International Journal of
Scientific Engineering and Technology Research

ISSN 2319-8885
Vol.03,Issue.43
December-2014,
Pages:8753-8757

www.ijsetr.com

Optimal and Scalable Performance in Social Wireless Network

ADABALA S G R SRINIVAS[1], YUVARAJU CHINNAM[2]

[1]PG Scholar, Dept of CSE, Sri Sai Madhavi Institute of Science & Technology, JNTUK, AP, India,
Email: adabala.bujji@gmail.com.

[2]Assoc Prof, Dept of CSE, Sri Sai Madhavi Institute of Science & Technology, JNTUK, AP, India,
Email: chinnamyuvraj@gmail.com.

Abstract: The applications based on social networking have brought revolution towards social life and are continuously gaining popularity among the Internet users. In these days with the advanced technologies came in to existence number of resources offered by different hardware. Today's world all are using mobile devices these social networks are offering many offers like games, apps and so on. Moreover, the mobile devices are considered more personal as compared to their desktop rivals, so there is a tendency among the mobile users to store sensitive data like contacts, passwords, bank account details, updated calendar entries with key dates and personal notes on their devices. The Project Social Wireless Network Secure Identification (SWIN) to explore the practicality of providing the secure mobile social networking portal with advanced security features to tackle potential security threats by extending the existing methods with more innovative security technologies. In addition to the extensive background study and the determination of marketable use-cases with their corresponding security requirements, this thesis proposes a secure identification design to satisfy the security dimensions for both online and offline peers. We have implemented an initial prototype using PHP Socket and OpenSSL library to simulate the secure identification procedure based on the proposed design. The design is in compliance with 3GPP"s Generic Authentication Architecture (GAA) and our implementation has demonstrated the flexibility of the solution to be applied independently for the applications requiring secure identification. Finally, the thesis provides strong foundation for the advanced implementation on mobile platform in future. The paper constructs analytical and simulation models for analyzing the proposed caching strategies in the presence of selfish users that deviate from network-wide cost-optimal policies. It also reports results from an Android phone based prototype SWNET, validating the presented analytical and simulation results.

Keywords: Social Wireless Networks, Cooperative Caching, Content Provisioning, Ad Hoc Networks.

I. INTRODUCTION

A. Motivation

Nowadays people walk around carrying all sorts of devices such as cellphones, PDAs, laptops, etc. Typically, these devices are able to communicate with each other in short distances by using communication technologies such as blue-tooth. These networks, also known as Pocket Switched Networks (PSN [1], [2]), can be a key technology to provide innovative services to the users without the need of any fixed infrastructure. Pocket Switched Networks fall in the class of the Delay Tolerant Networks (DTNs). In DTNs, messages can multi-hop from source to destination by using the forwarding opportunities given by the contacts between the nodes. These networks are usually disconnected, are characterized by social-based mobility and heterogeneous contact rate. Examples of such networks include people in working places, students in university campuses, and citizens in metropolitan areas. In this paper, we study how effective wireless social communities networks are, where we are in particular interested in the case where a wireless social community networks competes with traditional a licensed band cellular network. To do this, we first investigate how users decide whether or not to join the social community network, and study the evolution of the number of members in the community by modeling users' payoffs as a function of the subscription fee1, as well as the operators' provided coverage. For this case, we derive pricing strategies to maximize the coverage of the social community network. Next, we study the competition between a social community operator and cellular wireless network using a game-theoretic framework. For this case, we investigate the existence a Nash equilibrium, and characterize the number of users that subscribe to each service provider under a Nash equilibrium. Due to space constraints we present our results without proof and will focus at several instances on particular special cases, as discussed in the following.

Continuous developments of mobile technologies and use of devices such as smart phones in everyday life increase need to be continuously connected to others through WiFi and to the Internet, anywhere and at any time. In mobile environments user connectivity is mainly affected by wireless communications constraints and mobility of user. These boundary conditions do not allow us to design communication environments based on unique and fixed connected networks or assume a stable path between each pair of source and destination. [7] Any mobile node can exchange information opportunistically during their periods of contact with any other node, fixed or mobile. Network protocols are designed to be extremely resilient to events such as long partitions, node disconnections, etc, which are very features of this type of self-organizing, self adaptable mobile social networks. This is achieved by temporarily storing messages at intermediate nodes, waiting for future opportunities to forward messages towards their destination.

