Using Social Network Information into ICN

César Bernardini, Thomas Silverston, Olivier Festor

Using Social Network Information into ICN

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*—Online Social Networks carry extremely valuable information about their users and their relationships. We argue that this knowledge can help to drastically improve the efficiency of Information Centric Networks.**

In this paper, we propose a first step to include social information into ICN architectures. We conjecture a small number of users dominate the activity and receive most attention of others users in the social networks and we argue they produce content that will be more likely to be consumed, and in consequence their content must be replicated. We then propose a caching strategy based on prioritizing their content. We simulate a social network model where the proposed caching strategy is evaluated against common ICN caching strategies. Finally, we show that inclusion of social information into ICN networks may help to improve cache performances.

I. Introduction

Social Networking has exponentially grown in the last years. Nowadays, millions of users interact with each other through Facebook, Twitter, Digg, Google Plus, etc. People tweets about what they experience on their life with their community: president Obama's re-election picture became the single most-retweeted message in Twitter history with more than 800,000 re-tweets [1]; In one minute, football player Lionel Messi was mentioned 35,000 times in Twitter after he broke the record of goals in a calendar year [2]; french people issued 420,574 tweets regarding to the election of Miss France [3].Mobile phones' interfaces include functionalities to instantenously share information through the Social Networks. Most if not all the internet services improve your experience through the addition of social features to rapidly spread interesting content. Companies invest strongly into their Facebook pages to promote new products and profit as much as possible user's feedback [1]. News agencies dedicate employees to update their Twitter and Digg pages [4]. 90% of american hospitals use social media to attract new clients and one third has a formal social media plan [5] Users organize social events and keep updated on their friends and families pieces of news through the Internet. We believe these interaction model will continue to expand and to evolve. Internet is becoming social networks oriented.

At the same time and despite its remarkable success, several architectures have been proposed to overcome the limitations of the current Internet such as efficiency, availability, security and mobility. Information Centric Networking (ICN) is a promising new paradigm for the future Internet, where the communication depends on named-data, rather than host names. Indeed, content retrieval, content demand and content identification is lead by its name instead of its physical location. ICN features include in-network caching, multicast support, self-verification, encryption and protection against common current-Internet attacks. ICN architectures include Content Centric Networks (CCN)[2], NetInf[3] and Pursuit[4] among others, compared in [5].

As we said, All ICN nodes support in-networking caching to serve future requests which may help to reduce transport cost for the network and to enhance end-users delivery performances. The availability of different replicas depends on several factors such as cache replacement policies, cache size, content's popularity. Several caching schemes has been evaluated [2], [6], [7], [8], [9] and there are not concensus about the final caching scheme for ICN. We believe the caching scheme for ICN will be based on social network tendencies.

Online Social Networks carry extremly valuable informations about their users and their relationships. We argue that this knowledge can help to drastically improve the efficiency of Information Centric Networks.

In this paper, we propose a first step to include social information into ICN architectures. We conjecture a small number of users dominate the activity and receive most attention of others users in the social networks. We name them Influential users and we argue they produce content that is more likely to be consumed, and in consequence their content must be replicated. We then propose a caching strategy based on prioritizing content from influential users into the social network. We simulate a social network model where the proposed caching strategy is evaluated against common CCN caching strategies. Finally, we show that inclusion of social information into ICN networks help to improve cache performances.

This article is organized as follows. In section II, we give an insight into social networks and ICN. Then in section III we introduce our social network model used across the paper. Section IV details the simulation environment and section V

[1] http://mashable.com/2012/11/06/obama-wins-twitter/

[2] http://www.fcbarcelona.com/club/barca-2-0/detail/article/leo-messi-s-record-reverberates-throughout-social-networks

[3] http://www.peopleinside.fr/twitter-miss-france-signe-un-record/news/52782

[4] http://www.clarin.com/politica/cuenta-Twitter-Camioneros-dia-picante_0_814118751.html

[5] http://www.healthleadersmedia.com/content/TEC-245418/Few-Hospitals-Use-Social-Media-Effectively-Says-Study

presents the results. We discuss our model and results in Section VI and finally, our conclusions are exposed in Section VII.

