

Socially-Aware Caching Strategy for Content Centric Networking

César Bernardini^{*†}, Thomas Silverston^{*†} and Olivier Festor[†]

^{*}Université de Lorraine, LORIA, UMR 7503, Vandoeuvre-les-Nancy, F-54506, France

[†]Inria, Villers-les-Nancy, F-54600, France

Email: {name.surname}@inria.fr

Abstract—Content Centric Networking (CCN) emerged as a replacement architecture for the current Internet. CCN resorts to in-network caching to enhance end-user delivery performance. At the same time, Online Social Networks (OSN) have become the common paradigm to exchange information between users. OSNs carry extremely valuable information about their users and their relationships. This knowledge can help to drastically improve the efficiency of CCN.

We present a novel caching strategy for CCN based on social information. We conjecture a small number of users -the Influential users- dominate the activity, receive most attention from other users and produce content more likely to be consumed. Our caching strategy privileges the Influential users and cache pro-actively their content in the network.

Through extensive simulation experiments based on two social network scenarios, LastFM and Facebook, and substantial number of users in a CCN topology, we demonstrate the value of our approach. We also implemented and deployed our strategy on PlanetLab and it improves drastically the caching performances of CCN.

I. INTRODUCTION

Over the past few years Information Centric Networking (ICN) has become a promising new paradigm for the future Internet architecture. It is based on named data, where content address, content retrieval and the content identification is led by its name instead of its physical location. One of the ICN key concepts relies on in-network caching to store multiple copies of data in the network and serve future requests, which helps to reduce the load on servers, congestion in the network and enhance end-users delivery performances. ICN architectures include Content Centric Networking (CCN) [1], NetInf [2] and Pursuit [3] among others and are compared in [4]. Along this work, we focus on CCN due to its wide acceptance in the research community.

In parallel, Online Social Networks (OSN) have gained tremendous popularity on the Internet. Millions of users interact with each other through OSN such as Facebook or Twitter. Facebook announced one billion of users [5] while Twitter is massively used as a micro-blogging service. People create social relationships through OSN and exchange information about what they experience in their life within their community. As an example, the re-election picture of President Obama has become the single most-retweeted message in Twitter history with more than 800,000 re-tweets [6]. New ubiquitous devices (smartphones, tablets) appeared and include

functionalities to instantaneously share information through OSN. Most if not all the Internet services improve the users' experience through the addition of social features to rapidly spread interesting content. Companies invest strongly into their Facebook pages to promote new products and benefit from user's feedback [7]. 90% of American hospitals use social media to attract new customers and one third has a formal social media plan [8]. Users follow friends and families and organize social events through the Internet. The Internet is becoming a social-oriented network.

As a central component of CCN is in-network caching, the content's availability depends on several criteria such as cache strategies and replacement policies, cache size or content popularity. Several cache management schemes have been evaluated [1], [9]–[12] and there are no consensus about the appropriate caching scheme for CCN. OSN carry extremely valuable information about users and their relationships. We argue that this knowledge can help to drastically improve the efficiency of Content Centric Networks.

In this paper, we propose to include social information in the design of a new caching strategy for Content Centric Networking. We conjecture a small number of users counts a huge amount of social relationships, dominates the activity and receives most attention from other users [13]. We call such users *Influential users*, and we argue that they produce content that is more likely to be consumed by others, and in consequence their content must be favored and replicated in priority. Our novel caching strategy is therefore prioritizing content from Influential users of the social network.

The major contributions of this paper are :

- A model of social network over a CCN network together with the new caching strategy for Content Centric Networking;
- Extensive simulations of the interaction of users from two realistic social environments with thousands of users and an evaluation of the caching strategy;
- An implementation of our caching strategy in CCNx and a deployment on PlanetLab.

The results shows that our proposed socially-aware strategy improves drastically the caching performances of CCN.

The structure of this paper is organized as follows. We review in Section II the related work and discuss social networks, influence of users and caching schemes. Then, in Section III, we present the model used throughout this

paper. Section IV describes our novel socially-aware caching strategy designed for CCN. Section V details the simulation environment, and emphasize strongly on the simulation parameters, the users' social interactions and the simulation tool. In Section VI, we present the simulation experiments results and the benefits of our caching strategy for CCN. Section VII details the implementation of our caching strategy into CCNx and the experiments on PlanetLab. We then sum up our findings and discuss our model in Section VIII. Section IX concludes the paper and exposes the future work.

II. RELATED WORK

A. Online Social Networks

The study of Online social networks such as Facebook allows analyzing the relationships among interacting users [14]–[16]. Such studies have gained importance with the rise of the Web 2.0. The theory of *Efficient Hubs* or *Influentials* [13] shows that only a small number of users dominate the activity, *influence* other users and receive most of their attention. As a result, companies have considered the opinion leadership in order to improve the quality of their products through interpretation of media messages and increase their sales [7], [17]. Aiming at different targets, political analysis follows the same approach [18] to predict the preferences of a population. Eigenvector [19] and PageRank [20] are common centrality measures to calculate the importance of nodes in a graph. PageRank is a variant of the Eigenvector centrality measure, and it is used by Google to predict the most important pages across the Internet.

As social networks have become a very important trend in the Internet, there are currently only a limited number of studies about the use of social network paradigm with ICN. To our knowledge, [11] is the only work studying social networks and ICN at the same time. ICN, IP networks and Content Delivery Networks are compared with regard to the current Twitter architecture. They show that ICN is a natural architecture for deploying social networks' applications. In our paper, we take a different approach. We do not aim to compare architectures; we aim at using social networks' information in order to improve ICN architectures.

