IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received August 5, 2019, accepted September 9, 2019, date of publication September 23, 2019, date of current version October 3, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2943080

Employing Social Cooperation to Improve Data Discovery and Retrieval in Content-Centric Delay-Tolerant Networks

CLÁUDIO DIEGO TEIXEIRA DE SOUZA[1], DANIELLE LOPES FERREIRA[2], CARLOS ALBERTO VIEIRA CAMPOS[2], (Member, IEEE), ANTONIO CARLOS DE OLIVEIRA JÚNIOR[3,4], KLEBER VIEIRA CARDOSO[3], AND WALDIR MOREIRA[4]

[1]Federal University of Rio de Janeiro (UFRJ), Rio de Janeiro 21941-901, Brazil
[2]Federal University of the State of Rio de Janeiro (UNIRIO), Rio de Janeiro 22290-240, Brazil
[3]Federal University of Goiás (UFG), Goiânia 74690-900, Brazil
[4]Fraunhofer Portugal AICOS, 4200-135 Porto, Portugal

Corresponding author: Cláudio Diego Teixeira de Souza (disouza@cos.ufrj.br)

This work was supported in part by the Portugal 2020, in part by the European Regional Development Fund (ERDF) from European Union through the project Symbiotic technology for societal efficiency gains: Deus ex Machina (DEM), under Grant NORTE-01-0145-FEDER-000026, in part by the Rede Nacional de Ensino e Pesquisa (RNP), Brazil, and in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), Brazil.

ABSTRACT The paradigm of Content-Centric Networks (CCN) has been used in several networking application contexts in order to enhance the performance of data forwarding and retrieval. By disassociating device addressing from the packet routing process, CCN offers new message routing possibilities in network scenarios that are limited in this respect. An example of such scenarios concerns Delay/Disruption Tolerant Networks (DTNs), in which requesting and retrieving data and services may be even more unpredictable than simply sending any message from a source to a destination. With this in mind, we propose a DIscovery and REtrieval protocol based on Social Cooperation (DIRESC), designed specifically to perform data request and retrieval by applying the concept of CCN in the DTN scenario, an environment also known as Content-Centric Delay-Tolerant Networks (CCDTN). Our forwarding mechanism proposes the distributed sending of data based on the profile of each node, which is defined by its social interactions with other nodes. To evaluate our protocol, we used the Opportunistic Network Environment (ONE) simulator to compare DIRESC to other related CCDTN protocols, including our conceptual implementation of content-centric forwarding dubbed CCN broadcast. Results show that social cooperation has positive performance impact (i.e., improved delivery rate, delay, overhead, packet retransmission, and energy consumption) when compared to other social-oblivious CCDTN protocols.

INDEX TERMS Social cooperation, CCDTN, routing.

I. INTRODUCTION

By focusing on the data/content itself, alternative networking solutions for data forwarding/retrieval has gained the attention of the scientific community, such as Content-Centric Communications [1], Information-centric Networks (ICN), Named Data Networking (NDN) [2] from the NFD prototype [3] and the well-known and established Content-Centric Networking (CCN) architecture [4] from the CCNx project [5]. Moreover, considering environments where long delays and frequent disconnections are common, and there is no guarantee of end-to-end paths between source and destination nodes, we witness a demand for networking approaches in which the devices communicate opportunistically, i.e., taking advantage of the contacts between nodes to exchange data [6]. Such paradigm is explored by the Delay/Disruption Tolerant Networking (DTN), Pocket-Switched Networks (PSNs) and

The associate editor coordinating the review of this manuscript and approving it for publication was Mohammad Omer Farooq.

Opportunistic Networking (OppNet) in which the devices (e.g., smartphones, tablets, laptops, IoT devices, etc) establish direct connections between each other in order to exchange data [7].

Due to the advantages of implementing the content-centric network architecture (CCN) in the DTN paradigm [8], recent literature has focused on developing solutions that enhance data retrieval in the so-called Content-Centric Delay Tolerant Network (CCDTN). CCDTN comprises a DTN and its respective characteristics such as mobility, intermittency and partitioning, however, using the aspects of CCN that are beneficial to it, such as name-based routing, which facilitates the identification of content and its retrieval in dynamic scenarios; in-network caching, which reduces the data delivery delay; and asynchronous communication, which eliminates the need to maintain connection between communicating nodes (i.e,. end-to-end path). In summary, CCDTN solutions are not only concerned with the data exchange between nodes that do not have direct communication with each other, but with the proper functioning of the data request delivery processes in a DTN scenario.

Despite this positive feature, relevant issues are still open. In DTN, for instance, one of the major routing challenges is to determine the relay node selection strategy in order to increase the packet delivery probability, minimizing the number of replications in the network and reducing, if possible, the delivery delay. The well-known Epidemic protocol [9], which is based on replication at every contact opportunity, presents satisfactory results in terms of delivery rate and delay, but with the trade-off of a high network overhead [10]. Since the CCN communications have been designed to operate mainly in broadcast and multicast [4] behaving as the Epidemic, many recent works in CCDTN seek for the best trade-off between CCN operation policy and DTN limitations.

Hence, employing jointly DTN and CCN offers new opportunities for performance improvement [11], but also raises some relevant issues, for example, overcoming the broadcast of interests, but still balancing the forwarding of the corresponding data, avoiding to recurrently select the same set of nodes for that task in the network. With regard to data delivery, it is fundamental to value the possible multiple paths in DTN, since forwarding through the reverse path of interests can be hampered by network interruptions.

Still in the context of relay node selection strategy, which is one of the most promising in CCDTN, such strategy should avoid as much as possible the epidemic behavior by limiting the number of nodes that receive messages based on some criterion. This is where social awareness comes into play: forwarding data based on the different levels of social interaction among nodes has shown great potential to improve data delivery as well as reduce latency and overhead [12]–[14].

Within this scope, there are several DTN/OppNet routing proposals that consider knowledge about the social relationship among network nodes, such as Simbet [15], SimBetTS [16], Bubble Rap [17], dLife [18], SCORP [13], and STCR [19] to name a few. Hence, our work aims to improve relay node selection strategies such as [20], [21], and [22] by defining a local relationship between neighbor nodes in CCDTN environment.

Our proposal, named DIscovery and REtrieval protocol based on Social Cooperation (DIRESC), provides advertisement, discovery, and data retrieval in CCDTN based on cooperation between nodes that have a specific level of social relationship. The main goal of DIRESC is to define a selection strategy of relay nodes which increases the request satisfaction rate, reducing the content discovery and retrieval delays as well as the network overhead. DIRESC takes into account the social relationship factor between nodes aiming to reduce the distance (in terms of hop count and time) between a requester and the data requested. The social cooperation scheme proposed is based on the relationship between nodes, calculated through the encounter history. The proposed scheme is able to increase the message delivery probability since it determines whether a specific pair of nodes has chances of meeting again in the network, qualifying them as suitable or not to relay messages.

The benefits of DIRESC can be summarised as follows:

- The proposed DIRESC protocol significantly increases the request satisfaction rate, i.e., the relation between interests sent and attended, as well as significantly decreases the average data recovery delay.
- DIRESC avoids two drawbacks of the solution proposed in [23]: 1) the problem of the selection of the preferred nodes [24], and 2) partitioning of the content discovery process, which increases the delay for data recovering.
- DIRESC prevents the excessive message dissemination that represents a significant impact on the number of message retransmissions and duplicate packets.
- DIRESC avoids sending messages only to nodes with high centrality or with high rates of content retrieval, which overwhelms the demand on those nodes and depletes their resources. This undesired behavior is dealt with the utilization of a multiple criteria approach for ranking the nodes' ability of delivering packets, instead of using only a single criterion.
- Seeking a balance between the wide dissemination of packets in the network and the selection of a specific set of nodes for the sending of requests and data, the DIRESC protocol saves node resources such as energy and buffer capacity, which improves the overall network performance, as presented through our results.

Besides the proposed DIRESC protocol, this work has an additional contribution in which we have implemented the CCDTN [25] as described in Section IV-A for the Opportunistic Network Environment (ONE) simulator in order to evaluate our proposal.

This paper is organized as follows. Section II summarizes the related work focusing in the CCDTN solutions. Section III describes in detail the DIRESC protocol. Section IV presents the performance evaluation of DIRESC in comparison to specific benchmark solutions while the Section V discusses the results. Finally, Section VI presents the final remarks and directions for future works.

II. RELATED WORK

Nowadays, with increasing use of mobile devices, such as smartphones and tablets, the production and access of content on the internet has become distributed, i.e., users behave as both content producers and consumers, which leverages the use of these devices for data exchange between users and makes DTN an attractive paradigm for content distribution, i.e., CCDTN.

DTNs have been widely studied and present features that are challenging due to the nature of the application scenarios. Mobility, intermittency, and frequent disconnections between nodes raise some questions that are still addressed in the literature. Defining a message dissemination strategy that maximizes the probability of packet delivery and minimizes network overhead and delay are still few of the major challenges of this networking paradigm. In the last years, the literature has proposed several solutions for the packet delivery in DTN. In general, these solutions fall into three categories [12]: forwarding-based (such as Direct Transmission [26]), flood-based (such as Epidemic protocol [9]), and replication-based (such as Spray-and-Wait [27], Spray-and-Focus [28], Prophet [29], Avoiding Replication [30], Simbet [15], SimBetTS [16], Bubble Rap protocol [17], dLife [18]).

CCN/ICN has been proposed in the literature as an alternative, in order to mitigate DTN problems such as remote communication between nodes in an environment in which there is no end-to-end path [31]. This CCN/ICN networking paradigm is based on i) asynchronous communication between the provider-requester pair; ii) search of named data that facilitates content retrieval in high mobility environments and when the node location is unknown; and iii) controlled data replication that reduces the number of requests and the delay in data retrieval [8].