II. Related Work

A. Social Network

Social Network Analysis enables the examination of patterns in the relationships among interacting users. It gained importance with the raising of the Web 2.0. Theory of *efficient Hubs* or *influentials* says a small number of users dominate the activity and receive most attention of other users. Enterprises have focused their research into opinion leadership in order to improve quality of their products through interpretation of media messages[10], [1]. While their final goal is different, political analysis follows the same approach [11].

We support the idea of using Eigenvector and PageRank as metric to detect influentials nodes in a social network because Google determines most important pages across the Internet using PageRank [12]; PageRank is a variant of Eigenvector centrality measure [13].

To the best of our knowledge, [8] is the only one to deal with social networks and ICN at the same time; in order to compare IP, CDN and ICN, they use the same architecture Twitter servers have in the current Internet. In this paper, we differs on the fact we use social networks' information to improve ICN.

B. Caching 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 Operative Systems, Web-browsers and Proxy-servers, and -we believe- it's well-addressed in [14].

In the context of ICN, researched topics include different concepts such as replacement policies [2], [7], [15], alternative policies[6], [9], different cache sizes[16] over a broad range of topologies such as Binary Trees [2], [7] and common ISP structures [7], [17], [18], [6]. Most of the work focus on common IP client-server model [7], [2], [16], [17], [6], [9] and P2P model [18]. To our knowledge, we are the first to use a social networks model and at the same time to include social networks information to create a social-aware caching strategy.

III. Social Network Model

In this section, we introduce our model. First, we explain the social network model and how the content is produced and consumed. We then map the social users into an ICN network and present a small example to analyze model behavior. Finally, we introduce a social-aware caching strategy based on prioritizing content from influential users in the social network model.

A. Social Network Model

Social Networks allow users to publish new content at their own will. Their friends are always available to watch their friends updates through a news feed system. Every time, users find interesting content they may share it with their friends spreading and expanding the visible scope of information. This is what we call *social network model*: users publishing, consuming and sharing.

We define a *social network model* where users have personal preferences such as topic of interest and groups of friends. In the model, each user has two actions: publish and retrieve.

- Publish: production of new content, all the content issued is self-describing. I.e.: /userC/music/song1 means user C has published a song entitled song1. After retrieving messages, users may share their friends content if they find it interesting enough through a new publish message (*retweet*).
- Retrieve: gets last content issued by their friends. For example, in Figure 1 the user A has a friendship relationship (red dashed line) with users B, E and F. Everytime, A issues a Retrieve message A gets last B, E and F content.

We define a sequence of actions as a finite combination of publish and retrieve users' messages.

B. ICN Model: Content Centric Networks

We select Content Centric Networks (CCN) as the ICN Model due to his wide acceptance in the community. CCN communication architectures relies on two named primitives: *Interest* and *Data*. The name is self-describing and user-readable. A consumer requests content by broadcasting its interest all over the network; any node hearing the request and having the data can issue a response with a data message. As ICN architectures, nodes cache all the *Data* messages that have passed by them. Another fact worthy of mention is that caches have finite space which means replacement and acceptance policies must manage them.

As we said, every user has two actions: *Publish* and *Retrieve*. *Publish* consists of generating new personal content, it does not issue any message in the CCN network, it just update information of CCN nodes to future retrievals of the content. The *Retrieve* operation deals with getting last updates from his friends. The retrieve message generates an Interest for every of his friends. The Interest are then sent across the ICN sorthest path.

C. Example

We illustrate the model by an example where 6 users are distributed over a 9-CCN-nodes network depicting their relationships (red dashed lines) and physical connections (black lines) in Figure 1.

In Figure 1, we can see User E only cares about content from A because it's his only friend and B is concerned by content from Nodes A, C and D. Once B issues a retrieve message, last content from A, C & D users will be requested; when User E issues an retrieve message, only user A is requested by his last updates.

