Optimal Resource Allocation for Disaster Recovery

Interest-based cooperative caching in multi-hop wireless networks

Javed Iqbal
Department of Electronics and Telecommunications
Politecnico di Torino, Italy
Email: javed.iqbal@polito.it

Paolo Giaccone
Department of Electronics and Telecommunications
Politecnico di Torino, Italy
Email: paolo.giaccone@polito.it

***Abstract*—New communication protocols, as WiFi Direct, are now available to enable efficient Device-to-Device (D2D) communications in wireless networks based on portable devices. At the same time, new network paradigms, as Content-Centric-Networking (CCN), allow a communication focused on the content and not its location within the network, enabling a flexible location for the content, which can be cached in the nodes across the network.**

In such context, we consider a multi-hop wireless network adopting CCN-like cooperative caching, in which each user terminal acts also as a caching node. We propose an interest-based insertion policy for the caching, based on the concept of "social-distance" borrowed by online recommendation systems, to improve the performance of the overall network of caches; the main idea is to store only the contents which appear to be of interest for the local user. We show that our proposed scheme outperforms other well-known insertion policies, that are oblivious of such social-distance, in terms of cache hit probability and access delays.

I. Introduction

Mobile networks have been experiencing an impressive growth in the data traffic, mainly due to multimedia application; indeed, 2/3 of mobile traffic is expected to be video by 2016 [1]. Huge amount of this traffic is due to user-generated content and is distributed via popular Internet services, often based on an explicit, user-driven social network (e.g. Facebook and YouTube). This fast growth in traffic has been imposing a significant burden on the current wireless infrastructure, which must be periodically upgraded, increasing the profit gap between the network operators and the Over-The-Top content providers. Indeed, network operators continue to invest into the infrastructure to cope with the increasing traffic, while users spend more money for contents or cloud based services than for the network services. This dichotomy motivates new network paradigms in which the user's wireless terminals cooperate to decrease the load in the wireless network infrastructure: the Quality of Experience of the user is improved, even if the network infrastructure has not been upgraded. Note that these new paradigms are expected to complement but not to substitute the current wireless access network, since their effectiveness is not universal and depends on many factors (e.g., content popularity, user behavior, privacy, etc.)

The first factor which is enabling this change of paradigm is that users are interested in the content and not in its location (i.e. the particular server hosting such content). This motivates the novel Content Centric Network (CCN) paradigm, in which a content-based addressing (differently from the classical IP-based server-location addressing) allows to retrieve the requested content, independently from its actual location.

The second factor is the *content-reuse*, since it is well known [2] that in many contexts a large amount of traffic is caused by few popular contents (e.g., videos, music, apps, software updates, etc.). These popular contents are requested many times after being generated. This fact advocates the use of caching techniques to distribute popular content across the network and to avoid the access at the server storing the original copy of the content.

The third factor is the *content localization*, since the interest for some contents is spatially localized due their specific nature. For example, tourist/event informations, local news, shop advertisements show a clearly localized region of interest. This factor motivates the cooperation among nodes that are in proximity, since their spatial position increases the content reuse. To enable communications among neighboring nodes, without any infrastructure, layer-2 technologies (like Bluetooth [3] and WiFi Direct [4]) are supporting Device-to-Device (D2D) communications and enable peer-to-peer capabilities among terminal nodes, thanks also to specific middlewares (like AllJoyn [5]) to ease the development of applications.

Finally, modern mobile devices like smartphones and tablets are equipped with large storage capacity of many gigabytes. The storage capacity can be considered as a free resource nowadays, which the user is willing to share more preferably more than any other resource since it is not affecting directly its Quality of Experience when running applications.

All the previous factors advocate the adoption of *cooperative caching* techniques among neighboring nodes, exploiting D2D communications, multi-hop communications and the available free storage to share among the users. The large storage capacity enables each mobile device to act as a caching node of a wireless CCN; each node stores all the received contents (also the ones to be forwarded in a multi-hop fashion) in its cache. Contents are distributed across the nodes and a user can hopefully access the desired content in its proximity. Thus, cooperative caching can reduce both the delay to access the content (with satisfaction for the user) and the network load (with satisfaction for the wireless operator).