DTN and CCN are complementary in many aspects, especially with respect to mobility and intermittency [32]. Anastasiades *et al.* [33]–[35] propose an agent-assisted content retrieval scheme in which a requester finds potential agent nodes and delegates to one of them the content discovery and retrieval in an opportunistic CCN. The choice of these agents considers information such as location visited, planned destinations, or even randomly. The issue behind this kind of solution persists in the trust placed on a single node to deliver the request and the content.

The idea of ants behavior to obtain food is used by Nguyen *et al.* [36] to propose STIR, a mechanism in which a utility $U(c, n)$ is defined to compute and reflect the space-time proximity between a node and the consumer, thus composing a set of utilities for a given content to be requested and retrieved from consumer to provider. In addition, similar to the ant colony optimization, the utility between nodes is reinforced and aged. The process also generates a convergence for shorter paths between consumer and provider. Similarly, Nguyen *et al.* [37], [38] propose SIR, an intelligent routing scheme based on particle swarm optimization, also assuming the definition of a utility between the nodes and a gradient management mechanism. In addition, these bio-inspired proposals also use Spray-and-Wait [27] to propagate interests. The issue about them concerns the slow convergence process towards determining the forwarding paths.

Inspired by Prophet [29], Duarte et. al [39] propose PIFP, a probabilistic interest forwarding protocol, which computes the probability of nodes coming into contact with the requested contents, so that the requests are transmitted only to high probability nodes. Content delivery is performed through the reverse path of sending requests, i.e., using Pending Interest Table (PIT) information as ‘‘bread crumbs’’ [4]. The issue about this approach lies in increasing the probability of contacting the same set of nodes regarding specific contents. This results in recurrent selection of such nodes as forwarders, which leads to an unbalance load distribution and its respective aggravating factors. In addition, restricting data forwarding through the reverse path of the transmitted interest might make the protocol vulnerable to interruptions (i.e., lack of communication between nodes), and broadcasting can generate too much overhead and high discarding of unsolicited data.

Also based on the principle of forwarding interest to nodes with higher social centrality, assuming that the concept of centrality may be associated with knowledge about other nodes and contents in the network [7], Le *et al.* [40] propose SACR, a socially-aware content retrieval mechanism using random walks. With SACR, nodes generate and disseminate random-walk probing packets every second while counting the amount of these packets when received, assuming that nodes with higher counted value also have greater centrality in network. However, as stated earlier, the centrality metric tends to unbalance the dissemination of packets when the same node (i.e., with greater centrality) is always chosen to forward packets [41].

Using an approach that explores the social relationship between nodes, Lu *et al.* [23] propose STCR, a social-tie based opportunistic content routing protocol which associates each pair of network nodes with a social relation value, used to calculate centrality. This measure guides the K-means algorithm in the division of nodes into clusters (social hierarchy), through which interests are transmitted (always from the lowest to the highest hierarchy/centrality). Once a node of higher centrality discovers a provider, the routing of content to the requester is guided by the higher values of social-tie. Lu *et al.* [42] also propose the STCR for routing of packets between distinct communities (formed according to the social relations of the nodes), the STCRC. In it, the cluster head and providers identify and delegate to local nodes of closest contact with foreign communities to deliver interest and data packets, respectively, in those communities.

Despite the strategy of using network social knowledge to guide packet routing, these proposals neglect that current requester may be in the same hierarchical level of the provider or even have a high social relation with it, forcing the interest to go through other groups of nodes before actually reaching the provider, which can significantly increase the retrieval delay.

DIRESC is a multi-copy protocol that applies social metrics to define groups of candidate nodes to retransmit packets with distinct priorities. This prioritization is based on the propensity to retrieve content. Our proposal differs from the others because it defines different criteria for node selection, making the process of packet dissemination distributed rather than centralized, as it has been proposed in the literature. DIRESC makes full use of the multiple paths available in a DTN scenario, without resorting to broadcast. The nodes are selected to forward interest packets taking into consideration their ability to satisfy the requests. This ability is measured by the chance of the node on finding the content or its potential to act on a socially cooperative way in favor of the requester. Traditional approaches based on a metric of node centrality have a different behavior because the requests are not disseminated in other directions than the node of higher centrality.

Table 1 shows a comparison between DIRESC and the aforementioned proposals. As it is shown, the CCN broadcasting, which spreads interests in a broadcast manner when no Forwarding Information Base (FIB) information is found, does not consider social relationship between nodes or deliveries data packets through multi-paths, i.e., as described in [4], data only follows the reverse path of interests. Finally, no other criteria is defined by the native CCN to send interests besides using FIB information or broadcasting. The agent-based solution it is the most divergent since it does not rely on social tie, or provide distributed requesting or delivering, since it prioritizes and delegates the request and delivery of data to one single node. Also not valuing social relationships, bio-inspired solutions do not delivery data through multiple paths. Instead, such solutions focus on the use of single, shortest path. The social-based solutions STCR, STCRC and SACR, as expected value social relationships between nodes. However, by using centrality as their fundamental principle and not defining multi-criteria to select nodes, they result in a unbalanced requesting process. In its turn, the probabilistic solution PIFP does not consider social relationships. By recurrently valuing nodes with high probability to find contents, it does not employ a multi-criteria method to select nodes and forward packets. It also does not generate distributed requests or use multiple paths to deliver data.

Finally, DIRESC, which is based on social relationship between nodes, achieves distributed requesting using multi-copy and benefits from multi-path possibilities by defining multi-criteria not only to send interests to providers but also to deliver data to consumers. Thus, DIRESC balances the dissemination of packets ensuring that they do not flood the network or are only sent to a specific set of nodes that presents the same or similar attribute values.

TABLE 1. DIRESC compared to the literature.

	CCN Broadcasting [4]	Agent-based [33]	Bio-inspired (STIR [36], SIR [38])	STCR [23]	SACR [40]	PIFP [39]	DIRESC
Based on social-tie				X	X		X
Distributed requesting	X		X				X
Request by multi-copy	X		X	X	X	X	X
Delivery by multi-path				X	X		X
Multi-criteria forwarding		X					X

III. DISCOVERY AND RETRIEVAL BASED ON SOCIAL COOPERATION (DIRESC)

In this section, we present the DIRESC protocol. A preliminary version of this protocol was proposed and presented in [43]. DIRESC is a multi-copy based protocol whose main goal is to decrease the delay on data discovery and retrieval in CCDTN. DIRESC employs a distributed strategy that differs from the traditional approaches, in which the content are reached through the convergence of interest messages that go through preferred nodes [41] [24]. As detailed below, our protocol avoids this undesired behavior by employing multiple criteria for ranking the nodes' ability of delivering packets, instead of using only a single criterion.

A. SOCIAL RELATIONSHIP

DIRESC nodes take into consideration the social relationship between them by computing a social-tie value. This measurement represents how close a pair of nodes is in terms of social contact. Based on the history of contacts between the nodes, each node can compute the frequency and the freshness of its relationships. This allows, for instance, to infer that a pair of nodes has a high social-tie value due to the high frequency of their contacts in a short time interval.

Lu *et al.* [19], [23] proposed the social-tie concept as a combination of the concepts of frequency and freshness. Frequency is a metric to evaluate how frequently two nodes meet each other, while freshness is a metric to evaluate how recently two nodes have met each other. Formally, the social-tie value $R_i(j)$, between the pair of nodes i and j, is defined as (1).

$$R_i(j) = \sum_{k=1}^{N} W(t_{now} - t_{jk}), \tag{1}$$

where $W(x)$ is a contact weighting function that is computed over the time from a contact in the past t_k until the present time t_{now}. Thus, $t_{j1}, t_{j2}, t_{j3}, \ldots t_{jN}$ represent the ordered contact times between the nodes i and j and are kept on the contact array $W_i^N(j)$.

The influence of each contact between the nodes i and j is defined by the contact function $W(x)$ that computes the impact on $R_i(j)$ based on the freshness of the contacts, where x represents the time interval between the moment of a contact in the past and the present time. The contact function $W(x)$, defined by [44], is presented in (2).

$$W(x) = (\frac{1}{2})^{\lambda x}, \quad \text{where } 0 \leq \lambda \leq 1. \tag{2}$$

λ works as a control parameter that allows a trade-off between frequency and freshness in the participation to the social-tie value. Frequency contributes more than freshness as λ approaches 0, while freshness becomes more representative than frequency as λ approaches 1. The choice of an extreme value, 0 or 1, implies on the social-tie value be derived by only metric, frequency or freshness.

Unlike Lu *et al.* [19], [23], DIRESC ignores the global relationship in the network (i.e., the nodes centrality) and employs $R_i(j)$ only to define the local relationship between neighbor nodes. In other words, the social-tie array of the node i is kept locally.

Additionally, in DIRESC, every node i keeps a relationship array S_i^M with the computed social-tie values $R_i(h)$, where $h = 1, 2, \ldots, M$ and M is the total number of nodes with which the node i had contact until the present time t_{now}. Based on these individual social-tie values, the average social-tie $\bar{R}_i$ is computed as described by (3).

$$\bar{R}_i = \frac{1}{M} \sum_{h=1}^{M} R_i(h). \tag{3}$$

DIRESC defines node j as a friend of node i (i.e., nodes i and j can act in a socially cooperative way) if the inequality (4) is true to these nodes.

$$R_i(j) \geq \Omega \bar{R}_i, \tag{4}$$

where Ω is the density factor that must be adjusted according to the network density, i.e., the average number of neighbors perceived by nodes. The density factor Ω allows to control the number of friends of each node. Thus, the density factor allows, for example, to limit the number of friends in high-density scenarios and to increase the number of friends in low-density scenarios. As the value of Ω decreases the node becomes more friendly, i.e., socially cooperative.

