Hey, Influencer! Message Delivery to Social Central Nodes in Social Opportunistic Networks

Carlos Borrego[a,*], Joan Borrell[a], Sergi Robles[a]

[a]*Department of Information and Communications Engineering*
Universitat Autònoma de Barcelona
Barcelona, Spain

Abstract

This paper presents a new strategy to efficiently deliver messages to influencers in social opportunistic networks. An influencer node is an important node in the network with a high social centrality and, as a consequence, it can have some characteristics such as high reputation, trustfulness and credibility, that makes it an interesting recipient. Social network analysis has already been used to improve routing in opportunistic networking, but there are no mechanisms to efficiently route and deliver messages to these network influencers. The delivery strategy proposed in this article uses optimal stopping statistical techniques to choose among the different delivery candidate nodes in order to maximise the social centrality of the node chosen for delivery. For this decision process, we propose a routing-delivery strategy that takes into account node characteristics such as how central a node is in terms of its physical encounters. We show, by means of simulations based on real traces and message exchange datasets, that our proposal is efficient in terms of influencer selection, overhead, delivery ratio and latency time. With the proposed strategy, a new venue of applications for opportunistic networks can be devised and developed using the leading figure of social influencers.

Keywords: OppNet, Opportunistic Social Networks, Optimisation, centrality

1. Introduction

The global deployment of computer networks has permitted the creation of a wide range of different social network-based applications. These social applications allow collaborative behaviour among users to share their personal interests or hobbies, for example [17].

As mobile devices get smarter, Opportunistic Networking (OppNet) [18] has emerged as a solid network solution that allows mobile nodes to communicate with each other when no end to end connection is possible.

In the context of OppNet, research is directed towards a new network paradigm that evolves from the traditional node-to-node scheme to a more person-centric one. Under different terms, such as Proximity-Based Applications [46], Mobile Ad Hoc Social Networks (MASNs) [47], Offline Social Networks [49], or SmartPhone Adhoc Networks [49], the research community is pointing at the human social characteristics such as mobility patterns or personal interests, to provide new social applications and to improve networking decisions such as the message routing.

There are many reasons why these opportunistic social networks can be a useful alternative to traditional connected ones. One example for this is to preserve users' privacy. As explained in studies like [44], Internet service providers are using packet inspection techniques to read and store users' messages and personal data. Opportunistic social networks, in contrast with traditional connected ones, can provide their users with a way of socialising without renouncing to their privacy since OppNet nodes do not rely on a network infrastructure that identifies their users.

When OppNet is seen from a social perspective, new network roles come into view. For example, OppNet nodes can be characterised in terms of how socially connected they are. In social networks, an influencer node [31] is an important node in the network with a high social centrality and, as a consequence, can have interesting characteristics such as high reputation, trustfulness, and credibility. Having the possibility of communicating with influencer nodes in an OppNet and make them send a message with a given information may improve the social acceptance of this information. For example, in an emergency scenario with damaged network infrastructure, a critical advise from the emergency coordination to the population will have more chances of being followed when it comes (not just forwarded) from a trusted influencer.

However, finding strongly connected nodes in OppNet is far from being easy to achieve due to OppNet's dynamic change of topology and the lack of a global view of the network [30]. In this article, we propose a routing-delivery mechanism that allows to deliver messages to influencers of a network in an optimal way, statistically speaking.

*Corresponding author

Email addresses: cborrego@deic.uab.cat (Carlos Borrego), joan.borrell@uab.cat (Joan Borrell), sergi.robles@uab.cat (Sergi Robles)

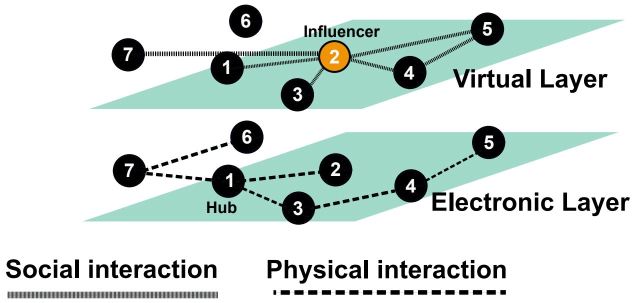


Figure 1: Two levels of the relationship of nodes in an OppNet: electronic and virtual.

The optimality of this mechanism is achieved by making, on one hand, optimal delivery decisions and, on the other, optimal routing decisions. Optimal delivery decisions involves deciding whether a node should be considered an influencer or not, whilst optimal routing decisions deals with deciding the optimal message path to discover influencers. To obtain this optimality, the routing-delivery strategy proposed in this article uses optimal stopping statistical techniques [6] to choose among the different delivery candidate nodes in order to maximise the social centrality of the delivered node. For the routing decision process, we propose a routing algorithm that takes into account other node's characteristics such as their physical centrality, that is, how central a node is in terms of its physical encounters.

The paper starts with all the relevant state of the art information, in Section 2, paying special attention to Opportunistic Social Networks, centrality metrics, and optimal stopping techniques in OppNet. Next, we provide a full description of our routing-delivery strategy in Section 3. The paper follows with Section 4, where a simulation-based experimentation is presented. Finally, Section 5 contains the conclusions we have drawn from this work.

2. Related Work

In this section, we study the state of the art of Opportunistic Social Networks. We make a summary of the different ways of studying the centrality of a node in classical networks and in ego networks. Additionally, a brief survey on articles that use Optimal Stopping statistical techniques for improving OppNet is presented. Finally, we present how to include complex routing and delivery algorithms in OppNet within the messages.

2.1. Opportunistic Social Networks

Opportunistic Networking [18, 51] is a network paradigm where end-to-end connection is not guaranteed. Messages in OppNet use intermediate nodes to be routed from their source to their destination in an opportunistic way.

Social Network Analysis (SNA) has been applied to OppNet to improve routing algorithms [35, 16, 29, 37, 26], focusing on the two key concepts of SNA: community and centrality. Social-based forwarding schemes outperform traditional schemes based on epidemic approaches or mobility-based predictions. Moreover, other proposals [15, 49] consider the existence of two levels of social networks in opportunistic environments. As depicted in Figure 1, there is a first level, an electronic (also called offline) social network, which refers to the physical wireless network in which users mobility and proximity are considered. Secondly, a virtual or online social network that reflects users friendships and influences [49]. That study also encourages the combination of the social awareness in both electronic and virtual social networks, in order to improve the efficiency of new data forwarding protocols in OppNets.

Under different terms such as Proximity-Based Applications [46], Mobile Ad Hoc Social Networks (MASNs) [47], Offline Social Networks [49], or SmartPhone Adhoc Networks [49], the scientific community proposes a natural evolution of OppNet where users, using mobile devices, are connected using opportunistic contacts to form virtual communities of similar personal interests. We use all along this article the term Opportunistic Social Network (OSN) to refer to this type of networks. There are some OppNet studies that propose that social network applications can become a driver for the proliferation of OppNet [28]. OSN is not merely for research: some recent proposals have developed solutions to build local mobile ad-hoc social networks on top of Android-based mobile terminals [48]. These solutions make opportunistic social networks a feasible network paradigm with promising social applications as pointed out in studies like [21].

2.2. Centrality in Opportunistic Networks

When OppNet is seen from a social perspective, as described in the previous section, nodes can be characterised in terms of how socially connected they are. In order to measure this sociability, Graph Theory can be very helpful. In Graph Theory in general, the term centrality identifies the most important vertices within a graph. Users in OppNet in particular, are structured in communities, and inside these communities, there are some more popular and influential than others. This influence is commonly measured using centrality indicators.

Many research studies [40, 41] have proposed to use the centrality of OppNet nodes to influence in the routing decision. The basic idea behind this concept is to choose nodes with high centrality to forward the custodied messages to increase the probability of delivering the message to its destination: centrality defines a node's ability to act as a communication hub.

There a many different centrality metrics proposed in the literature to measure the structural importance of a node in the network. There are many centrality metrics proposed by the scientific community [22]. The three most important are degree centrality, betweenness, and closeness centrality. Degree centrality [39] measures the number of direct edges that reach a given node. Betweenness centrality [2], instead, calculates the number of shortest paths connecting other nodes that use the node being measured as a hop. Finally, closeness centrality [38] studies the length of the shortest path connecting the rest of the nodes.