The effectiveness of the approach strongly depends on the caching scheme adopted in the nodes. A cooperative caching strategy usually consists of an *insertion policy* and a *replacement policy*. Whenever a content arrives, the node has

to decide whether to cache it or not; this is referred as insertion policy. On the other hand, when the memory devoted to the cache becomes full, the replacement policy has to remove one of the stored contents to make space for the new content. Replacement policies have been largely investigated in the past, and LRU (Least-Recently-Used) policy has been shown to be very effective and thus, in the rest of our work, we will always assume LRU as replacement policy. The intuition for its efficiency is that one content, that has been requested farthest in the past, probably will not be requested in the near future and is the best candidate to be removed from the cache.

On the contrary, in our work we will concentrate only in the insertion policy, which plays an important role especially in small caches. Indeed, note that in any finite cache it is completely useless to store any content for which the number of requests is one, independently from the size of the cache. Intuitively, for smaller caches the insertion policy must be more selective and should admit only those contents whose popularity is above some threshold, which is difficult to evaluate a-priori without a perfect knowledge of the requests pattern.

Our proposal is to consider the user interest for the content as the main metric for the insertion policy. The idea is that the node is caching locally a content in two cases: either when the corresponding user has requested the content, to allow a local access for future requests for the same content, or when the potential interest of the user for the content is high, to allow a local pre-fetching of the content, that may be accessed in future requests. Note that the second case is not trivial to be implemented since it requires to evaluate the interest for a content by a user, which requires some quantitative model. In the Internet, online recommendation systems have been already devised such kind of models, whose main flavor is to evaluate the similarity of interests among users and the user's interest for a content, based on the behavior of the user (e.g. when the user promotes a content with the "like" button) and on the user's position within the social network. Our contribution is to propose an interest-based insertion policy for cooperative caching and to show its efficiency with respect to other insertion caching policies, which are oblivious of the social distance between users and contents.

The rest of the paper is organized as follows. In Sec. II, we describe the system model for our social-aware cooperative caching. Sec. III is devoted to related work. In Sec. IV, we compare the performance of our caching policy with other alternative ones. Finally, in Sec. V we draw our conclusions.

II. System model

We consider a wireless network consisting of a *server* and U nodes interconnected through a multi-hop linear topology, as shown in Fig. 1. The choice of this topology is arbitrary; despite of its simplicity, it is able to highlight the main differences between insertion policies in large networks of caches, without taking into account the problem of data routing. In this network, each node is associated with a unique user, denoted as u. The server is equipped with a catalog of C different contents, which can be accessed by the users.

TABLE I. Main notation

Symbol	Meaning
U	Number of network nodes/users
u	User $\in \{1, \ldots, U\}$
C	Number of different contents
c	Content $\in \{1, \ldots, M\}$
B	Cache size in number of contents
x_u	Social position of user u
x_c	Social position of content c
$d_s(u, c)$	Social distance of user u from content c

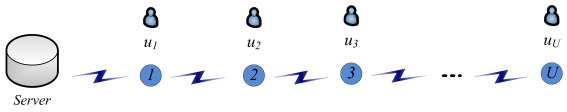

Fig. 1. Wireless multi-hop linear topology with one server and U nodes.

Fig. 2. "History" social space with 3 users and 3 books

A. Social space

We model the social relationship among users as "degree of similarity" (i.e., how much the users share common interests) and their interest for a specific content by "degree of interest". To evaluate quantitatively such values, we adopt a social space model. We start to describe it with a toy example, and then we will generalize the model.