B. Caching Schemes in ICN

The use of caches to increase content availability and to reduce perceived latency time has been deeply investigated in diverse environments such as Operating Systems, Web-browsers and Proxy-servers. There is already a large number of caching schemes in the literature and some of the most important are presented in [21].

In the context of ICN, caching has also been largely studied with a strong emphasis on content replacement policies [1], [10], [22] (e.g.: LRU, RAND, FIFO, etc.), the size of cache [23] and the impact of broad range of topologies such as Binary Trees [1], [10] and common ISP structures [9], [10], [24], [25]. In [26], it is proved that most of replacement policies can be grouped in equivalence classes and achieved the same performances.

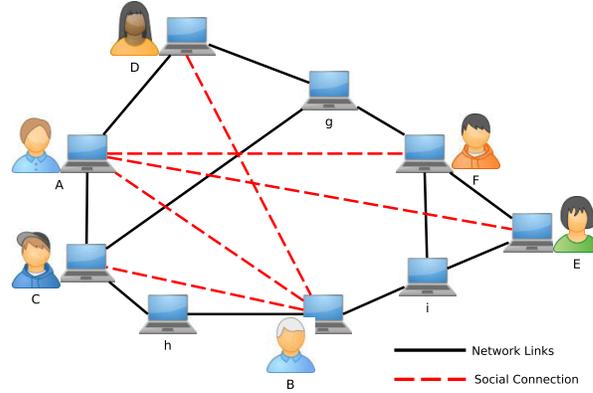


Fig. 1: Example of a social network on top of a CCN architecture. The black solid-lines represent the links between CCN nodes and the red dashed-lines represent the users' social relationships.

Recent studies show that the use of previously mentioned caching mechanisms does not improve performances [10], and huge caches up to 10 TB should be necessary to achieve acceptable levels of performance [27]. Nevertheless, [9] points out that caching indiscriminately does not guarantee high performances. They show that caching "less" content can achieve better performances because it reduces the load on the caches. Other studies [12], [28] follow the same approach and propose to discriminate content and cache only popular one in order to save the resources of the caches.

All these work treat equally content issued by any users whether they are Influentials or not. In our work, we propose a new caching strategy that gives priority to content issued by popular users.

III. NETWORK MODEL

In this section, we describe the model used throughout this paper. As a foundation, we first assume that the future Internet architecture is based on Content Centric Networking. Then, as the Internet is becoming a social oriented network, we propose a social network model built over the CCN network. In addition, we also model the interaction of users in social networks. The Figure 1 gives an overview of our model that we describe in details in this section. Finally, a use case scenario illustrates the interactions of users in a social network built over a CCN. We point out the limitations of the current caching strategies for CCN and the need for novel caching strategies.

A. Content Centric Networking

Among several ICN architectures [4], we choose Content Centric Networking (CCN) because it is well established in the research community [29]. The CCN communication architecture relies on two named primitives: *Interest* and *Data*. A consumer requests content by broadcasting its Interest messages all over the CCN network; any node hearing the request and having the data can issue a response with a Data message. As it is the most significant functionality in CCN,

nodes cache all the *Data* messages that have passed through them. Another fact worthy to mention is that caches have finite space. Hence an important feature for CCN is to manage the cache of nodes with caching strategies and replacement policies, which decide whether to cache and in case the cache is full, the element to be replaced respectively.

To this end, we design a novel socially-aware caching strategy adapted to CCN and improving the caching performances. This caching strategy is presented in Section IV.

B. Social Network Model

Online Social Networks (OSN) allow users to publish content at their own will and share it with their acquaintances (i.e., *friends*). Friends may always be updated through a news feed system. Each time a user finds interesting content (text, pictures or video), he may share it with his friends, spreading and expanding the visible scope of these information. Thus, we model a *social network* by a network where users can publish, retrieve and share information with their communities, according to their personal preferences. In our social model, each user has therefore two functionalities to interact with its community: *Publish* and *Retrieve*, as defined as follows:

- *Publish*: the production of new content. After retrieving a content, users may share it again with their friends (i.e., *re-tweet* a message).
- *Retrieve*: this function allows users to receive the last content issued by all their friends. For example in Figure 1, the user A has friendship relationships (red dashed-line) with users B, E and F. Each time A issues a Retrieve message, A will obtain the content from all its friends B, E and F.

C. Users' Interaction Model

In order to emulate the users' activities in the social network, we present the users' interaction model. Our interaction model is based on research studies on social networks [15], [16].

As stated before, a user can *Publish* or *Retrieve* content. The interactions of a user in a social network are depicted in Figure 2 and the different parameters introduced in the model are: NS , the number of sessions; t_{IS_i} , the inter-session time; NA_i , the number of activities per session; and $t_{IA_{ij}}$, the inter-activity time. The interactions of a user can therefore be summarized as follows: a user may start a certain number of independent and consecutive sessions (noted NS). The time interval between each session is calculated with an inter-session time t_{IS_i} from the time origin t_0 or the previous session ending time. In every session i , the user performs a finite number of activities, NA_i , such as publications or retrievals from friends. These activities are separated by an inter-activity time $t_{IA_{ij}}$. In the Figure 2, we illustrate an example of user interactions in which User A has two sessions ($NS = 2$); the first session starts after the inter-session time t_{IS_1} and contains two activities ($NA_1 = 2$), which are separated by three different inter-activity times $t_{IA_{11}}$, $t_{IA_{12}}$ and $t_{IA_{13}}$. The second session counts only one activity

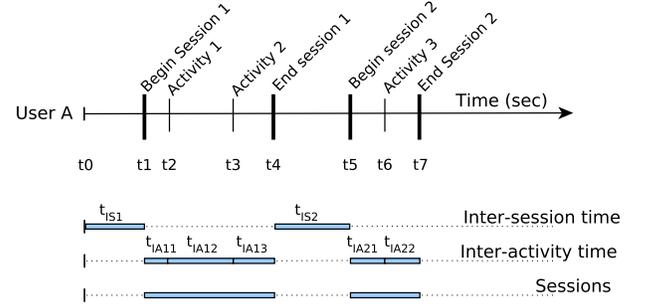


Fig. 2: Users' interaction model. Users can perform several activities within sessions.