Additionally, there has been a large effort in proposing social metrics to define network centrality from an egocentric point of view [23]. Ego networking is a network metric where that consists of defining a node together with the nodes to which this node is connected to and the links among those nodes. Some studies [36] show that ego network centrality is highly correlated to traditional centrality. This paradigm is very useful in Ad Hoc networks because of the lack of a central entity capable of calculating nodes' centrality.

2.3. Optimal stopping in OppNet

In this article, we propose a routing-delivery mechanism that allows, in an optimal way, to deliver messages to influencers, that is, to highly social-connected nodes in the network. This problem can be seen as the optimisation problem of deciding when to make a delivery decision. Probabilistically speaking, early delivery decisions will not guarantee optimal results, that is, messages will be delivered to poor socially connected nodes. Late decisions, moreover, will also fail since it is probable that good candidates will be discarded. Optimal Stopping Theory [43] deals exactly with this type of problems. It is a statistical solution for the problem of choosing the best moment to make a particular decision to maximise a certain reward.

One of the most popular problems in Optimal Stopping Theory is the secretary problem [20]. In this problem, a person must interview a group of n candidates, that can be ranked from best to worse, with the aim of selecting the best one. The difficulty of this problem lies in the fact that once a candidate is not selected he/she can not be recalled again. The solution to this problem, as presented in [20], is a selection strategy that discards the first n/e candidates interviewed (being e the mathematical constant) and selects the first one, if found, that is better than all of the previous ones.

There are different extensions of the secretary problem. A complete survey can be found in [24]. In this article, we will use a concrete variation of the secretary problem called the rank-based selection and cardinal payoffs variation of the secretary problem [3, 4]. This variation is more flexible than the traditional problem because it allows the selection of a candidate that is not necessarily the best, while trying to maximise the quality of the candidate. The solution to this problem is a similar strategy to the classical secretary problem: it discards the first $\sqrt{n}$ nodes instead of the first n/e, and then, as in the traditional problem, selects the next one, if found, that is better than all of the previous ones.

There are some proposals in literature that are using optimal stopping in OppNet. In [30], the authors propose to apply optimal stopping theories to make network decisions such as routing or node searching. However, these optimal stopping theories are difficult to be directly implemented in OppNet using traditional protocols because these protocols are not able to implement optimal strategies. The reason for this limitation is that optimal strategies need special delivery conditions that require keeping an internal state of the message with the purpose of remembering preceding events. In [6], the authors present Softwarecast, a general delivery scheme for group communications based on mobile code that helps to solve this limitation. In Softwarecast, messages carry a software code and a delivery state that permit them to perform refined delivery-decision-making methods based on optimal stopping theories to implement complex delivery decisions. Softwarecast uses Active-DTN [8], an OppNet solution that consists in extending the messages being communicated by incorporating software code for forwarding, delivery, lifetime control and prioritisation purposes. Active-DTN has been applied in different DTN scenarios such as opportunistic computing infrastructures [7], multi-application mobile node sensor networks [5] and disconnected emergency scenarios [8].

We base our proposal on the study OppNet Profile-cast scheme [27]. In OppNet Profile-cast, message destinations are users defined by certain profiles. These profiles provide very effective ways of characterizing nodes in terms of nodes' attributes. More concretely, we use a Profile-cast model [9] that allows messages to be sent to profiles defined in terms of relative delivery functions such as *best*, *maximum* or *over-the-average*. Additionally, in [9], the authors present Explore and Wait, a composite routing-delivery scheme that uses Optimal Stopping Theory-based delivery strategies to route messages to these special profiles.

Even though there has been a substantial number of proposals to solve routing and delivery problems in OppNet, there is not yet any proposal that gives a solution to the complex problem of sending messages to highly connected nodes in OppNet. In this article, we open a new avenue for creating new applications in OppNet by proposing a novel routing-delivery strategy that allows messages to be delivered to influencers. This strategy uses optimal stopping delivery techniques strategies on the basis of the studies presented in this section.

3. Influencer message delivery

In this section, we define the influencer message delivery problem and we describe an optimal routing-delivery strategy based on optimal stopping theory to solve this problem.

3.1. Influencer Nodes in OppNet

In network theory, nodes (vertex) are connected to one another in a symmetric or asymmetric way forming connections (edges). In OppNet, as introduced in Section 2,

Variable	Description
vc	Virtual centrality.
ec	Electronic centrality.
ms	Message scope.
$tlimit$	Time limit for message delivery.
iet	Inter-Explore Time.
xp	Number of nodes a message has explored.
$siet$	EWMA of iet.
α	$siet$ EWMA weight factor.
$lastxp$	List of nodes recently explored.
$lastxpvc$	List of nodes considered for the trend evaluation.
mvc	Median of the values in $lastxpvc$.
svc	EWMA of the explored vc.
β	svc EWMA weight factor.
vc_{trend}	The trend of the virtual centrality.

Table 1: Description of the acronyms used in Section 3.

this relationship among nodes can be seen from two different perspectives: virtual and electronic, following the terminology presented in [15].

A first perspective is a virtual one. Two nodes in an OppNet are virtually connected if they virtually interact with each other by, for example, exchanging messages or by explicitly being defined as friends in a given social network. A second perspective is an electronic one. In OppNet, two nodes are electronically connected if they physically come in contact with one another.

There are many proposals in literature for metrics that quantify how strongly connected two nodes are in a network. These metrics can be applied to both virtual and electronic planes [17]. Examples of these metrics include frequency of the virtual/electronic contacts, their longevity, their duration, reciprocity, etc.

An influencer node in OppNet is a node that has a high virtual centrality in comparison to the rest of the nodes from the network. As introduced in Section 1, an influencer node is a special role in the network that has interesting characteristics such as high reputation, trustfulness and credibility. Following, we define the centrality metrics we will use for the virtual and electronic social network.

3.2. Centrality Metrics

We need to define two different centralities metrics, the electronic centrality —how physically connected a node is —and the virtual centrality —how socially connected a node is.

3.2.1. Electronic Centrality Metric

For the electronic centrality, we have chosen to use a degree centrality metric. The degree centrality of a node [39] represents the number of physical encounters a node has made. We define two functions to update the electronic centrality metric. On one hand, the electronic centrality is increased whenever two nodes A and B meet, in the following way:

$$ec_{new_A} = ec_{old_A} + 1 \tag{1}$$

$$ec_{new_B} = ec_{old_B} + 1 \tag{2}$$

This action allows frequently-contacted nodes to have a high electronic centrality.

On the other hand, in order to make this centrality metric more adaptive to dynamical networks such as OppNet, nodes automatically age their values of their electronic centrality in order to decrease the impact of old physical contacts. A node A will update its electronic centrality using an exponentially decaying function in the following way:

$$ec_{new_A} = ec_{old_A} \times \gamma^k \tag{3}$$

where $\gamma \epsilon [0, 1]$ is an ageing constant and k is the number of time units that have passed since the last ageing update.

3.2.2. Virtual Centrality Metric

For the virtual centrality, we have chosen to maintain on every node its ego network. An ego network, as introduced in Section 2, is a network that consists of the single studied node together with the nodes this node is connected to and the links among those nodes. We have chosen to use ego networks to study social node's virtual centrality because it can be performed locally by individual nodes without complete knowledge of the entire network. As explained in [19], this betweenness calculation is very efficient in terms of CPU utilization. Additionally, as introduced in Section 2, some studies show that ego network centrality is highly correlated to traditional centrality [36]. The calculation of all the ego betweenness values for a given network would be one order of magnitude lighter rather than calculating the real betweenness scores.

In an ego network, nodes' virtual contacts are represented by an adjacency symmetric matrix where its rank, n, is the number of messages a given node has sent and received. Note that this virtual metric considers both sending and receiving messages. For these purposes, every node keeps a contact matrix (CM) that represents the node's virtual ego perspective of the network. In this contact matrix, the first row and the first column ($CM_{0,j}$ and $CM_{i,0}$) defines virtual contacts from the local node. Any other generic value $CM_{i,j}$, instead, defines virtual contacts between two nodes i and j. To obtain this, for every message $m(from\colon A, to\colon B)$ sent, received or forwarded, the matrix CM is updated in the following way:

$$CM_{a,b} = 1 \tag{4}$$

where a and b are the row position in matrix CM for nodes A and B.