B. Optimal Solution

For contents with varying level of popularity, a greedy approach for each node would be to store as many distinctly popular contents as its storage allows. This approach amounts to noncooperation and can give rise to heavy network-wide content duplications. In the other extreme case, which is fully cooperative, a node would try to maximize the total number of unique contents stored within the SWNET by avoiding duplications. In this paper, we show that none of the above extreme approaches can minimize the content provider's cost. We also show that for a given rebate-to-download-cost ratio, there exists an object placement policy which is somewhere in between those two extremes, and can minimize the content provider's cost by striking a balance between the greediness and full cooperation [26]. This is referred to as optimal object placement policy in the rest of this paper. The proposed cooperative caching algorithms strive to attain this optimal object placement with the target of minimizing the network-wide content provisioning cost.

C. User Selfishness

The potential for earning peer-to-peer rebate may promote selfish behavior in some users. A selfish user is one that deviates from the network-wide optimal policy in order to earn more rebates. Any deviation from the optimal policy is expected to incur higher network-wide provisioning cost. In this work, we analyze the impacts of such selfish behaviour on object provisioning cost and the earned rebate within the context of an SWNET. It is shown that beyond a threshold selfish node population, the amount of per-node rebate for the selfish users is lower than that for the nonselfish users. In other words, when the selfish node population is beyond a critical point, selfish behavior ceases to produce more benefit from a rebate standpoint.

D. Contributions

First, based on a practical service and pricing case, a stochastic model for the content provider's cost computation is developed. Second, a cooperative caching strategy, Split Cache, is proposed, numerically analyzed, and theoretically proven to provide optimal object placement for networks with homogenous content demands. Third, a benefit-based strategy, Distributed Benefit, is proposed to minimize the provisioning cost in heterogeneous networks consisting of nodes with different content request rates and patterns. Fourth, the impacts of user selfishness on object provisioning cost and earned rebate is analyzed. Finally, numerical results for both strategies are validated using simulation and compared with a series of traditional caching policies.

II. NETWORK, SERVICE, AND PRICING MODEL

A. Network Model

Fig. 1 illustrates an example SWNET within a University campus. End Consumers carrying mobile devices form SWNET partitions, which can be either multi-hop (i.e., MANET) as shown for partitions 1, 3, and 4, or single hop access point based as shown for partition 2. A mobile device can download an object (i.e., content) from the CP's server using the CSP's cellular network, or from its local SWNET partition. In the rest of this paper, the terms object and content are used synonymously. We consider two types of SWNETs. The first one involves stationary [1] SWNET partitions. Meaning, after a partition

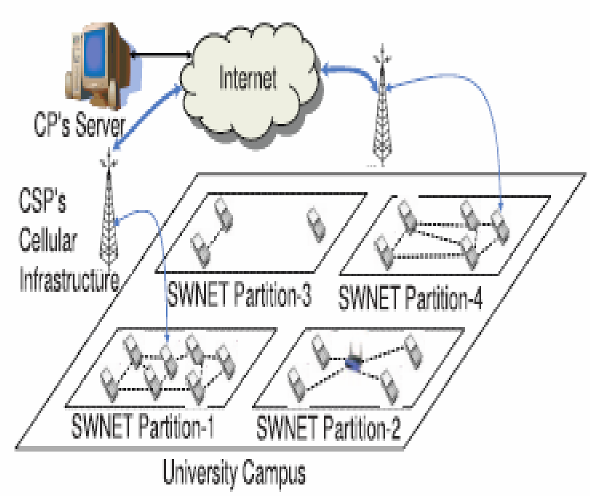

Fig1: Content access from an SWNET in a University Campus.

is formed, it is maintained for sufficiently long so that the cooperative object caches can be formed and reach steady states. We also investigate a second type to explore as to what happens when the stationary assumption is relaxed. To investigate this effect, caching is applied to SWNETs formed using human interaction traces obtained from a set of real SWNET nodes [2].

B. Search Model

After an object request is originated by a mobile device, it first searches its local cache. If the local search fails, it searches the object within its SWNET partition using limited broadcast message. If the search in partition also fails, the object is downloaded from the CP's server using the CSP's 3G/4G cellular network. In this paper, we have modelled objects such as electronic books, music, etc., which are time nonvarying, and therefore cache consistency is not a critical issue. We first assume that all objects have the same size and each node is able to store up to "C" different objects in its cache. Later, in Section 5.3, we relax this assumption to support objects with varying size. We also assume that all objects are popularity-tagged by the CP's server [3]. The popularity-tag of an object indicates its global popularity; it also indicates the probability that an arbitrary request in the network is generated for this specific object.