In a more concrete example, we define the sequence of actions as [A.publish(1), B.publish(2), C.publish(3), D.publish(4), E.publish(5), F.publish(6), A.retrieve(), E.retrieve()].

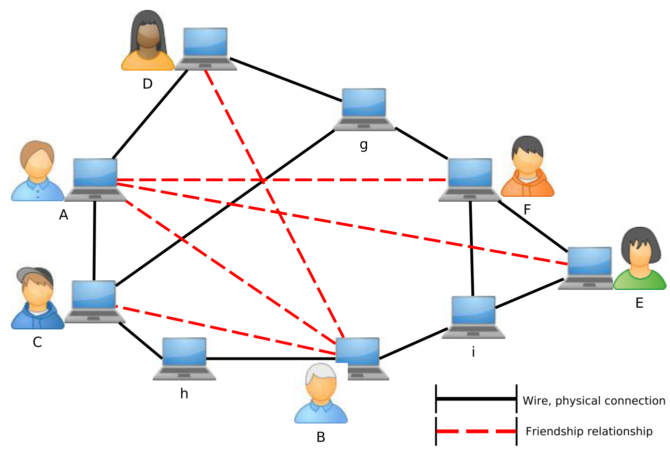

Fig. 1: Social Graph and Topological ICN Structure. The black lines represent connections of ICN nodes while red dashed lines symboline ICN nodes users' relationships.

User	Cached Content		
	Post- *.publish()	Post A.retrieve()	Post E.retrieve()
A	–	2, 5, 6	1, 2, 5, 6
B	–	2	2
C	–	2	2
D	–	5, 6	1, 5, 6
E	–	5	1, 5, 6
F	–	5, 6	1, 5, 6
g	–	5, 6	1, 5, 6
h	–	2	2
i	–	–	–

TABLE I: Content of the Caches after publish and retrieve messages' execution.

After every user publish new content sequentially (A:1, B:2, C:3, D:4, E:5, F:6), internal CCN nodes information is updated and after next-retrieve message, the CCN Data message is issued.

Then, user A and E issue a retrieve message respectivelly. First, A request for last updates of his friends, in Figure 1, users E, F and B. In the CCN architecture, it means an Interest message is issued from node A to node E across the network path [A, D, g, F, E] which stores Content 5; then, A issue another Interest across [A, E, g, F] to receive Content 6, which is stored across the path; finally last Interest for B, stores 2 across the path [A, C, h, B]. Then, E issue a retrieve message which means that path [E, F, g, D, A] caches Content 1. Finally, the post-execution content of the caches is shown in Table I. We can see in the table that non-influential users content (5 and 6) is stored 5 times in the caches while influential content (1, 2) is stored 4 and 5 times. We aim at increasing the availability of influentials contents and we define our social-aware strategy in the incoming section.

D. Eigenvector Social-aware Caching Strategy

In the model, we mentioned ICN nodes' caches are managed with a caching strategy. To compare caching strategies with social-aware caching strategies, we propose a social-aware caching strategy which prioritizes influential-nodes content. We call our strategy *Eigenvector*.

User	Eigenvector value	Influential
A	0.58	Yes
B	0.58	Yes
C	0.29	No
D	0.29	No
E	0.29	No
F	0.29	No
Average	0.38	N/A

TABLE II: Eigenvector values calculated over the graph of social relationships (red dashed lines into Figure 1).

The aim of Eigenvector strategy is to replicate content published by influential users before it's requested in order to improve availability and to reduce number of interest to get the content. We believe users with more friends are more influential than those with less number of friends and they produce content that will be more likely to be consumed.

We detect influential nodes using Eigenvector centrality metric. Eigenvector centrality metric is a measure of the influence of a node in a network; the nodes receives a score according to their importance into the graph; For instance, we first calculate the Eigenvector centrality metric into the social graph (red dashed lines in Figure1) and then we define a node/user as *Influential* if its eigenvector is greater than the average.