The (individual) social-tie value $R_i(j)$ and the average social-tie $\bar{R}_i$ are updated whenever two nodes come in contact. Thus, both, $R_i(j)$ and $\bar{R}_i$, are always kept available for decision making in our protocol. The confirmation of the friendship between two nodes, i.e., the verification of the inequality (4), is critical to DIRESC, because it allows the nodes to act in a socially cooperative way.

Finally, it is worth mentioning that, similar to the Epidemic and Prophet protocols, when contacted, two nodes first exchange summary vectors of their components (CS, PIT and FIB) to guide the processes of announcement, discovery, and retrieval of content. Each process corresponds to a phase in DIRESC, as we describe in detail in the next subsections.

B. ANNOUNCEMENT PHASE

As described in [4], the provider (or producer) nodes of a CCN announce the prefixes for which they are able to delivery the content and populating the FIBs of neighboring nodes with this information. In a static network scenario, this information is unlikely to be changed unless the provider explicitly announces that it does not have certain content. However, in a mobile environment such as DTN, information stored in FIB easily become out of date due to the varying frequency or freshness with which some nodes meet, making broadcast prefixes announces a disadvantageous option.

Thus, CCDTN demands for a strategy to announce prefixes that reduces the number of involved nodes, valuing contacts with greater chances of recurrence. By avoiding involving all nodes of the network in the process, this strategy avoids that information that is out of date to the sending of interests is maintained in these tables. Algorithm 1 summarizes the strategy employed in DIRESC for prefix announcing.

Algorithm 1 Announcement Phase of DIRESC

Input: $i, j, R_i(j), \bar{R}_i, \Omega$
1 **if** $R_i(j) \geq \Omega \bar{R}_i$ **then**
2   **if** *i is producer* **then**
3   **forall the** *prefix* $p \in CS_i$ **do**
4   **if** $p \notin FIB_j$ **then**
5   Update FIB_j: add p mapped to i
6   **end**
7   **end**
8   **end**
9   **forall the** *prefix* $p \in FIB_i$ **do**
10   **if** $p \notin FIB_j$ **then**
11   Update FIB_j: add p mapped to i
12   **else**
13   Update FIB_j: add i to the present mapping of p
14   **end**
15   **end**
16 **end**

The announcement between i and j is done only if they are friend (line 1). If node i generates any content (line 2) then it announces to j the contents in its cache CS_i (line 3) that the node j has not learned yet (line 4). Thus, the FIB_j is only updated with new entries (line 5). Node i also announces to node j everything it learned from other producers (line 6). If the prefix is new to node j (line 7), it adds a new entry in FIB_j (line 8); otherwise (line 9), it adds a new mapping to the already existing entry (line 10).

DIRESC implicitly creates groups, which we refer to as clusters, since they are composed of nodes that received the announcements of prefixes and that are able to help on delivering requests. This allows for fast propagation of the information (about the contents), but also avoids the network flooding, since the clustering limits the propagation to

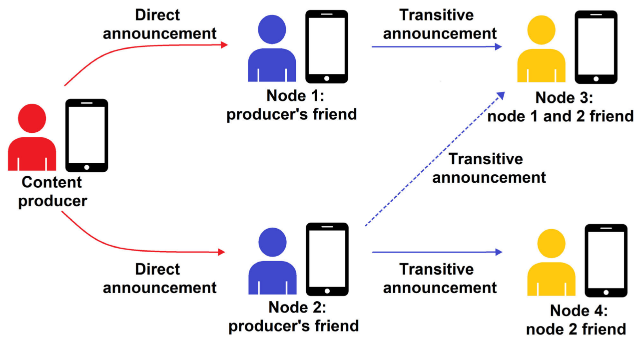


FIGURE 1. Direct and transitive announcements.

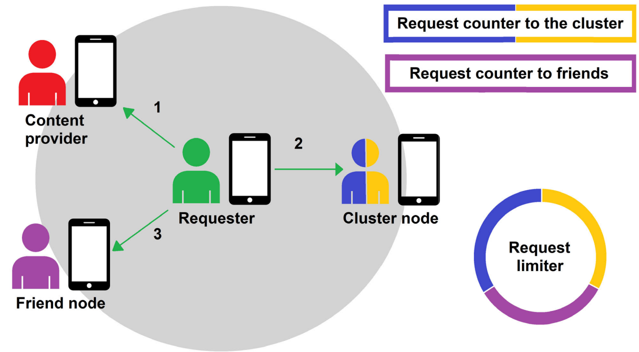


FIGURE 2. Possibilities of sending interest packets.

just friends. However, it is worth highlighting that our protocol does not employ any clustering algorithm.

Figure 1 illustrates a cluster and the two types of announcements employed in DIRESC: direct and transitive. Direct announcements are sent from a producer node to its friends (e.g., nodes 1 and 2), while transitive announcements are sent from a friend of the producer (e.g., node 2) to its own friends (e.g., nodes 3 and 4). As illustrated, a node (e.g., node 3) can have multiple friends in the same cluster (e.g., nodes 1 and 2), which makes this node receive announcements related to the same content from all them.

In this way, information about contents is propagated and maintained in FIBs of nodes with high chance of re-encounter, given their history of social relationship, regardless of whether its centrality in the network. To keep such information up to date, as well as to control the FIB size and entries, the equation is periodically tested, so records associated with a node that is no longer a friend are discarded.

C. DISCOVERY PHASE

In the discovery phase, each node must have a strategy for sending the interest packets, i.e., the requests for contents. DIRESC was designed to avoid the drawbacks of two traditional approaches: epidemic dissemination and preferred nodes. While the epidemic dissemination unnecessarily flood the network, the preferred nodes approach always drains the resources of a specific set of nodes. DIRESC employs a criteria set to create multiple groups of nodes to send the requests for contents. Thus, our protocol avoids flooding the network and it also distributes the burden of the interest forwarding among the nodes. For this, DIRESC controls locally the sending of requests in the network using the following elements:

- **Request counter to the cluster** (α) – number of requests for the same content that were sent from a node to a group of nodes that have information about that specific content in their FIB. This group of nodes is the cluster established during the announcement phase (cf., Section III-B). Thus, the counter α is increased by one for each request sent from node i to any node of the cluster, and whose the target content is c.
- **Request counter to friends** (β) – number of requests for the same content that were sent from a node to its friends, i.e., nodes with which the requester has a high social-tie value. Thus, the counter β is increased by one for each request sent from node i to any of its friends, and whose the target content is c.
- **Request limiter** (μ) – maximum number of requests for an content. In other words, a node i that is interested in the content c must send at most μ requests to the cluster or to its friends.

These elements allow DIRESC to control the number of interest packets sent by a requester node (primary or intermediary). Basically, our protocol compares the counters (α and β) against the limiter (μ) in order to avoid sending excessive number of packets.

In order to balance the sending of interests among distinct groups of nodes, valuing the social relation between intermediaries and providers or between intermediaries and the requester itself, DIRESC establishes three possibilities of sending these packages, which are illustrated in Figure 2 and ruled by the follow criteria:

1) **Contact with a provider** – the node met is a provider of the content that is being looked for, i.e., this node runs an application that generates the requested data or this node has a copy of the requested data in its cache (CS). Since this type of request does not involve intermediary nodes, the interest packet sent is not accounted.
2) **Contact with a cluster node** – the node met belongs to the group of nodes (or cluster) established during the announcement phase, Section III-B, and this node has information in its FIB concerning the content that is being looked for. Additionally, the number of requests sent to this cluster (α) must be lower than the maximum allowed (μ). If all these conditions are satisfied, the interest packet is sent and the request counter to the cluster (α) is incremented.
3) **Contact with a friend node** – the node met is considered as a friend, i.e., it has a high social-tie value, and the number of requests sent to friends (β) is lower than the maximum allowed (μ). If these conditions are satisfied, the interest packet is sent and the request

counter to friends (β) is incremented. The goal of this sort of request is to increase the distribution reach of interest requests with help from the friends.

Once nodes exchange summary vectors of their components, identifying a provider or an intermediary belonging to a cluster can be done immediately as soon as they come in contact. Algorithm 2 summarizes the strategy employed in DIRESC for content discovery.

Algorithm 2 Discovery Phase of DIRESC

Input: $i, j, R_i(j), \bar{R}_i, \Omega, \mu$

1 **forall the** *prefix* $p \in PIT_i$ **do**
2   **if** $p \in CS_j$ **then**
3   i send an interest packet concerning p to j
4   **else if** $(\alpha < \mu)$ **and** $(p \in FIB_j$ **or** j *is in mapping of* $p \in FIB_i)$ **then**
5   i send an interest packet concerning p to j
6     $\alpha \leftarrow \alpha + 1$
7   **else if** $(\beta < \mu)$ **and** $(R_i(j) \geq \Omega\bar{R}_i)$ **then**
8   i send an interest packet concerning p to j
9     $\beta \leftarrow \beta + 1$
10   **end**
11 **end**
12 **forall the** *interest concerning p received from j* **do**
13   **if** $p \in CS_i$ **then**
14   i send data packet concerning p to j
15   **else if** $p \in FIB_i$ **or** $R_i(j) \geq \Omega\bar{R}_i$ **then**
16   **if** $p \in PIT_i$ **then**
17   Update PIT_i: add j to the present mapping of p
18   Drop interest
19   **else**
20   Update PIT_i: add p mapped to j
21   Keep interest for future forwarding
22   **end**
23   **else**
24   Drop interest
25   **end**
26 **end**

The discovery phase is composed of two parts, one related to the interest packets potentially sent from i to j (lines 1–9), and other related to the interest packets received by i coming from j (lines 10–21). Node i sends an interest packet p to node j if any of the three follow conditions is satisfied. First, if node j has the content p in its cache CS_j (line 2), then an interest packet is sent from i to j (line 3). Second, if node j has an entry concerning p in its FIB_j and node i did not exceed the number of interest packets sent to the cluster, i.e, $\alpha < \mu$ (line 4), then an interest packet is sent from i to j (line 5) and the counter α is updated (line 6). Third, if node j is friend of i and node i did not exceed the number of interest packets sent to friends, i.e., $\beta < \mu$ (line 7), then an interest packet is sent from i to j (line 8) and the counter β is updated (line 9). Thus, we adopt a strategy different than the ones commonly employed in CCN. The reason is the context of CCDTN, in which the existence of requests in the PIT, right after the evaluation of the CS, would not guarantee **(i)** that the forwarder node can find quickly the content provider, **(ii)** neither that the requester node and its forwarder node can encounter each other again in the future.