($NA_2 = 1$) and is separated from the first session by a second inter-session time t_{IS_2} .

We tune the parameters of our model (NS , t_{IS_i} , NA_i and $t_{IA_{ij}}$) with realistic values extracted from research studies in the next simulation environment Section (SectionV).

D. Scenario of the Network Model

We illustrate the users' interactions over the CCN network with the following scenario example, depicted on Figure 1. In this example, 6 users are distributed over a 9 CCN-nodes network. The users' social relationships are depicted by the red dashed-lines and the links between CCN nodes by the black solid-lines. Note that on this simple example, a user and a CCN node can be on the same host.

In our model, when a user issues a Retrieve message, the user requests the last content from all his friends. In Figure 1, E has a single social relationship with A while B has several social relationships with A, C and D. Once B issues a Retrieve message, the last content from A, C and D are requested; when E issues a Retrieve message, only the last content from A is requested.

Consider the case where all the users publish a new content, referred as Content 1 to Content 6 respectively. Next, users A and E issue a Retrieve message. First, A requests for the last update of his three friends E, F and B. In the CCN architecture, it means that an *Interest* message is issued from node A to node E across the shortest path [A, D, g, F, E], and each node of the path will store the *Content 5* from E. A issues another Interest message across [A, E, g, F] to receive *Content 6* from F, which is stored at each node along the path. Finally, A issues its third Interest message for B, and *Content 2* will be stored along the path [A, C, h, B]. Then, E issues a Retrieve message, which means that *Content 1* will be cached at each node along the path [E, F, g, D, A]. The state of the caches is shown in Table I. With this simple scenario, we observe that content published by non-Influential users (Content 5 and Content 6 from node E and F) are not discriminated and are treated equally as content published by Influential users (content 1 and Content 2 from A and B).

| Users | Cache | Users | Cache | Nodes | Cache |
|-------|------------|-------|---------|-------|---------|
| A | 1, 2, 5, 6 | D | 1, 5, 6 | g | 1, 5, 6 |
| B | 2 | E | 1, 5, 6 | h | 2 |
| C | 2 | F | 1, 5, 6 | i | - |

TABLE I: State of the caches after the network model scenario in Section III-D.

In other words, there are as many copies of Content 6 as Content 1 in the network, whereas Content 1 has been generated by A, an Influential user, and this content is more likely to be consumed by others. Content 6 from non-Influential user E is wasting space in the network caches since it has a limited interest for other users. Then, we advocate the fact that content strategy in CCN should privilege content from Influential users.

In the following section, we present our novel socially-aware caching strategy that takes advantage of users' social information, and particularly the importance of users in the network.

IV. SOCIALLY-AWARE CACHING STRATEGY

The Previous section described our model and the interactions of a social network over a CCN network. In the case of a Future Internet based on CCN, we expect that users will still organize themselves into communities and exchange largely content through social networks, as it is already the case in the current Internet. A major limitation from the previous example (Section III-D) is that content is replicated into the CCN network whatever it comes from an Influential user (i.e.: a popular user with many social relationships) or not.

SACS: Socially-Aware Caching Strategy

We proposed a novel caching strategy for CCN based on the social information of users. Our socially-aware caching strategy gives priority to content issued by Influential users and cache it pro-actively into the CCN network. Indeed, users with more social relationships (*popular users*) are more influential than users with fewer relationships, and they produce content that is more likely to be consumed by others. For instance, when a popular Twitter user sends a message, it may count much more re-tweets than a message from a regular user. In the rest of the paper, we refer to our socially-aware caching strategy as *SACS*. To the best of our knowledge, *SACS* is the first proposed caching strategy designed for CCN using the social information of users.

In our strategy, the content published by Influential users is pro-actively replicated into the shortest path towards their social neighbors before it is requested. It improves the availability of Influential users' content and reduces the number of *Interest* messages in the CCN network. Thus, *SACS* privileges content issued by Influential users. It is therefore an important matter for *SACS* to detect the influential users in the social network.

| Users | Eigenvector | | PageRank | |
|-------|-------------|-------------|----------|-------------|
| | Score | Influential | Score | Influential |
| A | 0.58 | Yes | 0.29 | Yes |
| B | 0.58 | Yes | 0.29 | Yes |
| C | 0.29 | No | 0.11 | No |
| D | 0.29 | No | 0.11 | No |
| E | 0.29 | No | 0.11 | No |
| F | 0.29 | No | 0.11 | No |
| Avg. | 0.38 | N/A | 0.17 | N/A |

TABLE II: Scores of the centrality measures computed from the users' social relationship on Figure 1.

Influential Users Detection

We detect the influence of users within a social network by using the Eigenvector and PageRank centrality measures. These measures allow computing a score for every user in the network according to their *importance*. More details on the computation of these centrality measures can be found in [19] and [20] for Eigenvector and PageRank respectively. For instance, we compute on the Table II the score of users from the previous example (Figure 1) by using both the Eigenvector and PageRank centrality measure. We then define a user as being *Influential* if its score is greater than the average score of the overall social network.