We propose to use the betweenness egocentric centrality as virtual centrality. This metric is calculated by computing the number of nodes that are indirectly connected through the ego node. This calculation is done by obtaining a new adjacency matrix, (CM'), where every value $CM'_{i,j}$ is set to $CM_{i,j}$ if both $CM_{i,j}$ and $CM_{j,i}$ are set to 1, otherwise $CM'_{i,j}$ is set to 0. Then, the betweenness egocentric centrality is obtained by summing the reciprocals of the following matrix, as introduced in [19]:

$$CM'^2[1 - CM'] \tag{5}$$

3.3. Influencer Message Delivery Problem

The influencer message delivery problem can be formulated as a set of the following constraints:

- The goal of this problem is to deliver a message to a node with a high virtual centrality. The higher this virtual centrality is, the better.
- There is a limited and known time to deliver the message.
- There is an unknown number of nodes in the network.
- At a given time, at the hypothetical situation of having all the nodes in the network together, they can be ordered from best to worst according to their virtual centrality.
- When a node has accepted the custody of a message, it allows the message to explore its virtual centrality and the one from every neighbor not selected for routing.
- As a message is being routed from one node to another, it can be determined the relative ranks of the explored candidate nodes but it cannot be observed the absolute ranks of all of the nodes in the network.
- Once a candidate is rejected, it can be found again in the future.

While in connected networks this delivery action can be very easy to perform since network searches are easily conducted, in OppNet, this is not an easy task to accomplish because of the idiosyncrasy of the network. The search problem in OppNet is quite difficult to conduct, as described in proposals like [30].

This relative decision of delivering or not a message to a contacted node is a similar problem to optimal stopping problems [43]. As introduced in Section 2, optimal stopping theories deal with problems that aim to choose the optimal moment to take a particular action. The solution to these problems are complex strategies that, from a statistical point of view, offer an optimal result.

The influencer message delivery problem is very similar to the standard secretary problem. The main difference is that in the influencer message delivery problem, the payoff obtained when making a decision is equal to the selected node virtual centrality value, whereas in the classical secretary problem it is just 1 if the best overall candidate is selected or 0 otherwise. This problem can be seen as the rank-based selection and cardinal payoffs variation of the secretary problem [4]. This variation of the secretary problem fits better to the influencer message delivery problem than the traditional one because, as explained in Section 2, it includes the flexibility of allowing the selection of a candidate that is not necessarily the best, while trying to maximise the quality of the candidate in terms of its virtual centrality. In the influencer message delivery problem, we do not aim to send the message to the most influential node, but to a node which has a high virtual centrality (not necessarily the highest).

The influencer message delivery problem is extremely challenging. A straightforward solution would be to broadcast an influencer discovery message to the network and afterwards send the influencer message to the most social node found. However, this solution could fail because once the potential influencer is discovered, it could not receive the influencer message due to the low delivery ratios obtained in OppNet. Besides, this solution would have a high network overhead.

Instead, if we allow just one message to perform both the exploration and the delivery action some other problems may be found. First of all, the number of candidate nodes a message can explore in a period of time is difficult to estimate. Additionally, nodes have a dynamic value for their virtual centrality since it changes overtime. Also, influencer messages can be forwarded to one node more than once. Finally, the exploration action implies the necessity of storing information within the message that should be light in terms of storage and accurate in terms of the exploration.

In the following two sections, we explain our strategy to solve the influencer message delivery problem that covers all of these issues. We give a network oriented explanation for this strategy. First, in Section 3.4, we explain the message delivery decision process, that is, the decision on which node should be the destination of the influencer message. Secondly, in Section 3.5, we explain the message routing decision process, that is, the decision on whether or not to forward an influencer message to a certain contacted node. In Algorithm 1, the complete strategy is presented.

3.4. Delivery Protocol

In this section, we propose a delivery strategy for the influencer message delivery problem, as defined in the previous section.

The influencer message delivery problem delivery strategy consists of creating a single message that operates in two different phases: the Explore Phase and the Wait Phase, following the delivery scheme terminology presented in [9]. During the Explore phase, the influencer message is routed from a node to another. Messages during this phase are not delivered. Instead, the maximum value retrieved for the virtual centrality is kept within the message. This phase ends when the message has explored more than a certain number of nodes. During the Wait

Phase, the message will still be routed from one node to another, and it will be delivered to the first node found that has a greater virtual centrality than all of the previous ones.

The key issue in this delivery problem, as in all optimal stopping problems, is to decide when to stop exploring the network and switch to the Wait phase. As introduced in Section 2, we follow the variation of the secretary problem called the rank-based selection and cardinal payoffs variation [4]. This variation of the secretary problem considers $\sqrt{n}$ as the optimal value for the phase transition, being n the number of candidates. However, in OppNet, the number of nodes a message can explore in a given time can not be known in advance.

Accordingly, our optimal delivery strategy for the influencer message delivery problem is the following:

- Within the message, keep:
  - xp, the number of already explored nodes (updated in line 14 of Algorithm 1).
  - ms, an estimation of the total number of nodes this message could explore in the remaining time to be delivered (updated in line 7 of Algorithm 1).
  - $maxvc$, the maximum virtual centrality (vc) explored so far (updated in lines 8 and 9 of Algorithm 1).
- Allow the influencer message to be routed from node to node following the routing protocol that will be presented in Section 3.5 and reject for delivery the first $\sqrt{xp + ms}$ nodes a message has explored (line 19 of Algorithm 1).
- Select the first node (if found) that is better than all of the previous explored nodes in terms of its virtual centrality (line 20 of Algorithm 1).

In order to calculate the estimation of the number of nodes an influencer message can explore at a give time, a statistic of the time between explored nodes is kept within the message. We call this statistic the Smooth Inter-Explore Time ($siet$). To prevent recently forwarded nodes to distort this estimation, messages keep a list of the last nodes the message has explored. We call this list *last explored node list (lastxp)*. In Section 4, the size of this list is studied to understand its impact on the influencer delivery performance.

Every time a node containing an influencer message contacts another node not included in the $lastxp$ list, the observed Inter-Explore Time (iet) is calculated (line 5 of Algorithm 1):

$$iet = time.now() - lastexploretime, \tag{6}$$

where $time.now$ is the current time and $lastexploretime$ is the last time the influencer message explored a node.

Input Variable	Description
msg	Message to be delivered.
$tlimit$	Time limit for message delivery.
$swapmetric()$	A function that changes a metric from electronic to virtual and *vice versa*.
$getvc()$	A function returning the virtual centrality.
$getec()$	A function that returns the electronic centrality.
$neighbours$	The list of current neighbour nodes.
$lastxpT$	A message state variable with the time of the last explored node.
$lastxp$	A fixed-size unique-value LIFO array with the last explored nodes.
$lastxpvc$	A fixed-size LIFO array with the last vc values of explored nodes.
xp	The number of explored nodes.
$CurrentMetric$	The routing metric to use.
$maxvc$	The maximum vc explored.
α,β	$siet$ and svc EWMA weights.

Table 2: Description of the input variables from Algorithm 1.

Algorithm 1 Routing-Delivery strategy.

```
1:  procedure SENDTOINFLUENCER
2:      for all remote in neighbours do
3:          svc = svc + β × (svc - getvc(remote))
4:          mvc = median(lastxpvc)
5:          iet = getTime - lastexploretime
6:          siet = siet + α × (iet − siet) , α ∈ [0, 1]
7:          ms = (tlimit − getTime )/siet          ▷ nodes to explore
8:          if getvc(remote) > maxvc then
9:              maxvc=getvc(remote)                ▷ Update maxvc
10:         end if
11:         if mvc < svc then        ▷ negative trend, swap metric
12:             swapMetric(CurrentMetric)
13:         end if
14:         xp = xp + 1
15:         lastxpT = getTime()
16:         lastxp.add(local)
17:         lastxpvc.add(getvc(local))
18:     end for
19:     if xp > √(xp + ms) and getvc(local) > maxvc then
20:         deliver(msg); exit                 ▷ End of Wait phase
21:     else
22:         for all remote in neighbours do
23:             if CurrentMetric == electronic then
24:                 if getec(remote) > getec(local) then
25:                     forward(msg,remote); break
26:                 end if
27:             else if getvc(remote) > getvc(local) then
28:                 forward(msg,remote); break
29:             end if
30:         end for
31:     end if
32: end procedure
```

The *siet* statistic is updated using an EWMA (Exponential Weighted Mobile Average) in the following way (line 6 of Algorithm 1):

$$siet_{new} = siet_{old} + \alpha \times (iet - siet_{old}), \alpha \in [0, 1], \quad (7)$$

where $siet_{old}$ is the historical iet, α is a weight factor and iet the last iet time measured.

