreover, considering an opportunistic networking environment where the nodes have high mobility and frequent disconnections, the *contact duration* between them might be highly variable and this is a further potential reason for being selfish. A node, once assessed to have many short contacts, may decide to exploit a contact for first exchanging its messages and subsequently the messages belonging to other nodes in order to leave the possible fragmentation of a message due to the interruption of a wireless contact only to other nodes. Figure 1 depicts the three main reasons for deciding to act selfishly in an Opportunistic Network Environment.

Hypothesizing that a node acts in a selfish way for the reasons described above, the routing performance of opportunistic forwarding may severely degrade thus causing serious problems to the communications within the opportunistic network. In this section, we show through a preliminary analysis how routing performance is affected by node selfishness. For modeling node selfishness, we proceed as follows. Through a uniform distribution, we randomly choose a subset of network nodes labeling these nodes as selfish. We consider 25%, 50%, and 75% of selfish nodes and the limit case when all nodes are selfish. We hypothesize that each of these nodes acts selfishly for one or more of the three reasons described above. For this initial study, we implement a basic node selfishness behavior. When a selfish node encounters another node, they will exchange their messages as usually done in an opportunistic networking scenario with a certain routing scheme. However, the selfish node will drop the messages received for which it is not the destination. The following subsections describe the simulation setup and the results of this preliminary analysis.

3.1. Simulation Setup. In order to test the effect of node selfishness on opportunistic routing, we carried several simulations using the Opportunistic Network Environment (ONE) simulator [26]. This simulation environment models a node with a radio interface, persistent storage, several movement models (including the simulation of realistic mobility traces), several routing capabilities, a basic energy consumption model, and different application interactions.

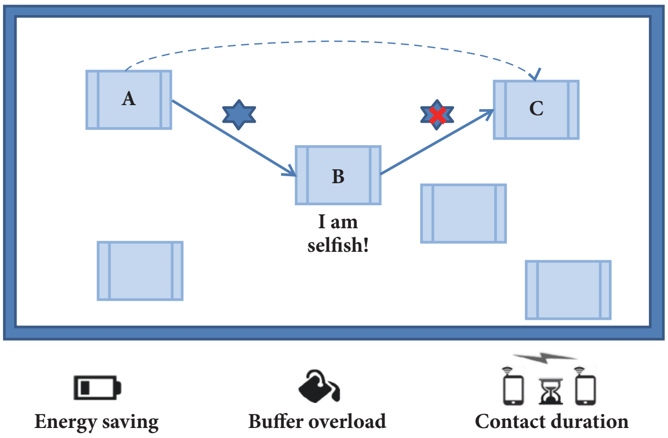


Figure 1: Node A wants to send a message to node C using node B as forwarder. Node A forwards the message to node B, but node B is selfish and will not forward the message to node C, because it wants to preserve his battery (*energy constraints*), it wants to leave space in the buffer deleting the messages belonging to other nodes (*buffer overflow*), and/or it wants to privilege its own messages during routing due to short contact durations (*contact duration*).

Table 1: Characteristics of the two experimental datasets.

Experimental dataset	SASSY	LAPLAND
Environment	Academic/Urban	Conference
Device type	T-Mote	Imote
Radio interface	ZigBee	Bluetooth
Granularity	6.67 s	[120-600] s
Duration	70 days	3 days
# of nodes	27	17

In the following subsections, we describe the mobility traces used, the routing protocols we simulated, and the performance measures analyzed.

3.1.1. Human Mobility Traces. We utilize two experimental datasets of human mobility where the mobile devices ran software logging contacts between them. Table 1 summarizes the main features of the selected datasets. SASSY dataset [27] contains the ZigBee encounter logs of 27 participants carrying T-Mote sensors and their social network, generated from Facebook data self-declared by candidates at the beginning of the experiment. The experiment took place at University of St. Andrews (United Kingdom) for an overall duration of 3 months between February 15, 2008 and April 29, 2008. For our analysis, we consider one week of encounters, focusing on the week having the highest number of encounters. LAPLAND [28] dataset spans a shorter period and was collected during the ExtremeCom09 workshop in Padjelanta National Park (Sweden). The Bluetooth colocation data of 17 conference attendees were gathered during 4 consecutive days of the experiment, from August 9, 2009 to August 12, 2009. Each candidate was asked to carry Imotes with him detecting devices in proximity. This dataset includes also each participant's Facebook friend list and interests in terms of scientific topics.

3.1.2. Routing Protocols. In simulations, we analyze three benchmark opportunistic routing protocols, Epidemic Routing [29, 30], Spray & Wait [31], and Bubble Rap [13]. In the following lines, a brief explanation motivates the reason for which we choose to consider these protocols for our study.

(i) *Epidemic Routing* is a flooding-based protocol. When two nodes encounter, they exchange all their messages so that these messages are spread like viruses by pairwise contacts between two nodes. This protocol is considered a reference for opportunistic routing since it determines an upper bound for message delivery.

(ii) *Spray & Wait* is a different kind of epidemic routing which floods the network with a fixed number of copies of a message. The source node "sprays" L message copies to L distinct encountered nodes and then "waits" hoping that one of these nodes will carry the message to the intended destination node. If the destination node is not found during the spray phase, each of the L nodes holding a message copy will forward the message only to the destination node. This algorithm has shown to have routing performance near to the classic epidemic routing with a lower overhead cost.

(iii) *Bubble Rap* [13] is a social-based protocol, considered one of the most efficient protocols in terms of messages delivered and delay in delivering them. Bubble Rap uses two centrality values associated with each node based on its global popularity in the whole network and the local popularity within its community or communities. The forwarding scheme uses these centrality values so that a message is transferred to nodes with higher global centrality values until the carrier node meets a node with the same community label as the destination node. Then, a message is forwarded to nodes with higher local rankings until successful delivery. Since opportunistic networks are characterized by a social nature due to human mobility, this protocol has shown to reduce the number of message replicas spread on the network while maintaining a good message delivery. For our simulations, we chose C-Window degree [13] for implementing node centrality since it can be computed in a fully distributed way providing a good estimate of the suitability of a node as message relay. This is a cumulative centrality measure averaging degree centrality (i.e., the number of unique contacts had with the encountered nodes) over a fixed number of temporal windows. For detecting communities, we chose k-Clique [32]. Even if several community detection methods are present in the literature, we chose this method since it finds overlapping communities, which are more similar to the communities formed in real life. For this community detection method, a community is defined as the union of all k-cliques (complete subgraphs with k nodes) that can reach each other through a series of adjacent k-cliques, where two k-cliques are said to be adjacent if they share $k-1$ nodes.

3.1.3. Performance Metrics. We use two important opportunistic performance metrics for our analyses. First, we study the *system throughput*, or delivery ratio, which is computed here as the number of delivered packets divided by the number of unique packets created in the system. Then we study the *system delay*, or delivery delay, which only considers the delivered messages. This metric measures the time it takes a packet to be delivered to the destination node.

We evaluated these metrics as a function of the TTL (Time To Live), which represents the maximum time a message can stay in the system after its creation. The TTL is fundamental for studying the ability of a routing protocol to find an adequate number of relay nodes within a certain time. The values of the main parameters used in our simulations are shown in Table 4. We repeated each simulation 20 times and took the average of each run as a result.

