Received: 25 July 2018 | Revised: 10 July 2019 | Accepted: 12 August 2019
DOI: 10.1002/spe.2750

SPECIAL ISSUE PAPER

WILEY

Socially aware microcloud service overlay optimization in community networks

Nuno Apolónia[1,2] | Felix Freitag[1] | Leandro Navarro[1] | Sarunas Girdzijauskas[2]

[1]Universitat Politècnica de Catalunya, Barcelona, Spain

[2]KTH Royal Institute of Technology, Stockholm, Sweden

Correspondence
Nuno Apolónia, Edifici C6-E208 C.Jordi Girona, 1-3, 08034 Barcelona, Spain.
Email: apolonia@ac.upc.edu

Summary

Community networks are a growing network cooperation effort by citizens to build and maintain Internet infrastructure in regions that are not available. Adding that, to bring cloud services to community networks (CNs), microclouds were started as an edge cloud computing model where members cooperate using resources. Therefore, enhancing routing for services in CNs is an attractive paradigm that benefits the infrastructure. The problem is the growing consumption of resources for disseminating messages in the CN environment. This is because the services that build their overlay networks are oblivious to the underlying workload patterns that arise from social cooperation in CNs. In this paper, we propose Select in Community Networks (SELECTinCN), which enhances the overlay creation for pub/sub systems over peer-to-peer (P2P) networks. Moreover, SELECTinCN includes social information based on cooperation within CNs by exploiting the social aspects of the community of practice. Our work organizes the peers in a ring topology and provides an adaptive P2P connection establishment algorithm, where each peer identifies the number of connections needed based on the social structure and user availability. This allows us to propagate messages using a reduced number of hops, thus providing an efficient heuristic to an NP-hard problem that maps the workload graph to the structured P2P overlays resulting in a number of messages close to the theoretical minimum. Experiments show that, by using social network information, SELECTinCN reduces the number of relay nodes by up to 89% using the community of practice information versus the state-of-the-art pub/sub notification systems given as baseline.

KEYWORDS

community networks, community of practice, microclouds, P2P overlay networks, social networks

1 | INTRODUCTION

Community networks (CNs) are a growing infrastructure for users to access the Internet and microcloud services. The CNs were created to fill a void where Internet providers considered areas too remote and not sufficiently profitable. Users in such areas have gathered together in social collaborations to provide free and open Internet to all members in such locations. Furthermore, CNs can be viewed as a community of practice (CoP), where users collaborate to fulfill common goals. In large networks with more than 35 000 nodes, such as Guifi.net,* collaboration is done mostly within areas or groups of people instead of the entire network. In addition, these networks present challenges to the members. Members contribute toward collaborative goals such as adding new devices (antennas and routers) to increase network capacity or adding new services to the network (Internet proxies, FTPs, cameras, and videos).†

A CoP is a common group of people who gather and complete tasks to achieve a common goal.[1] Collaboration is a key step in developing ideas or infrastructures that support their goals. In this respect, CNs are viewed as CoP applications when it comes to setting up network devices, augmenting network abilities or even supporting new cloud services at microcloud levels. The CoP can also provide social interaction between members of the network, which can help optimize the network routing and infrastructure built by the members.

Edge cloud computing is an emerging environment in which services are deployed at the edges of the network by taking advantage of edge devices to perform computational tasks without the need to go to the Internet or data centers. Therefore, edge cloud computing is an attempt to minimize the effect of long-distance communication. Edge cloud computing does not aim to replace traditional clouds but to complement them, where immediate and common computing is done at the edge while the most intensive work is still done in the traditional cloud. The idea of edge cloud computing is to introduce microclouds into CNs so that each area of the network collaborates to minimize the service requirements for outside resources. To further attempt to optimize such solutions, we review an adaptation for service deployment according to network properties.[2]

To complement microclouds in CNs, we need to consider the social behavior of the members to understand how people behave in the network where the CoP plays a role as an environment and to understand the expected behavior of the services with respect to the provided social properties. The level of interaction and collaboration between people must be respected and transposed toward the service behavior to have a fair system and motivate further interest in the services.

We find that building microclouds can be a solution to minimize the dependency on the outside network (Internet) within CNs. In fact, with the broader use of microclouds, such networks can minimize Internet interaction, favoring services that are already available on the CN and diminishing the traffic on the Internet. However, issues also arise as microcloud services come into play such as their performance with constrained devices, the communication latency in both networks, and the motivation to use newly created services rather than well-established Internet services. We can also establish that service routing is a key step toward optimizing services while making them attractive to members. However, such routing is only performed by assessing the network itself, which may not lead to collaboratively using devices on the network, disregarding social interactions.

Our work aids in the optimization of overlays within microclouds on CNs by using CoP information to build and optimize relay nodes when information is disseminated. Services within the CN microclouds focus on peer-to-peer (P2P) dissemination and run with constrained devices.[3,4] Therefore, with CoP information for social support, we aim to build network overlays that prioritize information according to how users contribute to the network. Nevertheless, all nodes should be guaranteed to receive their fair share of the services. The resulting overlay created through Select in Community Networks (SELECTinCN) disseminates the data through nodes that contribute more without losing those that contribute less.

When applying SELECTinCN on services in microclouds, we need to consider that each node (or user) contributes to the community in various ways, such as contributing network links, devices, or capital to construct area antennas or faster links. However, we see that each user may not be concerned about using all the services available in the microclouds in the same way. Therefore, we can establish a contribution as a component of disseminating data for each service.

In summary, we define our contributions as follows.

- Using the CoP information as the main source for relationships between users, we propose optimizing service overlays within microclouds in CNs that exploit both social graphs and cooperation between members.
- We describe and evaluate a reassignment process that projects the social graph onto the P2P overlay network, minimizing the distance of the overlay networks' ID space.
- The SELECTinCN algorithm defines the connections that allow for message propagation with a minimum number of hops when using CoP information instead of friendship graphs. It is adaptive to dynamic environments with the use of novel recovery mechanisms, applying rerouting when required.

*http://guifi.net
†http://guifi.net/node/3671/view/services

- To understand the value and viability of using CoP information to enhance service overlays in microclouds, we evaluate and analyze this using a simulation environment.

