Efficient Cooperative Cache Replacement Algorithm for Distributed Social Wireless Networks

K. Kiranmai[1], Y.Leela krishna[2],
[1] Pursuing M.Tech in CSE, E.V.M College of Engineering & Technology, Andhra Pradesh, India
[2] Asst. Professor, Department of CSE, E.V.M College of Engineering & Technology, Andhra Pradesh, India,
k.kiranmai111@gmail.com, leelakrishnayalavarthi@gmail.com [2]

Abstract: This paper introduces cooperative caching techniques for tumbling electronic content provisioning cost in Social Wireless Networks (SWNET).SWNETs are formed by mobile devices, such as modern cell phones etc. sharing common takings in electronic content, and actually meeting together in public places. Electronic object caching in such SWNETs are shown to be able to reduce the content provisioning cost which depends heavily on the service and price dependences among various stakeholders including content providers (CP), network service providers, and End Consumers (EC).This paper develops practical network, service, and pricing models which are then used for creating two object caching strategies for reducing content provisioning costs in networks with homogenous and heterogeneous object demands. The paper develops analytical and simulation designs 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: cooperative cache; cost-optimal policies; distributed search engines; selfish users

I. INTRODUCTION

1.1. Motivation

Latest emergence of data enabled mobile devices and Wireless-enabled data applications have fostered new content dissemination models in today's mobile ecosystem. A record of such devices includes iPhone, Android phones Amazon's kindle and electronic book readers from additional vendors. The set of data applications includes mobile phone apps. The level of propagation of mobile applications is indicated by the example fact that as of October 2010, apple's app store offered over 100,000 apps that are downloadable by the smart phone users. With the unadventurous download model, a user downloads contents directly from a content providers' (CP) server over a communication service providers' (CSP) network. Downloading data through CSP's group involves a price which must be paid either by customers or by the content provider. In this effort we take on Amazon kindle electronic book delivery selling model in which the CP (Amazon) pays to sprint, the CSP, for the cost of network usage due to downloaded electronic books by kindle clients. support and wpa supplicant for WiFi encryption.

When users carrying mobile devices physically gather in settings such as University campus, work place, Mall, Airport and other public places, Social Wireless Networks (SWNETs) can be formed using ad hoc wireless connections between the devices. With the existence of such SWNETs, an alternative approach to content access by a device would be to first search the local SWNET for the requested content before downloading it from the CP's server. The expected content provisioning cost of such an approach can be significantly lower since the download cost to the SP would be avoided when the content is found within the local SWNET. This mechanism is termed as cooperative caching.

1.2 Evolution of catching

In 1995, Jacobson proposed that caching could deal with the exponential growth of the internet. According to Jacobson, data has to find local sources near consumers rather than always coming from the place it was originally produced. To match with the exponential growth of the internet, cache has to grow as fast as the internet - exponentially. The question remains as to how should we architect lots of caches. The taxonomy of caching was identified in 1982. Dowdy et al. proposed that an acceptable File Assignment Problem (FAP) solution assigns files to the nodes in some "optimal" fashion. "optimality" was identified to be measured by cost and performance.

A. Single Cache and Multiple Caches

In 1999, Shim et al. introduced a single cache algorithm in. Albeit relatively simple with single cache proxy architecture, the LNC-R-W3-U algorithm took into consideration both cache replacement and consistency maintenance for web proxies. In 2000, Fan et al. described a multiple caching protocol - "summary cache". In the summary cache protocol, every proxy keeps a summary of the cache directory of each participating proxy and checks the summaries for potential hits before sending any queries. Fan et al. recognized that to fully benefit from caching, caches should cooperate and serve each other's misses to reduce the total traffic through the bottleneck.