From table II, A and B are Influential users because their score are greater than the average one ($0.58 \geq 0.38$ and $0.29 \geq 0.17$ respectively) while users C, D, E & F are non-influential users. Thus, whenever Influential user A publishes a new message, it is proactively cached into nodes along the shortest paths towards its social relationships B, E and F ($[A, C, h, B]$, $[A, D, g, F]$ and $[A, D, g, F, E]$). When F publishes a new message, it is not pro-actively cached as F is not an Influential user.

V. SIMULATION ENVIRONMENT

In this section, we present our simulation environment and the parameters we use to tune our model in order to evaluate *SACS*, our socially-aware caching strategy. A general overview of our simulation environment is provided in the Figure 3. As we describe the environment, all the parameters are also summarized in Table III.

Network Topology

We consider a future content-centric Internet built on the CCN architecture. Besides the caching capabilities at each node, the topological structure of this future Internet will be just like today's Internet. We then resort to *Inet* [30], a common tool to generate Internet topologies that we use to model the CCN topology. We use the *Inet* default parameters as presented in Table III and the network topology we use throughout this paper counts 3,037 nodes.

Social Network

In order to model the social relationships between users, we resort to two publicly available data sets : (i) LastFM data set [31] and (ii) Facebook data set [32]. LastFM is a music

recommender system, where users share their musical preferences with their community. From the LastFM data set, we extract a social graph counting 1,896 users, 12,717 bi-directional relationships and each user counts in average 13 relationships. Facebook is the most popular OSN and the data set consists of 4,039 users, 88,234 friend relationships and each user counts in average 44 relationships. Our simulation experiments are therefore performed with two different social network models.

The users of the social network are randomly mapped into the network topology.

Users' Activities

Now that we model social networks from publicly available data set over a CCN topology through *Inet*, we model the users' activities and interactions, i.e., the behavior of users and how they publish or retrieve pieces of information from their social relationships. Remember that in our model, users perform activities within sessions (Fig 2) and have two activity functions within the social network: *Publish* and *Retrieve* (Section III). We therefore extract from previous studies [15], [16] the distribution parameters to model the users' interactions such as the number of sessions, the number of activities by session, the inter-session time and the inter-activity time. From these measurement studies, we also found out that in average, 5% of the user's activities are Publication. This is consistent with the fact that most of the users only consult their timelines in the OSN and they barely publish new content such as status updates or share multimedia content. These distributions and their parameters are presented in Table III. Then, for each user from the two data sets (about 1,900 and 4,000 users respectively), we generate a sequence of activities (Publish and Retrieve) and sessions. These activities are timestamped and we obtain a realistic sequence of social activities for each user. By merging all the sequences of users' activities, we obtain synthetic social network activity traces for the LastFM and Facebook data sets. Following this procedure, we can generate as many traces as we need and for the next experiments we generate 20 different synthetic traces. On average, each trace counts 56,000 activities for the Facebook data set, and 25,547 activities for the LastFM data set. Our generator of social network activity traces will be publicly available.

Simulation Tool

Finally, in order to evaluate our socially-aware caching strategy for CCN, we implement a discrete-event simulation tool written in Python. We could not serve of the commonly used CCN simulators such as *ccnSim* [10] because these simulators are not designed to receive a realistic workload in input such as social network activity traces.

We also implement our novel caching strategy into our simulator. As our caching strategy relies on centrality measure to detect Influential users in the social network, we implement both Eigenvector and PageRank.

As stated before, CCN uses caching strategies and replacement policies to manage the cache of nodes. Our caching

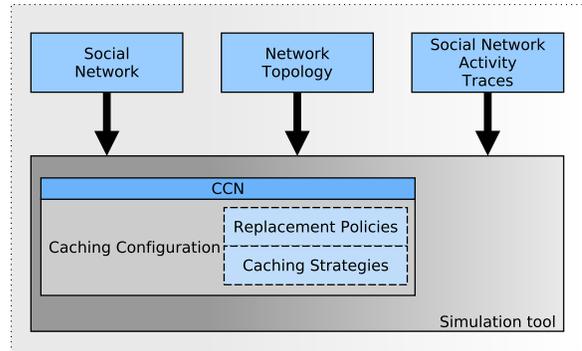


Fig. 3: General overview of the simulation environment.

| Social Network Activity Traces | |
|---------------------------------------|---|
| Number of sessions NS | Zipf ($\alpha = 1.792, \beta = 1.0$) |
| Number of activities NA_i | Zipf ($\alpha = 1.765, \beta = 4.888$) |
| Inter-session time t_{IS_i} (s) | LogNormal ($\mu = 2.245, \sigma = 1.133$) |
| Inter-activity time $t_{IA_{ij}}$ (s) | LogNormal ($\mu = 1.789, \sigma = 2.366$) |
| Publish/Retrieve | 5% |

| CCN Topology | |
|------------------|--------|
| #Nodes | 3,037 |
| Degree-one nodes | 0.3% |
| Plan size | 10,000 |
| Seed | 0 |

| Social Network Topology | |
|-------------------------|--------|
| LastFM | |
| #Users | 1,896 |
| #Links | 12,717 |
| Avg. Degree | 13 |
| Facebook | |
| #Users | 4,039 |
| #Links | 88,234 |
| Avg. Degree | 44 |

| Caching Configuration | |
|-----------------------|------------------------|
| Replacement Policies | LRU, FIFO, RAND |
| Centrality Measure | Eigenvector / PageRank |
| Cache Size | {1...20} |

TABLE III: Simulation parameters of the experiments.

strategy can therefore be used jointly with the traditional replacement policies for CCN such as *Last Recently Used* (LRU), *Random* (RAND) and *First In First Out* (FIFO).

The cache size at each node ranges from 1 to 20 elements and we show through experiments that larger cache size do not improve performances (see Section VI for further explanations).