To prove the viability and scalability of our proposed system, we conducted simulations on large-scale datasets with thousands to millions of users collected through Facebook and Twitter. We experimentally show that this social graph exploitation reduces the number of hops required for dissemination by over 64% and reduces the number of relay nodes by over 89% against the state-of-the-art approaches. Moreover, we compare our previous results with experiments with CoP information gathered from CNs. The results are in the same range as the social network information, thus demonstrating its viability for use within the microclouds in CNs.

The rest of the paper is organized as follows. In Section 2, we provide an information on the CoP and micro\clouds and the background of the structured P2P topology construction protocol. The design of SELECTinCN, our proposed solution for optimizing overlays in microclouds, is presented in Section 3. In Section 4, we present an extensive evaluation of our proposed approach against the state-of-the-art approaches and discuss the results in Section 5. In Section 6, we conduct a review of the related work. The last section concludes and specifies the direction for future work.

2 | BACKGROUND

2.1 | Community of practice

Communities of practice (CoPs) are a group of people socially involved in a given domain or goal. Community networks (CNs) can be perceived as CoPs because each individual person in the network aims to make their network sustainable for Internet or other services. In fact, in Guifi.net, one of the largest CNs, we observe several groups through the forum and mailing lists used for communication between members. Such groups are formed to tackle issues that are relevant within the same geographic zones. Each group is considered a CoP group that aims to gather information or even devices that can augment their capacity or performance with respect to the location. In addition, the CoP can be viewed as a social network with the same properties as small-world networks.

The activity in CNs can be considered a collaboration of individuals to guarantee that the network is stable, free, and open to anyone that wishes to join. Moreover, a CoP system within CNs is naturally established with goals, such as building main antennas, joining highly connected localities with fiber optics to increase Internet usability, or even making edge cloud devices available to be shared within the community for their own services.

The fact that CoPs can be considered social networks regarding the collaboration of individuals can be used to augment the knowledge of the network to improve cloud services. However, such an approach has not been addressed in the literature. Instead, social networks, such as Facebook, Twitter, and others, have been assumed to be the main point for social interactions.

2.2 | Microclouds

Edge cloud computing is a case of cloud computing by moving computations to the resources at the network edge, thus fully using the edge resources and relying less on Internet cloud resources. In this way, edge cloud computing works to share its own services and resources without going outside of the local network (ie., to the Internet) to use cloud services.

There are significant differences between classic cloud environments and edge clouds. An important characteristic is the use of distributed low-capacity devices instead of centralized data centers with powerful computing devices. Moreover, the network between devices has a higher variance in latency and bandwidth between devices than traditional data centers.

Our case study of edge cloud computing in CNs includes microclouds built with Cloudy.[‡] Cloudy is a distribution based on Debian GNU/Linux, which is intended to be used by common users. Therefore, Cloudy's development was driven by important aspects such as the ease of use, deployment in low-capacity devices, automated service discovery, and service preconfiguration.

Furthermore, community services are included in the distribution to facilitate the process for edge cloud computing (eg, PeerStreamer for peer-to-peer based live streaming, Tahoe-LAFS for decentralized storage service, and Syncthing[§] for data synchronization between various storage nodes, among others). In addition, the shared services within Cloudy are expected to be automatically announced to the network (published/unpublished) when initiated by the users.

[‡] http://cloudy.community

[§] https://www.syncthing.net/

In Cloudy, services can use an overlay network created through existing technologies, such as Serf,[¶] which specifically clusters nodes and manages service availability in the CN microclouds.

2.3 | Peer-to-peer networks

A P2P overlay network consists of a set of N peers $\mathcal{P}(|\mathcal{P}| = N)$. The identifiers of the peers are assigned from an ID space $\mathcal{I}$ on the unit interval $\mathcal{I} \in [0 \dots 1)$ using a uniform mapping function (SHA-1). A distance function $d_{\mathcal{I}}(u, v)$ indicates the distance between peer $u \in \mathcal{P}$ and peer $v \in \mathcal{P}$ in the ID space $\mathcal{I}$. Each peer $p \in \mathcal{P}$ maintains *short-range* links $\mathcal{R}_p^s \subset \mathcal{P}$ that are peers with the minimum distance $d_{\mathcal{I}}(u, v)$ in the ID space $\mathcal{I}$ and *long-range* links $\mathcal{R}_p^l \subset \mathcal{P}$ established with a probability that is inversely proportional to the distance between the peers. The *short-range* links $\mathcal{R}_p^s$ and *long-range* links $\mathcal{R}_p^l$ of each peer p form the routing table $\mathcal{R}_p = \mathcal{R}_p^s + \mathcal{R}_p^l$, where $|\mathcal{R}_p| \ll N$, usually $|\mathcal{R}_p| = logN$. This is the case in the latest P2P models.[5] To reach all peers within a minimum time, the optimization minimizes connections instead of establishing connections to all peers.

The lookup query from peer p for peer u is routed in a greedy fashion (ie, peer p selects the neighbor $w \in \mathcal{R}_p$ that minimizes the distance $d_{\mathcal{I}(w,u)}$ in the ID space $\mathcal{I}$ to forward the query). The lookup process forms an h-hop path $p \rightarrow^{+} u$ with $h = |p \rightarrow^{+} u| \geq 1$. Based on the selection process of the links $\mathcal{R}_p$, the P2P overlay networks can guarantee that the h-hop path is bounded in $O(logN)$.

Moreover, peers are heterogeneous in terms of their connectivity characteristics. Different peers present different bandwidth capabilities bw_p, which affects the rate at which a peer can send and receive packets. Therefore, the time data take to reach peer u from peer p can be given by $l(p, u) = \sum_{i=1}^{h} bw_i$. The propagation of messages from peer p to peer u is calculated based on the bandwidth capability that each peer $w \in h$ maintains in the h-hop path between peers p and u.

A routing tree RT_b is constructed to disseminate the message to all members $\mathcal{M}_b$. Thus, the edges of the routing tree connect social peers that receive and forward the message until all members receive it (in a publisher/subscriber fashion).

The relay nodes $r \in RT_b$ are viewed as the edges of the routing tree RT_b in the path between a publisher and its subscribers, thus allowing relay nodes to be subscribers themselves, depending on the social graph.

The focus of our work is to minimize the number of relay nodes used to disseminate a message on the P2P system within microclouds and to reduce the dissemination latency, even when including node churn. In addition, we aim to optimize the current overlay network that exists in microclouds by adding social information with the network status.