B. Hierarchical Caching

To increase the hit rate of web cache, large scale caching structure in which caches cooperate with each other were brought into sight - "hierarchical" and "distributed" caching. In hierarchical caching, caches are placed in different network levels. As early as 1993, polynomial time algorithm for hierarchical placement problem was presented. Harvest structure was considered to be the first hierarchical caching structure.It was named to illustrate its initial focus on reaping the growing crop of Internet information . Figure 1 is the overall Harvest architecture in which a Harvest Gatherer collects indexing information from across the internet, while a Broker collects information from many gatherers and provide query interface to the gathered information. Brokers can also collect information from other brokers to cascade the views from others.

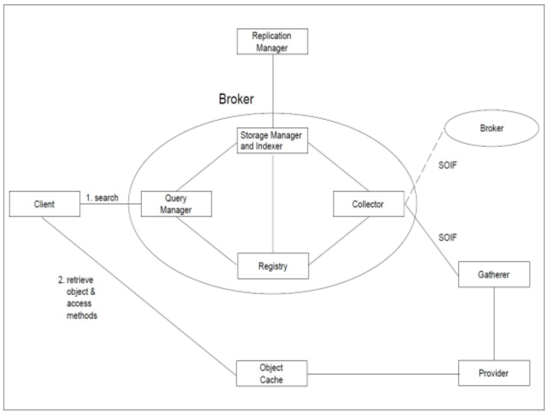

Fig. 1: Overall Harvest Architecture

1.3. Optimal Solutions

For contents with changeable level of popularity, a greedy approach for each node would be to store as many distinctly popular contents as its storage allows. This approach sums to noncooperation and can grow to heavy network-wide data duplications. In the other excessive case, which is fully cooperative, a terminal would try to make the best of the total number of single contents stored within the SWNET by avoiding duplication. In this paper, we show that none of the above excessive approaches can reduce the content provider's charge. We also show that for a given rebate-to-download-charge ratio, there is present an object placement policy which is somewhere in between those two ends, and can increase the content provider's cost by striking a stability between the greediness and full cooperation. This is referred to as optimal object placement policy in the rest of this paper. The proposed cooperative caching algorithms strive to attain this best object placement with the target of reducing the network-wide content provisioning price.

1.3 User Selfishness

The probability for earning peer-to-peer rebate may encourage selfish activities in some clients. A selfish client is one that diverges from the network-wide finest policy in order to receive more rebates. Any distinction from the optimal policy is expected to incur higher network-wide provisioning cost. In this work, we revise the impacts of such selfish behavior on object provisioning cost and the earned refund within the context of a SWNET. It is given that beyond a threshold selfish node population, the amount of per-node rebate for the selfish users is lower than that for the unselfish users. In supplementary terms, when the selfish terminal population exceeds a certain point, selfish actions discontinue producing more advantage from a refund standpoint.

1.4. Existing System

With the existence of such SWNETs, an alternative approach to content access by a device would be to first search the local SWNET for the requested content before downloading it from the CP's server. The expected content provisioning cost of such an approach can be significantly lower since the download cost to the CSP would be avoided when the content is found within the local SWNET. This mechanism is termed as cooperative caching. In order to encourage the End-Consumers (EC) to cache previously downloaded content and to share it with other end-consumers, a peer-to-peer rebate mechanism is proposed. This mechanism can serve as an incentive so that the end-consumers are enticed to participate in cooperative content caching in spite of the storage and energy costs. In order for cooperative caching to provide cost benefits, this peer-to-peer rebate must be dimensioned to be smaller than the content download cost paid to the CSP. This rebate should be factored in the content provider's overall cost.

1.5. PROPOSED SYSTEM

In this paper drawing motivation from Amazon's Kindle electronic book delivery business, this paper develops practical network, service, and pricing models which are then used for creating two object caching strategies for minimizing content provisioning costs in networks with homogenous and heterogeneous object demands. 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 We focus our attention on the blind recovery of secret data hidden in medium hosts via multi-carrier/signature direct-sequence spread-spectrum transform domain embedding.

II. . IMPLEMENTATION

Implementation is the stage of the project when the theoretical design is turned out into a working system. Thus it can be considered to be the most critical stage in achieving a successful new system and in giving the user, confidence that the new system will work and be effective. The implementation stage involves careful planning, investigation of the existing system and it's constraints on implementation, designing of methods to achieve changeover and evaluation of changeover methods.

