ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Special Issue 9, May 2016

Data Efficiency: A Cooperative Caching Approach

Shwetha Bhat M [1], Devidas Bhat [2]

PG Student, Dept. of Information Science and Engineering, NMAMIT, Udupi, Karnataka, India [1]

Assistant Professor, Dept. of Information Science and Engineering, NMAMIT, Udupi, Karnataka, India [2]

ABSTRACT: Data caching can significantly improve the efficiency of information access in a network by reducing the access latency and bandwidth usage. Cooperative caching is a technique used in networks to improve the efficiency of data access by reducing the access latency and bandwidth usage. Most of the current research efforts in networks are in data forwarding, but only limited work has been done on efficient data access. Here we propose an approach to support cooperative caching, which provides the sharing and coordination of cached data among several nodes and minimize data access delay. Our proposed design is to intentionally cache data at a set of Network central location (NCL). NCL can be easily accessed by other nodes in the network.

KEYWORDS: Cooperative Caching, Data Efficiency

I. INTRODUCTION

In Networks, due to the frequent changes in topology, only intermittent connectivity exists between the users. It is very difficult to maintain end to end communication path which makes it necessary to use store and forward methods for data transfer. Nodes in the network may not have large storage capacity and transfer data. The problem is how to determine the appropriate relay selection strategy. There is limited research on providing efficient data access to mobile users, despite the importance of data accessibility in many applications. The users can only request the data whenever needed and they do not know data locations in advance. The destination of data is, hence, unknown when data are generated.

A general technique used to improve the performance of data access is caching i.e., to cache data at appropriate network central locations(NCLs) based on query history, so that queries in the future can be responded with less delay. Although, to allow sharing and coordination among multiple caching nodes, it is difficult to be realized in networks due to lack of continuous network connectivity. First, the opportunistic network connectivity complicates the data transmission delay, and furthermore makes it difficult to determine appropriate caching locations for reducing data access delay. This difficulty is raised by the incomplete information at individual nodes about query history. Second, due to the uncertainty of data transmission, multiple data copies are to be cached at different locations to ensure data accessibility. The difficulty in coordinating multiple caching nodes makes it hard to optimize the trade-off between data accessibility and caching overhead.

In this paper, we propose a scheme to address the aforementioned challenges and to efficiently support cooperative caching in networks. Our basic idea is to intentionally cache data at a set of Network Central Locations (NCLs), each of which corresponds to nodes being easily accessed by other nodes in the network. Each NCL is represented by a central node, which has high popularity in the network and is prioritized for caching data. Due to the limited caching buffer of central nodes, multiple nodes near a central node may be involved for caching, and we ensure that popular data is always cached nearer to the central nodes via dynamic cache replacement based on query history.

Our detailed contributions are listed as follows: We develop an approach for NCL selection in networks based on a probabilistic selection metric. The selected NCLs achieve chances for response to user queries with low overhead in network storage and transmission. A data access scheme to probabilistically coordinate multiple caching nodes for responding to user queries.

ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Special Issue 9, May 2016

II. RELATED WORK

Data access in networks, on the other hand, can be provided in various ways [2]. Data can be disseminated to appropriate users based on their interest profiles[3]. Publish/ subscribe systems were used for data dissemination, where social community structures are usually exploited to determine broker nodes. In other schemes [4], [5]without brokers, data items are grouped into pre-defined channels, and are disseminated based on users' subscriptions to these channels.
Caching is another way to provide data access. Cooperative caching in networks was studied in [6], in which each node caches pass-by data based on data popularity, so that queries in the future can be responded with less delay. Caching locations are selected incidentally among all the network nodes. Some research efforts [7], [8] have been made for caching in networks, but they only improve data accessibility from infrastructure network such as Wi-Fi Access Points (APs) [8] or Internet [7]. Peer-to-peer data sharing and access among mobile users are generally neglected.
In [9], it is assumed that all the nodes contact each other with the same rate. In [10], users are artificially partitioned into several classes such that users in the same class are identical. In [11], data is intentionally cached at appropriate network locations with generic data and query models, but these caching locations are determined based on global network knowledge. Comparatively, in this paper we propose to support cooperative caching in a fully distributed manner in networks, with heterogeneous node contact patterns and behaviours.
A requester queries the network for data access, and the data source or caching nodes reply to the requester with data after having received the query. The key difference between with and without caching strategies in networks is illustrated in Figure 1. Note that each node has limited space for caching. Otherwise, data can be cached everywhere, and it is trivial to design different caching strategies. The existing design benefits from the assumption of existing end to-end paths among mobile nodes, and the path from a requester to the data source remains unchanged during data access in most cases. Such assumption enables any intermediate node on the path to cache the pass-by data. For example, in Figure 1(a), C forwards all the three queries to data sources A and B, and also forwards data $d1$ and $d2$ to the requesters. In case of limited cache space, C caches the more popular data $d1$ based on query history, and similarly data $d2$ is cached at node K. In general, any node could cache the pass-by data incidentally.
The proposed system, data is forwarded via opportunistic contacts, the query and replied data may take different routes, and it is difficult for nodes to collect the information about query history and make caching decision. For example, in Figure 1(b), after having forwarded query $q2$ to A, node C loses its connection to G, and cannot cache data $d1$ replied to requester E. Node$H$ which forwards the replied data to E does not cache the pass-by data $d1$ either, because it did not record query $q2$ and considers $d1$ less popular. In this case, $d1$ will be cached at node G, and hence needs longer time to be replied to the requester.

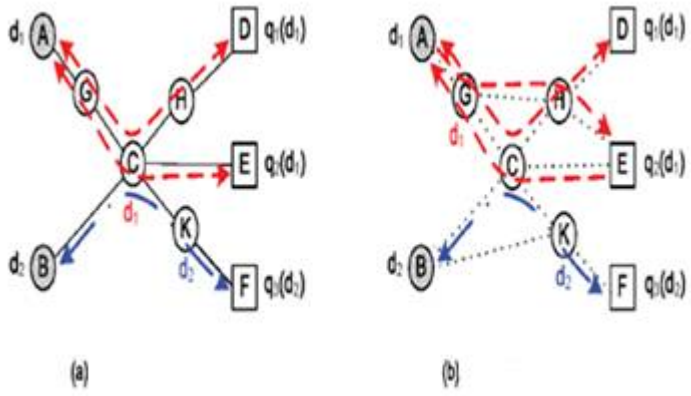

Figure 1: Existing And Proposed System

Our basic solution to improve caching performance in networks is to restrain the scope of nodes being involved for caching. Instead of being incidentally cached "anywhere", data is intentionally cached only at specific nodes. These nodes are carefully selected to ensure data accessibility, and constraining the scope of caching locations reduces the complexity of maintaining query history and making caching decision.

ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Special Issue 9, May 2016

III. PROPOSED SYSTEM

The basic idea is to intentionally cache data only at a spe-cific set of NCLs, which can be easily accessed by other nodes in the network. Queries are forwarded to NCLs for information access. The big picture of our proposed scheme is illustrated in Figure. 2. Each NCL is represented by a central node, which corresponds to a star. The push and pull caching strategies conjoin at the NCLs. The data source S actively pushes its generated data toward the NCLs, and the central nodes C1 and C2 of NCLs are pri-oritized for caching data. If the buffer of a central node C1 is occupied, data are cached at another node A near C1. Multiple nodes at a NCL may be involved for caching, and a NCL, hence, corresponds to a connected sub graph of the network contact graph G, as the dashed circles illustrated in Figure. 2. Note that NCLs may be overlapping with each other, and a node being involved for caching may belong to multiple NCLs simultaneously. A requester R pulls data by querying NCLs, and data copies from multiple NCLs are returned to ensure prompt data access. Particularly, some NCL such as C2 may be too far from R to receive the query on time, and does not act in response with data. In this case, data accessibility is determined by both node contact frequency and data lifetime.