Although the routing tree RT guarantees the dissemination of the message to all required members, it suffers from a high number of relay nodes. This happens because social users that are connected in the routing tree RT are not necessarily directly connected in the P2P overlay network. Hence, each edge in the routing tree consists of $O(logN)$ relay nodes. We must ensure that all social users will receive propagated messages regardless of the structure of the social network or CoP.

Therefore, a P2P substrate that minimizes the number of relay nodes in the routing tree RT is required. We define the problem of minimizing the relay nodes as follows.

Given a publisher b and a set of subscribers S_b, each peer $p \in \mathcal{P}$ aims to establish links in the routing table $\mathcal{R}_p$ such that the routing tree for publisher b and subscriber $s \in \mathcal{S}_b$ contains the minimum number of relay nodes $\mathcal{S}_b = \{s \in \mathcal{S} | f(s, b) = false\}$, granting a near theoretical optima minimal solution.

3 | SELECTINCN SYSTEM

SELECTinCN aims to construct a global P2P overlay network that establishes connections between peers that host social connections. Moreover, SELECTinCN seeks to organize close socially connected peers in the overlay network to reduce the number of hops required for the routing process. The intuition behind this is to provide a P2P substrate that reduces the number of hops between two socially connected peers and to maintain the minimum number of relay nodes of the routing tree RT_b for each publisher $b \in \mathcal{B}$. Finally, the goal of SELECTinCN is to optimize service overlays within microclouds in CNs while having a low latency influence.

3.1 | Applying the community of practice

The information gathered from the CoP differs from the usual social networks. Instead of relationships between users, such as friends and friends of friends, it uses the concept of interactions between people (cooperation between members).

[¶] https://serf.io/

D_p	the identifier for peer p
$\mathcal{R}_p$	a set of peer identifiers to which peer p is connected
$\mathcal{C}_p$	a set of peer identifiers that host the user relations of peer p
$\mathcal{L}_p$	a set of connections that peer $v \in \mathcal{R}_p$ maintains

TABLE 1 Local state of peer p and listing of local variables for a given peer

We exploit the mailing lists (as an alternative to social networks) to establish interactions between people and understand how people cooperate. Therefore, by establishing the relations as the cooperation between the users, it is an easier process to identify the main users who cooperate with others.

The mailing list information was gathered from the Guifi.net mailing lists.[#] We assume that the mailing lists are used exclusively for cooperation within the network and for network enhancement. Thus, users interact with each other in a variety of cooperation projects. For example, topics on the mailing lists include the installation of new routers/devices and services to be used by the community. From such lists, we identified the users (by email) and cross-referenced all the posts to establish common links between users. Therefore, we established the strength of cooperation for each user when they appear in different threads. In addition, the lists are primarily separate for geographically separate locations. Thus, we can add users that collaborate between different locations and those that only cooperate within the same region. Other types of CoPs should be usable as long as relations can be established between members, such as in the case of real-life meetings in which members come together to discuss and even deploy devices in the field.

The number of mailing list users is a small percentage of the total users of the network because only some of the people cooperate with others using mailing lists. We find that most people tend to install their devices without much guidance and let the network itself automatically configure routing and cooperation with other devices. Some of the network users also use real-life meetings for cooperation.

However, the CoP over CNs follows small-world properties, where the clustering coefficient is not small and the distance between nodes grows logarithmically.

Our system model consists of a set of peers $\mathcal{P}$ and a set of social users $\mathcal{V}$, where the social users are members of mailing lists. Each social user $u \in \mathcal{V}$ is mapped onto only one peer $p \in \mathcal{P}$. We assume that peers communicate with each other over reliable channels (eg, TCP connections) that bound the number of connections each peer maintains.

Each peer maintains a set of four local variables listed in Table 1. The first variable, D_p, is the identifier of the peer p and defines the position of peer p in the ID space $\mathcal{I} \in [0 \ldots 1)$. SELECTinCN organizes the socially connected peers within a close distance in the overlay network. Thus, each peer p modifies its identifier D_p to minimize the distance $d_{\mathcal{I}(p,u)}$ to its most important peer u. We measure the importance between two peers using the strength of cooperation between two users in the social graph. Using the mailing lists, we determine where each pair of members communicates with each other, strengthening their relationship in multiple threads $\mathcal{N}_t$. Therefore, we define the social strength between two peers p and u as follows:

$$s(p,u) = \frac{|\mathcal{C}_p \cap \mathcal{C}_u| * \mathcal{N}_t}{\mathcal{C}_p}, p, u \in \mathcal{V} \tag{1}$$

3.2 | System overview

The SELECTinCN system is built with three main processes:

- ***Projection***: SELECTinCN associates the position of each peer that hosts a social user in the overlay network. The position of each peer is used to define the distance between two socially connected peers in the ID space $\mathcal{I}$. When a peer joins the overlay network, the local variables of Table 1 are initialized.
- ***Identifier reassignment***: SELECTinCN evaluates the peer positions in the ID space and reassigns the identifiers on a round-based basis. Specifically, each peer leverages the social information and modifies its identifier to reduce its distance in the overlay network with users that have higher collaboration relations.
- ***Peer connection establishment and reassignment***: While SELECTinCN organizes the socially connected peers in the same area in the ID space $\mathcal{I}$, each peer establishes direct connections to peers that are part of the CoP. A link reassignment process is performed to ensure that each social peer communicates with the maximum number of CoP members in the minimum required hops with the minimum dissemination latency.

[#] https://llistes.guifi.net/sympa

Both the peer identifier and connection reassignment processes use a gossip-based peer-sampling methodology to evaluate the defined topology.

3.3 | Projection and identifier reassignment

The projection process determines the initial position of the peer that is perceived by the underlying overlay network. As shown in Algorithm 1, the projection of the social user in the overlay network is specified based on the relations of the user (line 1). Thus, the assigned identifier D_p is reduced in terms of the distance (line 3) between peer u and peer v when user v has a relationship with other users. Otherwise, a random identifier is assigned to peer v using a uniform hash function (line 5).