3.2. The Selfishness Effects

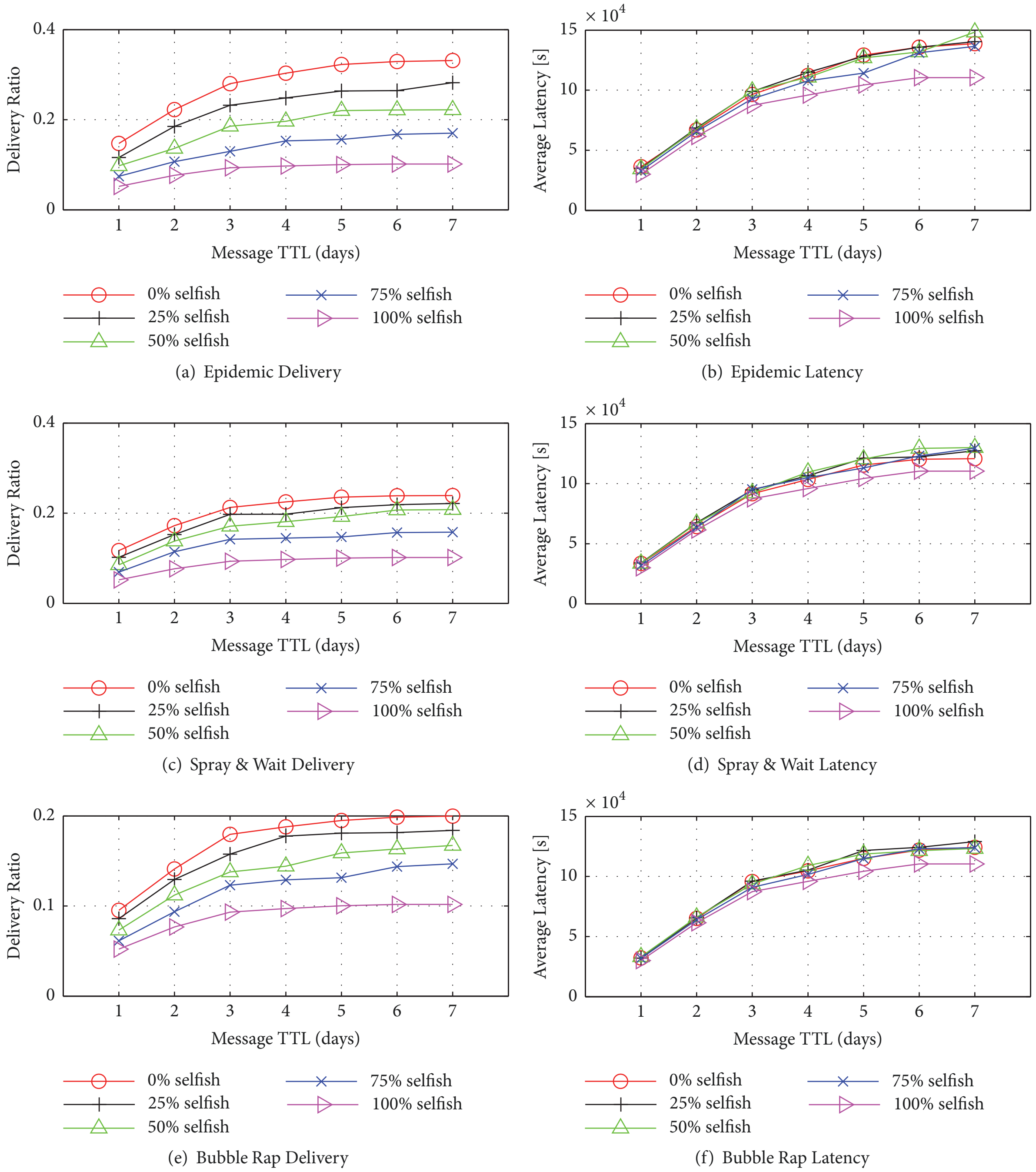
3.2.1. SASSY Results. We start evaluating the effects of node selfishness on opportunistic routing by analyzing the SASSY mobility trace. The main simulation parameters are listed in Table 2. Delivery ratio and average latency for this trace are shown in Figure 2. By analyzing delivery ratio, we observe that as the proportion of nodes acting selfishly in the network increases, this network performance decreases for all the routing protocols considered. This means that if the selfish behavior can be in some way detected and discouraged, it might be possible to achieve the same performance as if no nodes behave selfishly even if they have a propensity for being selfish. Spray & Wait delivery, compared to epidemic delivery achieving the highest message delivery, is slightly lower as it can be expected since it disseminates a limited number of message copies. As far as Bubble Rap evaluation concerns, we found an interesting result: as the percentage of selfish nodes increases, the system throughput does not vary significantly. For a message TTL equal to one week, for example, the protocol is characterized by the maximum difference between delivery with no selfish nodes and with all selfish nodes that is equal to 0.1. We consider this difference low compared to Epidemic Routing and Spray & Wait. This social-based protocol if on one hand is characterized by a lower delivery ratio due to its more restrictive forwarding rules, on the other hand it is able to better manage node selfishness. We note that by choosing the relay nodes that are more social, the selfishness effect can be better balanced. Here, we think that by choosing only the most central nodes as node relays, the probability to find a relay node that can be also selfish is reduced. In the case of Epidemic Routing, on the contrary, where each encountered node is chosen as relay node, the probability to find a selfish relay node is higher. A similar thing happens in Spray & Wait. It can be further observed that all algorithms deliver more packets to the destinations as the TTL increases. However, as the TTL becomes high, the increment in the delivery ratio is marginal, since the capacity of the network to forward packets becomes the performance bottleneck.

Table 2: Values for the simulation parameters.

	Parameter	Value
Network	Buffer size	2000 MB
	Message ∗ size	1 kB
	Intermessage creation interval	1800 s
Spray & Wait	L	5
Bubble Rap	C-Window duration	6 hours
	C-Window # of windows	5
	k (k-Clique)	3

∗ Each message is exchanged between randomly selected source-destination pairs.

On the system delay, node selfishness causes a degradation of this performance as the TTL increases. We note that for TTLs greater than 3 days, the average latency highly depends from the percentage of selfish nodes that are present within the network, while for lower TTLs node selfishness does not particularly degrade this protocol performance. This effect characterizes all the protocols considered. This happens because the longer TTLs result in higher probabilities for the packets to be relayed to selfish nodes during their path towards the destination. These packets will be discarded by the selfish nodes thus increasing the corresponding average latency. We can further note that Bubble Rap is less influenced by the percentage of selfish nodes. The average latency slightly varies from 0% to 100% of selfish nodes. Similarly for the delivery case, the social-based routing rules result in a better management of node selfishness. Finally, comparing the delay values achieved by the three protocols considered, Epidemic Routing has the highest average latency followed by Spray & Wait and Bubble Rap. Even if Epidemic Routing has usually the lowest delay due to the highest number of message replicas injected into the network, in this mobility scenario, the selfish behavior of some relay nodes together with the particular node encounters dynamics that often