VI. SIMULATION EXPERIMENTS RESULTS

A. Notations

Our objective is to evaluate the caching performances of CCN with regard to our socially-aware caching strategy, i.e., to compare CCN with or without using SACS. We performed simulations with each of the replacement policies (LRU, RAND, and FIFO). For sake of clearness, we present in this paper only the results with the LRU replacement policy and the other policies (RAND and FIFO) show similar results. Indeed, as stated in [26], replacement policies can be grouped into equivalent classes and have similar performances.

Thus, in order to clarify our notation, for the rest of the paper we refer to *CCN* as *CCN with the LRU replacement policy*. We also refer to *SACS* as *CCN with the LRU replacement policy and the SACS caching strategy*. In addition, as our caching-strategy relies on centrality measure, we implemented

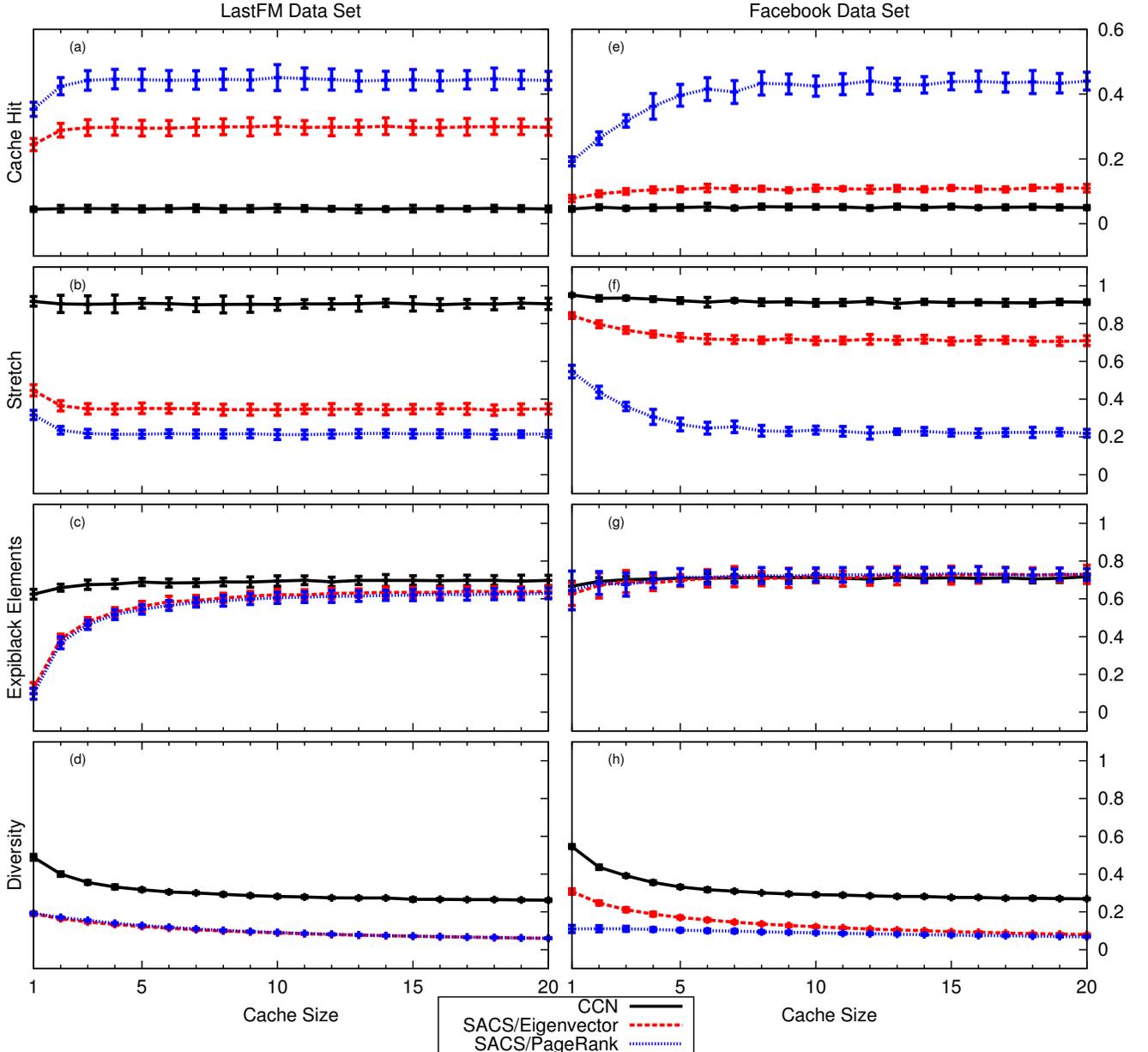


Fig. 4: Caching performances of CCN with or without using SACS, our proposed socially-aware caching strategy (Cache Hit, Stretch, Expired Elements and Diversity). The figures on the left side are for the LastFM data set and on the right side are for the Facebook data set.

both *Eigenvector* and *PageRank* and refer to them explicitly. We also perform the simulation experiments using two data sets (LastFM and Facebook, Section V) and refer to them directly.

B. Evaluation Metrics

We evaluate the performance of our socially-aware caching strategy according to the following metrics :

- *Cache Hit*: the probability to obtain a cache hit all along the path from a requester to a cache node;
- *Stretch*: the distance in number of hops that the data chunk has traveled in the network with respect to the distance between the node storing the original copy (ratio);
- *Expired Elements*: the ratio of content stored across the caches that are no longer requested by any users;
- *Diversity*: the number of distinct content stored across all the caches with respect to the total caching space.

