A New Distributed Environment for Cooperative Caching in Social Wireless

Kosuru Shalini #1, **B.Bangar Naidu** *2

#1M.Tech Scholar, *2 Assistant Professor
Department of Computer Science & Engineering,
Vizag Institute of Technology Engineering College,
Dakamarri, Visakhapatnam Dist, AP, India.

Abstract

During the last decade, Wireless Networks have emerged up to a great extend and matured at such point that they currently support several applications such as environment control, intelligent buildings, and target tracking in battlefields. This paper mainly introduces a new cooperative caching policies for minimizing electronic content provisioning cost in Social Wireless Networks (SWNT). SWNTs are mainly formed by collaborative collection of various mobile devices, such as data enabled phones, electronic book readers etc., sharing mostly all the common interests in electronic content, and physically gathering together in public places. The new method of object caching in SWNTs are shown to be able to reduce the content provisioning cost which depends heavily on the service and pricing dependences among various stakeholders including content providers (CP), network service providers, and End Consumers (EC). This paper mainly deals practically with 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. Finally we construct analytical and simulation models for analyzing the proposed caching strategies in the presence of selfish users that deviate from network-wide cost-optimal policies.

Keywords

Social wireless networks, cooperative caching, content provisioning, ad hoc networks

1. Introduction

Wireless Networks [1, 2] have rapidly increased the attention of many users during the last decade due to the advances in low-power hardware and the development of appropriate software. A wireless network mainly consists of wirelessly interconnected devices (each being able to compute, control and communicate with each other) that can also interact with their environment by controlling and sensing ‘‘physical’’ parameters. These networks have covered with a huge number of applications, such as disaster relief, environment control and biodiversity mapping, machine surveillance, target tracking in battlefields, and so on. There was a great demand with the emergence of data enabled mobile devices and wireless-enabled data applications in today’s mobile ecosystem which have mainly fostered new content dissemination models. A list of such mobile devices includes Apple’s iPhone, Google’s Android, Amazon’s Kindle, and electronic book readers from several other vendors. The level of each and every mobile application 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.

Further with the conventional download model which is proposed by mobile vendors, a user downloads contents directly from a Content Provider’s (CP) server over a Communication Service Provider’s (CSP) network. The downloading content which is done through CSP’s network mainly involves a small action like cost which must be paid either directly by end users or by the content provider who provides data for the end users. To research this work, we adopt Amazon Kindle electronic book delivery business model in which the Content Provider like (Amazon), pays to Sprint, who is the CSP, for the cost of network usage due to downloaded e-books by Kindle users.

For the users who carry mobile devices physically for places such as University campus, work premises, Shopping Mall, Airport and other public places, Social Wireless Networks (SWNTs) can be formed as a physical setting using ad hoc wireless connections between the devices. Due to the existence of such SWNTs, an alternative approach to content access by a device would be to first initially search the local SWNT for the requested content before downloading it from the CP's server. The expected content provisioning cost of such an approach can be gradually lowered, since the download cost to the CSP would be avoided when the content is found within the local SWNT. This mechanism is termed as **cooperative caching**.

Due to their limited storage facility for the mobile hand held devices, they are not expected to store all downloaded content for a very long time. This means after downloading and using a purchased electronic content, a device may remove it from the storage location. For example in Amazon Kindle clients (iPhone, iPad, etc.) an archive mode is available using which a user simply removes a book after reading it, although it remains archived as a purchased item in Amazon's cloud server. Under the above pricing model and data storage model a key question which is raised for cooperative caching is: *How to store contents in nodes such that the average content provisioning cost in the network is minimized*?

1.1 Optimal Solution

For some contents with different varying levels of popularity, a new greedy approach for each node would be used to store as many distinctly popular contents as long as storage space allows. This new recent approach amounts to noncooperation and can give rise to heavy network-wide content duplications. In the other extreme case, which is fully cooperative in nature, each and every node would try to maximize the total number of unique contents stored within the SWNT by avoiding duplications between them. In this paper, we mainly target to 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 [3]. This is referred to as optimal object placement policy in the rest of this paper. The proposed new cooperative caching algorithms strive to attain this optimal object placement with the target of minimizing the network-wide content provisioning cost.