Node i evaluates every interest received from j according to the follow rules. If node i has the content p in its cache CS_i (line 11), then a data packet is sent from i to j (line 12). If node i has an entry related to the content p in its FIB_i or node j is considered as a friend (line 13), then an additional check is necessary before processing the interest packet. If the content p is already in the PIT_i (line 14), then the entry is updated (line 15) and the interest is dropped (line 16). Otherwise, a new entry is added to the PIT_i (line 18) and the interest is kept for future forwarding (line 19). If the interest packet does not satisfy any of the previous checks then it is simply dropped (line 21).

D. RECOVERY PHASE

This phase deals with the data delivery, i.e., it occurs whenever a provider node of the requested content replies to a request. Similar to previous phases, in the recovery phase the goal is to keep the protocol operating in an efficient way, i.e., avoiding the epidemic behavior and enjoying the opportunities of content delivery offered by social relationships.

Algorithm 3 Recovery Phase of DIRESC

Input: $i, j, R_j(i), \bar{R}_j, \Omega$

1 **forall the** *prefix* $p \in PIT_i$ **do**
2   **if** $p \in CS_i$ **then**
3   **if** *j is requesting p* **or** $p \in PIT_j$ **or** $R_j(i) \geq \Omega\bar{R}_j$ **then**
4   i send data packet concerning p to j
5   **end**
6   **end**
7 **end**
8 **forall the** *data concerning p received from j* **do**
9   **if** $p \in PIT_i$ **then**
10   **if** *i is the primary requester of p* **then**
11   Send data to the requester application
12   **else**
13   Keep content in CS_i for future forwarding
14   **end**
15   **else if** $R_i(j) \geq \Omega\bar{R}_i$ **then**
16   Keep content in CS_i for future forwarding
17 **end**

Algorithm 3 summarizes the strategy employed in DIRESC for content recovery. The recovery phase is also composed of two parts, one related to the data packets potentially sent from i to j (lines 1–4), and other related to the data packets received by i coming from j (lines 5–12). If p is in CS_i, then node i will send the content (or data packet) to node j if any of the following conditions is satisfied: 1) node j is

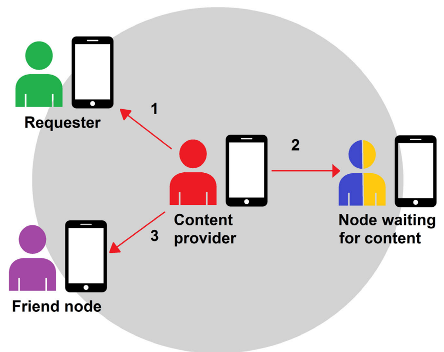


FIGURE 3. Possibilities of sending data packets.

requesting p; or 2) there is a pending request for p in PIT_j; or 3) node i is a *friend* of node j, i.e., $R_j(i) \geq \Omega\bar{R}_j$.

Node i evaluates every data concerning p received from j according to the follow rules. If node i has a pending request to p in its PIT_i (line 6) and this node is the primary requester of p (line 7), then it delivers the data to the requester application (line 8). If node i is an intermediary, i.e., it has a pending request to p in its PIT_i but it is not the primary requester of p (line 9), then it keeps the data in its CS_i for future forwarding (line 10). If node i does not have any pending request to p in its PIT_i (line 11), but is friend of j then it keeps the data in its CS_i for future forwarding (line 12).

In summary, the recovery phase takes into consideration any node interested on receiving a specific content, which is easily identified due to the exchange of summary vectors, and also nodes that can act in a socially cooperative way to help the primary requester. Figure 3 illustrates the possibilities of sending data packets, which are ruled by the follow criteria:

1) **Contact with the requester** – the node met is requesting the content. This criterion represents the case in which a node that has the requested content in its cache and the primary requester directly meet each other.
2) **Contact with *any* requester** – the node met has some number of pending requests in its PIT which can be satisfied by the provider. The node met did not send an interest packet directly to this provider, but the node was waiting for (at least) a content available in the provider.
3) **Contact with a friend** – the node met can act in a socially cooperative way (according to Equation 4) to favor the primary requester. Again, the social relationship between the nodes is employed, this time to help on the content recovery.

Thus, having presented the three phases of our protocol, we list in Table 2 the parameters and important variables for its proper functioning.

IV. EVALUATION METHODOLOGY

Our proposed DIRESC routing protocol has been implemented in Opportunistic Network Environment (ONE) simulator [45] and evaluated in terms of how efficient it increases data delivery rate and decreases delay and overhead, while maintaining energy efficiency against other opportunistic content routing protocols. Additionally, we evaluate DIRESC energy efficiency by investigating how the transmission of popular content in the network affects the energy of each node along the route.

TABLE 2. Parameters notation list.

Notation	Short description
λ	Control parameter for trade-off between frequency and freshness in nodes contact. Defined as e^{-4} [23]
Ω	Density factor that is adjusted according to the scenario
μ	Maximum limit of requests sent for an content to each group
α	Number of requests sent for an content to nodes with information in FIB about that specific content
β	Number of requests sent for an content to friend nodes

In our study, DIRESC was compared against three well-known routing protocols, namely: CCN broadcasting [4], Probabilistic Interest Forwarding Protocol (PIFP) [39] and Social-Tie based Content Retrieval (STCR) [23]. Our rationale for choosing these protocols for our comparative performance study of DIRESC is as follows. CCN broadcasting is based on the seminal works of the CCN architecture and is under specification in a number of documents of the IRTF ICN research group [46]. We use CCN broadcasting to attempt to understand the upper bounds of delivery rate once it broadcasts interest packets to all nodes in its range. This results in an 'uncontrolled' replication of interest packets to all nodes that do not yet have a copy. STCR and PIFP were selected as representatives of the class of social-assisted schemes that exploit the relationship between nodes to perform data forwarding in DTNs in our comparative analysis. PIFP presents better results for satisfied interest, average delay and total number of interest packets in dense scenarios, while STCR is better on the total number of data packets transmitted in the network [39]. DIRESC also follows in this category and was intended to overcome these limitations.

A. CCN ARCHITECTURE IN THE ONE

Before getting into the evaluation and results, it is worth noting that we have implemented a CCN architecture in ONE simulator in order to provide a realistic environment to test and evaluate DIRESC. Thus, this section presents an overview of the CCN architecture implemented, i.e., the CCN Broadcasting.

CCN has two types of messages, interest and data, which dictate the behavior of the network. In order to create messages of interest and data in the simulator, we generate two types of message, as defined by [39]: (i) interest messages, that holds the identifier for the source node, which is the node that originally generated the interest message (i.e., the requester), message size, and message name (i.e., prefix of searched content). During our simulations such messages are created at random nodes over the simulation time; and (ii) content messages, containing data uploaded by producer nodes. The content messages are defined in the amount of 500, whose sizes vary from 250 to 350 bytes, distributed among 15 producer nodes. These messages contain their content prefix, their size and the producer node identifier.

Both types of messages have a time to live (TTL) that varies according to the scenarios described in Section IV-B, and as shown in Table 4. Messages are discarded from buffer according to the FIFO (First In, First Out) criteria, at end of TTL interval. It is worth mentioning that this discard policy applies only to intermediate relay nodes (which replicate contents and operate as secondary providers), that is, data generated by producer nodes (primary providers) are not discarded from their caches, so that they are able to always feed the network with the content they produce, whenever requested.

The CCN architecture also takes into consideration: (i) a mechanism to avoid possible duplication of storage, as well as repeated responses to the same request; (ii) a producer node identification, that allows decisions such as deletion of data packets from cache; (iii) The CCN broadcasting implementation, which reproduces the standard behavior of CCN, that is, the decision making for the sending, processing and discarding of messages, according to [4]. It allows us to evaluate its operation in a DTN environment; (iv) an extension to the DTN environment to support CCN, including the creation, update and deletion of entries in CS, PIT and FIB, as well as deletion of expired messages according to their TTL value; (v) and the relationship between nodes in the network, as defined in Section III.

B. SIMULATION SCENARIOS

We have validated and evaluated DIRESC performance in two scenarios: the former is a synthetic scenario Helsinki [39] using the Shortest Path Map-Based Movement mobility model in order to represent the pedestrians and cars nodes movements, and Map-Based Movement mobility model to represent the trains movement. The latter is San Francisco cabs [47] real mobility trace. In this way, we are able to represent the behavior of the studied protocols, in environments with different technologies, transmission rates and mobility patterns.

Moreover, since DIRESC is based on social relationships, the evaluation in both scenarios allows to verify its performance 1) when the mobility of nodes represents the social interest of individuals (i.e., the San Francisco environment) and 2) when such mobility does not necessarily represent this behavior (i.e., the Helsinki scenario, in which node movements are random).

The Helsinki scenario applies mobility models to generate the movement of different nodes, including pedestrians, cars and trams. This scenario represents trajectories of 100 nodes, over a period of 24 hours. Table 3 summarizes the main characteristics of this scenario.

The San Francisco cabs scenario comprises GPS trajectories of 483 users, and was collected for 24 days with samples ranging from 1 to 3 minutes. However, similarly to the work presented in [48], we considered from the San Francisco trace only 2 hours, in which there are 100 nodes, for a fair comparison of all protocols considered in our evaluation.