MODULE DESCRIPTION

- Network Model
- Search Model
- Pricing Model

2.1 Network Model

Here the Fig.2 describes a model SWNET within a University grounds. People carrying mobile devices form SWNET partitions are the end consumers, which can be whichever multi-hop (i.e. MANET) as shown for partitions 1, 3, and 4, or single hop contact point based as shown for partition 2. A movable device can download some data (i.e., content) from the CP's server using the CSP's cellular system, or from its home SWNET partition. In the remaining paper, the terms object and content are used synonymously. We regard as two types of SWNETs. The foremost one involves motionless SWNET partitions. Meaning, after a partition is formed, it is maintained for sufficiently long so that the cooperative object caches can be formed and reach fixed states. We also consider a second type to explore as to what happens when the still 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.

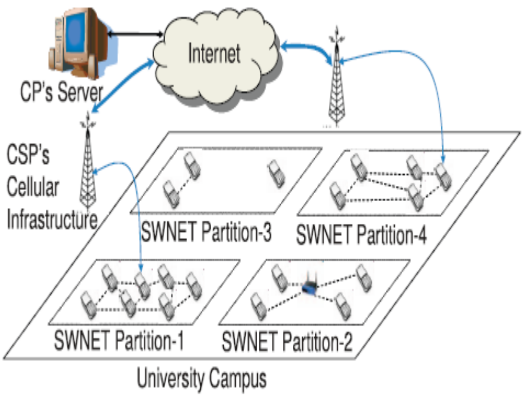

Fig. 2. Accessing content from an SWNET in a University Campus.

2.2 Search Model

After an object call is originated by a mobile tool, it first finds in its local cache. If the local search fails, it searches the object within its SWNET division using limited transmit note. If the search in division also fails, the data is downloaded from the CP's server using the CSP's 3G/4G cellular arrangement. In this paper, we have designed objects such as electronic books, music, etc., which does not vary on time, and therefore cache constancy is not a serious issue. We first suppose that all objects have the equivalent size and each terminal is able to store up to "C" dissimilar data in its cache. Later on, we let go this supposition to sustain objects with variable size. We also believe that all objects are popularity-tagged by the CP's server. The popularity-tag of an object points out its universal recognition; it also indicates the chances that a subjective request in the network is produced for this specific object.

2.3Pricing Model

We use a pricing model similar to the Amazon Kindle business model in which the CP 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 .The selling price is directly paid to the CP by an EC through an out-of-band secure payment system. 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. We assume the presence of these two mechanisms on which the proposed caching mechanism is built.

III.CACHING FOR OPTIMAL OBJECT PLACEMENT

3.1. Cache Replacement

Caching in wireless environment has unique constraints like scarce bandwidth, limited power supply, high mobility and limited cache space. Due to the space limitation, the mobile nodes can store only a subset of the frequently accessed data. The availability of the data in local cache can significantly improve the performance since it overcomes the constraints in wireless environment. A good replacement mechanism is needed to distinguish the items to be kept in cache and that is to be removed when the cache is full. While it would be possible to pick a random object to replace when cache is full, system performance will be better if we choose an object that is not heavily used. If a heavily used data item is removed it will probably have to be brought back quickly, resulting in extra overhead. So a good replacement policy is essential to achieve high hit rates. The extensive research on caching for wired networks can be adapted for the wireless environment with modifications to account for mobile terminal limitations and the dynamics of the wireless channel.

3.2. Cache Replacement Policies in Ad hoc networks

Data caching in MANET is proposed as cooperative caching. In cooperative caching the local cache in each node is shared among the adjacent nodes and they form a large unified cache. So in a cooperative caching environment, the mobile hosts can obtain data items not only from local cache but also from the cache of their neighboring nodes. This aims at maximizing the amount of data that can be served from the cache so that the server delays can reduced which in turn decreases the response time for the client. In many applications of MANET like automated highways and factories, smart homes and appliances, smart class rooms, mobile nodes share common interest. So sharing cache contents between mobile nodes offers significant benefits. Cache replacement algorithm greatly improves the effectiveness of the cache by selecting suitable subset of data for caching. The available cache replacement mechanisms for ad hoc network can be categorized in to coordinated and uncoordinated depending on how replacement decision is made. In uncoordinated scheme the replacement decision is made by individual nodes. In order to cache the incoming data when the cache is full, replacement algorithm chooses the data items to be removed by making use of the local parameters in each node.