As toy example, assume that all the contents in the catalog are books, which can be classified according to their "history" flavor. Any book is identified by a point on a segment $[0, 1]$ whose position represents the level of "history" present in the book, as shown in Fig.2. The extreme point 1 corresponds to a pure history book, whereas 0 to a book without any history content. Hence, a small distance between two contents implies a large degree of similarity between them, and vice versa. Similarly, the actual interest of a user for history books is represented by her position within the segment. In this way, a small distance between a user and a content implies a large degree of interest for it, and a small distance between two users implies a high degree of similarity in their interests. For example, regarding the 3 users u_1, u_2, u_3 in Fig. 2, we can claim that u_2 and, especially, u_3 show high interest toward history, whereas u_1 is not interested in this genre.

To model all the possible categories of contents, it would be necessary to consider a multi-dimensional space, in which each dimension corresponds to a particular category. After properly defining a norm on such a space, a smaller distance between two points represents a larger degree of similarity (between contents) or interest (by a user for a content). Due to the very large (actually, infinite) number of single categories for human interests, a multidimensional space approach is not practically feasible[1].

The approach we follow is to consider just a mono-dimensional social space, since it has been shown that this

[1]In practical recommendation systems, such space is compressed into a few dominant dimensions, thanks to a proper factorization of the so called "interests matrix", inferred from the content rating patterns [6].

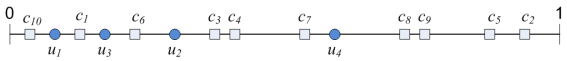

Fig. 3. Social space with 4 users and 10 contents on the wrapped segment $[0, 1]$.

simple model is able to capture some of main features of real social networks [7] with the minimum possible complexity. We assume that each user u and each content c is represented by a fixed point at position x_u and x_c, respectively, along the segment $[0, 1]$. To avoid border effects, the *social distance* between two generic entities a and b (being users and/or contents) is defined as the Euclidean distance on the wrapped segment: $d_s(a, b) = \min\{|x_a - x_b|, 1 - |x_a - x_b|\}$. For example, Fig.3 shows a social space with 4 users and 10 contents, in which the positions have been selected at random. Users u_1 and u_3 show a higher interest for c_1 as compared to u_4, because $d_s(u_1, c_1) < d_s(u_4, c_1)$ and $d_s(u_3, c_1) < d_s(u_4, c_1)$. Similarly, u_1 has a higher interest for c_2 compared to u_4, due to the wrapped segment.

B. Content request process

Given the social space with all the contents and users' positions in the social space, we model the fact that a user tends to request contents for which her interest is higher in the following way. We assume, for a generic user u, to rank all the contents in increasing distance $d_s(u, c)$, with $c = 1, \ldots, C$. When the user is requesting a content, the content is chosen on the basis of Zipf distribution with parameter α; more precisely, if q_k is the probability of requesting the kth content based on user's ranking: $q_k = \gamma / k^\alpha$ with $k = 1, \ldots, C$, where γ is a proper normalization factor. Note that a user can request the same content many times. We assume a continuous request process, in which a user requests a content just after the previous content has been received.

C. Cooperative forwarding and caching

As better discussed in Sec. III, we adopt a CCN approach for cooperative forwarding and caching. Whenever the user generates a request, the corresponding node sends a *request packet* with the identification of the requested content towards the server. The request is transmitted between neighboring nodes in a multi-hop fashion, until it reaches the server. Then, the server replies by sending back the *content data*, which reaches the requester's node in a multi-hop fashion. Note that in the case of more generic topologies (not considered in this work), the requests can be propagated through some standard controlled flooding protocols, whereas the data content is sent back to the requester through a single path, discovered during the initial request phase.

Each node is equipped with a local cache, denoted by Cache Storage (CS) and a Pending Interest Table (PIT) for storing requests for which the node has no content. When a node receives a content, this is forwarded back to the nodes for which it has stored requests in its PIT. We denote the overall cooperative caching approach we propose as "SOCIAL-CACHE-R_s", where R_s is a numerical parameter. Referring to the pseudo-code of Algorithm 1, when a content is received at a node, it is eventually cached according to our novel "social-aware" insertion policy. The main idea is