C Pricing Model

We use a pricing model similar to the Amazon Kindle business model in which the CP (e.g., Amazon) pays a download cost Cd to the CSP when an End-Consumer downloads an object from the CP's server through the CSP's cellular network. Also, whenever an EC provides a locally cached object to another EC within its local SWNET partition, the provider EC is paid a rebate Cr by the CP. Optionally, this rebate can also be distributed among the provider EC and the ECs of all the intermediate mobile devices that take part in content forwarding. Fig. 2 demonstrates the cost and content flow model. As it is shown in Fig. 2, Cd corresponds to the CP's object delivering cost when it is delivered through the CSP's network, and Cr corresponds to the rebate given out to an EC when the object is found within the SWNET (e.g., node A receives rebate Cr after it provides a content to node B over the SWNET). For a given Cr=Cd ratio, the paper aims to develop optimal object placement policies that can minimize the network-wide content provisioning cost.

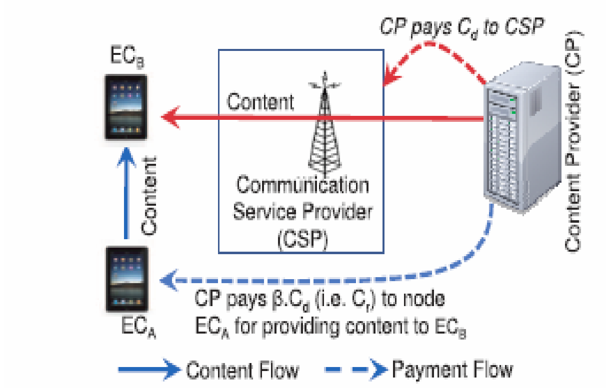

Fig 2: Content and cost flow model.

A digitally signed rebate framework needs to be supported so that the rebate recipient ECs can electronically validate and redeem the rebate with the CP. Also, a digital usage right mechanism [4] is needed so that an EC which is caching an object (e.g., an e-book) should not necessarily be able to open/read it unless it has explicitly bought the object from the CP. We assume the presence of these two mechanisms on which the proposed caching mechanism is built. Operationally, the parameters Cd and Cr are set by a CP and CSP based on their operating cost and revenue models. The end-consumers do not have any control on those parameters.

D. Request Generation Model

We study two request generation models, namely, homogenous and heterogeneous. In the homogenous case, all mobile devices maintain the same content request rate and pattern which follow a Zipf distribution. Zipf distribution is widely used in the literature for modeling popularity based online object request distributions [5]. According to Zipf law, the popularity of the ith popular object out of N different objects can be expressed as in the heterogeneous request model, each mobile device follows an individual Zipf distribution. This means popularity of object j is not necessarily the same from two different nodes standpoints. This is in contrast to the homogenous model in which the popularity of object j is same from the perspective of all network nodes. Also, the object request rate from different nodes is not necessarily the same in the heterogeneous model.

III. COST UNDER HOMOGENEOUS REQUEST MODEL

In this section, we compute the average object provisioning cost under a homogenous request model. Let PL be the probability of finding a requested object in the local cache (i.e., local hit rate), PV be the probability that a requested object can be found in the local SWNET partition (i.e., remote hit rate) after its local search fails, and PM be the probability that a requested object is not found in the local cache and in the remote cache (i.e., miss rate). We can write PM in terms of PV and PL as PM ¼ 1 _ PL _ PV : ð2Þ According to the pricing model in Section 2.3, the provisioning cost for an object is zero if it is found in the local cache, Cr when it is found in the SWNET, and Cd when it is downloaded from the CP's server through the CSP's network.

IV. OPTIMAL OBJECT PLACEMENT

For a given _, the cost in (6) is a function of the vector ~n ¼ <n1; n2; . . . ; nN>, where ni shows the number of copies of object "i" in the SWNET partition in question. An object placement ~n is optimal when it leads to minimum object provisioning cost in (6). In this section, we aim to determine the optimal ~n. An object should not be stored in a partition when at least one object of higher popularity is missing in that partition. That is, object i (i.e., ith popular object) cannot be cached while a higher popularity object k (k < i) is missing. Note that the above analysis does not help deciding the value of ‘, or the set of objects that need to be duplicated for the optimal object placement solution. It only shows that if the optimal solution requires duplication, it must be across all nodes. In the next section, we show how to determine the value of ‘.