3.3. 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 (λ fraction) and a unique segment (see Fig. 3).

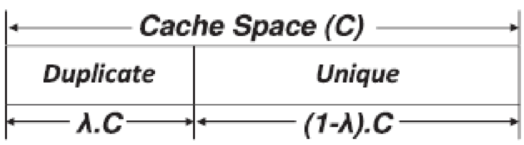

Fig. 3. Cache partitioning in split cache policy.

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 λ in Fig. 3 ($0 \leq \lambda \leq 1$) 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.

IV. AN EFFICIENT COORDINATED COOPERATIVE CACHE REPLACEMENT ALGORITHMS DISTRIBUTED WIRELESS NETWORKS

4.1 Coordinated Cache replacement Policies

Coordinated cache replacement strategy for cooperative caching schemes in mobile environments should ideally consider cache admission control policy. Cache admission control decides whether the incoming data is cacheable or not. Substantial amount of cache space can be saved by proper admission control, which can be utilized to store more appropriate data, thereby reducing the number of evictions. If a node doesn't cache the data that adjacent nodes have it can cache more distinct data items which increase the data availability. There is coordination between the neighboring nodes for the proper placing of data. Another feature of coordinated replacement is that the evicted data may be stored in neighboring nodes which have free space. Some of the replacement policies which make use of coordinated cache replacement are given below.

We are having different policies for cache replacement those are...

- **TDS**
- **LUV Mi**
- **ECORP**
- **COUNT VECTOR**

TDS

The cache replacement is based on two parameters distance (D) which is measured as the number of hops and access frequency. As the network is mobile the value of distance (D) may become obsolete. So the value is chosen based on the time at which it is last updated. The T value is obtained by the formula 1/tcur- tupdate. Distance is updated by looking at the value of T. Based on how the distance and time is selected three different schemes are proposed TDS_D,TDS_T and TDS_N.TDS_D considers distance as the replacement criteria. If two data items have the same distance least value of (D+T) is replaced. In TDS_T the replacement decision is made by selecting the data with lowest T value. In the third scheme product of distance and access frequency is considered. In these algorithms TDS_D has the lower success rate and TDS_T has the higher hit ratio.

LUV Mi

This replacement scheme has two parts replacement and migration. The replacement decision is based on a utility value formed by combining the parameters access probability, distance, size and coherency. In the migration part the replaced data is stored in the neighboring nodes which have sufficient space. For migration the data with highest utility value is given preference. Here even though the replacement decision is made locally migration is a coordinated operation. In order to save the cache space the data item is cached based on the location of the data source. If it is from the same cluster the data is not cached. The limitation of this scheme is that no checking is done whether the data is already present in the migrating node.

ECORP

Energy efficient cooperative cache replacement problem (ECORP) is an energy efficient cache replacement policy used in ad hoc networks. Here the replacement decision is done based on the energy utilization for each data access. For this, they considered the energy for in zone communication, energy for sending the object, energy for receiving and energy cost for forwarding the object. Based on this they proposed a dynamic ECORP DP and ECOPR _greedy algorithms to replace data. The neighboring nodes will not cache the same data item in its local cache which reduces the redundancy and increases hit ratio.

Count Vector

In this scheme, each data item maintains a count which gives the number of nodes having the same data. Whenever the cache is full, data item with maximum count is removed first as this will be available in the neighboring nodes. Whenever a data item is removed from the cache the access count will be decremented by one. Initially when the data is brought in to cache the count is set to zero. Table1 shows a comparison of different cache replacement policies.