1.2 User Selfishness

A selfish user among a several user is one that deviates from the network-wide optimal policy in order to earn more and more rebates. Any such deviation from the optimal policy is expected to incur higher network-wide provisioning cost. In this current research work, we mainly try to analyze the impacts of such selfish behavior on object provisioning cost and the earned rebate within the context of an SWNT. It is clearly shown that beyond a threshold selfish node population, the maximum amount of per-node rebate for the selfish users with in all users is lower than that for the nonselfish users. In other words, we can express that, when the selfish node population is beyond a certain critical point, selfish behavior ceases to produce more benefit from a rebate standpoint.

1.3 Our Contributions

Upon all the assumptions what we have discussed till now, we try to give following contributions for our proposed model. First, based on a practical service and pricing case what we observed, a new stochastic model for the content provider's cost computation is developed. Second, a cooperative caching strategy and a Split Cache, is proposed, numerically it was 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 individual 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.

2. Related Work

In this section we try to concentrate mainly on the network services and pricing model that was used for our proposed model.

2.1 Network Model

In this Network model, which is shown clearly in Fig. 1 illustrates an example SWNT within a University campus. End Consumers carrying mobile devices form SWNT 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 user mobile device can download an object (i.e., content) from the CP’s server using the CSP’s cellular network, or from its local SWNT partition.

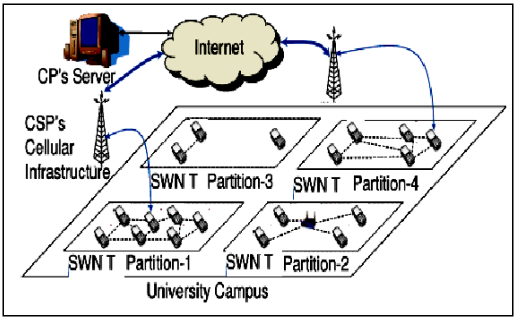


Fig. 1. Content access from an SWNT in a University Campus.

SWNTs is considered of two types. The first one involves mainly on stationary [5] SWNT partitions. Which means , after a partition is formed, it is maintained for sufficiently long so that the cooperative object caches can be formed and reach steady states. We also try to investigate a second type to explore as to what happens when the stationary assumption is relaxed. To investigate this effect, caching technique is applied to SWNTs formed using human interaction traces obtained from a set of real SWNT nodes [6].

2.2 Search Model

Once a mobile device originates an object request, it first searches in its local cache within the device. If the local search fails to find the object request, then it searches the object within its SWNT partition using limited broadcast message. If the search in partition also fails, then the object request is finally downloaded from the Content Provider’s server using the CSP’s 3G/4G cellular network. In this paper, we have modeled objects such as electronic books, music, etc., which are time non -varying, and therefore cache consistency is not a critical issue for such a cases. We first assume initially that all objects have the same size of data and each node is able to store up to “C” different objects in its cache. We also assume that all objects are popularity-tagged by the CP’s server [7]. The popularity-tag of an object in device indicates its global popularity; it also indicates the probability that an arbitrary request in the network is generated for this specific object.

2.3 Pricing Model

We use a pricing model in assumption similar to the Amazon Kindle business model in which the Content Provider (e.g., Amazon) pays a download cost C_d to the CSP when an End-Consumer downloads an object from the CP’s server through the CSP’s cellular network not from the local cache. Also, the pricing model is applied whenever an EC provides a locally cached object to another EC within its local SWNT partition, the provider EC is paid a rebate C_r by the CP. Optionally, this paid 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 clearly 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 C_r

corresponds to the rebate given out to an EC when the object is found within the SWNT (e.g., node A receives rebate C_r after it provides a content to node B over the SWNT). For a given $C_r = C_d$ ratio, the paper aims to develop optimal object placement policies that can minimize the network-wide content provisioning cost.