In other words, publish messages from non-influential users do not generate replication of content while publish messages from Influential users copy its brand-new content to his friends across the CCN caches in the shortest path of the topology.

Eigenvector works in combination with a common caching strategy (e.g. *LRU*, *FIFO*, *RAND*). For example, *LRU+Eigenvector* refers to Eigenvector with Last Recently Used (LRU) replacement policy. The common caching strategy manages the cached elements: everytime it receives a Data messages, it's stored and when the cache is full, the strategy decides which item to discard to make room for the new ones. In addition, every time influential nodes publish new content it is automatically replicated into the shortest ICN path to their Social Neighbours.

In Table II, we calculate the Eigenvector centrality metric over the graph of social relationships (red dashed lines) in Figure 1. Users A and B are influentials because its Eigenvector value is greater than Eigenvector averaged of all users ($0.58 > 0.38$) while users C, D, E and F are non-influentials ($0.29 < 0.38$). It means when User A publish new content it's copied to his friends across the topology path to E, F and B ([A, D, g, F, E], [A, D, g, F] & [A, C, h, B] respectivelly). In the other hand, E publish message does not generate copies due his non-influential nature.

IV. Simulation Environment

We introduced a model where users publish and request their friends content. Users are distributed across a CCN network. In this section, we show the environment where the simulations were carried out. We developed a simulation tool in Python to represent our model.

To simulate the model, we generate a social network graph and topology structure for CCN. Then, we randomly map social network nodes to CCN nodes. We configure different caching strategies into the CCN nodes. Finally, we analyze the caching performance using a sequence of publish/retrieve messages.

Social networks are well-modeled by small world graphs. That's the reason we represent social networks structures with randomly generated graphs using small-world Newman-Watts-Strogatz algorithm[19] with 50 nodes.

The topological structure of the ICN Internet has not been decided yet. In this context, we choose randomly generated connected graphs to model the topology. The number of nodes is fixed at 100 and the number of edges is randomly selected. All the graph manipulation is handled with Networkx library [20].

We consider three well-known replacement strategies: *Last Recently Used (LRU)*, *Random replacement (RAND)* and *FIFO*. We then compared the results of caching performance with social-aware *Eigenvector* caching strategies: *LRU+Eigenvector*, *Rand+Eigenvector*, *FIFO+Eigenvector*.

Every user has personal preferences and a random location in the CCN network, and according to his personal preferences: he requests, publishes and shares his thoughts in a twitter-like manner. The sequence of actions is generated according to a uniform distribution of 40 events per node (in average) to simulate caching performance of one day. In 2010, Twitter revealed that it has 105 millions registered users and 3,000 millions requests a day; which in simple calculations stand for almost 29 requests per day per user. We argue 2 years has passed from there, and we estimate 40 requests per day per user. [6]. After analyzing their friends' content he might decide to publish again his friend's content. This decision follows an uniform distributed variable.

V. Results

In this section, we assess the performance of social-aware caching strategies against common-used caching strategies. As we pointed out in previous section, we defined a sequence of publish/retrieve messages and then we analyze the performance of CCN caches using several strategies and cache sizes.

First, we introduce the metrics to evalute the performance of the caches:

- *Cache Hit Ratio*: the probability to obtain a cache hit all along the path from a requester to a cache node;
- *Stretch*: the number of hops that the data chunk has travelled in the network with respect to the server storing original copy;
- *Diversity*: express the ratio of distinct content stored across all the caches.
- *Expired-chunks*: express the ratio of non-longer-valable content stored across the caches.

[6]http://www.businessinsider.com/twitter-stats-2010-4?op=1

We then analyze the results. From Figure 2a, we observe that all *Eigenvector* social-aware strategies perform 20% -in average- better than common-used caching strategies in terms of *Cache Hit*. It is noteworthy that bigger cache sizes do not improve performances due to the highly large amount of new publish data.