V. CACHING FOR OPTIMAL OBJECT PLACEMENT

A. Split Cache Replacement

To realize the optimal object placement under homogeneous object request model we propose the following Split Cache policy in which the available cache space in each device is divided into a duplicate segment (_ fraction) and a unique segment (see Fig. 3). In the first segment, nodes can store the most popular objects without worrying about the object duplication and in the second segment only unique objects are allowed to be stored. The parameter _ in Fig. 3 (0 _ _ _ 1) indicates the fraction of cache that is used for storing duplicated objects. With the Split Cache replacement policy, soon after an object is downloaded from the CP's server, it is categorized as a unique object as there is only one copy of this object in the network. Also, when a node downloads an object from another SWNET node, that object is categorized as a duplicated object as there are now at least two copies of that object in the network. For storing a new unique object, the least popular object in the whole cache is selected as a candidate and it is replaced with the new object if it is less popular than the new incoming object. For a duplicated object, however, the evictee candidate is selected only from the first duplicate segment of the cache. In other words, a unique object is never evicted in order to accommodate a duplicated object. The Split Cache object replacement mechanism realizes the optimal strategy established in Section 4. With this mechanism, at steady state all devices' caches maintain the same object set in their duplicate areas, but distinct objects in their unique areas. The pseudocode of Split Cache replacement policy is shown in Algorithm 1.

VI. CACHING UNDER HETEROGENEOUS REQUESTS

The Split Cache policy in Section 5 may not able to minimize the provisioning cost for nonhomogenous object requests where nodes have different request rates and request patterns. In this section, we propose and analyze a benefitbased heuristics approach to minimize the object provisioning cost in a network with nonhomogenous request model. The probability that a node "i" finds the requested object in its own cache is P j2si pj i , where si indicates the set of stored object in node "i" and pj i shows the probability that a generated request in node "i" is for object "j." The probability that a request is found in the network after its local search fails is equal to P j2ðS_siÞ pj i , where _ represents the set of all objects stored in the network. Finally, the probability that an object is not available in the network and needs to be downloaded from the CP's server is 1 _ P j2S pj i.

VII. USER SELFISHNESS AND ITS IMPACTS

In Section 5, we computed the cost and rebate in a cooperative SWNET with homogeneous requests where all nodes run the split replacement policy with optimal _. The impacts of user selfishness on object provisioning cost are analyzed in this section. Note that the following study is limited only for homogenous content requests and it assumes that there is no collusion among nodes that behave in a selfish manner.

Selfishness: A node is defined to be selfish when it deviates from optimal caching in order to earn more rebates. A rational selfish node stores an object only if that object increases the amount of its own potential rebate. The set of objects in such a node is expected to be different from that of a nonselfish node. To analyze the impacts of user selfishness we first need to know which policy maximizes the rebate for a selfish node. In the absence of collusion, each selfish node assumes that there is no other such node in the network. Therefore, all selfish nodes choose the same policy, namely, Split Cache, but with a split factor _ that is different from the optimal _ as established for nonselfish operation in Section 5. If a selfish node decides to store a duplicated object, the other selfish nodes will do the same thing since they are not colluding and are forced to run the exact same policy when they use the same exact information. Therefore, there is no partially duplicated object in the network. Degree of selfishness in an SWNET is modeled by parameters _, which is the number of selfish nodes, and _S which is the non-optimal split-factor chosen by those nodes.

$$P_L^C = H_D^C + \frac{(1-\lambda_o)C}{\mu C} H_U$$

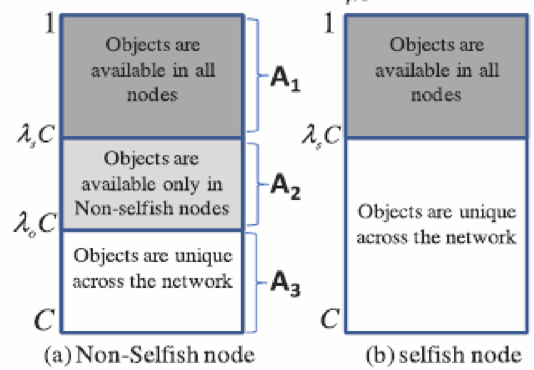

Fig 3: Cache status at steady state.

VIII. PERFORMANCE WITH HOMOGENEOUS REQUESTS

The performance of Split Cache was evaluated using the analytical expressions in Section 5, and then via ns2 network simulation. For simulation, a flooding-based object search mechanism has been implemented using the baseline AODV [6] route discovery syntaxes. Baseline experimental parameters are summarized in Table 1.

A. Hit Rates and Provisioning Cost

Fig. 3 depicts the impacts of _ on the hit rates. The _ ¼ 0 case represents zero duplication, leading to the maximum number of unique objects in the partition. The _ ¼ 1 case causes maximum duplication. In this case, all nodes cache the same set of C (cache size) most popular objects. Smaller _ values lead to very few copies of the popular objects within the local cache and the subsequent low local hit rates. Since with larger _, more and more popular objects are duplicated, the likelihood of finding objects locally improves, leading to higher PL values.