Algorithm 1 SOCIAL-CACHE-Rs at node u

```
if (content c arrives)
    if (d_s(x_c, x_u) < R_s) // Check social-distance
        if (c is not in CS) // Check if to cache
            if (CS is full)
                remove a content according to LRU
            store c in CS
if (request r_c for content c arrives)
    if (c is in CS) // Cache hit
        send c to the requester
    elseif (r_c is not present in PIT)
        add r_c into PIT
        send r_c to the other neighboring nodes
```

to store only those contents that would be of possible interest for the corresponding user. Thanks to the social space model, the insertion policy at user u's node simply consists of storing a content c if their social distance $d_s(c, s)$ is below a threshold R_s. Furthermore, if the cache is already full, Least Recently Used (LRU) replacement policy is adopted. If a request packet arrives at a node which does not have the requested content in its cache, a "miss" is experienced; then the request is stored in the PIT and forwarded to the neighboring nodes. Instead, if the requested content is present in the cache, a "hit" is experienced and the content is sent back to the requester in a multi-hop fashion.

Our social-aware policy caches only the most requested (i.e., most popular) contents for each user, based on the threshold R_s. The actual number of such contents varies for each user since it reflects the random distances between the user and the different contents. Note that R_s allows a simple implementation since it does not require to know in advance the overall ranking of all the contents, which is a priori unknown for a user, and could be only measured a posteriori. This social threshold is instead based on a different metric, which is the expected interest for the content, deduced with different approaches, as the ones used in recommendation systems. In this work we do not investigate such approaches, and we assume that each node is able to evaluate the social distance between any content and the corresponding user.

III. Related work

Caching replacement policies, as LRU and some variants of it, have been vastly studied [8], [9], [10]. On the other hand, *insertion* policies are less studied. Different authors have proposed different insertion policies to efficiently utilize cooperative caching. Later, we will compare our interest-based policy with the other policies described below. In the specific context of CCN, [8] proposed a universal caching approach, in which each node caches every new content, which we will refer as "ALL-CACHE" policy in this work. To cache only those contents which are frequently requested, in [11] the insertion decision is taken uniformly at random with a fixed probability $p \in [0.75, 0.9]$, which will be referred as "PROB-CACHE-p" policy in this work. Both ALL-CACHE and PROB-CACHE-p require no information exchange for caching decisions and have no overhead to the network. However, as they cache contents on a fix random probability only and do not consider any knowledge about the contents, they might cache contents

with very few requests (just one in the worst case) that are useless to be stored. Differently from these two approaches, our proposed insertion scheme is based on the interest for a content by each node. Finally, [12] compared ALL-CACHE and PROB-CACHE-p policies against the Leave Copy Down (LCD) policy, which was initially proposed in [13] for hierarchical web caches. According to LCD policy, a node, located along the path to the requester, inserts a new arrived content into the cache only if a hit has occurred in the cache one hop away. This reduces the redundancy of content cached in the network nodes as compared to ALL-CACHE scheme. PROB-CACHE-p was shown [12] to outperform LCD. Because of this, in the following section we will not consider LCD as term of comparison.

In the context of wireless ad-hoc networks, [14] considers a content dissemination scheme among the nodes based on the *channels* each user has subscribed. Whenever a node receives a content, if this belongs to a subscribed channel, it is stored in the node *private cache*. Otherwise, it is eventually stored in the node *public cache* that is present in the node to allow cooperative dissemination among the nodes. The paper compares two insertion policies for the public cache: (i) a random policy in which the node stores only the contents belonging to some random channels; (ii) a policy in which the node stores just the most popular channels. The performance of such policies are shown to depend strongly on the mobility patterns. Note that in our work, the social space is a generic model that is able to capture the "channel" concept, since all the contents of a single channel can be associated to the same position in the content space. Differently from [14], we do not distinguish between private and public cache and use only the threshold R_s to control the insertion into the cache. Finally, we consider a fixed communication topology and we do not consider the effect of mobility.