(a) Epidemic Delivery

(b) Epidemic Latency

(c) Spray & Wait Delivery

(d) Spray & Wait Latency

(e) Bubble Rap Delivery

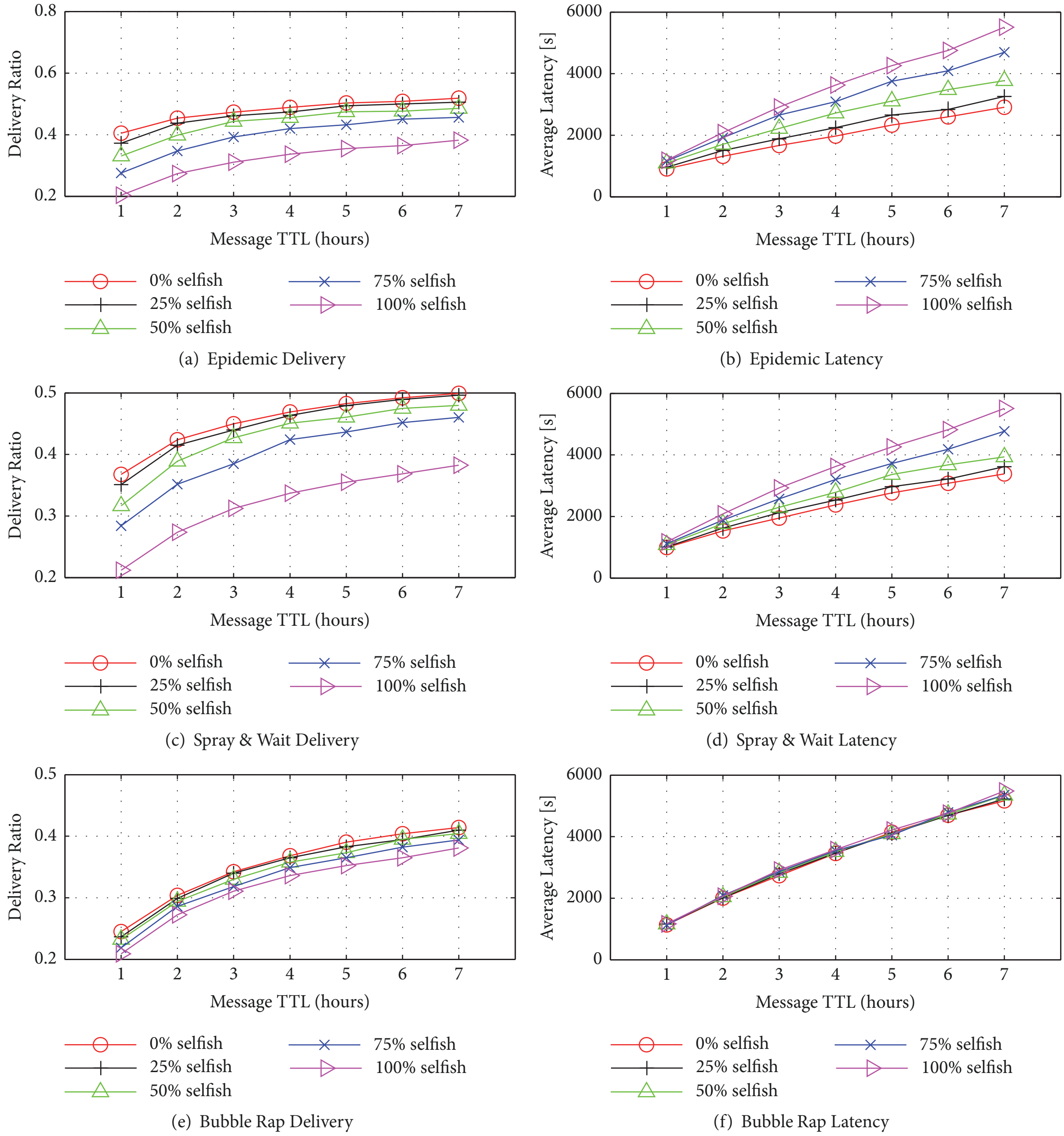
(f) Bubble Rap Latency

Figure 2: SASSY routing performance under different % of selfish nodes.

assign packets to these relays resulting in packets delivered with high delays.

3.2.2. LAPLAND Results. LAPLAND dataset covers a shorter experimental period and has a smaller number of nodes compared to SASSY. As such, the TTLs have been varied from 1 to 7 hours. As it can be observed from the results in Figure 3, similarly to SASSY, the increase in the proportion of selfish nodes results in a lower delivery ratio for all the protocols considered. We thus conclude that also in a smaller dataset this feature is present. However, Bubble Rap shows again to better manage node selfishness as it can be observed by the trend of the curves in Figure 3(c). We can further note that again all the algorithms deliver more packets to the destinations as the TTL increases.

The average latency results confirm that the system delay increases as the TTL and the percentage of selfish nodes increase. In particular, for these lower TTL values, the average latency is characterized by an almost linear trend, which we also found in SASSY for TTLs lower than 3 days.



(a) Epidemic Delivery

(b) Epidemic Latency

(c) Spray & Wait Delivery

(d) Spray & Wait Latency

(e) Bubble Rap Delivery

(f) Bubble Rap Latency

Figure 3: LAPLAND routing performance under different % of selfish nodes.

Here, the interesting results deal again with Bubble Rap: for this social-based protocol, the values of average latency are similar both if the system contains selfish nodes and if the system has all altruistic nodes. This result again confirms that the social-based protocols offer a promising research direction concerning nodes selfishness and opportunistic routing. Comparing the average latency results achieved by the protocols, we found an opposite trend with respect to SASSY. Here, Epidemic Routing has the lower average latency, followed by Spray & Wait and Bubble Rap. These results are more similar to those usually found in the literature since in most cases, the epidemic delivery results in a lower latency (even if at a higher overhead cost) due to its flooding nature.

4. SORSI: Social-Based Opportunistic Routing with Selfishness Detection and Incentive Mechanisms

After being assessed through simulations that selfishness can severely degrade opportunistic routing, we now describe how this problem can be mitigated through a selfish detection mechanism incentivizing node collaboration. Our scheme, named SORSI (Social-based Opportunistic Routing with Selfishness detection and Incentive mechanisms), exploits node sociality and history of encounters to detect selfishness. Once selfishness is detected, nodes do not forward messages to selfish nodes. As such, selfish nodes are incentivized to

Table 3: List of symbols used to define SORSI social metric.

List of symbols	
i	i^{th} node
j	j^{th} node
d	destination node
t	t^{th} time slot
T	number of time slots
M_i	number of neighbors of node i
M_j	number of neighbors of node j
l	l^{th} online social network layer
L	number of online social network layers
e	encounter event
C_{Degree}	temporal degree centrality
$C_{CDegree}$	cumulative degree centrality
TS	online tie strength
TS_{TOT}	total online tie strength
LP	link predictor (common neighbors)
CS	centrality utility score
TSS	tie strength utility score
LPS	link predictor utility score
MLS	SORSI utility score
ESS	SORSI encounter-based selfishness score

participate in forwarding if they want their messages to be routed.