IX. CONCLUSION

The objective of this work was to develop a cooperative caching strategy for provisioning cost minimization in Social Wireless Networks. In this paper, we presented our design and implementation of cooperative cache in wireless P2P networks. In our asymmetric approach, data request packets are transmitted to the cache layer on every node; however, the data reply packets are only transmitted to the cache layer on the intermediate nodes which need to cache the data. This solution not only reduces the overhead of copying data between the user-space and the kernel space, but also allows data pipeline to reduce the end-to-end delay. We have developed a prototype to demonstrate the advantage of the asymmetric approach. Since our prototype is at a small scale, we evaluate our design for a large scale network through simulations. Our simulation results show that the asymmetric approach outperforms the symmetric approach in traditional 802.11 based ad hoc networks by removing most of the processing overhead. In mesh networks, the asymmetric approach can significantly reduce the data access delay compared to the symmetric approach due to data pipelines. It was shown that selfishness can increase user rebate only when the number of selfish nodes in an SWNET is less than a critical number.

X. FUTURE WORK

It was shown that with heterogeneous requests, a benefit based heuristics strategy provides better performance compared to split cache which is proposed mainly for homogeneous demand. Ongoing work on this topic includes the development of an efficient algorithm for the heterogeneous demand scenario, with a goal of bridging the performance gap between the Benefit Based heuristics and the centralized greedy mechanism which was proven to be optimal in Section 6.4. Removal of the no-collusion assumption for user selfishness is also being worked on.

XI. REFERENCES

[1] M. Zhao, L. Mason, and W. Wang, “Empirical Study on Human Mobility for Mobile Wireless Networks,” Proc. IEEE Military Comm. Conf. (MILCOM), 2008.

[2] “Cambridge Trace File, Human Interaction Study,” http:// www.crawdad.org/download/cambridge/haggle/Exp6.tar.gz, 2012.

[3] E. Cohen, B. Krishnamurthy, and J. Rexford, “Evaluating Server- Assisted Cache Replacement in the Web,” Proc. Sixth Ann. European Symp. Algorithms, pp. 307-319, 1998.

[4] S. Banerjee and S. Karforma, “A Prototype Design for DRM Based Credit Card Transaction in E-Commerce,” Ubiquity, vol. 2008, 2008.

[5] L. Breslau, P. Cao, L. Fan, and S. Shenker, “Web Caching and Zipf- Like Distributions: Evidence and Implications,” Proc. IEEE INFOCOM, 1999.

[6] C. Perkins and E. Royer, “Ad-Hoc On-Demand Distance Vector Routing,” Proc. IEEE Second Workshop Mobile Systems and Applications, 1999.

[7] S. Podlipnig and L. Boszormenyi, “A Survey of Web Cache Replacement Strategies,” ACM Computing Surveys, vol. 35, pp. 374- 398, 2003.

[8] A. Chaintreau, P. Hui, J. Crowcroft, C. Diot, R. Gass, and J. Scott, “Impact of Human Mobility on Opportunistic Forwarding Algorithms,” IEEE Trans. Mobile Computing, vol. 6, no. 6, pp. 606-620, June 2007.

[9] “BU-Web-Client - Six Months of Web Client Traces,” http:// www.cs.bu.edu/techreports/1999-011-usertrace-98.gz, 2012.

[10] A. Wolman, M. Voelker, A. Karlin, and H. Levy, “On the Scale and Performance of Cooperative Web Caching,” Proc. 17th ACM Symp. Operating Systems Principles, pp. 16-31, 1999.

[11] S. Dykes and K. Robbins, “A Viability Analysis of Cooperative Proxy Caching,” Proc. IEEE INFOCOM, 2001.

[12] M. Korupolu and M. Dahlin, “Coordinated Placement and Replacement for Large-Scale Distributed Caches,” IEEE Trans. Knowledge and Data Eng., vol. 14, no. 6, pp. 1317-1329, Nov. 2002.

[13] L. Yin and G. Cao, “Supporting Cooperative Caching in Ad Hoc Networks,” IEEE Trans. Mobile Computing, vol. 5, no. 1, pp. 77-89, Jan. 2006.

Author’s Profile:

Adabala S G R Srinivas, PG Scholar, Department of Computer Science & Engineering, Sri Sai Madhavi Institute of science &Technology, JNTUK A.P, India.
Email: adabala.bujji@gmail.com**.**

Yuvaraju Chinnam Associate Professor, Department of Computer Science & Engineering, Sri Sai Madhavi Institute Of Science & Technology, JNTUK A.P, India.
Email: chinnamyuvraj@gmail.com.