TABLE 3. Helsinki nodes.

Type	Amount	Speed
Pedestrians	62	0.5-1.5 m/s
Cars	32	10-50 km/h
Trams	6	25-36 km/h

TABLE 4. Simulation parameters.

Parameter	Helsinki	San Francisco
Duration	24 hours	2 hours
Message TTL	8 hours	40 min
Storage capacity	100 MB	100 MB
Throughput	1 Mbps	3 Mbps
Transmission Range	30 meters	300 meters
Interests generated	about 5700 interests	about 500 interests
Area (km x km)	15.3 km^2	37.6 km^2
Density factor (Ω)	Ω: 2	Ω: 1
Request limiter (μ)	μ: 1	μ: 1

In both scenarios simple network interfaces with constant bit rate are used. Table 4 summarizes other parameters used in these experiments.

For the proposed assessment scenarios, we tested different values for Ω and μ in order to verify the impact resulting from the assignment of different values for these parameters. As expected, high density factor values (Ω) limit the establishment of friendship relations between nodes. On the other hand, low values for this parameter allow more friendship relations to be established in the network. In fact, by the Equation 4, it is known that Ω is a multiplicative factor for the average of the social relationships of any node, whose the resulting value determines the occurrence or not of friendship between a pair of nodes.

Thus, through the evaluation tests, for the Helsinki scenario, considered dense in terms of number of nodes per area of movement, the density factor Ω should be adjusted to a higher value, since in this environment the contacts between nodes tend to occur more frequently and recently. In turn, the San Francisco scenario, considered less dense, required a lower density factor value. Thus, after tests with different values for the factor Ω, we confirmed that, for the scenarios, the values described in Table 4 implied a better trade-off between the evaluation metrics for DIRESC.

Similarly, we tested different values for the request limiter μ. However, for both scenarios, the assignment of high values for this parameter represents a negative impact on the protocol performance in terms of increase of overhead, since it increases the permissiveness of sending requests to the nodes described in Section III-C. Thus, we set the value of μ as shown in Table 4.

In order to evaluate and compare the performance of DIRESC against the other protocols, we measured our metrics over the arithmetic mean of 30 simulation runs in each of the scenarios. The performance metrics are:

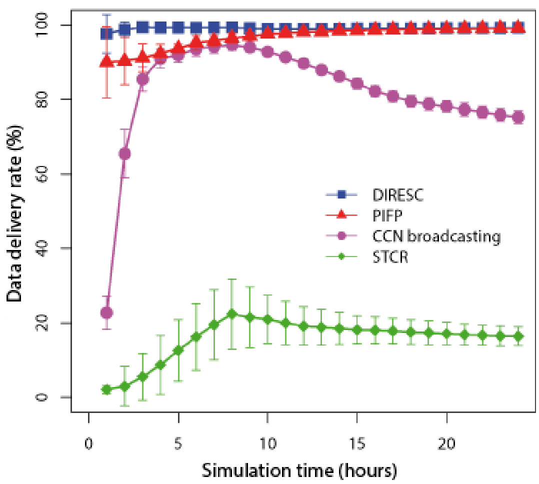


FIGURE 4. Data delivered rate in Helsinki scenario.

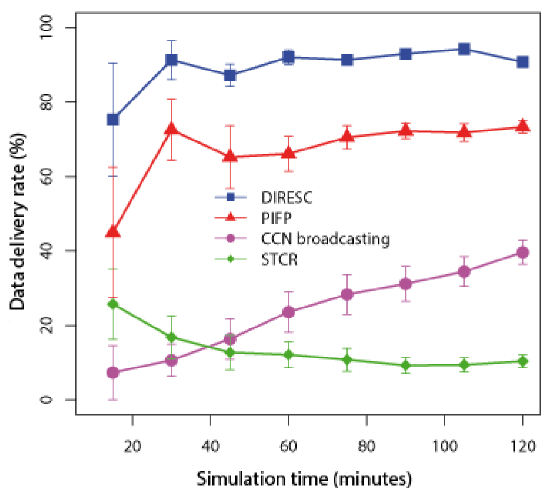


FIGURE 5. Data delivered rate in San Francisco scenario.

- **Data delivery ratio**: the ratio between the total number of satisfied requests and the total number of content requests (or interest packets) transmitted by all nodes.
- **Total average delay**: the time elapsed between the instant an interest packet is sent by the primary requester and the instant the content is delivered ($d_{timeDelivered} - i_{timeSent}$) to this same requester. Then, averaged for all requests served in the network.
- **Overhead**: the number of packets replicated in the network, that is, the number of packets sent by intermediate nodes (relays). Thus, it is computed as the total packet transmitted in the network (p_{sent}) minus the number of packets delivered ($p_{delivered}$) for the first time to the destination. The result is then normalized by the total amount of packets sent. In other words, it indicates how many packets should be sent in order to successfully deliver one.
- **Average number of packets retransmitted per node**: the average number of packets retransmitted by each node.
- **Percentage energy consumption**: the average energy consumption percentage of the nodes at the end of the simulation time.

V. RESULTS

Results are reported using the Helsinki and San Francisco mobility traces with a 95% confidence interval over 30 runs by R Software [49]. The Helsinki scenario used 1 hour of sampling period, generating about 250 interest packages in each of them. The San Francisco scenario used sampling period of 15 minutes, generating 60 interest packets in each of them. The generation of the interest packets was discussed in detail in Section IV-A.

A. DATA DELIVERY RATE

Figures 4 and 5 present the data delivery rate obtained from experiments with the four protocols evaluated in Helsinki and San Francisco scenarios. In those figures, we note that DIRESC has a better performance compared to the others, even though PIFP achieves a close result as shown in Figure 4. The results obtained from DIRESC can be explained due to the dynamics of the protocol, that chooses the best relay node based on different criteria, increasing the probability of message delivery, while the other protocols use only a single criterion. In addition, DIRESC increases the chances of content request and data delivery thanks to the proposed package delivery prioritization scheme, explained in Section III.

We can also see that CCN broadcasting presents the lower delivery rate, in both scenario. It is due to the excess of messages relayed in the network, which causes a buffer overflow preventing the routing of messages in the network, as the simulation goes. On other hand, the PIFP and STCR are more restrictive in choosing the forwarder node, decreasing the chances of message delivery. This behavior becomes more evident in the San Francisco scenario (Figure 5), in which are observed smaller contact and message exchange opportunities between the nodes due to their high mobility and wide area of movement. Thus, the higher the restrictions of the routing protocols for sending messages, the lower the chances that messages be delivered.

B. TOTAL AVERAGE DELAY

The simulation results of the total average delay for data discovery and retrieval are shown in Figures 6 and 7. Regarding delay, DIRESC performs better than the other protocols most of the time. It can be explained by the fact that DIRESC uses three possibilities for forwarding interest and data packets, increasing the chances of content discovery and retrieval. In addition, the DIRESC package delivery prioritization scheme ensures that messages are sent to nodes that most likely meet the requirements (i.e., sending of interests or delivering data to the requester).

We can note in Figure 6 that CCN broadcasting protocol has a higher average delay only up to 3 hours of simulation, when it starts to decrease and approaches zero. This happens

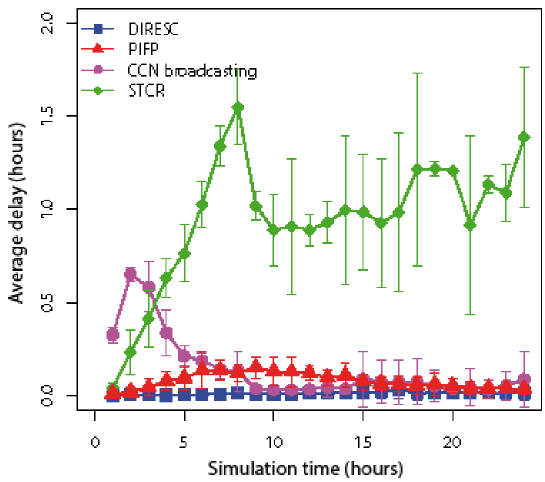


FIGURE 6. Total average delay in Helsinki scenario.

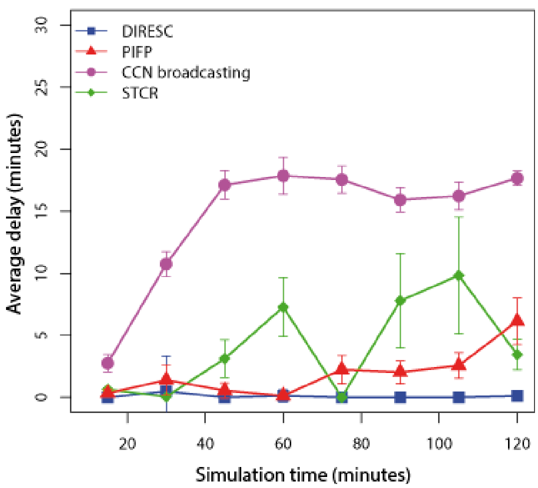


FIGURE 7. Total average delay in San Francisco scenario.

because at the beginning of the simulation, the data packets have not yet been properly replicated over the network, requiring some time until they are retrieved from the primary providers. Then, copies of these packets are generated in the caches of intermediate nodes, reducing the distance between the requester and the secondary providers, and also the delay. In Figure 7, the delay of CCN broadcasting is the highest, during all simulation time. However, we can infer that this represents the same process. That is, it takes some time until the contents are distributed on the network so that they can be retrieved with a low delay. Considering the sparseness of the San Francisco scenario, this time may be longer, so we observe such behavior.