Social-based rules for opportunistic forwarding have been shown to result in low-cost forwarding paths (see, for example, [6, 7, 33]). Moreover, our previous analysis shows that Bubble Rap, which is a reference for social-based opportunistic routing protocols, is able to better sustain the selfishness effect. As such, we choose to build SORSI routing scheme on a social-based logic. However, differently from Bubble Rap only considering offline sociality reconstructed from node encounters, SORSI is based on a routing metric, namely MLS, which exploits three social dimensions: wireless proximity, online friendships, and interests. Since each of these dimensions is able to represent a node's sociality, we choose to consider this set of social features in order to have a wider view of a node's social behavior. As a matter of fact, the way in which we move, interact online with our friends, and share our interests represents our sociality. MLS metric is thus computed using a combination of three measures: (1) node centrality computed on the DSN (Detected Social Network, i.e., the social network detected through nodes' encounters) graph layer, (2) tie strength computed on the OSN (Online Social Network, i.e., Facebook, Twitter, etc.) graph layer(s), and (3) a link predictor computed on the Interest network layer (i.e., a social layer constructed on node interests). As such, we model the network nodes' sociality as a multilayer social network where each layer is a social graph representing a social dimension and computing MLS metric. Table 3 lists the symbols used to define the social metric.

We consider centrality as one of the most important factors to choose a good message relay. In graph theory and network analysis, centrality quantifies the structural importance of a vertex within the graph. A central node has usually a stronger capability of connecting other network nodes. We therefore compute centrality at the DSN layer, where the corresponding social graph is leveraged through encounters between mobile devices. Here, the DNS social graph of the multilayer social network is a dynamic graph where an edge between two nodes represents a wireless contact. There are several ways to measure centrality [18]. SORSI social metric computes node centrality for a node i, $C_{CDegree}(i)$, using a long-term cumulative estimate of degree centrality. Degree centrality basically quantifies the number of connections a node has. The advantage in using this measure is that it can be easily computed locally considering only a node's ego network, while other centrality measures (e.g., betweenness, closeness, or eigenvector centrality) require global knowledge of the network. More specifically, SORSI computes the number of unique nodes seen throughout a specific time slot and then averages this measure with a set of previous measures. Degree of centrality for a node i during a time slot t is computed as follows:

$$C_{Degree}(i,t) = \sum_{j=1}^{M_i} e(i,j,t) \tag{1}$$

where

$$e(i,j,t) = \begin{cases} 1 & if\ i\ encounters\ j\ during\ time\ slot\ t \\ 0 & otherwise \end{cases} \tag{2}$$

representing an edge between node i and node j on the DSN graph corresponding to the time slot considered (considering that the DSN graph is a temporal graph, we form a static graph for each time slot by amalgamating all contacts in that time interval), and M_i is the number of nodes in i's range. The cumulative degree, $C_{CDegree}(i)$, is then calculated by averaging the node's degree values over a set of T time slots including the most recent time slot and all the previous ones:

$$C_{CDegree}(i) = \frac{1}{T}\sum_{t=0}^{T} C_{Degree}(i, T-t) \tag{3}$$

In that way, SORSI provides a fully decentralized approximation for a node's degree centrality, which is easy to be computed.

Centrality described above is measured using the history of contacts and does not consider future links availability. Considering that the links in the network are time-varying, an existing link to a central node may not be highly available. We therefore include a *tie strength* indicator into SORSI social metric. This indicator is able to identify the links that have a higher probability to be activated and is measured by considering online social ties between the individuals carrying the mobile devices. This choice is driven by the consideration that social ties on online social networking websites, such as Facebook, Twitter (here we consider a tie between a user A and a user B, if A follows B and vice versa) or LinkedIn, are more stable and hence stronger than contact network ties. Typically, an online tie between two

Table 4: Values for the simulation parameters.

	Parameter	Value
Network	Buffer size	2000 MB
	Message ∗ size	1 kB
	Intermessage creation interval	1800 s
	TTL (SASSY)	4 days
	TTL (LAPLAND)	240 s
SORSI-NS	Time slot	6 hours
	T	5
SORSI	Selfishness threshold *thr*	50
	Initial selfish score ESS (altruistic node)	0
	Initial selfish score ESS (selfish node)	100
	Time slot	6 hours
	T	5
Bubble Rap	C-Window duration	6 hours
	C-Window # of windows	5
	k (k-Clique)	3
SPRINT-SELF	w_1	0.7
	w_2	0.3
	k (k-Clique)	3
	Cache size (I and O)	100 MB

∗ Each message is exchanged between randomly selected source-destination pairs.

users does not change over time. In Facebook, for example, "intermittent" friendships are highly improbable. Moreover, it has been shown in [18, 20] that nodes encounter other online socially connected nodes with a high probability. Consequently, online ties can be considered a good measure of whether a link on DSN will be activated. SORSI calculates tie strength between node i and node j at OSN layer l as

$$TS(i, j, l) = \begin{cases} 1 & if\ i\ and\ j\ are\ connected\ at\ layer\ l \\ 0 & otherwise \end{cases} \tag{4}$$

The total tie strength between two nodes is the sum of the indicators measured at each OSN layer:

$$TS_{TOT}(i, j) = \sum_{l=1}^{L} TS(i, j, l) \tag{5}$$

where L is the total number of online social networking websites considered.

SORSI social metric takes into account a third measure useful to predict future collaborations between two nodes. A *link predictor* is computed on Interest network layer, where a link between two nodes exists if they have at least one interest in common. Examining common neighbors of a pair of nodes on Interest network layer, we can predict future encounters to which the transfer of information may arise. Several works on coauthorship or collaboration networks demonstrated that the probability of two nodes being connected by a link is higher when the nodes in question have common neighbors. In [34], for example, a network of scientific collaborations was analyzed, showing that examining coauthors of authors helps in predicting future collaborations. Since coauthorship networks are networks of scientific collaborations and hence networks of shared scientific interests, we chose Interest network layer for link prediction, assuming that scientific interest networks and more general interest networks have a similar behavior. SORSI computes the link predictor $LP(i, j)$ of a possible future collaboration between node i and node j as a common neighbor measure based on Jaccard coefficient:

$$LP(i, j) = \frac{|M_i \cap M_j|}{|M_i \cup M_j|} \tag{6}$$

where M_i is the number of nodes in i's range and M_j is the number of nodes in j's range.