C. Results

The results of our simulation experiments are presented in Figure 4. This figure consists in fact of 8 figures: the four figures on the left column (Fig. 4(a)-(b)-(c)-(d)) are for the results obtained with the LastFM data set, while the four on the right column are for the Facebook data set (Fig. 4(e)-(f)-(g)-(h)). Four rows of two charts depict each of our metrics: Cache Hit (Fig. 4(a)-(e)), Stretch (Fig. 4(b)-(f)), Expired Elements (Fig. 4(c)-(g)) and Diversity (Fig. 4(d)-(h)). All the figures share the same axes: the x-axis is the cache size ranging from 1 to 20 elements; The y-axis is the probability. For clarity reasons, we show the x-axis only on the two bottom figures ((Fig. 4(d)-(h)), and the y-axis on the figures of the right column (Fig. 4(e)-(f)-(g)-(h)). For each simulation experiment, we performed 20 runs of each simulation using different social network activity traces and provide the average value and the confidence intervals.

Figures 4(a) and 4(e) show the Cache Hit performances of CCN with or without using *SACS*, our socially-aware caching strategy (for Eigenvector and PageRank measures). Without our strategy, the Cache Hit of CCN achieves low values and it barely reaches 5%. Differently, *SACS* increases significantly the Cache Hit and reaches 30% for *SACS/Eigenvector* and up to 40% for *SACS/PageRank* with the LastFM data set, while it reaches 10% and up to 40% using Facebook data set.

The low performances for CCN are due to the use of large-scale and realistic social and network topologies. Indeed, the long routes traversed by the content affect the Cache Hit and its computation works as follows. In every hop passed by an *Interest* message, the Cache Hit gets updated with a *Hit* or a *Miss*. If the requested content is present at hop n , we obtain a single *Hit* and there has been $n - 1$ hops without the requested content (i.e., $n - 1$ *Miss*). As the Cache Hit metric is the ratio of *Hit* with regard to the number of *Miss*, the longer the path to find the content is, the lower the Cache Hit is. In our case, our strategy pro-actively caches content from Influential users in the network paths. Our strategy *SACS* succeeds to make the content more available and improves drastically the caching performances of CCN.

The Stretch metric is presented in Figures 4(b) and 4(f). This metric is in direct correlation with the Cache Hit. Without our strategy, the Interest messages traverse 90% of the shortest path to get the content. As expected, the distance to get the content has greatly reduced with *SACS*: Interest messages traverse only 35% and 72% of the shortest path with *SACS/Eigenvector* and 22% and 26% of the path with *SACS/PageRank* (LastFM and Facebook data set respectively).

Figures 4(c) and 4(g) show the ratio of Expired Elements in the caches. This metric is important to verify if the content stored in the caches is still requested or not (i.e., outdated). One could have expected that CCN performs better than our socially-aware caching strategy because *SACS* favors the content from Influential users. However, as seen on Figures 4(c) and 4(g), *SACS* achieves the same level of performances as CCN. Indeed, our strategy gives priority to content from

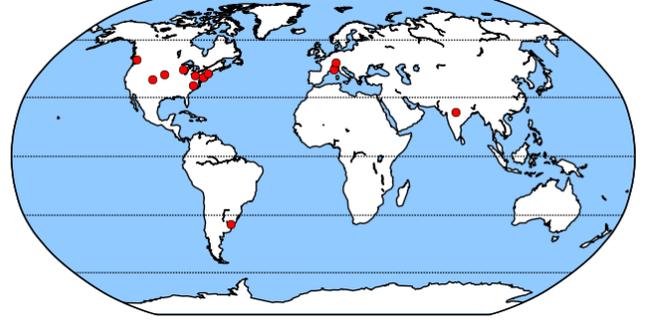


Fig. 5: Location of the selected PlanetLab nodes.

Influential users, and these popular users produce content that is more likely to be consumed by others. Thus, content cached by *SACS* is still mainly requested and it keeps the level of expired elements at the same level as CCN.

Last, the Diversity metric is presented in Figures 4(d) and 4(h). Since this metric stands for the number of distinct content stored across all the caches with respect to the total caching space, Diversity decreases as measure as the cache size grows. There is less diversity in the caches with *SACS* and this result was expected since our strategy discriminates content from Influential users. Hence *SACS* deliberately creates multiple copies of Influential's content, reducing the variety in the caches at the same time. However, even though *SACS* does not provide as high Diversity as CCN, it still drastically increases the caching performances in CCN.

It is noteworthy to mention that larger cache sizes have no impact on the Cache Hit. On Fig. 4(e), the Cache Hit for *SACS/PageRank* remains stable from a cache size of 6, while it is stable at any Cache size for *Sacs/Eigenvector* or for the other data set (Fig. 4(a)).

Regarding both data sets, the number of users has no impact on the performances of our strategy. Indeed, Facebook data set counts almost twice the number of users as the LastFM one (4,039 and 1,896 respectively, Table III), and our strategy achieves high level of performances with each social environment. It is especially the case with the PageRank centrality measure (blue-dashed line on Fig. 4(a)-(e)). With the LastFM data set, *SACS* shows similar performances for both centrality measures. For the Facebook data set, *SACS/PageRank* reaches high performances while *SACS/Eigenvector* is lower and slightly improves CCN. The average degree can explain this trend (13 for LastFM and 44 for Facebook, Table III) and the PageRank measure exhibits better performances than Eigenvector with a highly connected social environment. As the two centrality measures succeed to detect influential users in the network, the PageRank measure is a better choice for our strategy.

VII. EXPERIMENTS ON PLANETLAB

Besides simulation experiments, we evaluate *SACS*, our socially-aware caching strategy, into a real testbed. To this

end, we use CCNx [29], the most advanced prototype of CCN, and we perform experiments on PlanetLab, a planetary-scale testbed platform that supports the development of new network services [33].