Procedure 1 Peer identifier assignment

```
Input: $v \in \mathcal{V}$ newly registered social users
Output: $D_p$ peer identifier
1: if $\mathcal{C}_v \neq \emptyset$ then
2:     u ← the peer of the social user that invited $v$
3:     $D_p \leftarrow min_D d_{\mathcal{I}}(u, v)$
4: else
5:     $D_p \leftarrow$ UNIFORMHASH($v$)
6: end if
7: Return $D_p$
```

As the CoP grows, social users interact more often, and the social strength in Equation (1) between two users is modified. The goal is to reduce the distance in the ID space $\mathcal{I}$ between social users. Each peer modifies its identifier and moves closer to the peer that hosts a collaboration (social peer) with higher social strength.

The new position choice is the centroid of all its social relations. To address the problem using only the highest connected users, we use the centroid between the two social peers that maintain the highest social strength value, as presented in Algorithm 2, to avoid congestion and to avoid centering the network on only the most active members.

Procedure 2 Peer identifier reassignment

```
Procedure EVALUATEPOSITION()
    $u \leftarrow$ the peer with the highest social strength in $\mathcal{C}_p$;
    $v \leftarrow$ the peer with the second highest social strength in $\mathcal{C}_p$;
    $D_p \leftarrow \frac{|d_{\mathcal{I}(u,v)}|}{2}$;
end Procedure
```

Complexity analysis: For each peer $p \in \mathcal{P}$, the initial position in the overlay network is calculated in $O(1)$ because the identifier is assigned either uniformly or based on a user connection. Thus, the initial projection of the social graph in the P2P topology requires $O(N)$ complexity. The reassignment of the peer identifiers based on the peer-sampling protocol requires $O(|\mathcal{C}_p|)$ complexity for each peer, where $|\mathcal{C}_p| \ll N$. In modern social networks, $|\mathcal{C}_p|$ is usually in the range of hundreds of social friends, while the size of the network N is billions of users.[6] For CNs, the numbers are smaller in comparison; however, they have grown over the last few years. Thus, N is around 30 000, whereas the social interactions on a local level may be less than a hundred. Summarizing, the total complexity of the projection and identifier reassignment algorithm is as follows:

$$O(N \cdot |\mathcal{C}_p|) \tag{2}$$

3.4 | Peer connection establishment and reassignment

SELECTinCN uses a gossip-based peer-sampling service to construct the topology in the overlay network. Each peer periodically acquires its social neighbor's connections in the overlay network and evaluates its current established connections. Specifically, each peer p seeks to establish direct connections with the maximum number of peers in its social neighborhood $\mathcal{C}_p$.

To create a ring topology, each peer is allowed to accept only K incoming links while maintaining two *short-range* outgoing links $\mathcal{R}_p^s$ with its successor and predecessor in the overlay network and K *long-range* outgoing links $\mathcal{R}_p^l$ with its social peers. The intuition behind the K incoming links is to avoid having peers that have too many connections because other peers seek to connect to them, creating more traffic than others. When the K incoming links are established, the peer accepts a new incoming connection if the new connection has a better bandwidth capacity than the already-existing connections. The K outgoing *long-range* links are selected by applying the locality sensitive hashing (LSH) technique to the social neighbor's connections retrieved from the peer-sampling service (lines 3-6 and 2-8 in Algorithms 3 and 4, respectively). The LSH technique is used to identify the peers that maintain different connections and to avoid any link overlap in the overlay network.

Procedure 3 Peer-sampling - Active thread

```
1: Procedure EXCHANGERT()
2:     socialFriend ← getRandomSocialFriendPeer();
3:     Send < $\mathcal{C}_p, \mathcal{R}_p$ > to socialFriend;
4:     Receive < $nMutual, M$ > from socialFriend;
5:     socialFriend.nMutual = nMutual;
6:     socialFriend.M = M;
7:     $D_p$ ← evaluatePosition();
8:     $\mathcal{R}_p$ ← createLinks();
9: end Procedure
```

Procedure 4 Peer-sampling - Passive thread

```
1: Procedure RESPONSEEXCHANGERT()
2:     Receive < $\mathcal{C}_u, \mathcal{R}_u$ > from socialFriend;
3:     nMutual ← $|\mathcal{C}_u$.merge($\mathcal{C}_p$)$|$;
4:     socialFriend.nMutual ← nMutual;
5:     $M$ ← $\mathcal{C}_u$.constructFriendshipBitmap($\mathcal{R}_p$);
6:     Send < $nMutual, M$ > to socialFriend;
7:     $M'$ ← $\mathcal{C}_p$.constructFriendshipBitmap($\mathcal{R}_u$);
8:     socialFriend.bitMap = $M'$;
9:     $D_p$ ← evaluatePosition();
10:    $\mathcal{R}_p$ ← createLinks();
11: end Procedure
```

The connection establishment mechanism is shown in Algorithm 5. We begin by indexing the bitmaps of the social neighborhood in $\mathcal{H}$ buckets in the LSH index (lines 2-4). In our algorithm, we consider the number of buckets to be equal to the number of defined *long-range* links ($|\mathcal{H}| = K$). The reason for selecting $|\mathcal{H}| = K$ buckets in the LSH index is to simplify the selection process of the direct connections. Peers whose connections are similar will be indexed in the same bucket. This results in selecting only one peer of each bucket to establish at most K long-range links.

The bitmap of $u \in \mathcal{C}_p$ is an array of size $|\mathcal{C}_p|$, the values of which define the link existence in $\mathcal{R}_u$ between two socially connected peers $u \in \mathcal{C}_p$ and $v \in \mathcal{C}_p$, where $u \neq v$, as follows:

$$bitmap(u, v) = \begin{cases} 1 & \text{if } (u, v) \in \mathcal{R}_u \\ 0 & \text{if } (u, v) \notin \mathcal{R}_u \end{cases}$$

The bitmaps are indexed in the $\mathcal{H}$ buckets of LSH in lines 5-18, and we aim to select one peer from each bucket $h \in \mathcal{H}$ to establish a connection. To establish connections with the maximum number of peers p in the social neighborhood $\mathcal{C}_p$ with the minimum dissemination latency, we select the peer that presents the maximum value in the bitmap (line 8). In doing so, we ensure that each peer achieves the maximum number of social connections with the minimum number of peer connections (not all K connections are used). Moreover, we drop an already-established connection $(p, u) \in \mathcal{R}_p$ with a peer u that presents similar connections with the newly established connection $(p, v) \in \mathcal{R}_p$ (lines 12-16) in Algorithm 5.