Stretch is further analyzed in Figure 2b, social caching schemes improves common replacement strategies. Common-used strategies reduces distance to reach content by 33% (stretch of 66%), but our social-aware caching strategy reduce it by 66%. In other words, our social-aware strategy requires to traverse in average 33% of the path to the content while common-used caching strategies need to pass through 66% of the path. Even more, the distance is shortened 50% in comparison with common-used caching strategies. For example using common-used strategies, some user requires to traverse 7-hops in average in order to access content located 10-hops away. while using our social-aware strategy the user requires just to pass through 3-hops.

In the Figure 2c, we depict that social-aware strategies reduce by 6% the ratio of expired elements stored. At first sight, it may seem an slight improvement but every outdate elements represent wasted space into the caches which means we are taking advantage of 6% more of the cache sizes using our approach.

In terms of diversity, common-used strategies keep diverse contents in their caches in comparison with social-aware caching strategies. Even though in Figure 2d common-used strategies are sligthly better in diversity terms (2% in average), our approach drastically increases the performance of 20% and 30% regarding cache hit and stretch respectivelly. For sake of clearness, we only illustrate *LRU* strategy against *LRU+Eigenvector* in the Figure 2d.

To sum up, social-aware caching strategies helps to improve caching performance.

VI. Discussion

We consider Social Networking has arrived to completly change the way we use the Internet.

The aim of the work, it's to show that inclusion of social networks' information is beneficial for ICN. In this context, we introduce a caching strategy based on social properties. We give an first step into social-aware caching strategies and start discussing about new ways of including social patterns and preferences into the ICN architecture.

We detail some limitations of the study. Shortest path algorithm was chosen as routing algorithm because there are not agreement about the final routing protocol for ICN, and in particular por CCN. We insist on the fact that we aim to keep the work as simple as possible, some statistical values represent simplifications of the complex reality of the interaction across the social networks.

VII. Conclusion & Future Work

In this paper we present a social networks-like model: users publish and retrieve content in an ICN environment.