For each measure, SORSI determines the utility score of node i for delivering a message to node d compared to node j as follows:

$$CS(i, j) = \frac{C_{CDegree}(i)}{C_{CDegree}(i) + C_{CDegree}(j)} \tag{7}$$

$$TSS(i, j, d) = \frac{TS_{TOT}(i, d)}{TS_{TOT}(i, d) + TS_{TOT}(j, d)} \tag{8}$$

$$LPS(i, j, d) = \frac{LP(i, d)}{LP(i, d) + LP(j, d)} \tag{9}$$

The SORSI social metric is given by the combination of the contributing score values as follows:

$$MLS(i, j, d) = CS(i, j)\left[1 + TSS(i, j, d) + LPS(i, j, d)\right] \tag{10}$$

```
procedure ENCOUNTERNODE($M_i$)
  exchangeCentralityValues()
  exchangeOnlineContactsLists()
  exchangeInterestNodeList()
  exchangeForwardingHistoryList()
  for every message $m$ in message_buffer do
    $D \longleftarrow m$.destination()
    $myMLS \longleftarrow$ computeMLScore()
    $encounterMLS \longleftarrow$ computePeerMLScore()
    if $encounterMLS \geq myMLS$ || $M_i$==$D$ then
      forwardMessage($m$,$M_i$)
    end if
  end for
  for every message $m$ in forwarding_history_list do
    if $timestamp$ > last encounter with $M_i$ && $msgSourceID$==$myID$ then
      if last encounter with $forwarderID$ > last encounter with $M_i$ && $forwarderID$.receivedMessage($msgID$)==false then
        $ESS \longleftarrow forwarderID$.selfishScore()
        $forwarderID$.setSelfishScore($ESS$+1)
      end if
    end if
  end for
end procedure
```

Algorithm 1: SORSI message forwarding with selfishness detection.

```
procedure RECEIVEMESSAGE($m$,$M_i$)
  if $M_i$! = $m$.source() then
    $ESS \longleftarrow M_i$.selfishScore()
    $M_i$.setSelfishScore($ESS$-1)
    receive($m$)
  else
    if $M_i$.selfishScore() < $thr$ then
      receive($m$)
    else
      discard($m$)
    end if
  end if
end procedure
```

Algorithm 2: SORSI message reception with selfishness detection.

As can be observed, MLS captures the relay significance of a node when compared to an encountered node across all social network layers, in terms of centrality, tie strength, and link predictor. Note also that node centrality is considered as the predominant factor in message forwarding. Both tie strength and tie predictor utility scores are weighted with centrality utility score and then added to centrality utility score. In that way, tie strength and link predictor utility scores will reflect the centrality utility score (e.g., high, low, or medium) between the sender node and the encountered node.

The forwarding process in SORSI is given by Algorithm 1. The two nodes having an encounter exchange also their forwarding history lists containing the forwarding activity of the encountered nodes. Each past forwarding included in the list of a node is logged through the ID of the encountered node forwarding the message to that node, the message ID, the source ID of the forwarded message, and the time when the message has been received. A history list is thus implemented as a list containing *[forwarderID, msgID, msgSourceID, timestamp]* tuples. Then, after having computed the MLS metric to decide whether forwarding messages to the encountered node N is according to its sociality and selfishness, the node checks the encountered node forwarding history list to detect eventual selfish nodes. Note that the messages in the buffer are received messages that have successfully passed the selfishness check. If the node detects another node as selfish (i.e., N's history does not contain certain messages of the detecting node that should have been passed by the selfish node), the detecting node increments the encounter-based selfishness score *ESS* for the selfish node. Similarly, when a node receive (and will potentially forward) a message for which it is not the source (see Algorithm 2), its altruism is awarded decrementing its selfish score. In this way, an incentive mechanism to forward messages is realized. Nodes do not receive the messages of nodes that have a selfish score greater than a threshold. Consequently, according to SORSI scheme, nodes that have been labeled as selfish are obliged to forward the other nodes' messages for forwarding their own messages.

5. SORSI Performance Evaluation

5.1. Simulation Setup. We validate our proposal again through the Opportunistic Network Environment (ONE) simulator using SASSY and LAPLAND as human mobility traces. In simulations, we compare SORSI to its version

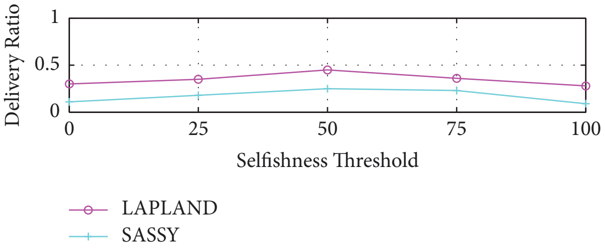


Figure 4: The effect of selfishness threshold on message delivery.

without selfishness detection, namely SORSI-NS, in order to quantify the improvements brought by selfishness detection and the mechanism incentivizing participation, to Epidemic Routing, Bubble Rap, and SPRINT-SELF. For this analysis, we do not analyze anymore Spray & Wait since we only focus on social-based schemes considering that we intend to demonstrate the improvements brought to opportunistic mobile social routing by SORSI. However, we still consider Epidemic Routing to have an upper bound on message delivery.

As previously specified, SPRINT-SELF protocol is social-based and employs a selfishness detection mechanism discouraging node selfishness. Here, we briefly describe its features. Similarly to SORSI, SPRINT-SELF uses social network information, future node behavior prediction, and a selfishness detection mechanism. This is the reason why we chose to compare SORSI to this protocol. However, differently to SORSI, SPRINT-SELF does not consider a multilayer social structure to model the opportunistic network and makes use of communities knowledge. Communities, since most of the nodes are more likely to interact with the members of its own community [20], are a fundamental part of this protocol. They are both computed on-the-fly through community detection algorithms such as k-Clique or through the social communities/groups directly extracted by online social networks (e.g., Facebook groups). However, in a recent study [18], we demonstrate that not always online communities correspond to the offline ones. As such, online community information is not always usable and may lead to suboptimal forwarding paths. Moreover, due to the computation of communities through community detection algorithms and of a utility value taking into account several information on contact and message exchange history, the cost of computing this utility value is sensibly higher than SORSI MLS value. Specifically, SPRINT-SELF uses a utility function which uses two parts one of

For this protocol, when a node is labeled as selfish we suppose its battery level is greater than a tolerance threshold (e.g., for 2% of battery level a node is not considered selfish if it does not forward messages, while for 30% of battery level it is).

For our analyses, we use three opportunistic performance metrics. First, we study again the *system throughput*. Then, we study the *system cost*, or overhead cost, which measures the number of packets transmitted across the air divided by the number of unique packets created. This performance index is important for our study since the duplicated messages injected into the network consume resources, such as battery and memory. Finally, similarly to the previous analysis, we study the *system delay*.

As first experiment, through a uniform distribution, we randomly choose a subset of network nodes labeling these nodes as selfish. We consider again 25%, 50%, and 75% of selfish nodes and the limit case when all nodes are selfish, hypothesizing that selfishness is driven by one or more of the three motivations described in Section 3. For Epidemic Routing, Bubble Rap, and SORSI-NS, a selfish node always drops the messages for which it is not the destination. We repeated each simulation 20 times and took the average of each run as a result. The values of the main parameters used in our simulations are shown in Table 4. Simulating different selfishness thresholds (see Figure 4), we have chosen a threshold equal to 50 for which we obtained the best SORSI delivery performance in both datasets. As expected, as the threshold increases being less restrictive with selfish nodes, the delivery ratio starts decreasing.

As second experiment, we test and compare the effect of energy consumption and buffer occupancy on node selfishness and hence, on opportunistic routing. For each node, we consider the energy consumption model adopted.