Finally, the estimation of the number of nodes an influencer message could explore in a given time t is calculated using the following expression (line 7 of Algorithm 1):

$$ms = t/siet. \quad (8)$$

3.5. Routing Protocol

In this section, we introduce a routing algorithm that makes use of both electronic and virtual centrality metrics to drive information in social OppNets toward nodes with a high virtual connectivity (influencers).

This algorithm uses the electronic centrality (in what extent a node is physically connected to others) and the virtual centrality (in what extent a node is socially connected to others). One may think that because the sought destination (the influencer) is the best social connected node in the network, exclusively taking into consideration the virtual centrality would suffice for the sake of the routing. Unfortunately, virtual and electronic centrality are not necessarily correlated. A physically highly-connected node is a good candidate to forward messages to other nodes (whatever their virtual connectivity, which might be null). On the other hand, a socially highly-connected node looks like a good candidate to forward messages towards influencers, although this last forwarding might not take place due to a lack of physical connectivity. Therefore, an on-the-fly detection of the current trend towards nodes with high virtual centrality is going to be considered. If there is an upward trend of the virtual centrality (the message is progressing towards an influencer), the routing algorithm will keep using the same centrality (virtual or electronic) as a selection criterion for choosing the next forwarding node. Otherwise, it will use the other centrality (the trend has to be reverted to get closer to influencers).

To make this routing possible, there are three required elements: the current virtual centrality in a node, its electronic centrality, and the general trend of virtual centrality considering all nodes, updated periodically and stored in the message. The algorithm will use these three values to make the routing decision for a particular message.

The electronic centrality of a node in a given time can be easily calculated, for example, by counting how many neighbour nodes are at physical reach during a window of time. Every node keeps this metric, and can be accessed from the routing algorithm. The virtual centrality of a node at a given time can be worked out by ascertaining the number of social contacts during the same window of time.

The trend of the virtual centrality, vc_{trend}, indicates whether the tendency of this metric along the journey of a message is upward (in this case the trend value is 1), or downward (trend value of -1). To calculate this trend, the algorithm uses two variables: another EWMA of the virtual centralities of the explored nodes that we call the smooth virtual centrality (*svc*) and the median of the metric considering the last n updates (*mvc*). These two values are stored in the message, and are updated every time step by the node, as seen in lines 3-4 and 12 of Algorithm 1:

$$svc_{new} = svc_{old} + \beta \times (vc - svc_{old}), \beta \in [0, 1] \quad (9)$$

$$mvc = median(vc_t, vc_{t-1}, vc_{t-2}, \ldots, vc_{t-n}) \quad (10)$$

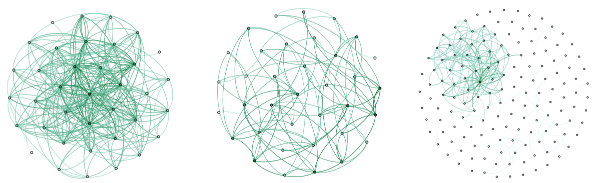
$$vc_{trend} = \begin{cases} 1 & svc \leqslant mvc \\ -1 & svc > mvc \end{cases} \quad (11)$$

We define *currentmetric* as an element in the set $electronic, virtual$. It is first intitialised to *virtual* and will be changed when vc_{trend} is negative.

In the routing process, a neighbour is selected for forwarding if its ec or vc, depending on the *currentmetric*, is higher than the local one. This routing decision can be seen in lines 23-28 of Algorithm 1.

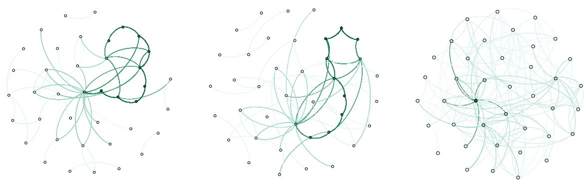
4. Evaluation

The routing-delivery strategy presented in Section 3 is our approach for delivering messages to socially influencers in an OppNet. We have conducted several experiments using simulations based on real mobility traces and real social interaction behaviour in three different scenarios. In these scenarios, we prove that our approach, in comparison to state of the art proposals, improves the influencer grade of the selected node for delivery, the network latency, and the delivery ratio. We claim that our approach is highly performant to solve the influencer message delivery problem in OppNet as presented in Section 3.3. In this section, we describe the evaluation experimentation in detail.



(a) Info5. (b) Cambridge. (c) Taxis.

Figure 2: Electronic social network for Info5, Cambridge and Taxis scenarios.



(a) Info5. (b) Cambridge. (c) Taxis.

Figure 3: Virtual social network for Info5, Cambridge and Taxis scenarios.

4.1. Simulation Environment

The experiment has been conducted over the Opportunistic Network Environment (TheONE) simulator [32]. For the sake of simulation accuracy, we have performed the simulations presented in this sections following the behaviour and performance of a real Bundle Protocol implementation capable of reproducing the influencer delivery strategy. We base our simulations on Active-DTN [8], an OppNet solution[1] that consists in an extension of the Bundle Protocol where the messages being communicated incorporate software and message state for forwarding, delivery, lifetime control and prioritisation purposes.

In the following simulations, the routing-delivery strategies presented in Section 3 are included in messages as bundle forwarding and/or delivery extensions following the bundle extension introduced in [8]. We simulate messages sent to influencers as messages that carry a delivery extension with a software mobile code. The simulator is programmed to allow messages, when carrying the delivery extensions, to be delivered in terms of mobile software codes that are included in these delivery extensions.

As introduced in Section 3, there are several variables that the message must keep to implement the Influencer delivery strategy: the Smooth Inter-Explore Time (*siet*), the last explored node list (*lastxp*), the number of explored nodes (*xp*) and the last *vc* values explored values list (*lastxpvc*). These variables must be kept when being forwarded from one node to another and represent historical information that allows our delivery strategy to be capable of performing appropriately upon arriving at new nodes. For these purposes, we have adapted the simulator to extend its messages to include these variables.

In Table 3, details related to Active-DTN considered for the simulations such as the mobile code extension size, delivery mobile code compilation and execution time are listed.

4.2. Simulation Model and scenarios

We have chosen three different scenarios to analyse the performance of our proposal. The electronic social network, as defined in Section 3, of all proposals are defined by physical contacts obtained from real mobility traces from the Crawdad database[2], a community resource for collecting wireless data at Dartmouth College, United States.

General	**Value**
Transmission Speed	250Kb/s
Transmit Range	10-50m
# Nodes (Info5)	41
# Nodes (Cambridge)	51
# Nodes (Taxis)	370
# Messages	1000
# random seeds	50
Buffer Size	1GB
Electronic layer	
Trace duration (Info5)	2.97 days
Trace duration (Cambridge)	6 days
Trace duration (Taxis)	30 days
# Contacts (Info5)	22459
# Contacts (Cambridge)	10641
# Contacts (Taxis)	449226
Virtual layer	
Message Payload	100 KB
# Message Events	18600
tlimit	20’-180’
Active-DTN Settings	
Mobile Code Extension Size	300 bytes
Compilation Time	54884 ns
Delivery Execution Time	360 ns
Routing/Delivery Settings	
γ (electronic ageing constant)	0.98
Time unit size	30 s
α (*siet* EWMA weight factor)	0.75
β (*svc* EWMA weight factor)	0.75

Table 3: Simulation settings default values for all scenarios.

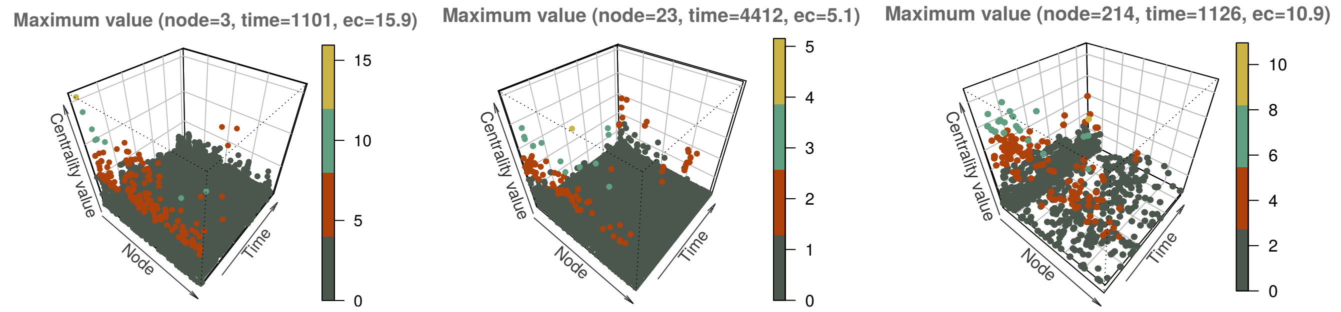
The first scenario, the Info5 scenario, is based on real mobility traces, as published in [42]. These traces were retrieved during the 2005 edition of the Infocom conference in the course of 2.97 days. Contacts from this mobility traces represent 41 students carrying *iMote* platforms . The total of physical encounters provided in these traces is 22459.