IV. Simulation results

We developed a simulator in Omnet++ [15], which is an open source event-driven simulator. We have considered a network with $U = 10$ nodes (or users) with the linear topology shown in Fig. 1. For each communication link, we assume a communication bandwidth of 2 Mbps and a propagation delay of 0.33 μs. We also assume a catalog of $C = 200$ contents present in the server, each of size 30 MBytes, which is the size of a typical YouTube movie clip at $480p$ resolution with the typical duration of 4 minutes [16]. The social position of all the contents are distributed uniformly in $[0, 1]$ as in [7]. Regarding the social position of the users, we consider two scenarios, denoted as UI and SI, representative of the extreme cases. Realistic scenarios reside in the middle of the two.

- *Uniform Interests* (UI), in which the social position x_u of user u is chosen uniformly at random in $[0, 1]$. This scenario is representative of a population of users with completely different interests, for which we could expect that caching may be less effective due to the limited content reuse.
- *Same Interests* (SI), in which $x_u = 0$ for any user u. This scenario refers to a population of users with the same interests, and hence we can expect a large degree of content reuse and higher efficiency of the caching system.

The exponent in the Zipf law for the popularity of the contents was set equal to $\alpha = 1.3$, equivalent to an exponent $\beta = 1.77$ in the corresponding Pareto distribution. This is compatible with the values $\beta \in [1.5, 2.5]$ that have been observed for content-on-demands in the Internet [2].

In the following, we will compare our SOCIAL-CACHE-R_s policy with ALL-CACHE (which admits any content into the cache) and PROB-CACHE-p (which admits a content into the cache with probability p, as discussed in Sec. III) and NO-CACHE (which does not allow caching at the nodes).

A. Performance metrics

We evaluate the performance of different caching schemes based on two main metrics, the cache hit probability and the distance in terms of hops to retrieve the content. The hit probability is relevant especially for the network operator, since a higher number of hits implies that the cache is more effective to "filter" the requests directed to the server; in this way a lower number of transmissions is required and the network congestion is reduced. Furthermore, the server load is reduced by a factor proportional to the overall hits in the network. Of course, the beneficial effect of lower network/server congestion is experienced also by the user.

To evaluate the hit probability, we distinguish between local and remote requests at a node. The former are generated by the user residing on the node, the latter are generated by all the other users. For node u, let r_u^L be the total number of local requests and let r_u^R be the total number of remote requests. Each request can generate either a miss or an hit; let h_u^L be the total number of hits due to local requests and let h_u^R be the ones due to remote requests. We define the *local hit probability* at node u as $P_{hit,u}^T = (h_u^L + h_u^R)/(r_u^L + r_u^R)$. By averaging across all the nodes, we define the corresponding *average total hit probability* as P_{hit}^T.

On the contrary, the distance is relevant especially for the user, who wishes to access the content with the minimum latency, achievable with the minimum number of hops. Indeed, the average distance can be considered a good estimation of the access delay to the content, in the case the network is not congested. Let δ_u be the average distance, measured in terms of hops from the requesting node, to reach the content. Due to the specific topology and user sequence ids, we have $\delta_u \in [0, u]$; indeed, as extreme cases, the content may be already present locally at user's cache or, in the worst case, it must be accessed at the server.

B. Numerical results

Fig. 4 shows the average total hit probability obtained by SOCIAL-CACHE-R_s for small cache size ($B = 5$) and large one ($B = 25$), when the users' social positions are distributed as either UI or SI. The plots are obtained by varying R_s. When R_s approaches to 0, the cache becomes ineffective since almost no content is admitted into the cache. For small R_s, the hit probability is independent from B, since the actual number of contents admitted into the cache is very small and the "effective" cache size is smaller than the maximum allowed B. For larger R_s, the number of contents admitted into the cache is larger than B and the hit probability is strongly affected by the cache size. The different behavior between the two interest (UI and SI) scenarios is due to the social distance between the users and their closest contents in terms of social distance.