5. DISCUSSION ABOUT ALL POLICIES

Most of the replacement algorithms used in ad hoc networks is LRU based which uses the property of temporal locality.

This is favorable for MANET which is formed for a short period of time with small memory capacity. Frequency based algorithms will be beneficial for long term accesses. It is better if the function based policies can adapt to different workload condition. In these schemes if we are using too many parameters for finding the value function, which are not easily available the performance can be degraded. Most of the replacement algorithms mentioned above uses cache hit ratio as the performance metric. In wireless network the cost to download data item from the server may vary. So in some cases this may not be the best performance metric. Schemes which improve cache hit ratio and reduce access latency should be devised. In cooperative caching coordinated cache replacement is more effective than local replacement since the replacement decision is made by considering the information available in the neighboring nodes. The area of cache replacement in cooperative caching has not received much attention. Lot of work needs to be done in this area to find better replacement policies.

6. CONCLUSION AND ONGOING WORK

The objective of this work was to develop a cooperative caching strategy for provisioning cost minimization in Social Wireless Networks. The key contribution is to demonstrate that the best cooperative caching for provisioning cost reduction in networks with homogeneous content demands requires an optimal split between object duplication and uniqueness. Such a split replacement policy was proposed and evaluated using ns2 simulation and on an experimental testbed of seven android mobile phones. Furthermore, we experimentally (using simulation) and analytically evaluated the algorithm's performance in the presence of user selfishness. It was shown that selfishness can increase user rebate

Table 1 : Comparison of cache replacement policies for cooperative caching

Algorithm	Parameters Considered	Eviction	Performance measure	Advantage	Disadvantage
TDS	Distance and access frequency	Low access rate and lowest distance	Success rate, cache hit ratio.	Value of distance is updated.	Doesn't consider recency of data item.
LUV Mi	Access probability, size, coherence and distance.	Low access probability, bigger size, low consistency, lowest distance	Byte hit ratio, average query latency, message overhead	Evicted data is stored in adjacent nodes	No checking is done before storing data.
ECORP	Energy for in zone communication, sending object, receiving object	Lowest energy value	Cache hit ratio, average access delay	Energy is taken as the important parameter	Computing energy for each task is not easy
Count Vector	Access count	Maximum access count	Average access time	Coordinated simple to implemen	Data redundancy is high.

only when the number of selfish nodes in an SWNET is less than a critical number. 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. Removal of the no-collusion assumption for user selfishness is also being worked on.

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.

[14] C. Chow, H. Leong, and A. Chan, “GroCoca: Group-Based Peerto- Peer Cooperative Caching in Mobile Environment,” IEEE J. Selected Areas in Comm., vol. 25, no. 1, pp. 179-191, Jan. 2007.

[15] F. Sailhan and V. Issarny, “Cooperative Caching in Ad Hoc Networks,” Proc. Fourth Int’l Conf. Mobile Data Management, pp. 13- 28, 2003.

[16] C. Chow, H. Leong, and A. Chan, “Peer-to-Peer Cooperative Caching in Mobile Environments,” Proc. 24th Int’l Conf. Distributed Computing Systems Workshops, pp. 528-533, 2004.

[17] A. Schrijver, Theory of Linear and Integer Programming. Wiley- Interscience, 1986.

[18] H.K. Kuhn, “The Hungarian Method for the Assignment Problem,” Naval Research Logistics, vol 52, no. 1, pp. 7-21, 2005.

[19] N. Laoutaris et al., “Mistreatment in Distributed Caching: Causes and Implications,” Proc. IEEE INFOCOM, 2006.

[20] B. Chun et al., “Selfish Caching in Distributed Systems: A Game- Theoretic Analysis,” Proc. 23th ACM Symp. Principles of Distributed Computing, 2004.

[21] M. Goemans, L. Li, and M. Thottan, “Market Sharing Games Applied to Content Distribution in Ad Hoc Networks,” IEEE J. Selected Areas in Comm., vol. 24, no. 5, pp. 1020-1033, May 2006.