Procedure 5 Peer link reassignment

```
1:  **Procedure** CREATELINKS()
2:      **for** $u \in C_p$ **do**
3:          LSHIndex($u.bitMap$);
4:      **end for**
5:      **for** $h \in \mathcal{H}$ **do**
6:          $\mathcal{P}_h \leftarrow$ peers assigned in the same bucket $h$;
7:          **if** $\mathcal{P}_h \neq \emptyset$ **then**
8:              $u \leftarrow$ picker($\mathcal{P}_h$);
9:              **if** $(p, u) \notin \mathcal{R}_p$ **then**
10:                 $\mathcal{R}_p \leftarrow (p, u)$
11:             **end if**
12:             **for** $v \in \mathcal{P}_h, v \neq u$ **do**
13:                 **if** $(v, p) \in \mathcal{R}_p$ **then**
14:                     $\mathcal{R}_p$, remove(v);
15:                 **end if**
16:             **end for**
17:         **end if**
18:     **end for**
19: **end Procedure**
```

Procedure 6 Picker peer connection

```
**Procedure** PICKER($\mathcal{P}_h$)
    $\mathcal{PS}_h \leftarrow$ sort peers($\mathcal{P}_h$)
    **if** ($|\mathcal{PS}_h| > 0$) && ($\mathcal{PS}_h(0).bw < \mathcal{PS}_h(1).bw$) **then**
        **return** $\mathcal{PS}_h(1)$
    **end if**
    **return** $\mathcal{PS}_h(0)$
**end Procedure**
```

Complexity analysis: The complexity analysis of the *connection establishment and reassignment* algorithm is analogous to the number of social peers $|C_p|$ that each user maintains and the number of buckets $|\mathcal{H}|$ assigned on the LSH index. The peer-sampling protocol aggregates the bitmaps with $O(|C_p|)$ complexity. The index of the bitmaps in $|\mathcal{H}| = K$ buckets requires $O(|C_p| \cdot log(|C_p|) \cdot K)$ complexity. The K *long-range* links are selected using the LSH index at $O(K)$ cost. Summarizing, the total complexity of the *connection establishment and reassignment* algorithm for each peer $p \in \mathcal{P}$ is

$$O(|C_p|^2 \cdot log(|C_p|) \cdot K^2) \tag{3}$$

4 | EVALUATION

For our evaluation, we consider previous experiments done with SELECTinCN and social networks, such as Facebook and Twitter, as the baseline[7] and the experiments with Guifi.net mailing lists as the CoP information. Therefore we show that our proposed solution can be applicable to other community networks, which exhibit the underlying user interaction patterns similar to that of Guifi.net (eg, in collaboration networks such as FunkFeuer, AWMN or Freifunk). The simulation experiments were performed using the Gelly Graph API[‖] running over the Apache Flink[**] distributed data processing

[‖] https://ci.apache.org/projects/flink/flink-docs-master/libs/gelly_guide.html

[**] http://flink.apache.org/

Dataset	Users/Peers	Connections/Links	Average Degree
Facebook	63 731	817 090	25.642
Twitter	3 990 418	294 865 207	73.89
CoP-Guifi	3016	16 471	10.9

TABLE 2 Three real-world datasets of social networks that include user information, such as social connections and the average degree of social connection

framework. The experiments were run on a Flink cluster with 20 virtual machines to provide a distributed discrete event simulator suitable for conducting large-scale experiments. For each experimental round, we used all the users/peers from each dataset.

The implementation of SELECTinCN is performed using the vertex-centric iterative model.[8] Specifically, in synchronized iteration steps, each peer produces messages to other peers and updates their identifiers and connections in the overlay network using the SELECTinCN algorithms.

Experiments are performed in evolving networks, where users join the overlay network at different phases. We initiate our experiments by selecting a social peer $u \in \mathcal{V}$ from the dataset at random. Thereafter, we insert a portion of the relationships of user u into the social network, following the model by the work of McCune et al.[9] Therefore, at each iteration step, we select a registered social user and insert a number of the user's social peers into the social graph, which preserves the exponentially decreasing rate of the model. The same process as in the social networks is performed with the CoP graph in which relationships are added to the users with each iteration. In the experiments, we use all the users as peers from each dataset to create the overlay.

4.1 | Datasets

Our evaluation is performed with three real-world datasets listed in Table 2. These datasets cover a wide range of social graph features from less connected graphs, such as Facebook,[10] to highly connected graphs, such as Twitter,[6] which enhances the evaluation of our proposed approach for several graph types. Moreover, we conduct experiments on the CoP-Guifi dataset, based on the gathered mailing list information. The number of nodes in the simulations corresponds to the number of users in the datasets, and the links correspond to the number of connections.

4.2 | Metrics

To measure the efficiency of SELECTinCN, we used the following metrics.

- **Number of hops:** The average number of overlay hops within the path between two peers.
- **Number of relay nodes:** The average number of relay nodes in the pub/sub routing tree.

To validate our analysis, for each metric, we report the average result out of 100 independent trials to decrease the risk of statistical error. We consider these metrics to be important for understanding the behavior of SELECTinCN when different social information is used and for comparing the results with other work while also providing feedback on using SELECTinCN for the domain of CNs.

4.3 | Simulation experiments

We compared SELECTinCN with several existing pub/sub systems of different categories: (i) *Symphony*, a pub/sub system over a P2P overlay network without any further modification on the P2P topology; (ii) *Bayeux*, a pub/sub system that organizes peers into a DHT P2P overlay and builds a spanning tree for each topic to propagate the messages; (iii) *Vitis*, a gossip-based pub/sub system that organizes the subscribers into clusters; and (iv) *OMen*, that constructs topic-connected overlays (TCOs) to disseminate information on each topic.

As the number of direct connections increases, we observe a substantial reduction of over 90% on the average number of hops required for communication between two socially connected peers. However, as the number of links overcomes the logarithmic number of peers in the overlay network, no further improvement is achieved. Based on the above observation, for the rest of the experimentation, we assign log_2N direct connections on each peer to construct a P2P topology.

Figures 1A and 1B present the average number of hops required for a publisher to propagate information to each of the subscribers with respect to Facebook and Twitter information. As the network grows, the average number of hops increases logarithmically. However, SELECTinCN performs with 76% and 83% fewer hops compared to the pub/sub mechanism built over the Symphony overlay network and for the Facebook and Twitter datasets, respectively. This occurs because Symphony's construction of long-range links is completely oblivious to the social graph and publication workload.