In Figure 6, STCR presents the highest delay when compared against the other protocols. It can be explained due to the many characteristics of the protocol's operation, among which the announcement process of STCR, that is only performed for nodes with the highest centrality. Thus, nodes having a high social relationship with the providers and

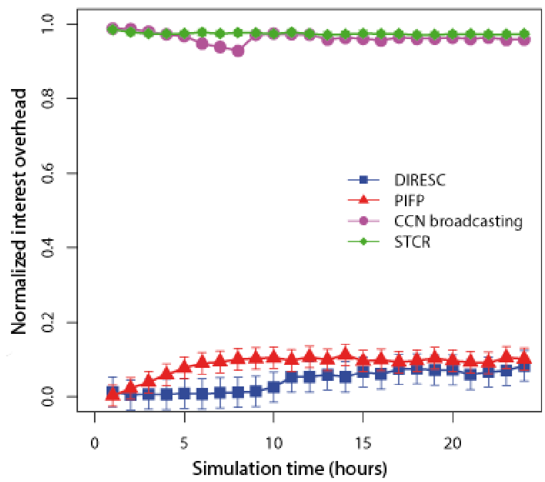


FIGURE 8. Normalized interest in Helsinki scenario.

belonging at the same social level, or even lower, may not have any knowledge of the prefixes that such providers come into contact, which significantly reduces the chances of data retrieval and increases delay. In addition, the disjoint discovery process forces interest packets to take a longer path to providers, which greatly increases the delay for data retrieval. In Figure 7, the low delay obtained from STCR is a result of the low data delivery rate, which implies in few processes of discovery and retrieval per interval (sometimes none) to calculate the average delay, behavior that can be observed on the slopes shown in the graph starting at 60 and repeating at 105 minutes of simulation.

The PIFP presents the second lowest delay obtained in both scenarios. This protocol rely only on the computation of the prediction of the nodes to find contents, increasing their chances of retrieving the desired content and reducing the associated delay. Thus, it achieves better performance in terms of the delay than CCN broadcasting, that spread messages to all nodes they encounter, and STCR protocol that only forwards message to nodes having higher centrality than itself.

C. OVERHEAD

Figures 8, 9, 10 and 11 show the interest and data overheads obtained from the experiments with the four protocols evaluated in Helsinki and San Francisco scenarios, respectively. In both scenarios, DIRESC shows the lowest overhead rates, for the pair interest and data packet, followed by PIFP. Both protocols use a minimum number of relay nodes to forward messages, when compared with others protocols, reducing unnecessary message propagation. DIRESC has better performance also due to the package delivery prioritization scheme, whereby the discovery and retrieval process can be terminated without involving relay nodes, resulting in low overhead of both interests and data. It is important to mention that, PIFP performs better than DIRESC in some points of interest overhead, in San Francisco scenario. This can be explained by the few restrictions imposed by our protocol for

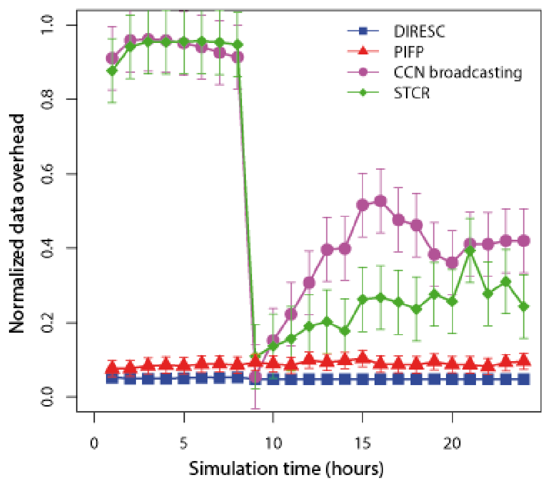


FIGURE 9. Normalized data overhead in Helsinki scenario.

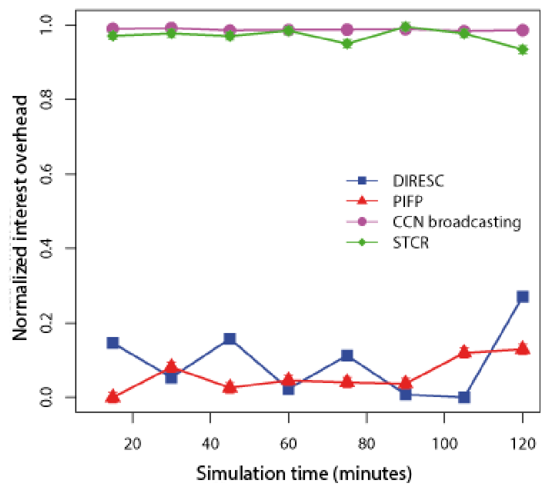


FIGURE 10. Normalized interest in San Francisco scenario.

choosing relays, especially in this lower density scenario. So, DIRESC was configured to allow the establishment of more friendship relationships, in order to increase data delivery, and reduce delay.

CCN broadcasting and STCR show the highest overhead. The observed behaviour of CCN broadcasting is due to the epidemic nature of the protocol that causes an excessive load of interest and data packets to be replicated in the network. In the case of STCR, the need to use many relays between requester and provider, since protocol neglects the direct contact between them, explains the high overhead.

D. AVERAGE NUMBER OF RETRANSMITTED PACKETS PER NODE

The average number of retransmitted packets per node is given in Figures 12 and 13. We can note that all protocols perform very similar, with the exception of CCN broadcasting. As expected CCN broadcasting presents the highest value for this metric in both scenarios due to its epidemic behavior of sending messages to all nodes in the network.

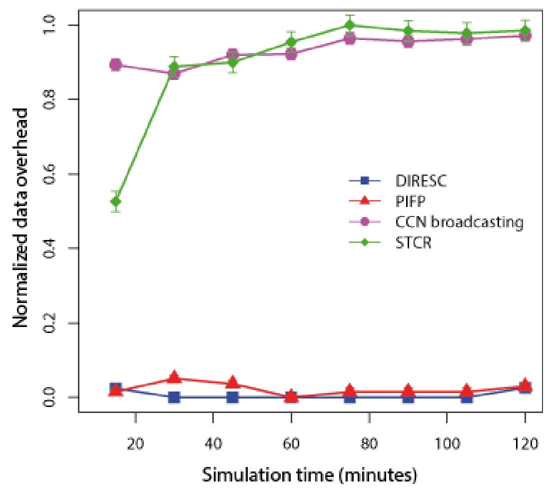


FIGURE 11. Normalized data overhead in San Francisco scenario.

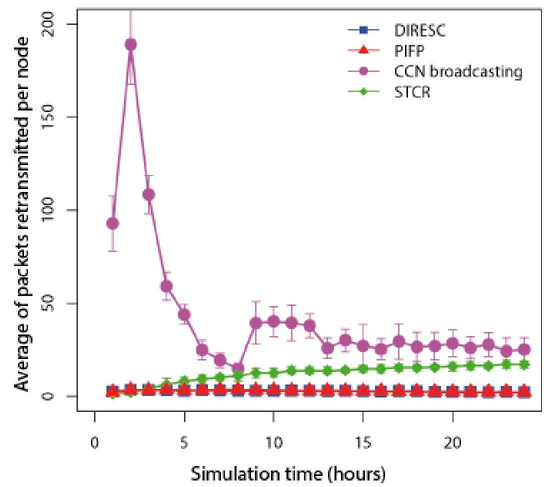


FIGURE 12. Average number of packets retransmitted per node in Helsinki scenario.

The other curves can be explained by the message forwarding mechanisms employed by each protocol that control the message replication in network. DIRESC uses the prioritization scheme, PIFP uses a message retrieval probability scheme, and STCR uses node social centrality for forwarding messages.

E. ENERGY CONSUMPTION

The percentage of energy consumption obtained from our experiments is presented in Figures 14 and 15. We can observe that all the protocols have the battery consumption quite alike, with DIRESC being slightly better in both scenarios. Therefore, we state that despite the more sophisticated message forwarding mechanism proposed by DIRESC, it did not increase the energy consumption of devices when compared to the other protocols evaluated. In other words, DIRESC was able to reduce overhead and delay in package delivery, and to increase the delivery rate, without incurring further battery consumption.

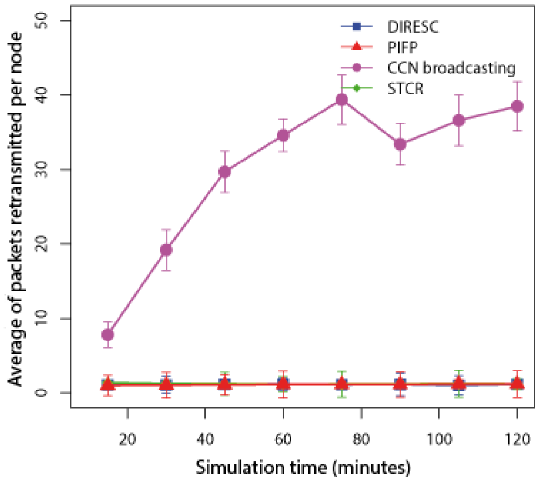


FIGURE 13. Average number of packets retransmitted per node in San Francisco scenario.

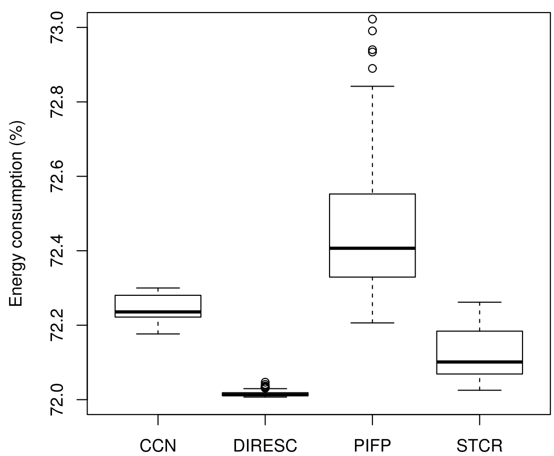


FIGURE 14. Percentage energy consumption in Helsinki scenario.