The second scenario, the Cambridge scenario, is based on real Bluetooth traces from students from the System Research Group of the University of Cambridge carrying small devices for six days [33]. Additionally, some stationary nodes were placed in various points of interest such as grocery stores, pubs, market places, and shopping centers all around the city of Cambridge, UK. A stationary device was also placed at the reception of the Computer Lab, in which most of the experiment participants were students. The number of contacts provided in these traces is 10641.

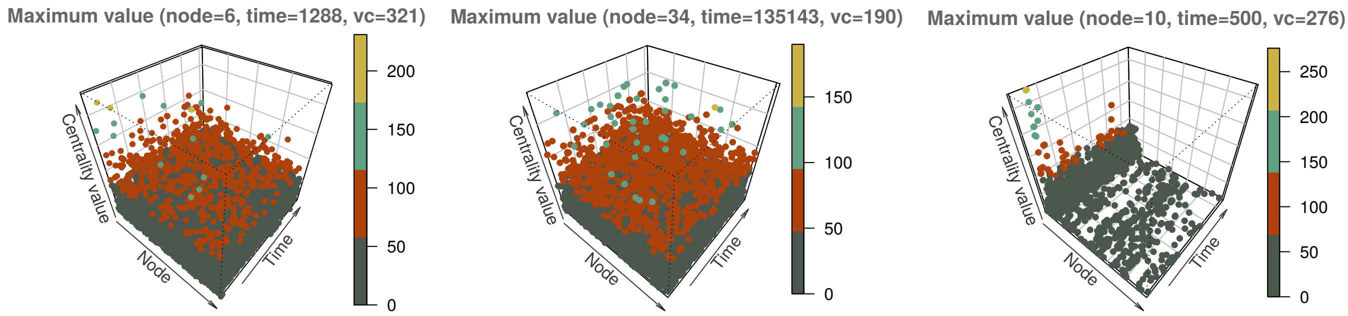
Info5 and Cambridge are long-established scenarios which have been used in many papers in the area of Op-

[1]Active-DTN source code can be found at https://github.com/SeNDA-UAB/aDTNPlus.

[2]*http://crawdad.org/keyword-DTN.html.*



(a) Info5: electronic centrality. (b) Cambridge: electronic centrality. (c) Taxis: electronic centrality.



(d) Info5: virtual centrality. (e) Cambridge: virtual centrality. (f) Taxis: virtual centrality.

Figure 4: Evolution of the electronic centrality (contact degree centrality) and virtual centrality (betweenness ego centrality) of the different nodes in the network every time two nodes connect as a function of the simulation time for the scenarios Info5, Cambridge and Taxis, respectively.

portunistic Networking to compare the performance of different algorithms. Nevertheless, there are other more recent and realistic scenarios which consider larger temporal and spatial scales, and with a larger number of participants, which are ordinary people instead of volunteer students. An interesting example is the mobile phone location dataset provided by the company Airsage to the MIT's SENSEable City lab [13], which contains 200 million anonymous location measurements of one million mobile phones, collected in 2009 in East Massachusetts. From this dataset, [11] constructs a scenario of 1900 users in a Boston area of 150 km^2 during 29 days of October 2009, very interesting to explore the human mobility properties from a networking point of view. Also from this dataset, [12] focuses on the Boston Independence Day Celebration on the evening of July 4th 2009, in the Boston city area (15 km^2, 700 users), as a crowded event where to analyse the feasibility of disseminating emergency information among the mobile phones in absence of network infrastructures.

Therefore, in addition to Info5 and Cambridge, we analyse the performance of our proposal on a third scenario, which is larger in time, space and number of participants, and which is also publicly available from the Crawdad database. The *Taxis* scenario, as published in [10], contains mobility traces from the ordinary activity of 370 taxi cabs in the city centre of Rome over 30 days (February-March 2014), covering an area of 64 km^2. The number of contacts provided in these traces is 449226.

All three scenarios run applications with the same application traffic based on real message exchange from real communication datasets taken from the Stanford Large Network Dataset Collection[3]. We have simulated application messages that define the virtual social network, as explained in Section 3, with real dataset collections. The virtual social network was generated based on anonymised real message exchange from different users belonging to a large European research institution as presented in [34]. These real message exchange social network has been adapted to be integrated into the three above-mentioned scenarios[4].

In Figure 2(a), Figure 2(b), Figure 2(c), Figure 3(a), Figure 3(b), and Figure 3(c) the electronic and virtual social network for all scenarios are represented using the Fruchterman-Reingold [25] algorithm to improve the placement of neighbouring nodes.

In Figure 4(a), Figure 4(b) and Figure 4(c) we show the evolution of the electronic centrality (contact degree centrality) of the different nodes in the network as a function of the simulation time for the scenarios Info5, Cambridge and Taxis, respectively. As it can be seen, the overall electronic social network, from the point of view of the nodes, is different from a scenario to another. In the Info5 scenario the contacts are more frequent, and the electronic

[3]Datasets https://snap.stanford.edu/data/

[4]TheOne Events for this Datasets can be found at *https://senda.uab.cat/wiki/aDTN* in Section "TheOne resources".

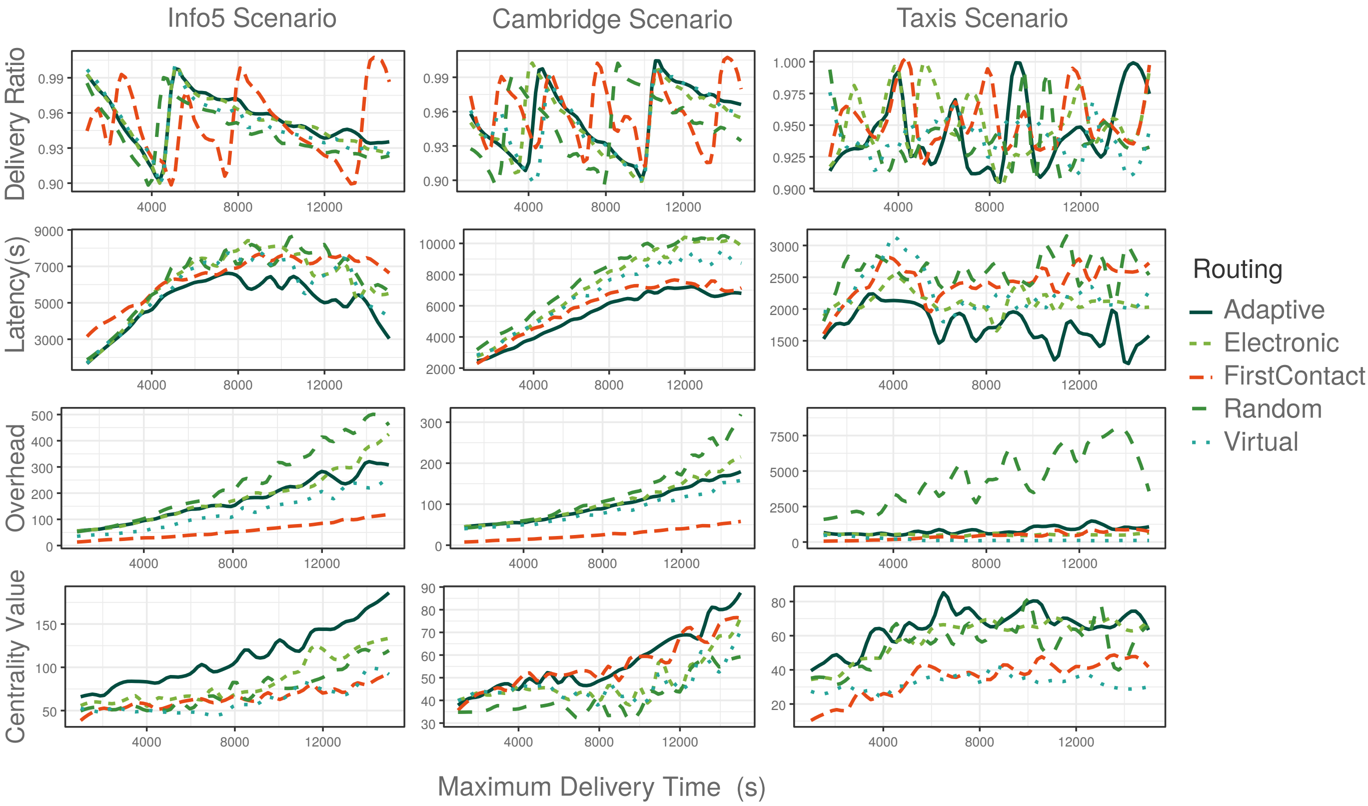


Figure 5: Routing performance for the metrics delivery ratio, latency time, overhead and centrality value for the scenarios Info5, Cambridge and Taxis using different routing algorithms: random routing, first contact routing, electronic routing, virtual routing and adaptive routing.

centrality values are very dynamical. In the Cambridge scenario the contacts are less frequent, and the electronic centrality values do not vary so much. Finally, in Taxis we can see the diversity that comes from such a complex scenario.