5.2. Results. We start evaluating our proposal by analyzing the SASSY mobility trace for the first experiment where we randomly choose a node as selfish. Delivery ratio, overhead cost, and average latency for this trace are shown in Figure 5. As expected, when the proportion of nodes acting selfishly in the network increases, the delivery ratio decreases for all the routing protocols considered. Similarly, the increase in the percentage of selfish nodes degrades the average latency, while the overhead cost decreases since the selfish behaviors result in a lower number of forwarded packets. However, Epidemic Routing is more influenced by the proportion of selfish nodes, showing a more rapid degradation of its routing performance indexes as the percentage of selfish nodes present within the network increases. We further note that when all nodes are selfish, the protocols deliver the same number of messages, resulting in equal costs and delays. This happens because every node forwards only its own messages. When it is present, there is a certain percentage of selfish nodes (25%, 50%, 75%); on the contrary, Epidemic Routing performs the best achieving the highest delivery ratio values, followed by SORSI which is able to outperform all the other social-based schemes (its classic version SORSI-NS, Bubble Rap, and SPRINT-SELF). However, SORSI is characterized by delivery ratios and average latencies comparable to Epidemic Routing with a much lower overhead cost. This result shows that SORSI is able to manage node selfishness while sensibly limiting the number of message replicas and, hence, the energy consumption avoiding the development of selfish behaviors. Moreover, SORSI with 50% of selfish nodes, for example, is able to achieve almost the same delivery ratio of Bubble Rap with no selfish nodes. Its multilayer social metric together with selfish management is thus able to achieve comparable routing performance of a reference social-based protocol with no selfish nodes. Compared to its version

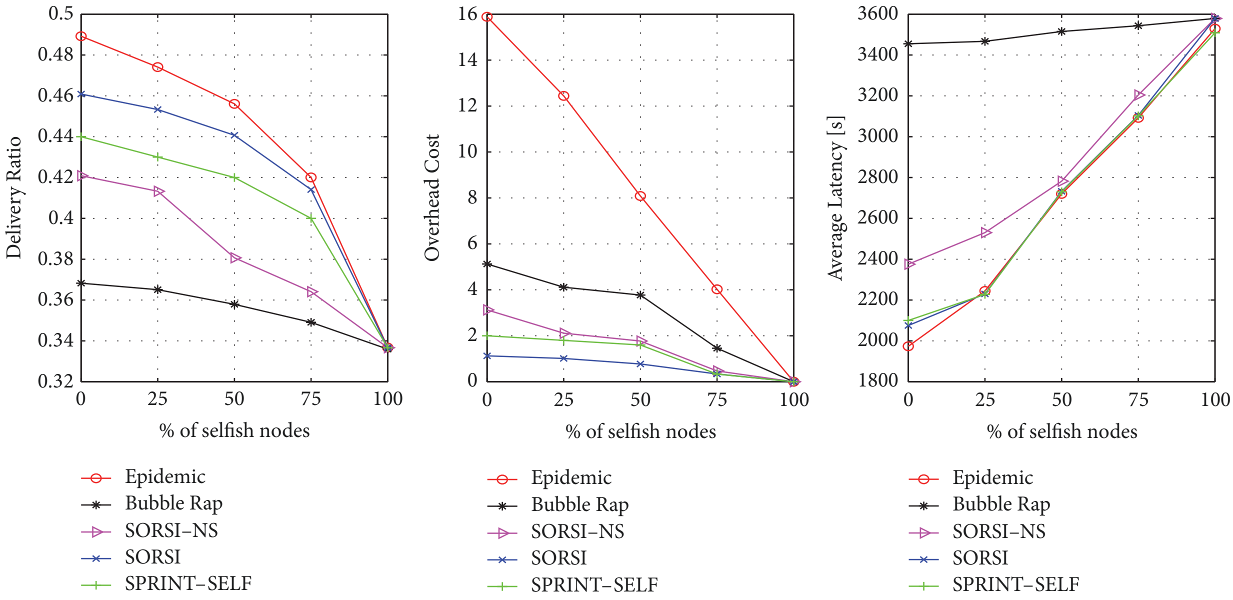


Figure 5: SORSI routing performances compared to Epidemic Routing, Bubble Rap, SORSI-NS, and SPRINT-SELF for the SASSY human mobility trace.

Table 5: Protocols performance when nodes have 50% of residual energy (SASSY).

	Epidemic	Bubble Rap	SORSI-NS	SORSI	SPRINT-SELF
Delivery Ratio	0.412	0.317	0.331	0.397	0.375
Overhead Cost	7.121	3.136	1.951	1.875	1.056
Average Latency	2406.54	3229.35	2451.779	2394.832	2398.569

Table 6: Protocols performance when nodes have 50% of residual energy (LAPLAND).

	Epidemic	Bubble Rap	SORSI-NS	SORSI	SPRINT-SELF
Delivery Ratio	0.151	0.091	0.129	0.147	0.137
Overhead Cost	5.162	4.125	3.165	1.551	2.945
Average Latency	$1.147 \cdot 10^5$	$1.211 \cdot 10^5$	$1.171 \cdot 10^5$	$1.145 \cdot 10^5$	$1.149 \cdot 10^5$

Table 7: Protocols performance when nodes have 75% of residual energy (SASSY).

	Epidemic	Bubble Rap	SORSI-NS	SORSI	SPRINT-SELF
Delivery Ratio	0.461	0.387	0.404	0.443	0.427
Overhead Cost	10.182	6.183	4.889	4.751	3.559
Average Latency	2009.573	3275.596	2354.773	2010.551	2009.762

without selfishness detection, we observe that SORSI is able to reduce the effect of node selfishness behaving similarly to SORSI-NS with a significantly lower number of selfish nodes.

Figure 6 shows the results obtained for the LAPLAND mobility trace. This trace covers a shorter experimental period and has a smaller number of nodes compared to SASSY. As such, we chose a shorter TTL of 4 minutes. Similarly to SASSY, the increase in the proportion of selfish nodes results in a lower delivery ratio for all the protocols considered. Again, SORSI outperforms Bubble Rap, SORSI-NS, and SPRINT-SELF achieving a message delivery similar to Epidemic Routing. We thus conclude that also in a smaller dataset this feature is present. The overhead cost trends confirm that SORSI is able to reduce the number of message replicas achieving the lowest costs without compromising its efficiency in terms of end-to-end delay.

In Table 5 SORSI outperforms SORSI-NS and Bubble Rap with performance close to Epidemic Routing in terms of delivery ratio. Also average latency is lower in SORSI in comparison with the other diffusion strategies. However, overhead cost of SORSI is higher than SPRINT-SELF. This means that the number of node involved in data dissemination is grater in SORSI and SORSI-NS than SPRINT-SELF. The same trend can be observed in Table 6 where SORSI in LAPLAND scenario outperforms all data diffusion techniques in all performance metrics. When the number of nodes switching in selfish behavior increases such as that in Tables 7 and 8, the performance metrics of delivery ratio and average latency