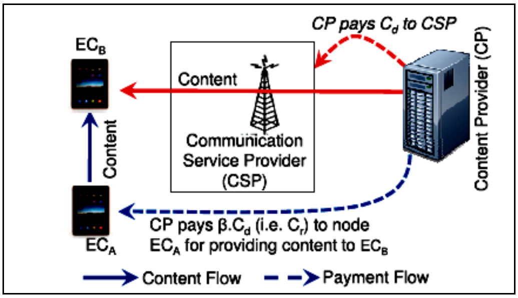


Fig. 2. Content and cost flow model.

Operationally, the parameters like C_d and C_r are set by a CP and CSP [4] based on their operating cost and revenue models. The end-consumers do not have any control on those parameters.

3. Cost under Homogeneous Request Model

In this section, we mainly compute the average object provisioning cost under a homogenous request model where each and every device is having same functionalities. Let us assume for observation P_L be the probability of finding a requested object from the local cache (i.e., local hit rate) and, P_V be the probability that a requested object can be found in the local SWNT partition (i.e., remote hit rate) after its local search fails, and P_M 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 P_M in terms of P_V and P_L as

In this section we mainly discuss about following four ways of giving centralized solution for our energy optimization framework. The following are the various solutions for our proposed framework.

1. Energy Optimization Framework
2. Base Case Problem
3. Static Tree Construction
4. Node Insertion

Now let us discuss about each concept in detail.

3.1 Energy Optimization Framework

The Optimal Mobile Relay Configuration (OMRC) problem is one of the major challenging problems in our proposed frame work because of the dependence of the obtained solution on various multiple factors such as on the routing tree topology and the amount of data transferred through each link. For example, when we try to transfer little data, the optimal configuration is required to use only some relay nodes at their original positions instead of using many relay nodes. As the amount of data to be transferred increases, following three changes mainly occur: firstly the topology may change by adding new relay nodes or removing any relay nodes, the topology may also change by changing which edges are used, and the relay nodes may move closer together. In many of the cases, we may have restrictions such as no mobility for certain relay nodes or we must use a predefined fixed routing tree. These constraints may also affect the optimal configuration.

We now finally present a centralized approach to solve OMRC that breaks the problem into three distinct individual steps:

1. initial tree construction,
2. node insertions, and
3. tree optimization.

For each individual step, we present an algorithm to solve the corresponding sub problem. Our algorithm for initial tree construction is optimal for the static environment where nodes cannot move. Our new greedy heuristic principles for improving the routing tree topology by adding a node exploits the mobility of the newly added nodes. Finally our tree optimization algorithm improves the routing tree by relocating its nodes without changing its existing topology. Our approach is not guaranteed to produce an optimal configuration because we do not necessarily find the optimal topology, but our simulation results show that it performs well.

3.2 Base Case Problem

According to our proposed energy models, the total transmission and movement energy cost incurred by the mobile relay node s_i is

$$\boxed{c_i(U) = k\|u_i - o_i\| + am + b\|u_{i+1} - u_i\|^2 m}$$

We also define

$$\boxed{C_i(U) = c_i(U) + am + b\|u_i - u_{i-1}\|^2 m}$$

This corresponds to the transmission cost of node s_{i-1} plus the total cost of node s_i, which is the total cost of the final configuration in this example.

3.3 Static Tree Construction

Different applications may have different individual constraints on the routing tree. When only optimizing energy consumption problem is declared, a shortest path strategy yields a very optimal routing tree given no mobility of nodes. However, when we compared the same with various applications, we do not have the freedom of selecting the routes of our own. Instead, initially only before the routing starts, they are predetermined according to some other factors (such as delay, capacity, etc). In Some other rare cases, we may be able to update the given routes provided we keep the main structure of the tree in our assumption. Depending on the route constraints decided initially by the application, we start our solution at different phases of the algorithm.