Additionally, in Figure 4(d), Figure 4(e) and Figure 4(f) we show the evolution of the betweenness social ego centric centrality of the different nodes in the network as a function of the simulation time for the scenarios Info5, Cambridge and Taxis, respectively. In this case, even if the social interaction defined for the three scenarios follows the same social trace, it can be seen that the social centrality perception is different for the three scenarios.

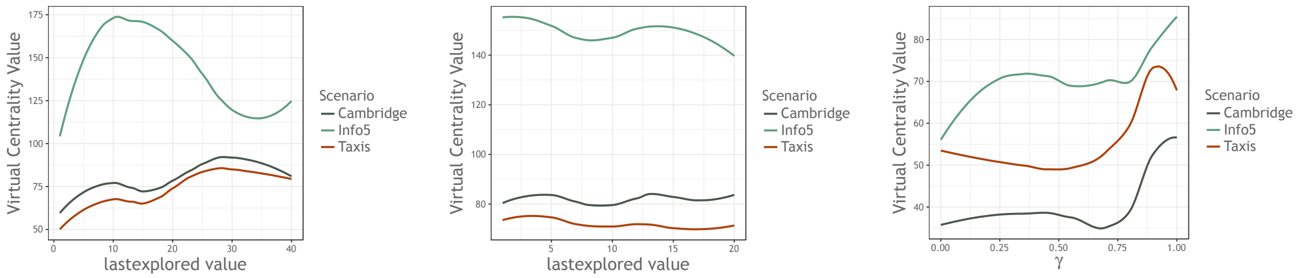
4.3. Experimentation Results

In this section, we describe the results of the different simulations performed. We have included two types of experimentations: routing experimentation and delivery experimentation.

4.3.1. Routing Experimentation Results

We have performed a series of simulations where we have studied the performance of our routing protocol. As presented in Section 3.5, our routing protocol aims to forward messages delivered to influencers. The following experimentations compare five different routing protocols: our routing proposal, three well-known ones and a random protocol. These routing protocols are:

- Our routing protocol presented in Section 3.5 (Adaptive Routing in the figures). If there is an upward trend of the virtual centrality, the routing algorithm will keep using the same centrality as a selection criterion for choosing the next forwarding node and it will user the other centrality, otherwise.
- Virtual Routing. Messages are forwarded to a contacted node if its virtual centrality is bigger than the custodian of the message. This routing behaviour has been used in recent proposals like [6].
- Electronic Routing. Messages are forwarded to a contacted node if its electronic centrality is bigger than the custodian of the message. This routing behaviour has been applied in many proposals such as [1] following the principle that hubs are good candidates for message relays.
- First Contact Routing. A very popular routing protocol used in studies like [50] where messages are forwarded to the first available contact for routing.
- Random Routing. Messages are forwarded to a contacted node following a random criterion as used in [45].

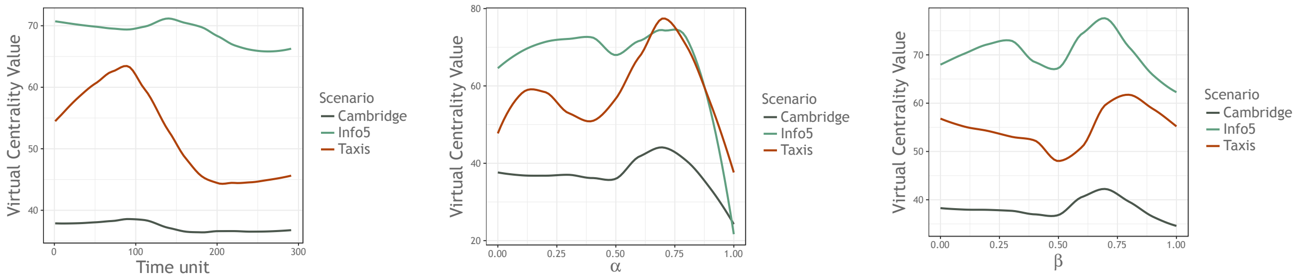


(a) Impact of the size of the last explored node list (*lastxp*).

(b) Impact of the size of the last virtual centrality updates (*lastxpvc*).

(c) Impact of γ *ec* ageing constant.

Figure 6: *lastxp*, *lastxpvc* and γ impact on the virtual centrality performance.



(a) Impact of the time unit size for *ec* ageing.

(b) Impact of α (*siet* EWMA weight factor).

(c) Impact of β (*svc* EWMAweight factor).

Figure 7: Time unit size, α and β impact on the virtual centrality performance.

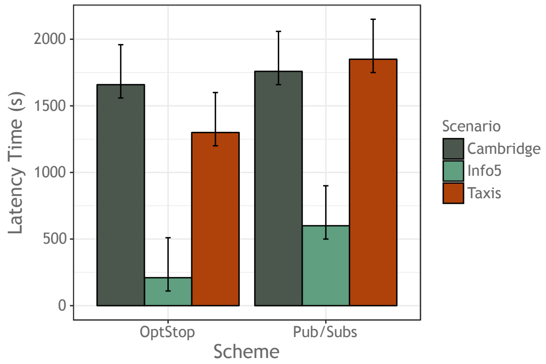
We have simulated the behaviour of 1000 messages sent to influencers in the Info5, Cambridge and Taxis scenarios. We study in this experimentation the virtual centrality value obtained, the delivery ratio, the latency time and its overhead. In Figure 5, the results for these experimentations are depicted. As it can be seen, our proposal, for all scenarios, improves the virtual centrality value obtained and the latency time. Messages for the four routing algorithms have very similar and very high delivery ratio performance. The reason for this high delivery ratio not related to any routing scheme lies in the fact that Active-DTN, the OppNet solution in which the simulations are based on, is capable of evaluating the delivery conditions of the custodied messages even if the OppNet node has not contacted any new node. As a consequence of this, the delivery strategy, independently from the routing protocol, will eventually decide to locally deliver the message, even if there are no new contacts. As it can be seen in Figure 5, messages are almost always delivered. The ones that they are not delivered, then, are the ones being dropped. Finally, concerning the overhead, the best results are obtained by the First Contact routing protocol. Our routing protocol instead has a very similar overhead performance to the Virtual Router and Electronic Router.

In Figure 6(a), we show the impact of the last explored node list (*lastxp*) size on the virtual centrality value obtained on average. As it can be seen, for the three scenarios, the evolution of this performance is similar. However, for the Info5 scenario this impact is much more clear than for the other two scenarios.

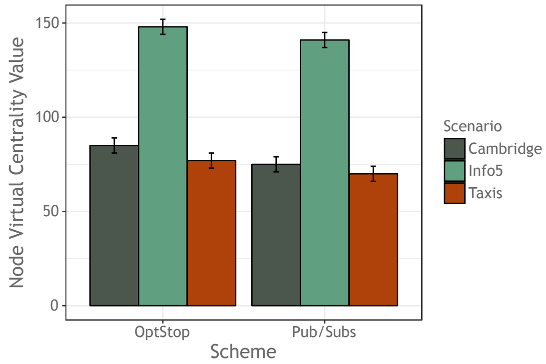
Additionally, in Figure 6(b), we show the impact of the size of the list of the nodes considered for trend evaluation (*lastxpvc*) on the virtual centrality value obtained on average. We see that, for the three scenarios, the maximum value on average for the virtual centrality is obtained when this list has a small size. However, for the Cambridge and the Taxis scenarios, this variable has a smaller effect on the performance.