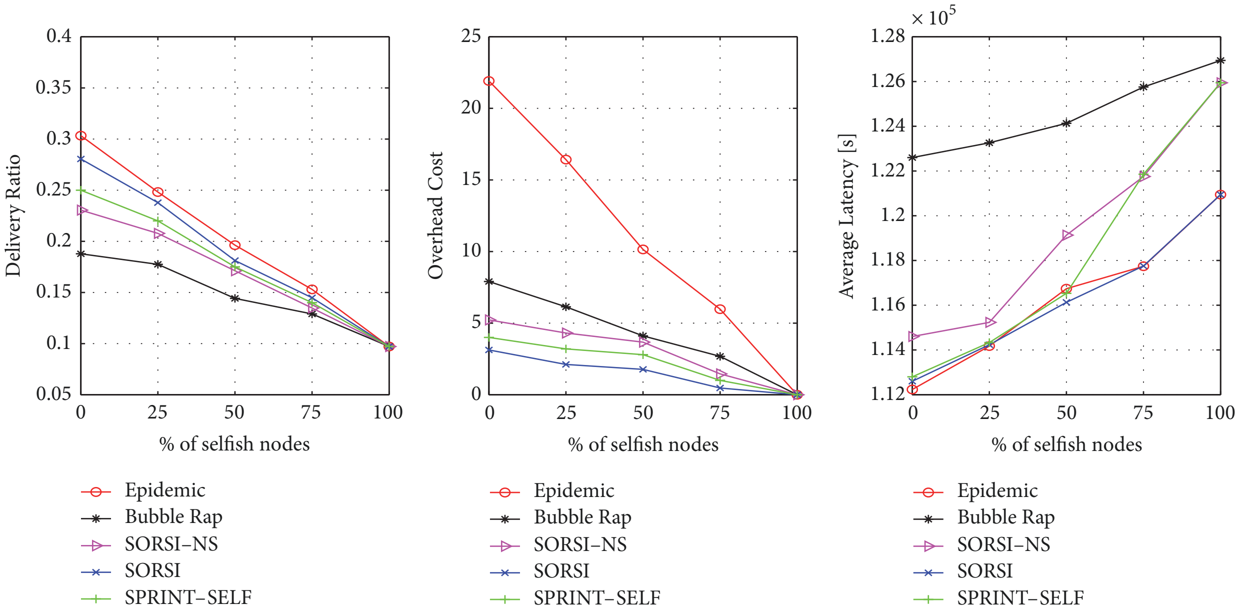


FIGURE 6: SORSI routing performances compared to Epidemic Routing, Bubble Rap, SORSI-NS, and SPRINT-SELF for the LAPLAND human mobility trace.

TABLE 8: Protocols performance when nodes have 75% of residual energy (LAPLAND).

	Epidemic	Bubble Rap	SORSI-NS	SORSI	SPRINT-SELF
Delivery Ratio	0.231	0.189	0.201	0.229	0.211
Overhead Cost	13.769	5.125	4.751	3.161	4.125
Average Latency	$1.122{\cdot}10^5$	$1.209 \cdot 10^5$	$1.131 \cdot 10^5$	$1.122 \cdot 10^5$	$1.123 \cdot 10^5$

TABLE 9: Protocols performance when nodes have 50% of full buffer (SASSY).

	Epidemic	Bubble Rap	SORSI-NS	SORSI	SPRINT-SELF
Delivery Ratio	0.391	0.311	0.334	0.375	0.359
Overhead Cost	6.125	2.566	1.956	1.873	1.018
Average Latency	2389.24	3119.775	2399.161	2355.22	2356.594

TABLE 10: Protocols performance when nodes have 50% of full buffer (LAPLAND).

	Epidemic	Bubble Rap	SORSI-NS	SORSI	SPRINT-SELF
Delivery Ratio	0.141	0.094	0.114	0.132	0.121
Overhead Cost	5.093	4.037	3.05	1.467	2.895
Average Latency	$1.11{\cdot}10^5$	$1.192 \cdot 10^5$	$1.153 \cdot 10^5$	$1.116 \cdot 10^5$	$1.147 \cdot 10^5$

are always better than other strategies. However, the overhead cost of SORSI is worst in SASSY and it is better in LAPLAND scenario. This means that when few TTLs are allowed and the scenario is small, SORSI does not have enough time to involve more nodes in data dissemination. Similar considerations can be applied when buffer occupancy threshold is used. It can affect the selfish behavior such as presented in Tables 9–12. In these cases, SORSI always outperforms in delivery ratio and average latency the other data dissemination strategies with the exception of Epidemic Routing that is considered as benchmark for the delivery ratio. Moreover, the overhead cost slightly increases in the SASSY scenario where SPRINT-SELF is more performing than SORSI and SORSI-NS.

6. Conclusions

In this paper, we have presented a routing mechanism for mitigating the effect of node selfishness on opportunistic network routing. First, we have delineated the main reasons for which an opportunistic node may act selfishly and evaluated the effects of node selfishness on opportunistic routing. Then, we have proposed SORSI, a selfishness-aware opportunistic routing scheme using multilayer social network information and the history of forwarding for driving routing decisions. Simulating two real-world mobility traces, we have demonstrated that SORSI is able to outperform Bubble Rap, SORSI version without selfishness detection

Table 11: Protocols performance when nodes have 75% of full buffer (SASSY).

	Epidemic	Bubble Rap	SORSI-NS	SORSI	SPRINT-SELF
Delivery Ratio	0.452	0.385	0.404	0.449	0.421
Overhead Cost	9.332	5.957	3.146	1.161	2.593
Average Latency	2002.765	2754.925	2012.48	1984.275	1999.734

Table 12: Protocols performance when nodes have 75% of full buffer (LAPLAND).

	Epidemic	Bubble Rap	SORSI-NS	SORSI	SPRINT-SELF
Delivery Ratio	0.261	0.199	0.219	0.259	0.231
Overhead Cost	5.125	4.165	3.165	1.447	2.169
Average Latency	$1.139 \cdot 10^5$	$1.172 \cdot 10^5$	$1.154 \cdot 10^5$	$1.137 \cdot 10^5$	$1.138 \cdot 10^5$

and SPRINT-SELF achieving epidemic delivery ratios with the lowest overhead cost. Moreover, SORSI outperforms SPRINT-SELF and Bubble Rap also in cases where buffer thresholds or energy thresholds can affect the selfish behavior increasing or decreasing the number of nodes acting selfishly. In these cases, SORSI continues to perform better in terms of delivery ratio while reducing the average latency in comparison with the other social-aware data dissemination techniques. As a result, SORSI is able to increase the number of nodes cooperating within the opportunistic network when some nodes have an initial attitude to act selfishly.

In future work, we plan to further validate SORSI extending the simulations to other datasets investigating also the impact of other parameters like TTL and the message generation rate. We also intend to analyze the theoretical reasons behind SORSI performance.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] K. Fall, "A delay-tolerant network architecture for challenged internets," in *Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications. SIGCOMM '03*, pp. 27–34, ACM, New York, NY, USA, 2003.

[2] V. Cerf, S. Burleigh, A. Hooke et al., "Delay-tolerant networking architecture," 2007, http://tools.ietf.org/html/rfc4838, http://tools.ietf.org/html/rfc4838.

[3] A. V. Vasilakos, Y. Zhang, and T. Spyropoulos, *Delay tolerant networks: Protocols and applications*, CRC Press, 2012.

[4] L. Pelusi, A. Passarella, and M. Conti, "Opportunistic networking: data forwarding in disconnected mobile ad hoc networks," *IEEE Communications Magazine*, vol. 44, no. 11, pp. 134–141, 2006.

[5] C. Boldrini, M. Conti, and A. Passarella, "Autonomic behaviour of opportunistic network routing," *International Journal of Autonomous and Adaptive Communications Systems*, vol. 1, no. 1, pp. 122–147, 2008.