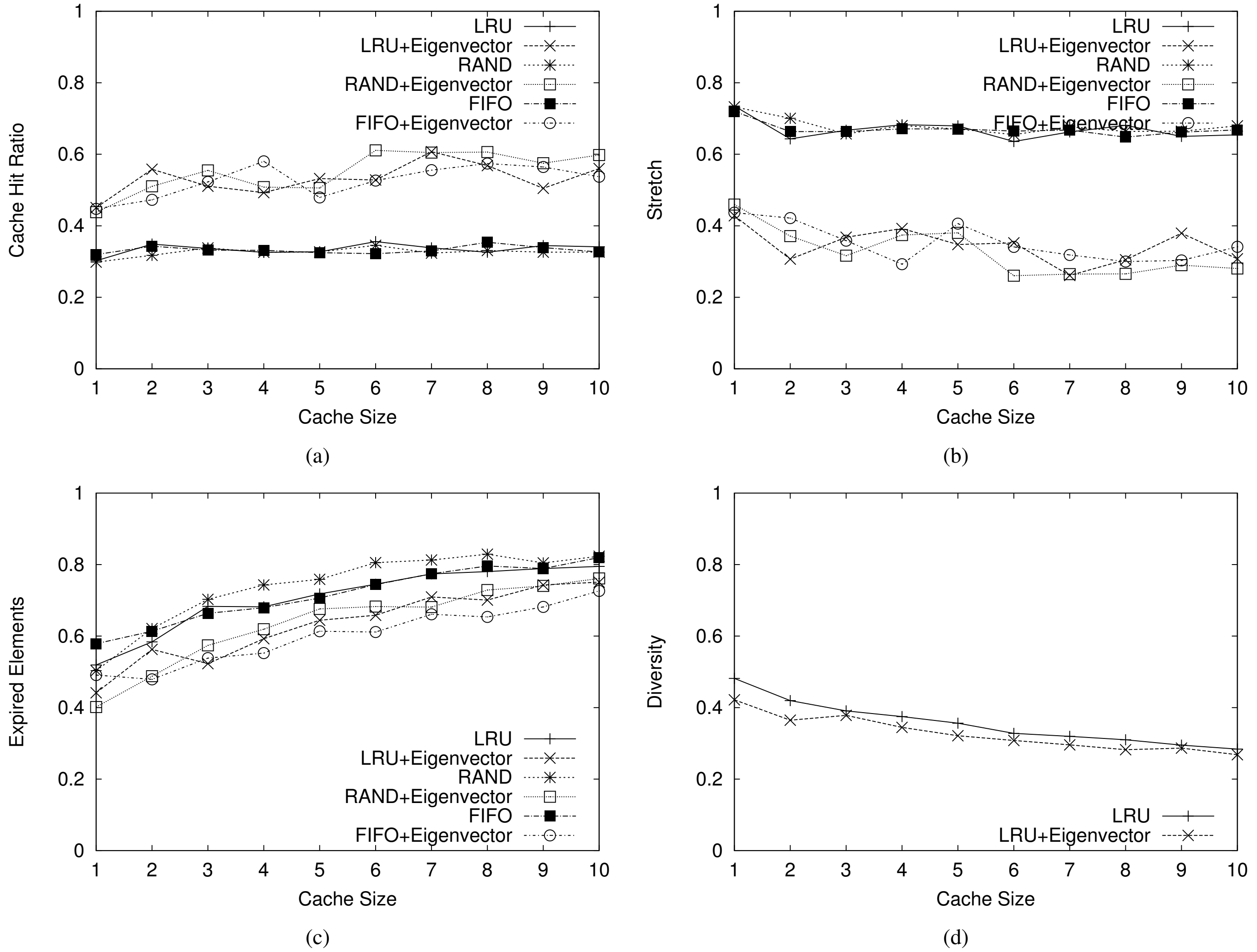

Fig. 2: Comparison between social-aware strategies and common-used ICN strategies according to cache hit, stretch, expired-elements and diversity using different cache sizes.

We then propose to include social networks information into caching strategies in order to improve in-network caching performances to the best of our knowledge, this is the first attempt to merge social networks and ICN.

The social-aware caching strategies show a significant improvement in terms of caching performance. In particular, Eigenvector caching strategy improves results in terms of cache hit (20%), stretch (33%) and at the same time they reduce the number of expired elements into the caches (6%). We consider the Eigenvector strategy as the first step in the research of a social-aware caching strategy for ICN.

Finally, we point out that the inclusion of social information should not be limited to ICN caching related problems. Routing alternatives may as well consider social features to select the best path. Analysis of communities might be a key-point to point ICN routing strategies, we intend to deploy routing and caching mechanisms community-oriented.

References

[1] K. Song, D. Wang, S. Feng, and G. Yu, "Detecting opinion leader dynamically in chinese news comments," in *Proceedings of the 2011 international conference on Web-Age Information Management*, ser. WAIM'11. Berlin, Heidelberg: Springer-Verlag, 2012, pp. 197–209.

[2] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. H. Briggs, and R. L. Braynard, "Networking named content," in *Proceedings of the 5th international conference on Emerging networking experiments and technologies*, ser. CoNEXT '09. New York, NY, USA: ACM, 2009, pp. 1–12. [Online]. Available: http://doi.acm.org/10.1145/1658939.1658941

[3] C. Dannewitz and T. Biermann, "Prototyping a network of information," October 2009.

[4] M. Särelä, T. Rinta-aho, and S. Tarkoma, "RTFM: Publish/subscribe internetworking architecture." ICT Mobile Summit, 2008.

[5] B. Ahlgren, C. Dannewitz, C. Imbrenda, D. Kutscher, and B. Ohlman, "A survey of information-centric networking," *Communications Magazine, IEEE*, vol. 50, no. 7, pp. 26–36, Jul. 2012. [Online]. Available: http://dx.doi.org/10.1109/MCOM.2012.6231276

[6] W. K. Chai, D. He, I. Psaras, and G. Pavlou, "Cache "less for more" in information-centric networks," in *Proceedings of the 11th international IFIP TC 6 conference on Networking - Volume Part I*, ser. IFIP'12. Berlin, Heidelberg: Springer-Verlag, 2012, pp. 27–40.