We initially construct the tree for our starting configuration using a shortest path strategy. We first define a weight function **w** specific to our communication energy model. We observe that using this weight function, the optimal tree in a static environment coincides with the shortest path tree rooted at the sink. So we apply Dijkstra's shortest path algorithm starting at the sink to all the source nodes to obtain our initial topology.

```
function LOCALPOS($o_i, u_i, u_{i-1}, u_{i+1}$)
    ▷ Consider case $s_i$ moves right
    $valid \leftarrow FALSE;$
    $x_i \leftarrow \frac{1}{2}(x_{i-1} + x_{i+1}) - Y_i;$
    if $x_i > p_i$ then
        $valid \leftarrow TRUE;$
    else
        ▷ Consider case $s_i$ moves left
        $x_i \leftarrow \frac{1}{2}(x_{i-1} + x_{i+1}) + Y_i$
        if $x_i < p_i$ then
            $valid \leftarrow TRUE;$
        end if
    end if
    ▷ Record if new position is different from previous one
    if $valid$ then
        $y_i \leftarrow \frac{(x_{i-1}+x_{i+1}-2p_i)}{(y_{i-1}+y_{i+1}-2q_i)}(x_i - p_i) + q_i;$

        $u'_i = (x_i, y_i);$

        if $\left\|u'_i - u_i\right\| >$ threshold then
            return $(u'_i, \text{TRUE});$
        end if
    end if
    ▷ not beneficial to move, stay at original position
    return $(o_i, \text{FALSE});$
end function
```

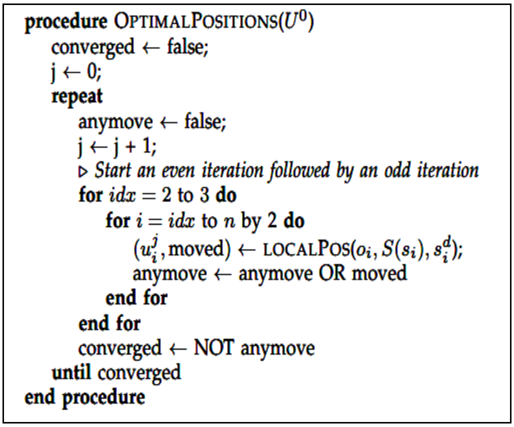
Fig. 3. Algorithm to compute the optimal position of a relay node that receives data from a single node and transmits the data to a single node.

3.4 Node Insertion

We improve the routing tree efficiency by greedily adding new nodes to the existing routing tree exploiting the mobility of the inserted nodes. For each node say s_{out} that is not previously available in the tree and for each and every tree edge s_i s_j, we compute the reduction (or increase) in the total cost along with the optimal position of s_{out} if s_{out} joins the tree such that data is routed from s_i to s_{out} to s_j instead of directly from s_i to s_j. We continuously insert the outside nodes with the highest reduction value modifying the topology to include the selected node at its optimal position, though the node will not actually move until the completion of the tree optimization phase. After each and every node insertion takes place, we compute the reduction of tree in total cost and optimal position for each and every remaining outside node for the two newly added .Finally at the end of this procedure, we find that the topology of the routing tree is fixed in its position and its mobile nodes can able to start the tree optimization phase to relocate to their optimal positions.

4. Tree Optimization

In this section, we initially consider the sub problem of finding the optimal positions of relay nodes for a routing tree given assuming that the topology is fixed. We assume the topology is a directed tree in which the leaves are nothing but sources and the root nothing but as the sink. We also assume that separate messages cannot be compressed or merged, that is, suppose if two distinct messages of lengths m_1 and m_2 use the same sink node (s_i, s_j) on the path from a source node or leaves to a sink node , the total number of bits that must traverse link (s_i, s_j) is $m_1 + m_2$



```
procedure OPTIMALPOSITIONS($U^0$)
    converged ← false;
    j ← 0;
    repeat
        anymove ← false;
        j ← j + 1;
        ▷ Start an even iteration followed by an odd iteration
        for $idx = 2$ to 3 do
            for $i = idx$ to $n$ by 2 do
                ($u_i^j$, moved) ← LOCALPOS($o_i$, $S(s_i)$, $s_i^d$);
                anymove ← anymove OR moved
            end for
        end for
        converged ← NOT anymove
    until converged
end procedure
```

Fig. 4. Centralized algorithm to compute the optimal positions in a given tree

Our above algorithm in figure 4, starts by an odd/even labeling step followed by a weighting step. To obtain very best consistent labels for nodes, we start the labeling process from the root using a breadth first traversal of the tree. The root gets labeled as even. Each of its children gets labeled as odd. Each subsequent child is then given the opposite label of its parent. We define m_i , the weight of a node s_i , to be the sum of message lengths over all paths passing through s_i. This computation starts from the sources or leaves of our routing tree. Initially, we know $m_i = M_i$, for each source leaf node s_i. For each intermediate node s_i , we compute its weight as the sum of the weights of its children.