Moreover, in Figure 6(c) and Figure 7(a), we study the performance impact of γ, the electronic centrality ageing constant and the size of its time unit, as presented in Equation 3 from Section 3.2.1. As it can be seen in Figure 6(c), the best performances for the three scenarios are obtained when γ values are greater than 0.9. Concerning the size of the time unit, there is no general value that fits well for the three scenarios. However, only in the Taxis scenarios, this parameter seems to have a high impact on the virtual centrality value obtained, on average.

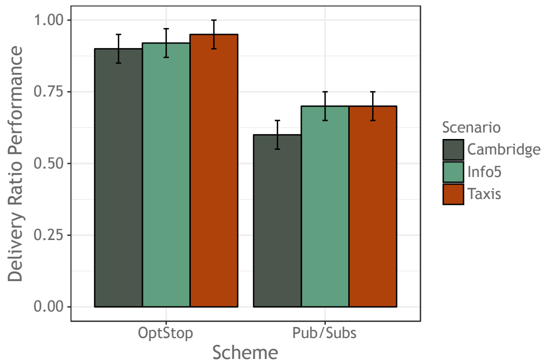
Finally, we study in Figure 7(b) and 7(c), the impact of the two EWMA weight factors for *siet* and *svc* (α and β). As it can be seen in these figures, giving a weight of 70% to new measured information and the remaining 30% to the old measured one seems to be a good compromise value that performs well for the three scenarios and both exponential moving averages.



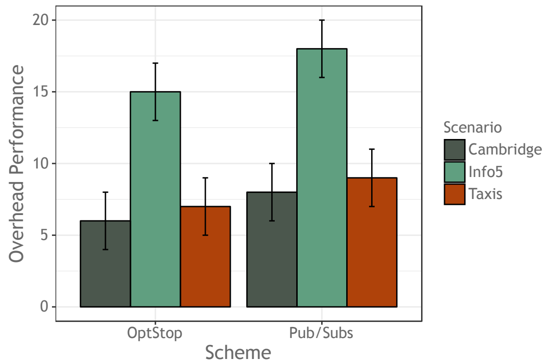
(a) Latency average.



(b) Virtual centrality value average.



(c) Delivery ratio average.



(d) Network overhead average.

Figure 8: Delivery Performance of 1000 influencer messages in 4 hours of time for two different strategies: Influencercast and Publish/Subscribe strategies.

4.3.2. Delivery Experimentation Results

As far as we know, this paper is the only publication that studies the problem of sending messages to influencers in OppNet. However, we still wanted to compare our proposal with another scheme that could also implement this delivery problem. We compare our proposal (*OptStop* in the figures) with a publish/subscribe paradigm called the Onside algorithm [14]. This proposal is a message dissemination algorithm that takes advantage of node's social connections, its interests and the history of contacts, in order to decrease congestion and required bandwidth. In this publish/subscribe paradigm, nodes willing to send messages to influencers are subscribed to virtual centrality values from other nodes that have already published their virtual centrality values. Messages sent to influencers are sent to nodes whose centrality is the highest received.

We analyse the latency time, delivery ratio and virtual centrality value obtained as a function of the maximum time for delivery (*tlimit*) when 1000 messages to influencers are sent to the two different scenarios using the two different ways of solving the influencer message delivery problem. As it can be seen in Figure 8(a), Figure 8(b), Figure 8(c) and Figure 8(d), our strategy performs better than the publish/subscribe one in terms of latency time, delivery ratio, virtual centrality value obtained on average and network overhead.

5. Conclusions

To pave the way for new social applications in OppNet, in this paper, we analysed the possibility of contacting influencer nodes in such networks. Finding strongly connected nodes in OppNet is far from being easy to achieve due to OppNet's dynamic change of topology and the lack of a global view of the network. Therefore, we had to use optimal stopping statistical techniques to devise a novel routing-delivery strategy to allow messages to be delivered to influencers in OppNet.

We firstly formulated the influencer message delivery problem, whose goal is to deliver a message to a node with a high virtual centrality, along with all its restrictions. In particular, the fact that there is an unknown number of nodes in an OppNet. Our strategy to solve the influencer message delivery problem follows an adaptation of the variation of the well-established secretary problem known as the rank-based selection and cardinal payoffs.

Our solution to this problem operates in in two different phases: the Explore Phase and the Wait Phase. In the Explore Phase, the influencer message is routed from node to node while estimating the number of nodes this message could explore. Our routing is adaptive as it takes into account three different elements: both the virtual and electronic centralities in each explored node, and the general trend of the virtual centrality considering the last explored nodes. After exploring the first square root of the estimated number of nodes for exploration, our strategy enters the Wait Phase, where the message is delivered to the first node that is better than all the previous explored nodes in terms of its virtual centrality.

We evaluated our strategy using three scenarios based on real mobility traces with real message exchange from real communication datasets. In our experimentation we proved that our routing protocol performs better than state of the art ones in terms of latency time and virtual centrality obtained with very similar delivery ratios. Additionally, we compared our general strategy with a publish/subscribe scheme. The simulation results showed that our proposal performs better than the publish/subscribe one in terms of latency time, virtual centrality value, delivery ratio and network overhead.

Acknowledgements

This work was partly supported by the Catalan AGAUR 2017 SGR 00463 project and the Spanish Ministry of Science and Innovation TIN2017-87211-R project.

References

[1] Shabbir Ahmed and Salil S Kanhere. HUBCODE: hub-based forwarding using network coding in delay tolerant networks. *Wireless Communications and Mobile Computing*, 13(9):828–846, 2013.

[2] Marc Barthelemy. Betweenness centrality in large complex networks. *The European Physical Journal B-Condensed Matter and Complex Systems*, 38(2):163–168, 2004.

[3] MohammadHossein Bateni, Mohammadtaghi Hajiaghayi, and Morteza Zadimoghaddam. Submodular secretary problem and extensions. *ACM Transactions on Algorithms (TALG)*, 9(4):32, 2013.

[4] J Neil Bearden. A new secretary problem with rank-based selection and cardinal payoffs. *Journal of Mathematical Psychology*, 50(1):58–59, 2006.

[5] Carlos Borrego, Sergio Castillo, and Sergi Robles. Striving for sensing: Taming your mobile code to share a robot sensor network. *Information Sciences*, 2014.

[6] Carlos Borrego, Gerard Garcia, and Sergi Robles. Softwarecast: A code-based delivery manycast scheme in heterogeneous and opportunistic ad hoc networks. *Ad Hoc Networks*, 55:72–86, 2017.

[7] Carlos Borrego and Sergi Robles. A store-carry-process-and-forward paradigm for intelligent sensor grids. *Information Sciences*, 222(0):113 – 125, 2013.

[8] Carlos Borrego, Sergi Robles, Angela Fabregues, and Adrián Sánchez-Carmona. A mobile code bundle extension for application-defined routing in delay and disruption tolerant networking. *Computer Networks*, 87:59–77, 2015.

[9] Carlos Borrego, Adrián Sánchez-Carmona, Zhiyuanand Li, and Sergi Robles. Explore and wait: A composite routing-delivery scheme for relative profile-casting in opportunistic networks. *Computer Networks*, 123:51–63, 2017.

[10] Lorenzo Bracciale, Marco Bonola, Pierpaolo Loreti, Giuseppe Bianchi, Raul Amici, and Antonello Rabuffi. CRAWDAD dataset roma/taxi (v. 2014-07-17). Downloaded from https://crawdad.org/roma/taxi/20140717, July 2014.

[11] Angela Sara Cacciapuoti, Francesco Calabrese, Marcello Caleffi, Giusy Di Lorenzo, and Luigi Paura. Human-mobility enabled networks in urban environments: Is there any (mobile wireless) small world out there? *Ad Hoc Networks*, 10(8):1520 – 1531, 2012. Special Issue on Social-Based Routing in Mobile and Delay-Tolerant Networks.

[12] Angela Sara Cacciapuoti, Francesco Calabrese, Marcello Caleffi, Giusy Di Lorenzo, and Luigi Paura. Human-mobility enabled wireless networks for emergency communications during special events. *Pervasive and Mobile Computing*, 9(4):472 – 483, 2013.

[13] F. Calabrese, G. Di Lorenzo, and C. Ratti. Human mobility prediction based on individual and collective geographical preferences. In *13th International IEEE Conference on Intelligent Transportation Systems*, pages 312–317, Sept 2010.