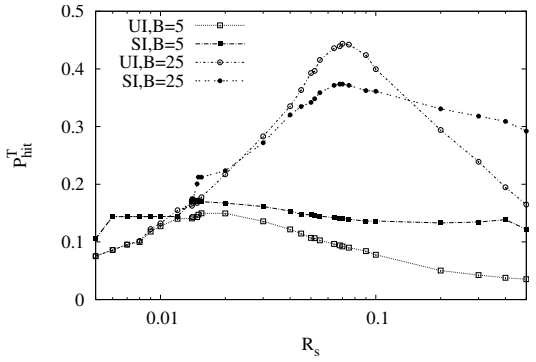

Fig. 4. Average total hit probability for SOCIAL-CACHE-R_s under UI and SI scenarios and for different cache size B

TABLE II. P^T_{hit} FOR DIFFERENT ALOGORITHMS

Insertion policy	UI		SI	
	B=5	B=25	B=5	B=25
ALL-CACHE	0.036	0.166	0.121	0.292
PROB-CACHE-0.9	0.040	0.182	0.126	0.302
PROB-CACHE-0.5	0.052	0.321	0.150	0.341
SOCIAL-CACHE-$\hat{R}_s$	0.147	0.444	0.169	0.373

When R_s is close to the maximum 0.5, all the contents are admitted into the cache and the policy degenerates into ALL-CACHE.

Interestingly, the hit probability shows a maximum for a specific value of R_s, which depends mainly on B. We claim the following:

Property 1: In UI and SI scenarios, the optimal value of social distance $\hat{R}_s$ that maximizes the hit probability in SOCIAL-CACHE is equal to the social distance that on average comprises B contents[2]:

$$\frac{1}{U}\sum_{u=1}^{U}\sum_{c=1}^{C} 1_{\{d_s(u,c)<\hat{R}_s\}} = B$$

For UI scenario, the average number of contents within a distance R_s from a generic user can be approximated by $2R_sC$ (this because $R_s = 0.5$ must include all the C contents in the catalog). By setting this number equal to B, we obtain the optimal value $\hat{R}_s = B/(2 * C)$. For our specific scenarios, for small cache ($B = 5$) we get $\hat{R}_s = 0.0125$ and for large cache ($B = 25$) we get $\hat{R}_s = 0.0625$, which are actually very good approximations of the values of R_s that maximize the hit probability in Fig. 4.

Table II compares the average total hit probability achievable by different caching policies under SI and UI scenarios and under small/large cache sizes. We compare the different caching policies described before; for our social-based policy, we have chosen the optimal value $\hat{R}_s$ for each B, according to the computation of the previous paragraph.

From the aggregate point of view, SOCIAL-CACHE is able to achieve 4 times better hit probability than ALL-CACHE for UI scenario and 20% better for SI scenario. Indeed, in SI scenario all the users share the same interests and they are

[2]Here $1_{\{x\}}$ is a binary indicator function, equal to 1 iff the event x is true

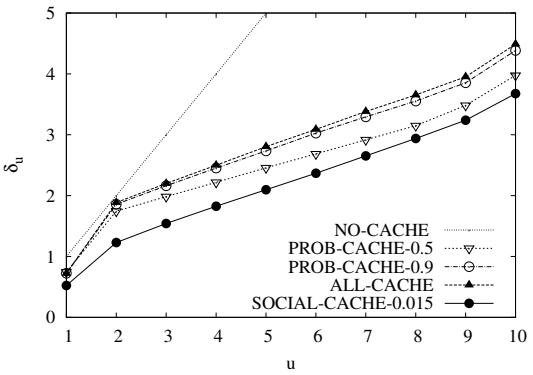

Fig. 5. Average distance to the content under SI case for small cache ($B = 5$)

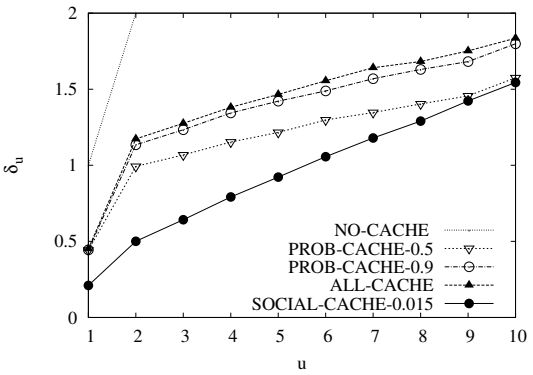

Fig. 6. Average distance to the content under SI case for large cache ($B = 25$)

requesting mainly the same contents. ALL-CACHE stores all the contents, which are the most popular among all the users, with a beneficial effects for all the users, even if the caching is not efficient as for SOCIAL-CACHE, since also less popular content is admitted into the cache.