We implemented our socially-aware caching strategy into CCNx v0.7.1. and the PageRank centrality measure to detect Influential users as it has shown the best performances in the simulation experiments (Section VI). Our modified CCNx prototype has been deployed into 14 PlanetLab nodes geographically distributed. During our experiments, as PlanetLab is a global research network, we could not handle more stable nodes at the same time. The location of the PlanetLab nodes is indicated in the Figure 5.

For these experiments, we use the LastFM data set presented in Section V that counts 1,896 users and for which we obtained synthetic social network activity traces to model the users' activities. Each of the 1,896 users was assigned to one of the 14 PlanetLab nodes. The built CCN topology consists of a fully connected topology between the 14 PlanetLab nodes. We perform three runs of the experiment for each cache size with different synthetic traces. We evaluate the Cache Hit performances of CCN with regards to our strategy. The results of the experiments are presented in Figure 6 and we provide the average value and the standard deviation.

In the previous section, we used additionally Stretch, Diversity and Expired Elements. Due to the high complexity to implement these metrics in the PlanetLab platform, we decide to only implement Cache Hit and to use it to assess the previously shown results.

Our socially-aware caching strategy improves drastically the performances of CCNx. CCNx Cache Hit barely reaches 20%, while it reaches about 50% with our strategy. *SACS* enhances the performances of the default caching strategies in CCNx by 2.5 times.

The planet-scale experiment results match the previously obtained simulation results (Section VI), and it confirms that our socially-aware caching strategy *SACS* improves the caching performances of CCN. It also points out that the use of social information (i.e.: the importance of users in a social network) improves the performances of a Future Internet based on the Content Centric Networking architecture.

VIII. DISCUSSION

From our previous experiments, we show that *SACS* improves drastically the caching performances of CCN and naturally reduces the distance to get the content. In addition, our PlanetLab experiments demonstrate that *SACS* enhances the caching performances of CCN by 2.5 times. Besides, we discuss here some open issues regarding our caching strategy.

SACS discriminates the content from popular users and caches it pro-actively in the network. There is a trade-off between the caching performances and the diversity of elements in the caches. By privileging content from important users, *SACS* reduces the number of sources of content, and thus the number of distinct elements stored in the network. However, regarding the number of Expired Elements, *SACS* shows the

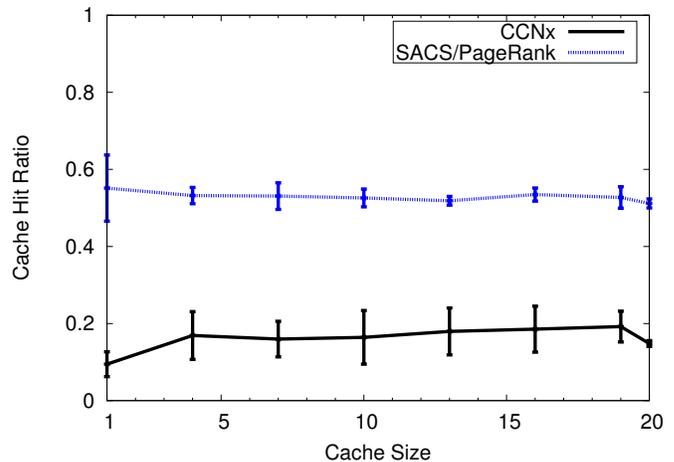


Fig. 6: Cache Hit performances for CCNx and our socially-aware strategy (*SACS* with PageRank centrality measure) on Planetlab.

same level of performances as CCN. With *SACS*, the replicated content is the one from popular users. These pieces of content are still requested frequently by other users and multiple copies into caches does not induce larger waste of space than using CCN.

Our strategy also requires detecting Influential users. It is therefore essential to compute accurately the centrality measure in the social network. In order to avoid a centralized computation of this measure and to have a complete knowledge of the social network, several studies [34]–[36] address the topic and in particular, in [34], they propose a decentralized version of the Eigenvector algorithm in which scores are calculated locally by every node and normalized by exchanging messages. [35], [36] also shows that it is feasible to compute a good estimation of the PageRank without the entire knowledge of the network.

Another matter of importance is the privacy of users. Computing a centrality measure involves the exchange of users' relationships. Thus, users are exposing sensitive information and partially unveiling their privacy. As the management of privacy is a subject of high interest, it has been investigated in other research areas. For instance, in the context of databases, Differential Privacy [37] hides sensitive information while it still allows answering queries on databases. Such kind of mechanism can also be used with our strategy in order to preserve users' privacy.

For large bulks of data (such as video), congestion control mechanisms could be put together with our strategy. For instance, [28] decides the number of chunks to be cached according to the content popularity. *SACS* may use congestion control mechanisms along with its proactive content replication functionality in order to avoid overloading the network.

IX. CONCLUSION

We proposed in this paper *SACS*, a Socially-Aware Caching Strategy for Content Centric Networks. *SACS* uses social

information and privileges Influential users in the network by pro-actively caching the content they produce.

Based on extensive simulation experiments, we showed that our caching strategy improves significantly the caching performances of CCN with regard to the Cache Hit and Stretch, while it keeps a similar level of performances for the Expired Elements. Furthermore, we implemented SACS on CCNx and performed experiments on PlanetLab. SACS improved the caching performances of CCN by 2.5 times in a real testbed. These results point out that social information are relevant pieces of information to improve a future content-centric Internet.

Finally, the inclusion of social information should not be limited to CCN caching related issues. As future work, routing alternatives may as well consider social features to select the best path. Increasing the cache diversity is also a topic to be investigated.

ACKNOWLEDGEMENT

The authors would like to thank the *Conseil Régional de Lorraine* for its financial support.