[7] D. Rossi and G. Rossini, "Caching performance of content centric

networksunder multi-path routing (and more)," Telecom ParisTech, Tech. Rep., 2011.

[8] B. Mathieu, P. Truong, W. You, and J.-f. Peltier, "Information-centric networking: a natural design for social network applications," *IEEE Communications Magazine*, vol. 50, no. 7, pp. 44–51, Jul. 2012. [Online]. Available: http://dx.doi.org/10.1109/MCOM.2012.6231278

[9] C. Bernardini, T. Silverston, and O. Festor, "Mpc: Popularity-based caching strategy for
content centric networks," Inria and Université de Lorraine, Tech. Rep., 2012.

[10] F. Bodendorf and C. Kaiser, "Detecting opinion leaders and trends in online social networks," in *Proceedings of the 2nd ACM workshop on Social web search and mining*, ser. SWSM '09. New York, NY, USA: ACM, 2009, pp. 65–68. [Online]. Available: http://doi.acm.org/10.1145/1651437.1651448

[11] N. Agarwal, M. Lim, and R. Wigand, "Online collective action and the role of social media in mobilizing opinions: A case study on womens right-to-drive campaigns in saudi arabia," in *Web 2.0 Technologies and Democratic Governance*, ser. Public Administration and Information Technology, C. G. Reddick and S. K. Aikins, Eds. Springer New York, 2012, vol. 1, pp. 99–123.

[12] M. Franceschet, "Pagerank: standing on the shoulders of giants," *Commun. ACM*, vol. 54, no. 6, pp. 92–101, Jun. 2011. [Online]. Available: http://doi.acm.org/10.1145/1953122.1953146

[13] A. N. Langville and C. D. Meyer, "A Survey of Eigenvector Methods for Web Information Retrieval," *SIAM Rev.*, vol. 47, no. 1, pp. 135–161, Jan. 2005. [Online]. Available: http://dx.doi.org/10.1137/S0036144503424786

[14] S. Podlipnig and L. Böszörmenyi, "A survey of web cache replacement strategies," *ACM Comput. Surv.*, vol. 35, no. 4, pp. 374–398, Dec. 2003. [Online]. Available: http://doi.acm.org/10.1145/954339.954341

[15] S. B. Luca Muscariello, Giovanna Carofiglio, "Accelerated lru replacement with probabilistic caching: A fast algorithm to approximate lfu," CCNxConf, Sep 2012.

[16] D. Rossi and G. Rossini, "On sizing CCN content stores by exploiting topological information," in *Computer Communications Workshops (INFOCOM WKSHPS), 2012 IEEE Conference on*. IEEE, Mar. 2012, pp. 280–285. [Online]. Available: http://dx.doi.org/10.1109/INFCOMW.2012.6193506

[17] J. Choi, J. Han, E. Cho, T. Kwon, and Y. Choi, "A Survey on content-oriented networking for efficient content delivery," *Communications Magazine, IEEE*, vol. 49, no. 3, pp. 121–127, Mar. 2011. [Online]. Available: http://dx.doi.org/10.1109/MCOM.2011.5723809

[18] 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 *Proceedings of the 21st international Conference on Computer Communication Networks*, Munich, Germany, 2012.

[19] M. E. J. Newman, D. J. Watts, and S. H. Strogatz, "Random graph models of social networks," *Proc. Natl. Acad. Sci. USA*, vol. 99, pp. 2566–2572.

[20] A. A. Hagberg, D. A. Schult, and P. J. Swart, "Exploring network structure, dynamics, and function using NetworkX," in *Proceedings of the 7th Python in Science Conference (SciPy2008)*, Pasadena, CA USA, Aug. 2008, pp. 11–15.