On the contrary, in UI scenario all the users show very different interests, but ALL-CACHE tends to distribute contents among all the nodes, even if the only requesting user has already a local copy in its own cache; this results in a lower hit probability. In this scenario, our very "selective" policy is much more efficient, since it stores only the contents that are locally the most popular. At high level, PROB-CACHE-p behaves in an intermediate way with respect to ALL-CACHE and SOCIAL-CACHE. For $p = 0.9$ probabilistic caching does not show a meaningful advantage with respect to ALL-CACHE, even if this value was included in the range $[0.75, 0.9]$ suggested by [11]. For smaller $p = 0.5$ (which is outside this suggested range), probabilistic caching is still behaving only slightly better than ALL-CACHE.

To understand better the behavior of the different caching policies, in Fig. 5 we show the average distance to access the content from each node under SI scenario and for small caches. We plot as a reference NO-CACHE, for which, by construction, the distance is exactly equal to the node position u along the topology. Generally, for the first node (i.e., the closest to the server), it holds $\delta_1 < 1$, since there are cases in which the user is able to access the content directly in the local cache. In general, any caching policy is always able

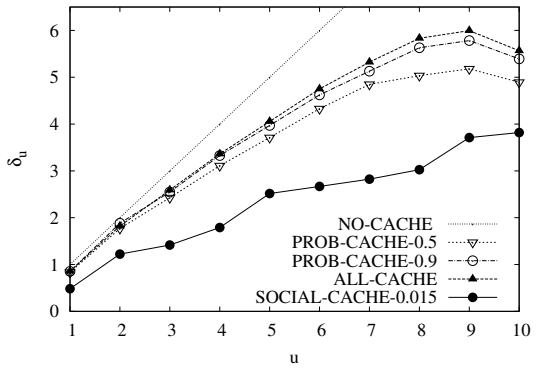

Fig. 7. Average distance to the content under UI case for small cache ($B = 5$)

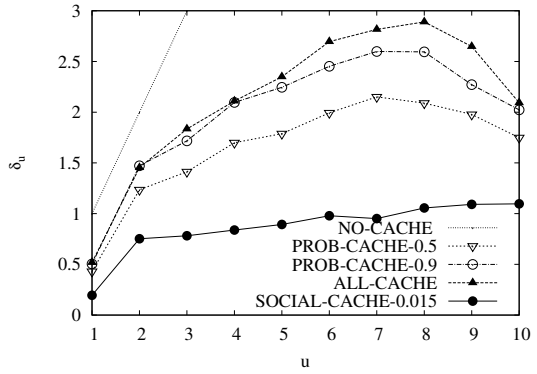

Fig. 8. Average distance to the content under UI case for large cache ($B = 25$)

to reduce the distance (i.e. the latency) by a factor at least 2, with respect to NO-CACHE. In general, SOCIAL-CACHE is outperforming all the other caching policies for any node, reducing the distance also by one hop.

Fig. 6 reports the average distance for SI scenario and for large cache size; the behavior is qualitatively identical to Fig. 5. Notably, with respect to the small cache scenario, the average distance is less than half, thanks to the higher available storage, and the performance gain due to SOCIAL-CACHE is higher especially close to the server, where the user's cache is "polluted" by the other users' content requests.