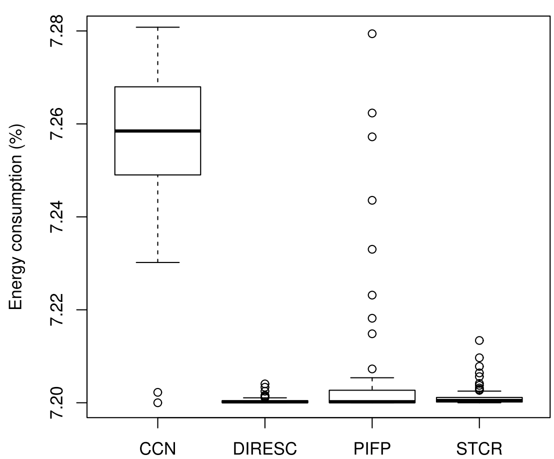


FIGURE 15. Percentage energy consumption in San Francisco scenario.

VI. CONCLUSION

In this work, we have proposed DIRESC, a protocol for data discovery and retrieval based on social cooperation. DIRESC was designed to operate in Content-Centric and Delay-Tolerant Networks (CCDTNs) and to pursue the following goals: increase the rate of data delivery; reduce delay and overhead; and balance the burden of forwarding, i.e., avoid the problem of the *preferred nodes*. Our proposal employs the concept of *friends*, i.e., nodes with high social-tie values, to maximize the chances of success of both requests (interest) and responses (data or content). This strategy is combined with dissemination thresholds and a packet forwarding priority scheme, which allows controlling the network overhead.

In order to evaluate the DIRESC and compare it with some other state-of-the-art solutions, we have implemented a CCN architecture in ONE simulator. Based in two scenarios, a synthetic mobility model from Helsinki city and real-world mobility trace from San Francisco city, the performance of the DIRESC protocol was compared to CCN broadcasting [4], PIFP [39], and STCR [23]. The DIRESC outperforms all the considered protocols, with delivery data rate gains ranging from 17.1% up to 24.9% when compared to the other solutions in Helsinki and San Francisco scenarios, respectively.

As future work, we plan to investigate the potential of using the similarity of requests from nodes as a factor of influence for relationship between nodes. This could further improve the DIRESC performance since it allows the protocol to estimate the *friendship* more precisely. Additionally, we aim to make automatic adjustments to Ω and μ parameters. Thus, DIRESC protocol could dynamically adapt itself to the environment, i.e., to the network's density and node's mobility.

REFERENCES

[1] M. Amadeo, C. Campolo, A. Molinaro, and G. Ruggeri, "Content-centric wireless networking: A survey," *Comput. Netw.*, vol. 72, pp. 1–13, Oct. 2014.

[2] L. Zhang, D. Estrin, J. Burke, V. Jacobson, J. D. Thornton, D. K. Smetters, B. Zhang, G. Tsudik, D. Massey, and C. Papadopoulos, "Named data networking (NDN) project," PARC, Palo Alto, CA, USA, Tech. Rep. NDN-0001, 2010.

[3] NFD. (2013). *Named Data Networking Forwarding Daemon*. Accessed: Apr. 19, 2018. [Online]. Available: http://named-data.net/doc/NFD/current/overview.html

[4] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. H. Briggs, and R. L. Braynard, "Networking named content," in *Proc. 5th ACM CoNEXT*, 2009, pp. 1–12.

[5] CCNx. (2009). *The CCNx Project*. Accessed: Apr. 19, 2018. [Online]. Available: https://wiki.fd.io/view/Cicn

[6] M. Conti, C. Boldrini, S. S. Kanhere, E. Mingozzi, E. Pagani, P. M. Ruiz, and M. Younis, "From MANET to people-centric networking: Milestones and open research challenges," *Comput. Commun.*, vol. 71, pp. 1–21, Nov. 2015.

[7] P. Santi, *Mobility Models for Next Generation Wireless Networks: Ad Hoc, Vehicular and Mesh Networks*, 1st ed. Hoboken, NJ, USA: Wiley, 2012.

[8] X. Liu, Z. Li, P. Yang, and Y. Dong, "Information-centric mobile ad hoc networks and content routing: A survey," *Ad Hoc Netw.*, vol. 58, pp. 255–268, Apr. 2016.

[9] A. Vahdat and D. Becker, "Epidemic routing for partially connected ad hoc networks," Dept. Comput. Sci., Duke Univ., Durham, NC, USA, Tech. Rep. CS-200006, 2000.

[10] K. Zhu, W. Li, X. Fu, and L. Zhang, "Data routing strategies in opportunistic mobile social networks: Taxonomy and open challenges," *Comput. Netw.*, vol. 93, pp. 183–198, Oct. 2015.

[11] W. Moreira and P. Mendes, "Social-aware forwarding in opportunistic wireless networks: Content awareness or obliviousness?" in *Proc. IEEE Int. Symp. World Wireless, Mobile Multimedia Netw.*, Jun. 2014, pp. 1–6.

[12] W. J. Moreira and P. Mendes, "Social-aware opportunistic routing: The new trend," in *Routing in Opportunistic Networks*. New York, NY, USA: Springer, 2013, pp. 27–68.

[13] W. Moreira, P. Mendes, and S. Sargento, "Social-aware opportunistic routing protocol based on user's interactions and interests," in *Proc. Int. Conf. Ad Hoc Netw.* Cham, Switzerland: Springer, 2013, pp. 100–115.

[14] W. Moreira and P. Mendes, "Impact of human behavior on social opportunistic forwarding," *Ad Hoc Netw.*, vol. 25, pp. 293–302, Feb. 2015.

[15] E. M. Daly and M. Haahr, "Social network analysis for routing in disconnected delay-tolerant MANETs," in *Proc. ACM Int. Symp. Mobile Ad Hoc Netw. Comput.*, 2007, pp. 32–40.

[16] E. M. Daly and M. Haahr, "Social network analysis for information flow in disconnected delay-tolerant MANETs," *IEEE Trans. Mobile Comput.*, vol. 8, no. 5, pp. 606–621, May 2009.

[17] P. Hui, J. Crowcroft, and E. Yoneki, "BUBBLE rap: Social-based forwarding in delay-tolerant networks," *IEEE Trans. Mobile Comput.*, vol. 10, no. 11, pp. 1576–1589, Nov. 2011.

[18] W. Moreira, P. Mendes, and S. Sargento, "Opportunistic routing based on daily routines," in *Proc. IEEE Int. Symp. World Wireless, Mobile Multimedia Netw.*, Jun. 2012, pp. 1–6.

[19] Y. Lu, W. Cheng, L.-C. Tung, and M. Gerla, "Social-tie based content retrieval for delay-tolerant mobile ad-hoc networks," in *Proc. 8th ACM MobiCom Workshop Challenged Netw.*, 2013, pp. 51–52.

[20] C. Yi, A. Afanasyev, I. Moiseenko, L. Wang, B. Zhang, and L. Zhang, "A case for stateful forwarding plane," *Comput. Commun.*, vol. 36, no. 7, pp. 779–791, Jul. 2013.

[21] Y.-T. Yu, R. B. Dilmaghani, S. Calo, M. Y. Sanadidi, and M. Gerla, "Interest propagation in named data manets," in *Proc. IEEE Int. Conf. Comput., Netw. Commun.*, Jan. 2013, pp. 1118–1122.

[22] Y. Lu, B. Zhou, L.-C. Tung, M. Gerla, A. Ramesh, and L. Nagaraja, "Energy-efficient content retrieval in mobile cloud," in *Proc. ACM SIGCOMM Workshop Mobile Cloud Comput.*, 2013, pp. 21–26.

[23] Y. Lu, X. Li, Y.-T. Yu, and M. Gerla, "Information-centric delay-tolerant mobile ad-hoc networks," in *Proc. IEEE INFOCOM*, Apr./May 2014, pp. 428–433.

[24] N. M. Junior and C. A. V. Campos, "SOCLEER: A social-based energy-efficient forwarding protocol for opportunistic networks," in *Proc. Int. Wireless Commun. Mobile Comput. Conf. (IWCMC)*, Aug. 2015, pp. 757–762.

[25] C. D. Souza and C. A. V. Campos. (2017). *Implementation of the DIRESC Protocol and the CCDTN in the ONE Network Simulator*. [Online]. Available: https://github.com/diegodisouzza/CCONE

[26] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Efficient routing in intermittently connected mobile networks: The single-copy case," *IEEE/ACM Trans. Netw.*, vol. 16, no. 1, pp. 63–76, Feb. 2008.

[27] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Spray and wait: An efficient routing scheme for intermittently connected mobile networks," in *Proc. ACM SIGCOMM Workshop Delay-Tolerant Netw.*, 2005, pp. 252–259.

[28] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Spray and focus: Efficient mobility-assisted routing for heterogeneous and correlated mobility," in *Proc. 5th Annu. IEEE Int. Conf. Pervas. Comput. Commun. Workshops*, Mar. 2007, pp. 79–85.

[29] A. Lindgren, A. Doria, and O. Schelén, "Probabilistic routing in intermittently connected networks," *ACM SIGMOBILE Mobile Comput. Commun. Rev.*, vol. 7, no. 3, pp. 19–20, 2003.

[30] S. Ali, J. Qadir, and A. Baig, "Routing protocols in delay tolerant networks—A survey," in *Proc. IEEE 6th Int. Conf. Emerg. Technol. (ICET)*, Oct. 2010, pp. 70–75.

[31] H. M. A. Islam, D. Chatzopoulos, D. Lagutin, P. Hui, and A. Yla-Jaaski, "Boosting the performance of content centric networking using delay tolerant networking mechanisms," *IEEE Access*, vol. 5, pp. 23858–23870, 2017.