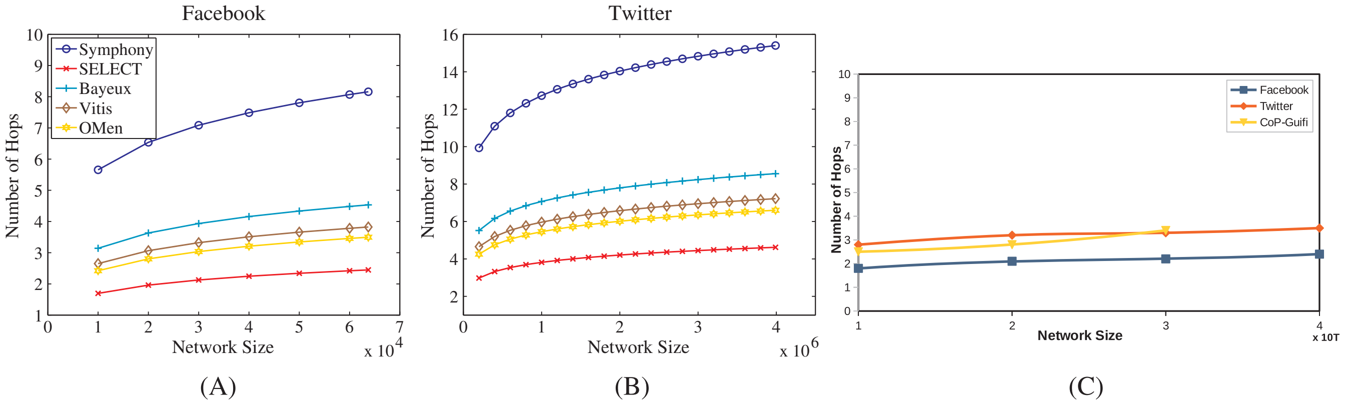

FIGURE 1 Number of hops per social lookup for (A) Facebook and (B) Twitter datasets compared with different algorithms. (C) A comparison using Select with different datasets (Facebook, Twitter, and CoP-Guifi) [Colour figure can be viewed at wileyonlinelibrary.com]

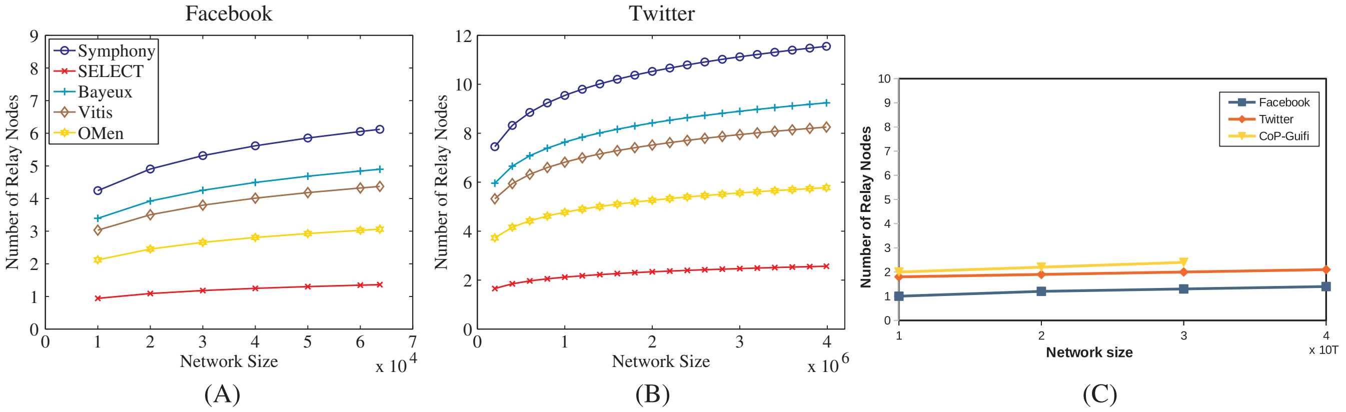

FIGURE 2 Number of relay nodes per pub/sub routing path for (A) Facebook and (B) Twitter datasets. (C) A comparison using Select with different datasets (Facebook, Twitter, and CoP-Guifi) [Colour figure can be viewed at wileyonlinelibrary.com]

In contrast, SELECTinCN establishes connections between socially connected peers, and as such, subscribers are one or two hops away from the publisher. Compared to the state-of-the-art pub/sub approaches, SELECTinCN achieves more than 43% and 61% reduction for Facebook and Twitter datasets, respectively. This happens because the peer identifiers on SELECTinCN are mutable, and socially connected peers are clustered in the same region in the ID space. Hence, a small-world network is accomplished on SELECTinCN, in contrast to the presented approaches, where an immutable identifier policy is applied. Figure 1C illustrates using the CoP information compared to using social networks. The average number of hops is within the same values as for social networks.

Figures 2A and 2B present the effect of SELECTinCN on the number of relay nodes in the routing path between the publisher and subscriber with respect to Facebook and Twitter information. SELECTinCN presents an over 98% reduction in the number of relay nodes for all datasets in comparison to the approaches for Symphony, Bayeux, Vitis, and OMen. In Symphony, Bayeux, Vitis, and OMen, this happens because the probability of connection between two social users in the overlay network is extremely low.

In contrast to the existing pub/sub systems used as a baseline, SELECTinCN leverages the social graph and establishes connections between socially connected peers that reduce the number of relay nodes in the routing path between the publisher and subscriber. The main overhead of SELECTinCN regarding the baseline system consists of the cost of gathering the social information, which is dependent on the specific social network application. In the case of applying SELECTinCN in the assessed CoPs, we built a tool to construct a social graph from the member interactions in the community.

Figure 2C presents the comparison between using Facebook and Twitter data against using the CoP information, and the average relay nodes are also within the same range as in the social networks. Thus, the CoP information can be used with the same results as these social networks. However, the users that are not connected through the CoP are not represented within the information, which can make the number of relays and hops higher on average.

5 | LESSONS LEARNED

Our approach to P2P overlay optimization in CNs relies on the CoP connections provided by the SELECTinCN algorithm. Although it is not possible to provide a fair cost comparison between the state-of-the-art approaches and the SELECTinCN algorithm because all of them rely on different aspects of the network, we clearly see from our experimental results that the actual costs are the added social information necessary to create the relationship graph to augment the global overlay.