Finally, Fig.s 7-8 report the average distance under UI scenario, for different cache sizes. In both cases, SOCIAL-CACHE stores just the locally most popular contents and this fact allows to keep the average distance very small, since the content is accessed either at the local cache (at distance zero) or the server (at distance u for node u). This allows to achieve much smaller distances, especially for large cache sizes; notably, in Fig. 8 user u_{10} achieves a distance 10 times smaller than NO-CACHE. In both figures, PROB-CACHE outperforms ALL-CACHE, especially for nodes further from the server. The non-monotonic behavior of ALL-CACHE and PROB-CACHE is due to the fact that for small u the maximum distance to the server is by construction small, whereas for large u the cache is not "polluted" by the contents that are traveling towards their requesting node. In the extreme case, for ALL-CACHE and PROB-CACHE, user u_{10} stores only the contents requested by herself, similarly to SOCIAL-CACHE, but with a lower efficiency due to the fact that also less popular contents are admitted into the cache.

V. Conclusions

We have considered a CCN paradigm applied to multi-hop wireless networks exploiting D2D communications. We have proposed a new social-aware insertion policy for the cache that admits into the cache only contents for which the corresponding user has some "interest". The evaluation of the level of interest is based on the concept of "social-distance" borrowed by online recommendation systems.

We show that it is possible to maximize the performance of the our insertion policy by properly setting a threshold parameter R_s; in this way, our scheme outperforms other state-of-art insertion policies that have been devised for the same CCN scenario.

VI. Acknowledgment

This work was supported by PeopleNET, a PRIN project funded by the Italian Ministry for Instruction, University and Research (MIUR).

References

[1] A. Fehske, G. Fettweis, J. Malmodin, and G. Biczok, "The global footprint of mobile communications: The ecological and economic perspective," *IEEE Communications Magazine*, vol. 49, no. 8, pp. 55–62, 2011.

[2] Cha, H. Kwak, P. Rodriguez, Y.-Y. Ahn, and S. Moon, "Analyzing the video popularity characteristics of large-scale UGC systems," *IEEE/ACM Transactions on Networking*, vol. 17, no. 5, pp. 1357–1370, 2009.

[3] Bluetooth. [Online]. Available: http://www.bluetooth.com

[4] Wi-Fi Direct. [Online]. Available: http://www.wi-fi.org

[5] AllJoyn. [Online]. Available: http://www.alljoyn.org

[6] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *IEEE Computer*, vol. 42, no. 8, pp. 30–37, 2009.

[7] M. Boguñá, R. Pastor-Satorras, A. Díaz-Guilera, and A. Arenas, "Models of social networks based on social distance attachment," *Physical Review E*, vol. 70, no. 5, pp. 56–122, 2004.

[8] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. H. Briggs, and R. L. Braynard, "Networking named content," in *ACM Co-NEXT*, 2009, pp. 1–12.

[9] Y. Abdelmalek and T. Saadawi, "Collaborative multimedia content caching algorithms for mobile ad-hoc networks," in *IEEE MILCOM*, 2009, pp. 1–7.

[10] H. Gomaa, G. Messier, R. Davies, and C. Williamson, "Media caching support for mobile transit clients," in *IEEE WIMOB*, 2009, pp. 79–84.

[11] S. Arianfar, P. Nikander, and J. Ott, "On content-centric router design and implications," in *ACM Proceedings of the Re-Architecting the Internet Workshop*, 2010, p. 5.

[12] G. Rossini and D. Rossi, "Evaluating CCN multi-path interest forwarding strategies," *Computer Communications*, 2013.

[13] N. Laoutaris, H. Che, and I. Stavrakakis, "The LCD interconnection of LRU caches and its analysis," *Performance Evaluation*, vol. 63, no. 7, pp. 609–634, 2006.

[14] L. Hu, J.-Y. Le Boudec, and M. Vojnoviae, "Optimal channel choice for collaborative ad-hoc dissemination," in *IEEE INFOCOM*, 2010, pp. 1–9.

[15] A. Varga, "Omnet++," in *Modeling and Tools for Network Simulation*. Springer, 2010, pp. 35–59.

[16] Inside YouTube Videos. [Online]. Available: http://www.sysomos.com/reports/youtube