[32] H. M. A. Islam, A. Lukyanenko, S. Tarkoma, and A. Yla-Jaaski, "Towards disruption tolerant ICN," in *Proc. IEEE Symp. Comput. Commun. (ISCC)*, Jul. 2015, pp. 212–219.

[33] C. Anastasiades, W. E. M. El Alami, and T. Braun, "Agent-based content retrieval for opportunistic content-centric networks," in *Proc. WWIC*, 2014, pp. 175–188.

[34] C. Anastasiades, T. Braun, and V. A. Siris, "Information-centric networking in mobile and opportunistic networks," in *Wireless Networking for Moving Objects*. Cham, Switzerland: Springer, 2014, pp. 14–30.

[35] C. Anastasiades, T. Schmid, J. Weber, and T. Braun, "Information-centric content retrieval for delay-tolerant networks," *Comput. Netw.*, vol. 107, pp. 194–207, Oct. 2016.

[36] A.-D. Nguyen, P. Sénac, and M. Diaz, "STIgmergy Routing (STIR) for content-centric delay-tolerant networks," in *Proc. LAWDN-Latin-Amer. Workshop Dyn. Netw.*, 2010, p. 4.

[37] A. D. Nguyen, P. Sénac, V. Ramiro, and M. Diaz, "Swarm-based intelligent routing (SIR): A new approach for efficient routing in content centric delay tolerant networks," in *Proc. 9th ACM Int. Symp. Mobility Manage. Wireless Access*, 2011, pp. 137–142.

[38] A. D. Nguyen, P. Sénac, V. Ramiro, and M. Diaz, "Pervasive intelligent routing in content centric delay tolerant networks," in *Proc. IEEE 9th Int. Conf. Dependable, Autonomic Secure Comput.*, Dec. 2011, pp. 178–185.

[39] P. Duarte, J. Macedo, A. D. Costa, M. J. Nicolau, and A. Santos, "A probabilistic interest forwarding protocol for named data delay tolerant networks," in *Proc. 7th AdhocNets*, Sanremo, Italy, vol. 155, 2015, pp. 94–107.

[40] T. Le, H. Kalantarian, and M. Gerla, "Socially-aware content retrieval using random walks in disruption tolerant networks," in *Proc. IEEE 16th Int. Symp. World Wireless, Mobile Multimedia Netw. (WoWMoM)*, Jun. 2015, pp. 1–6.

[41] K. Xu, V. O. K. Li, and J. Chung, "Exploring centrality for message forwarding in opportunistic networks," in *Proc. IEEE Wireless Commun. Netw. Conf.*, Apr. 2010, pp. 1–6.

[42] Y. Lu, M. Gerla, T. Le, V. Rabsatt, and H. Kalantarian, "Community aware content retrieval in disruption-tolerant networks," in *Proc. 13th MED-HOC-NET*, 2014, pp. 172–179.

[43] C. D. Souza, D. L. Ferreira, and C. A. V. Campos, "A protocol for data discovery and retrieval in content-centric and delay-tolerant networks," in *Proc. IEEE 86th Veh. Technol. Conf. (VTC)*, Sep. 2017, pp. 1–5.

[44] D. Lee, J. Choi, J.-H. Kim, S. H. Noh, S. L. Min, Y. Cho, and C. S. Kim, "LRFU: A spectrum of policies that subsumes the least recently used and least frequently used policies," *IEEE Trans. Comput.*, vol. 50, no. 12, pp. 1352–1361, Dec. 2001.

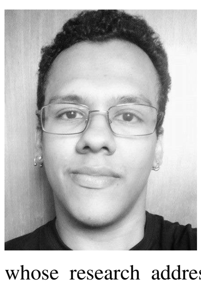
[45] A. Keränen, J. Ott, and T. Kärkkäinen, "The ONE simulator for DTN protocol evaluation," in *Proc. 2nd Int. Conf. Simulation Tools Techn. SIMUTools*, 2009, Art. no. 55.

[46] M. Mosko, I. Solis, and C. A. Wood, *CCNx Semantics*, document draft-irtf-icnrg-ccnxsemantics-10, Internet Engineering Task Force, Jan. 2019. [Online]. Available: https://datatracker.ietf.org/doc/html/draft-irtf-icnrg-ccnxsemantics-10

[47] M. Piorkowski, N. Sarafijanovic-Djukic, and M. Grossglauser. (Feb. 2009). *CRAWDAD Dataset EPFL/Mobility (v. 2009-02-24)*. [Online]. Available: http://crawdad.org/epfl/mobility/20090224

[48] T. Le, Y. Lu, and M. Gerla, "Social caching and content retrieval in disruption tolerant networks (DTNs)," in *Proc. IEEE Int. Conf. Comput., Netw. Commun. (ICNC)*, Feb. 2015, pp. 905–910.

[49] *R: A Language Environment for Statistical Computing*, R Found. Stat. Comput., R Core Team, Vienna, Austria, 2016. [Online]. Available: https://www.R-project.org/



CLÁUDIO DIEGO TEIXEIRA DE SOUZA received the bachelor's degree in computer science from the Federal University of Pará (UFPa), Brazil, in 2014, and the M.Sc. degree in the Computer Science Graduate Program from the Federal University of State of Rio de Janeiro (UNIRIO), Brazil, in 2017. He is currently pursuing the Ph.D. degree in the Systems Engineering and Computer Science Graduate Program with the Federal University of Rio de Janeiro (UFRJ), Brazil, whose research addresses applications running over vehicular networks combined with information-centric networks. His current research interests include wireless networks, opportunistic networks, vehicular networks, information-centric networks, mobile computing, and performance analysis.

DANIELLE LOPES FERREIRA received the B.S. degree in engineering from the State University of Rio de Janeiro (UERJ) and the M.S. degree in computer science from the Federal University of Rio de Janeiro (UFRJ), Brazil. She is currently pursuing the Ph.D. degree in Computer Science Graduate Program with the Federal University of the State of Rio de Janeiro (UNIRIO), Brazil. Since 2018, she has been with the Internetwork-Research Group (i-NRG), Computer Engineering Department, University of California in Santa Cruz's, as a Visiting Ph.D. Researcher. Her current research interests include wireless networks, mobile computing, opportunistic networks, software defined networking, the Internet of Things, and intelligent transportation systems.

CARLOS ALBERTO VIEIRA CAMPOS (S'09–M'12) received the bachelor's degree in computer science from the State University of Montes Claros, Brazil, in 2000, and the M.Sc. and D.Sc. degrees in systems engineering and computer science from the Federal University of Rio de Janeiro, Brazil, in 2003 and 2009, respectively. From 2018 to 2019, he was a Visiting Professor with the Department of Electrical and Computer Engineering, University of Toronto, Canada. He is currently an Associate Professor with the Federal University of the State of Rio de Janeiro (UNIRIO), Rio de Janeiro, Brazil. He has published several articles. His current research interests include user mobility modeling, wireless networks, vehicular networks, mobile computing, and intelligent transportation systems. He served as a technical committee member for the ISCC, ICCVE, VTC, and WCNC. In addition, he served as a Reviewer for the *IEEE Communications Magazine*, the *IEEE Vehicular Technology Magazine*, *Ad Hoc Networks*, and *Computer Communications*.

ANTONIO CARLOS DE OLIVEIRA JÚNIOR was born in Goiânia, Goiás, Brazil, in 1979. He received the B.S. degree in computing and the M.S. degree in electrical engineering from the Federal University of Uberlandia, Brazil, in 2000 and 2003, respectively, and the Ph.D. degree in computer science (Doctoral Programme in Computer Science) from Minho, Aveiro, and Porto Universities (MAP-i), Portugal, in 2014. From 2007 to 2008, he was a Researcher of Telecommunications Institute (IT-Porto) and from 2008 to 2010, he was a Researcher of INESC-TEC, Porto. From 2011 to 2014, he was a Researcher of SITILabs and COPELABS (Association for the Research and Development of Cognition and People-centric Computing), Lisbon. Since 2004, he has been a Professor with the Institute of Informatics (INF), Federal University of Goiás (UFG) and also a Postdoctoral Researcher (Invited Senior Scientist) with Fraunhofer Portugal AICOS. He is currently an Adjunct Coordinator of GT-SOFTWAY4IoT funded by RNP and a member of LABORAtory (Computer Networks and Distributed Systems) mainly involved in networking area, such as the Internet of Things, smart city, wireless networks, software defined networking, and energy efficiency.

KLEBER VIEIRA CARDOSO received the degree in computer science from the Federal University of Goiás, in 1997, and the master's and Ph.D. degrees in electrical engineering from COPPE, Federal University of Rio de Janeiro, in 2002 and 2009, respectively. In 2015, he spent his sabbatical at Virginia Tech, USA. He has participated in some international research projects, involving Brazil and European countries, and has coordinated several Brazilian research and development projects. He is currently an Associate Professor with the Institute of Informatics, Federal University of Goiás, where he has been teaching and researching, since 2009. His current research interests include wireless networks, software defined networks, virtualization, resource allocation, and performance evaluation.

WALDIR MOREIRA received the bachelor's and master's degrees in computer science, in 2005 and 2008, respectively, and the Ph.D. degree in telecommunications from the Universidades do Minho, Aveiro e Porto, in 2014. He started the research career at GERCOM, UFPA, Brazil, in 2006, passing by INESC-Porto, Portugal, in 2009, COPELABS, Portugal, in 2010, and Regional Catalão, UFG, Brazil, in 2015, where he was involved in different national and European projects either as a Team Member or a Coordinator. He is currently a Senior Scientist with the Fraunhofer Portugal Research Center for Assistive Information and Communication Solutions (AICOS) involved in the field of information and communication technologies for development (ICT4D). His current research interests and publications are related to wireless adhoc, mesh, social-aware, cooperative, opportunistic, information-centric, and software-defined networking and routing.