5. Efficiency and Optimality

We first consider efficiency of our proposed algorithms. Our initial tree construction algorithm is essentially a single source shortest path algorithm. Using Dijkstra's algorithm, the time complexity is $O(n^2)$, where n is the number of nodes. According to our second algorithm, it needs to compute the reduction in cost for each pair of node and tree edge, so the time complexity is $O(n^2)$.Finally our tree optimization algorithm runs until the change in position for each node falls below a predefined threshold. The value of this threshold represents a tradeoff between precision and cost. As the threshold decreases, more iterations are needed for convergence.

With respect to optimality problem that is present in our tree construction, our resulting configuration is not necessarily optimal because we do not necessarily find the optimal topology in our tree. However, of all the three, two of our algorithms, the initial tree construction algorithm and the tree optimization algorithm, are very nearer to optimal for their respective sub problems. That is, our initial tree construction algorithm is optimal in a static environment where nodes cannot move so that only the original positions of the nodes are considered; this clearly tells that once nodes are fixed in initial state can't be move as they are static in nature. Likewise, for our tree optimization algorithm, we prove that the final configuration where no node can move by itself to improve the overall cost (within the threshold bound) is globally optimal; that is, no simultaneous relocation of multiple nodes can improve the overall cost. We present the proof of optimality in the appendix.

6. Conclusion

In this paper, we proposed a very new holistic approach to minimize the total energy consumed by both mobility of relay nodes and wireless transmissions. Most of our existing work on these techniques ignored the energy consumed by moving mobile relays. Initially, we start with the optimal initial routing tree in a static environment where no nodes can move from its positions as this is fixed in their positions due to static nature. However, our approach can work with less optimal initial configurations including one generated using only local information such as greedy geographic routing. Our proposed new approach improves the initial configuration of our nodes using two iterative schemes. The first scheme inserts new nodes into the tree. The second scheme computes the optimal positions of relay nodes in the tree given a fixed topology. This algorithm is appropriate mainly for a variety of data-intensive wireless sensor networks. It allows some nodes to move while others do not because any local improvement for a given mobile relay is a global improvement. Our simulation results tell that our proposed method is substantially reduces the energy consumption by up to 45%.

7. References

[1] Akyildiz, I., Su, W., Sankarasubramaniam, Y., & Cayirci, E. (2002). A survey on sensor networks. IEEE Communications magazine, 40(8), 102–114.

[2] Karl, H., & Willig, A. (2006). Protocols and architectures for wireless sensor networks. New York: Wiley.

[3] M. Taghizadeh, A. Plummer, A. Aqel, and S. Biswas, "Optimal Cooperative Caching in Social Wireless Networks," Proc. IEEE Global Telecomm. Conf. (GlobeCom), 2010.

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

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

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

[7] 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.

8. About the Authors

Kosuru Shalini is a student of the Department of Computer Science & Engineering of Vizag Institute of Technology, Visakhapatnam. Presently she is pursuing her M.Tech from this college. She completed her B.Tech (IT) at Nova college of Engg. and Technology for women, Vijayawada. Her area of interest includes Mobile Computing and Networks.

B.Bangar Naidu received his M.Tech degree in Computer Science & Technology (Computer Networks) from Andhra University A.P, India and B.Tech degree in Computer Science & Information technology from Saint Theressa Institute of Engineering and technology, A.P, India. Currently he is working as an Assistant Professor in Vizag Institute of Technology in the department of Computer Science & Engineering. He has 7 years of teaching experience in various engineering colleges. His research interests include Information Security, Computer Organization, Neural Networks and Network Security.