REFERENCES

- [1] V. Jacobson, D. Smetters, J. Thornton, M. Plass, N. Briggs, and R. Braynard, "Networking named content," in *ACM CoNEXT '09*.
- [2] C. Dannewitz and T. Biermann, "Prototyping a network of information," in *IEEE LCN*, 2009.
- [3] M. Särelä, T. Rinta-aho, and S. Tarkoma, "RTFM: Publish/subscribe internetworking architecture." ICT Mobile Summit, 2008.
- [4] B. Ahlgren, C. Dannewitz, C. Imbrenda, D. Kutscher, and B. Ohlman, "A survey of information-centric networking," *IEEE Communications Magazine*, 2012.
- [5] "Business week article. <http://goo.gl/0qutg>."
- [6] "<http://mashable.com/2012/11/06/obama-wins-twitter>."
- [7] K. Song, D. Wang, S. Feng, and G. Yu, "Detecting opinion leader dynamically in chinese news comments," in *WAIM'11*, 2012.
- [8] "Health leaders media article. <http://goo.gl/ljbwd>."
- [9] W. Chai, D. He, I. Psaras, and G. Pavlou, "Cache "less for more" in information-centric networks," in *IEEE IFIP '12*.
- [10] D. Rossi and G. Rossini, "Caching performance of content centric networks under multi-path routing (and more)," Telecom ParisTech, Tech. Rep.
- [11] B. Mathieu, P. Truong, W. You, and J. F. Peltier, "Information-centric networking: a natural design for social network applications," *IEEE Communications Magazine*, 2012.
- [12] C. Bernardini, T. Silverston, and O. Festor, "Mpc: Popularity-based caching strategy for content centric networks," in *IEEE ICC 2013*.
- [13] D. J. Watts and P. S. Dodds, "Influentials, Networks, and Public Opinion Formation," *Journal of Consumer Research*, vol. 34, 2007.
- [14] A. Friggeri, G. Chelius, and E. Fleury, "Egomunities, Exploring Socially Cohesive Person-based Communities," in *NetSci 2011*.
- [15] F. Benevenuto, T. Rodrigues, M. Cha, and V. Almeida, "Characterizing user behavior in online social networks," in *ACM SIGCOMM*, ser. IMC '09, 2009.
- [16] L. Gyarmati and T. Trinh, "Measuring user behavior in online social networks," *Netwrk. Mag. of Global Internetwkg.*, 2010.
- [17] F. Bodendorf and C. Kaiser, "Detecting opinion leaders and trends in online social networks," in *ACM SWSM '09*.
- [18] N. Agarwal, M. Lim, and R. T. Wigand, "Online collective action and the role of social media in mobilizing opinions: A case study on women's right-to-drive campaigns in saudi arabia," in *Web 2.0 Technologies and Democratic Governance*, 2012.
- [19] A. Langville and C. Meyer, "A Survey of Eigenvector Methods for Web Information Retrieval," *SIAM Rev.*, 2005.
- [20] M. Franceschet, "Pagerank: standing on the shoulders of giants," *Commun. ACM*, 2011.
- [21] S. Podlipnig and L. Böszörményi, "A survey of web cache replacement strategies," *ACM Comput. Surv.*, 2003.
- [22] L. Muscariello, G. Carofiglio, and S. Borst, "Accelerated lru replacement with probabilistic caching: A fast algorithm to approximate lfu," *CCNxConf*, 2012.
- [23] D. Rossi and G. Rossini, "On sizing CCN content stores by exploiting topological information," in *2012 IEEE INFOCOM WKSHPS*, 2012.
- [24] J. Choi, J. Han, E. Cho, T. Kwon, and Y. Choi, "A Survey on content-oriented networking for efficient content delivery," *IEEE Communications Magazine*, 2011.
- [25] G. Tyson, S. Kaune, S. Miles, Y. El-Khatib, A. Mauthe, and A. Taweel, "A trace-driven analysis of caching in content-centric networks," in *IEEE ICCCN*, 2012.
- [26] E. Rosensweig, D. Menasche, and J. Kurose, "On the steady-state of cache networks," in *IEEE INFOCOM 2013*.
- [27] C. Fricker, P. Robert, J. Roberts, and N. Sbihi, "Impact of traffic mix on caching performance in a content-centric network," in *IEEE NOMEN 2012*, 2012.
- [28] K. Cho, M. Lee, K. Park, T. Kwon, Y. Choi, and S. Pack, "WAVE: Popularity-based and collaborative in-network caching for content-oriented networks," in *IEEE INFOCOM WKSHPS*, 2012.
- [29] "<http://www.ccnx.org>."
- [30] J. Winick and S. Jamin, "Inet-3.0: Internet topology generator," Tech. Rep., 2002.
- [31] I. Cantador, P. Brusilovsky, and T. Kuflik, "Hetrec," in *ACM RecSys 2011*, 2011.
- [32] J. McAuley and J. Leskovec, "Learning to discover social circles in ego networks," in *NIPS 2012*, 2012.
- [33] "<http://www.planetlab.org>."
- [34] S. D. Kamvar, M. T. Schlosser, and H. Garcia-Molina, "The eigentrust algorithm for reputation management in p2p networks," in *WWW2003*, 2003.
- [35] K. Avrachenkov, N. Litvak, D. Nemirovsky, and N. Osipova, "Monte carlo methods in pagerank computation: When one iteration is sufficient," 2005.
- [36] S. Acharyya and J. Ghosh, "Outlink estimation for pagerank computation under missing data," in *WWW 2004*.
- [37] C. Dwork, "Differential privacy: A survey of results," in *Theory and Applications of Models of Computation*, 2008.