With respect to the CoP information, the difference in the number of users is because of the network having fewer members than online social networks. Therefore, in our experiments, the comparison is as close as possible to the maximum number of users in the CoP graph. However, our approach has growth potential even when using CoP information. Furthermore, an extended study on different types of collaborative users is necessary when including all members of CNs, such that all social peers can minimize their own routing toward other peers without being randomized within the overlay. We also assume that other CNs have the same core principle of user collaboration and therefore follow the same pattern of CoP. Thus, the connectivity between members would be similar to that observed from other CNs.

The use of SELECTinCN with microcloud services in CNs is an approach that should be further studied because our experiments were designed to understand the viability of applying CoP information in overlays. In addition, social peers maintain their underlying network connectivity, even though an overlay may optimize the path in which data should flow to peers. Therefore, if this research were applied in a real environment, the network connectivity (eg, latency and underlying hops) should be included as a metric for the creation of the overlay to account for network status and social status. The introduction of network connectivity should be a priority in CNs because minimizing the number of connections in the overlay does not mean it minimizes the underlay connections. In fact, collaborators can be in different parts of the underlay network, whereas collaborators can be at a one-hop distance in the overlay network.

Although the algorithm was designed with pub/sub systems in mind, it can work for microcloud services because these services use pub/sub methods for disseminating data. Thus, applying SELECTinCN in microcloud environments would guarantee optimization for the dissemination of data when coupled with network information, such as latency, bandwidth capacity, and resources. Moreover, by adding network and social information in an evolving environment, we can enhance data dissemination for services.

We ensure the correctness of our approach by grounding it in the ring topology because it allows us to continue sending messages to all peers and guarantees that all peers are able to receive the messages. Other topologies, such as mesh, tend to create isolated communities of nodes. Therefore, using other topologies may not guarantee the same results when applying different social networks or CoPs. In the CoP, some users do not communicate by mailing lists or forums, as used in our experiments; therefore, pockets of users without connections would be a possibility with other topologies and thus should be avoided by using a ring topology.

A more extensive study on the CoP would be necessary to include other members and information that are not available through the mailing lists and forums (eg, meetings where members meet face-to-face without the use of email). Moreover, the CoP information and relationships must be explored to account for the aspect of collaboration over time, where people can collaborate for a given time until a task is completed, with no further collaboration needed.

6 | RELATED WORK

In,[11] the authors reported on how the deployment of the cloud model on top of guifi.net was undertaken. They elaborated on a system where users can benefit from cloud-based services inside the network without having to consume services from the Internet. Instead, they can use the resources that already exist on their CN while also granting access to cloud-based services. Their work is based on the same infrastructure and environment; however, they do not account for social networks or the CoP as the primary source to create an overlay to optimize routing.

In other work, the construction of P2P pub/sub systems aims to minimize the number of relay nodes. The proposed approaches are divided into two main categories: (i) the design of a routing tree, the construction of which relies on the routing process of the underlying P2P overlay network[12] and (ii) the construction of a P2P topology, such that the paths in the routing tree contain the minimum number of relay nodes.[13-15]

In the first category, Bayeux[12] organized peers into a DHT, where each peer maintains $O(logN)$ connections. Then, a routing tree was built for each topic with a rendezvous node at the root, which delivers the events to the peers that join the tree. This approach forces many nodes to relay messages for which they have not been subscribed. Consequently, the Bayeux-based systems suffer from high traffic overhead because they fail to minimize the number of relay nodes.

Rahimian et al[14,16] proposed a gossip-based hybrid P2P overlay called Vitis for pub/sub systems. Peers in Vitis are organized in a ring structure and run a gossip-based peer-sampling algorithm to identify the subscription and establish connections so that peers that are interested in similar topics are organized into clusters. Although Vitis manages to reduce the number of relay nodes, peers with a high degree of social connections create high traffic overhead because the rest of the peers aim to connect with the social users that maintain the most social friends in common.

Finally, OMen[17] is one of the most recent approaches that emphasize the design of the P2P overlay network to provide a P2P pub/sub system. OMen[17] incorporated the design of a TCO,[18] which is an approximation of the GM algorithm.[13] Forming a P2P small-world overlay network,[15] each peer in the OMen pub/sub system maintains a *shadow set*, which is a subset of backup peers that maintain the information to repair the TCO when churn occurs. Although OMen provides a fast recovery mechanism while maintaining a low number of relay nodes, the online activity of peers is not monitored, thus causing high traffic overhead for peers that establish connections to peers with an extremely low amount of online behavior.

Other research works[19,20] can be considered improvements to the above approaches in their respective fields. However, each of these studies has still not considered building the overlay with social information, only establishing the social strength between peers as the main part for connectivity or considering using the CoP as the source for social information.

The CoPs are not well documented. In the work of De Marsico et al,[21] they described CoPs as a vertical evolution of social networks, where members share common interests and cooperate with each other to reach certain goals. Based on their study, such CoPs can achieve the same metrics for betweenness, centrality, and closeness as other social networks.

In the work of Xia et al,[22] social awareness was introduced as an emerging paradigm. The authors described how several studies have exploited the knowledge of social properties for building socially aware routing and forwarding protocols. More specifically, in the work of Xia et al,[23] community-based load balancing was proposed, which leverages the knowledge of similar interests of brokers to improve load balancing for brokers in a pub/sub system. In turn, this made the system more robust toward failures.

Therefore, an important and challenging problem that we consider is to build an optimized overlay structure, where our work differs in designing an overlay network that leverages the social graph topology and interactions to organize both the peers and their established connections in the overlay network. Specifically, in our work, the peers are placed in the same area in the overlay network based on their social proximity, and a bounded and adaptive number of connections are established with peers that are also connected in the social graph, avoiding congestion with peers with a high degree of social connections. In contrast to other studies, our approach does not consider brokers or hierarchical overlays. Instead, it provides a solution through community-aware flat P2P overlays that are capable of scaling to an arbitrarily large number of nodes. To guarantee an opportunity to be developed into CNs, we augment the social information with the CoP, which has not been fully explored in other work.