[6] A. Socievole, E. Yoneki, F. De Rango, and J. Crowcroft, "ML-SOR: Message routing using multi-layer social networks in opportunistic communications," *Computer Networks*, vol. 81, pp. 201–219, 2015.

[7] F. De Rango, A. Socievole, and S. Marano, "Exploiting online and offline activity-based metrics for opportunistic forwarding," *Wireless Networks*, pp. 1–17, 2014.

[8] K. Xu, P. Hui, V. O. K. Li, J. Crowcroft, V. Latora, and P. Lio, "Impact of altruism on opportunistic communications," in *Proceedings of the 2009 1st International Conference on Ubiquitous and Future Networks, ICUFN 2009*, pp. 153–158, June 2009.

[9] P. Hui, K. Xu, V. O. K. Li, J. Crowcroft, V. Latora, and P. Lio, "Selfishness, altruism and message spreading in mobile social networks," in *Proceedings of the IEEE INFOCOM Workshops*, pp. 1–6, IEEE, April 2009.

[10] Y. Li, P. Hui, D. Jin, L. Su, and L. Zeng, "Evaluating the impact of social selfishness on the epidemic routing in delay tolerant networks," *IEEE Communications Letters*, vol. 14, no. 11, pp. 1026–1028, 2010.

[11] A. Mei and J. Stefa, "Give2Get: Forwarding in social mobile wireless networks of selfish individuals," *IEEE Transactions on Dependable and Secure Computing*, vol. 9, no. 4, pp. 569–582, 2012.

[12] A. Mtibaa, M. May, C. Diot, and M. Ammar, "PeopleRank: Social opportunistic forwarding," in *Proceedings of the IEEE INFOCOM 2010*, pp. 1–5, 2010.

[13] P. Hui, J. Crowcroft, and E. Yoneki, "BUBBLE Rap: social-based forwarding in delay-tolerant networks," *IEEE Transactions on Mobile Computing*, vol. 10, no. 11, pp. 1576–1589, 2011.

[14] A. Socievole, F. De Rango, and S. Marano, "Face-to-face with facebook friends: Using online friendlists for routing in opportunistic networks," in *Proceedings of the 2013 IEEE 24th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications, PIMRC 2013*, pp. 2989–2994, September 2013.

[15] S. Gaito, E. Pagani, and G. P. Rossi, "Strangers help friends to communicate in opportunistic networks," *Computer Networks*, vol. 55, no. 2, pp. 374–385, 2011.

[16] G. Bigwood and T. Henderson, "Bootstrapping opportunistic networks using social roles," in *Proceedings of the 2011 IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks, WoWMoM 2011*, pp. 1–6, June 2011.

[17] R. I. Ciobanu, C. Dobre, and V. Cristea, "SPRINT: Social prediction-based opportunistic routing," in *Proceedings of 7th*

IEEE WoWMoM workshop on autonomic and opportunistic communications, (AOC 2013), pp. 1161–1166, Madrid, Spain, 2013.

[18] A. Socievole, F. De Rango, and A. Caputo, "Opportunistic mobile social networks: From mobility and Facebook friendships to structural analysis of user social behavior," *Computer Communications*, vol. 87, pp. 1–18, 2016.

[19] A. Socievole, F. De Rango, A. Caputo, and S. Marano, "Simulating node selfishness in opportunistic networks," in *Proceedings of the 2016 International Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS 2016)*, pp. 1–6, July 2016.

[20] R. I. Ciobanu, C. Dobre, V. Cristea, and D. Al-Jumeily, "Social aspects for opportunistic communication," in *Proceedings of the 11th International Symposium on Parallel and Distributed Computing (ISPDC '12)*, pp. 251–258, IEEE, June 2012.

[21] M. Berlingerio, M. Coscia, F. Giannotti, A. Monreale, and D. Pedreschi, "Foundations of multidimensional network analysis," in *Proceedings of the 2011 International Conference on Advances in Social Networks Analysis and Mining, ASONAM 2011*, pp. 485–489, July 2011.

[22] A. Lupia and F. De Rango, "Evaluation of the energy consumption introduced by a trust management scheme on mobile adhoc networks," *Journal of Networks*, vol. 10, no. 4, p. 240, 2015.

[23] E. Hernández-Orallo, M. D. S. Olmos, J.-C. Cano, C. T. Calafate, and P. Manzoni, "CoCoWa: a collaborative contact-based watchdog for detecting selfish nodes," *IEEE Transactions on Mobile Computing*, vol. 14, no. 6, pp. 1162–1175, 2015.

[24] S. Ahmed and S. S. Kanhere, "Hubcode: message forwarding using hub-based network coding in delay tolerant networks," in *Proceedings of the 12th ACM international conference on Modeling, analysis and simulation of wireless and mobile systems. MSWiM '09*, pp. 288–296, ACM, New York, NY, USA, October 2009, http://doi.acm.org/10.1145/1641804.1641853.

[25] A. Urpi, M. Bonuccelli, and S. Giordano, "Modelling cooperation in mobile ad hoc newtorks: a formal description of selfishness. WiOpT03: Modeling and Optimization in Mobile," *Ad Hoc and Wireless Networks*, 2003.

[26] A. Keränen, J. Ott, and T. Kärkkäinen, "The ONE simulator for DTN protocol evaluation," in *Proceedings of the 2nd International Conference on Simulation Tools and Techniques (Simutools '09)*, pp. 1–55, ICST, Brussels, Belgium, March 2009, http://dx.doi.org/10.4108/ICST.SIMUTOOLS2009.5674.

[27] G. Bigwood, D. Rehunathan, M. Bateman, T. Henderson, and S. Bhatti, CRAWDAD trace set st_andrews/sassy/mobile (v. 2011-06-03), http://crawdad.cs.dartmouth.edu/st_andrews/sassy/mobile, 2011.

[28] E. Yoneki and F. B. Abdesslem, "Finding a data blackhole in bluetooth scanning," *ExtremeCom*, 2009.

[29] A. Vahdat and D. Becker, "Epidemic routing for Partially-Connected ad hoc networks," Tech. Rep., Duke University, April 2000.

[30] F. De Rango, S. Amelio, and P. Fazio, "Enhancements of epidemic routing in delay tolerant networks from an energy perspective," in *Proceedings of the 2013 9th International Wireless Communications and Mobile Computing Conference, (IWCMC)*, pp. 731–735, Italy, July 2013.

[31] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Spray and wait: an efficient routing scheme for intermittently connected mobile networks," in *Proceedings of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking (WDTN '05)*, pp. 252–259, ACM, New York, NY, USA, 2005, http://doi.acm.org/10.1145/1080139.1080143.

[32] G. Palla, I. Derényi, I. Farkas, and T. Vicsek, "Uncovering the overlapping community structure of complex networks in nature and society," *Nature*, vol. 435, no. 7043, pp. 814–818, 2005.

[33] F. De Rango, A. Socievole, A. Scaglione, and S. Marano, "Novel activity-based metrics for efficient forwarding over online and detected social networks," in *Proceedings of the 2013 9th International Wireless Communications and Mobile Computing Conference, (IWCMC)*, July 2013.

[34] M. Newman, "Clustering and preferential attachment in growing networks," *Physical Review E-Statistical, Nonlinear, and Soft Matter Physics*, vol. 64, no. 2, pp. 251021–251024, 2003.