[14] Radu-Ioan Ciobanu, Radu-Corneliu Marin, Ciprian Dobre, Valentin Cristea, and Constandinos X Mavromoustakis. ONSIDE: Socially-aware and interest-based dissemination in opportunistic networks. In *Network Operations and Management Symposium (NOMS), 2014 IEEE*, pages 1–6. IEEE, 2014.

[15] Marco Conti, Mohan Kumar, et al. Opportunities in opportunistic computing. *Computer*, 43(1):42–50, 2010.

[16] Elizabeth M Daly and Mads Haahr. Social network analysis for routing in disconnected delay-tolerant manets. In *Proceedings of the 8th ACM international symposium on Mobile ad hoc networking and computing*, pages 32–40. ACM, 2007.

[17] Elizabeth M Daly and Mads Haahr. Social network analysis for information flow in disconnected delay-tolerant manets. *IEEE Transactions on Mobile Computing*, 8(5):606–621, 2009.

[18] Mieso K Denko. *Mobile Opportunistic Networks: Architectures, Protocols and Applications*. CRC Press, 2016.

[19] Martin Everett and Stephen P Borgatti. Ego network betweenness. *Social networks*, 27(1):31–38, 2005.

[20] Thomas S Ferguson. Who solved the secretary problem? *Statistical science*, pages 282–289, 1989.

[21] Mohamed Amine Ferrag, Leandros Maglaras, and Ahmed Ahmim. Privacy-preserving schemes for ad hoc social networks: A survey. *arXiv preprint arXiv:1610.06095*, 2016.

[22] Linton C Freeman. Centrality in social networks conceptual clarification. *Social networks*, 1(3):215–239, 1978.

[23] Linton C Freeman. Centered graphs and the structure of ego networks. *Mathematical Social Sciences*, 3(3):291–304, 1982.

[24] PR Freeman. The secretary problem and its extensions: A review. *International Statistical Review/Revue Internationale de Statistique*, pages 189–206, 1983.

[25] Thomas MJ Fruchterman and Edward M Reingold. Graph drawing by force-directed placement. *Software: Practice and experience*, 21(11):1129–1164, 1991.

[26] Wei Gao, Qinghua Li, Bo Zhao, and Guohong Cao. Social-aware multicast in disruption-tolerant networks. *IEEE/ACM Transactions on Networking (TON)*, 20(5):1553–1566, 2012.

[27] Wei-jen Hsu, Debojyoti Dutta, and Ahmed Helmy. Profile-cast: Behavior-aware mobile networking. In *Wireless Communications and Networking Conference, 2008. WCNC 2008. IEEE*, pages 3033–3038. IEEE, 2008.

[28] Xiping Hu, Terry HS Chu, Victor CM Leung, Edith C-H Ngai, Philippe Kruchten, and Henry CB Chan. A survey on mobile social networks: Applications, platforms, system architectures, and future research directions. *IEEE Communications Surveys & Tutorials*, 17(3):1557–1581, 2015.

[29] Pan Hui, Jon Crowcroft, and Eiko Yoneki. Bubble rap: Social-based forwarding in delay-tolerant networks. *IEEE Transactions on Mobile Computing*, 10(11):1576–1589, 2011.

[30] Esa Hyytiä, Suzan Bayhan, Jörg Ott, and Jussi Kangasharju. Searching a needle in (linear) opportunistic networks. In *Proceedings of the 17th ACM international conference on Modeling, analysis and simulation of wireless and mobile systems*, pages 187–196. ACM, 2014.

[31] Zsolt Katona, Peter Pal Zubcsek, and Miklos Sarvary. Network effects and personal influences: The diffusion of an online social network. *Journal of marketing research*, 48(3):425–443, 2011.

[32] Ari Keränen, Jörg Ott, and Teemu Kärkkäinen. The ONE simulator for DTN protocol evaluation. In *Proceedings of the 2nd international conference on simulation tools and techniques*, page 55. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2009.

[33] Jérémie Leguay, Pan Hui, Jon Crowcroft, James Scott, Anders Lindgren, and Timur Friedman. CRAWDAD dataset upmc/content (v. 2006-11-17). Downloaded from http://crawdad.org/upmc/content/20061117, November 2006.

[34] Jure Leskovec, Jon Kleinberg, and Christos Faloutsos. Graph evolution: Densification and shrinking diameters. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 1(1):2, 2007.

[35] Anders Lindgren, Avri Doria, and Olov Schelén. Probabilistic routing in intermittently connected networks. *ACM SIGMOBILE Mobile Computing and Communications Review*, 7(3):19–20, 2003.

[36] Peter V Marsden. Egocentric and sociocentric measures of network centrality. *Social networks*, 24(4):407–422, 2002.

[37] Mehul Motani, Vikram Srinivasan, and Pavan S. Nuggehalli. Peoplenet: Engineering a wireless virtual social network. In *Proceedings of the 11th Annual International Conference on Mobile Computing and Networking*, MobiCom '05, pages 243–257, New York, NY, USA, 2005. ACM.

[38] Kazuya Okamoto, Wei Chen, and Xiang-Yang Li. Ranking of closeness centrality for large-scale social networks. In *International Workshop on Frontiers in Algorithmics*, pages 186–195. Springer, 2008.

[39] Tore Opsahl, Filip Agneessens, and John Skvoretz. Node centrality in weighted networks: Generalizing degree and shortest paths. *Social networks*, 32(3):245–251, 2010.

[40] Jiho Park, Junyeop Lee, Sun-Kyum Kim, Kiyoung Jang, and Sung-Bong Yang. A forwarding scheme based on swarm intelligence and percolation centrality in opportunistic networks. *Wireless Networks*, 22(8):2511–2521, 2016.

[41] Florin Pop, Radu-Ioan Ciobanu, and Ciprian Dobre. Adaptive method to support social-based mobile networks using a pagerank approach. *Concurrency and Computation: Practice and Experience*, 27(8):1900–1912, 2015.

[42] James Scott, Richard Gass, Jon Crowcroft, Pan Hui, Christophe Diot, and Augustin Chaintreau. CRAWDAD dataset cambridge/haggle (v. 2006-01-31). Downloaded from http://crawdad.org/cambridge/haggle/20060131, January 2006.

[43] Albert N Shiryaev. *Optimal stopping rules*, volume 8. Springer Science & Business Media, 2007.

[44] Sophie Stalla-Bourdillon, Evangelia Papadaki, and Tim Chown. From porn to cybersecurity passing by copyright: how mass surveillance technologies are gaining legitimacy the case of deep packet inspection technologies. *Computer Law & Security Review*, 30(6):670–686, 2014.

[45] Sergio Martínez Tornell, Carlos Miguel Tavares Calafate, Juan-Carlos Cano, and Pietro Manzoni. DTN Protocols for Vehicular Networks: An Application Oriented Overview. *IEEE Communications Surveys and Tutorials*, 17(2):868–887, 2015.

[46] Sacha Trifunovic, Sylvia T Kouyoumdjieva, Bernhard Distl, Ljubica Pajevic, Gunnar Karlsson, and Bernhard Plattner. A decade of research in opportunistic networks: Challenges, relevance, and future directions. *IEEE Communications Magazine*, 55(1):168–173, 2017.

[47] Jie Wu and Yunsheng Wang. *Opportunistic Mobile Social Networks*. CRC Press, 2014.

[48] Daqiang Zhang, Daqing Zhang, Haoyi Xiong, Ching-Hsien Hsu, and Athanasios V Vasilakos. BASA: Building mobile Ad-Hoc social networks on top of android. *IEEE Network*, 28(1):4–9, 2014.

[49] Yanru Zhang, Lingyang Song, Chunxiao Jiang, Nguyen H Tran, Zaher Dawy, and Zhu Han. A Social-Aware Framework for Efficient Information Dissemination in Wireless Ad Hoc Networks. *IEEE Communications Magazine*, 55(1):174–179, 2017.

[50] Haojin Zhu, Suguo Du, Zhaoyu Gao, Mianxiong Dong, and Zhenfu Cao. A probabilistic misbehavior detection scheme toward efficient trust establishment in delay-tolerant networks. *IEEE Transactions on Parallel and Distributed Systems*, 25(1):22–32, 2014.

[51] Haojin Zhu, Xiaodong Lin, Rongxing Lu, Yanfei Fan, and Xuemin Shen. Smart: A secure multilayer credit-based incentive scheme for delay-tolerant networks. *IEEE Transactions on Vehicular Technology*, 58(8):4628–4639, 2009.