7 | CONCLUSION

In this paper, we approached an optimization for service overlays within microclouds in CNs and proposed SELECTinCN, a distributed publish/subscribe system. We designed a P2P overlay network that exploits the social graph of the CoP in CNs to organize peers in an overlay network and establish connections between socially connected peers. Using a gossip-based peer-sampling service, SELECTinCN reduces the number of hops required to communicate between two socially connected peers. Moreover, the constructed routing trees exhibit the minimum number of relay nodes. We evaluated SELECTinCN in a simulated environment using three real-world datasets and highlighted the performance of SELECTinCN with state-of-the-art approaches as a baseline to compare the use of CoP information.

With respect to our evaluation, using the CoP information is a way to handle social information in CNs to reduce the number of relay nodes in the overlay network, which will be beneficial to enhancing services in CNs.

For future research directions, we consider that the CoP information must be gathered from sources other than mailing lists and should be added to the social information to augment the social information across multiple domains. In addition, the consequences and influence of the overlay creation when running under intensive node churn should be included to study the effects of network devices in the reassignment of nodes in the overlay.

ORCID

Nuno Apolónia https://orcid.org/0000-0002-6590-0179

REFERENCES

1. Serrat O. *Building Communities of Practice*. Singapore: Springer Singapore; 2017.
2. Selimi M, Cerdà-Alabern L, Sánchez-Artigas M, Freitag F, Veiga L. Practical service placement approach for microservices architecture. Paper presented at: 17th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing; 2017; Piscataway, NJ.
3. Apolónia N, Sedar R, Freitag F, Navarro L. Leveraging low-power devices for cloud services in community networks. Paper presented at: 3rd International Conference on Future Internet of Things and Cloud (FiCloud); 2015; Rome, Italy.
4. Saovapakhiran B, Devetsikiotis M. Enhancing computing power by exploiting underutilized resources in the community cloud. Paper presented at: IEEE International Conference on Communications (ICC 2011); 2011; Kyoto, Japan.
5. Dhara K, Guo Y, Kolberg M, Wu X. *Overview of Structured Peer-to-Peer Overlay Algorithms*. Boston, MA: Springer US; 2010.
6. Leskovec J, Krevl A. SNAP datasets: Stanford large network dataset collection. 2014. http://snap.stanford.edu/data
7. Apolónia N, Antaris S, Girdzijauskas S, Pallis G, Dikaiakos M. Select: a distributed publish/subscribe notification system for online social networks. Paper presented at: IEEE International Parallel and Distributed Processing Symposium (IPDPS); 2018; Vancouver, Canada.
8. McCune RR, Weninger T, Madey G. Thinking like a vertex: a survey of vertex-centric frameworks for large-scale distributed graph processing. *ACM Comput Surv*. 2015;48(2).
9. Zhu K, Li W, Fu X. Modeling population growth in online social networks. *Complex Adapt Syst Model*. 2013;1(1).
10. Viswanath B, Mislove A, Cha M, Gummadi K. On the evolution of user interaction in Facebook. In: Proceedings of the ACM Workshop on Online Social Networks; 2009; Barcelona, Spain.
11. Jimenez J, Baig R, Esrich P, et al. Supporting cloud deployment in the Guifi.net community network. Paper presented at: 5th Global Information Infrastructure and Networking Symposium (GIIS 2013); 2013; Trento, Italy.
12. Zhuang SQ, Zhao BY, Joseph AD, Katz RH, Kubiatowicz JD. Bayeux: An architecture for scalable and fault-tolerant wide-area data dissemination. Paper presented at: 11th International Workshop on Network and Operating Systems Support for Digital Audio and Video; 2001; Port Jefferson, NY.
13. Chockler G, Melamed R, Tock Y, Vitenberg R. Constructing scalable overlays for pub-sub with many topics. Paper presented at: Annual ACM Symposium on Principles of Distributed Computing; 2007; Salerno, Italy.
14. Rahimian F, Girdzijauskas S, Payberah AH, Haridi S. Vitis: A gossip-based hybrid overlay for internet-scale publish/subscribe enabling rendezvous routing in unstructured overlay networks. Paper presented at: IEEE International Conference on Parallel Distributed Processing Symposium (IPDPS); 2011; Anchorage, AK.
15. Chen C, Tock Y. Design of routing protocols and overlay topologies for topic-based publish/subscribe on small-world networks. Paper presented at: Industry Track of the 16th ACM/IFIP/USENIX Middleware Conference (Middleware Industry 2015); 2015; Trento, Italy.
16. Rahimian F, Le Nguyen Huu T, Girdzijauskas S. Locality-awareness in a peer-to-peer publish/subscribe network. Paper presented at: IFIP International Conference on Distributed Applications and Interoperable Systems; 2012; Stockholm, Sweden.
17. Chen C, Vitenberg R, Jacobsen H-A. OMen: Overlay mending for topic-based publish/subscribe systems under Churn. Paper presented at: 10th ACM International Conference on Distributed and Event-Based Systems; 2016; Irvine, CA.
18. Chen C, Jacobsen H-A, Vitenberg R. Algorithms based on divide and conquer for topic-based publish/subscribe overlay design. *ACM/IEEE Trans Netw*. 2014;24(1):422-436.
19. Chockler G, Melamed R, Tock Y, Vitenberg R. SpiderCast: A scalable interest-aware overlay for topic-based pub/sub communication. Paper presented at: Inaugural International Conference on Distributed Event-Based Systems; 2007; Toronto, Canada.
20. Setty V, van Steen M, Vitenberg R, Voulgaris S. PolderCast: Fast, robust, and scalable architecture for P2P topic-based pub/sub. Paper presented at: 13th International Middleware Conference; 2012; Montreal, Canada.
21. De Marsico M, Limongelli C, Sciarrone F, Sterbini A, Temperini M. *Social Network Analysis and Evaluation of Communities of Practice of Teachers: A Case Study*. Cham, Switzerland: Springer International Publishing; 2014.
22. Xia F, Liu L, Li J, Ma J, Vasilakos AV. Socially aware networking: a survey. *IEEE Syst J*. 2015;9(3):904-921.
23. Xia F, Ahmed AM, Yang LT, Luo Z. Community-based event dissemination with optimal load balancing. *IEEE Trans Comput*. 2015;64(7):1857-1869.

How to cite this article: Apolónia N, Freitag F, Navarro L, Girdzijauskas S. Socially aware microcloud service overlay optimization in community networks. *Softw: Pract Exper*. 2020;50:675–687. https://doi.org/10.1002/spe.2750