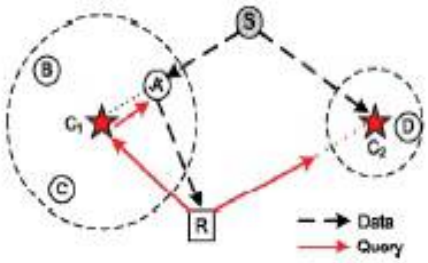

Figure 2: Intentional Caching

A. Multihop Opportunistic Connection on Network :

The data transmission delay between two nodes A and B, indicated by the random variable Y, is measured by the weight of the shortest opportunistic path between the two nodes. In practice, mobile nodes maintain the information about shortest opportunistic paths between each other in a distance-vector manner when they come into contact.

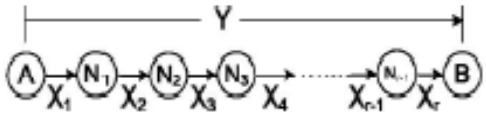

Figure 3: Multihop Opportunistic Connection

B. Caching scheme:

In this section, we present cooperative caching scheme. The basic idea is to intentionally cache data at a set of NCLs, which can be promptly accessed by other nodes. This scheme consists of the following three components:

1. When a data source generates data, it pushes data to central nodes of NCLs, which are prioritized to cache data. One copy of data is cached at each NCL. If the caching buffer of a central node is full, one more node near the central node will cache the data. Such decision are by design made based on buffer conditions of nodes involved in the pushing process.
2. A requester multicasts a query to central nodes of NCLs to pull data, and a central node forwards the query to the caching nodes. Multiple data copies are returned to the requester, and optimize the tradeoff between data accessibility and transmission over-head by controlling the number of returned data copies.
3. Utility-based cache replacement is conducted whenever two caching nodes contact and ensures that popular data are cached nearer to central nodes.

ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Special Issue 9, May 2016

C. Pushing Scheme

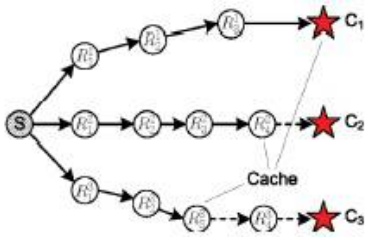

Figure 4: Pushing Scheme

The determination of caching location is illustrated in Figure. 4, where the solid lines indicate opportunistic contacts used to forward data, and the dashed lines indicate data forwarding stopped by node buffer constraint. Central node C1 is able to cache data, but data copies to C2 and C3 are stopped and cached at relays R24 and R33 , respectively, because neither C2 nor R34 Have enough buffers to cache data.

D. Pulling Scheme

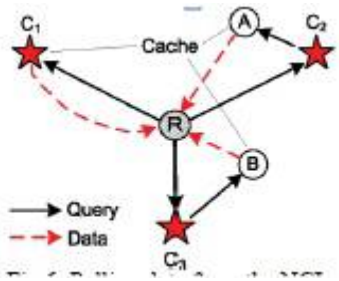

Figure 5: Pulling Scheme

While the central node C1 is able to return the cached data to R immediately, the caching nodes A and B simply reply to R after they receive the query from central nodes C2 and C3, respectively is illustrated in fig 5. The query broadcast finishes when query expires.
Multiple data copies are replied to the requester from NCLs to ensure that the requester receives Data before query expires. However, only the first data copy received by the requester is useful, and all the others are essentially useless and waste network resources. The major challenge for solving this problem arises from the intermittent network connectivity in networks.

IV. SYSTEM MODEL

When the data is requested by the user then the source node sends data by initially processing the data and then selects the appropriate NCL as central node in the network to cache the data in it by NCL selection Mode. The NCL can cache the data to be delivered faster when compared to the conventional memory storage. If the current NCL's buffer is completely full, then it is replaced with nearby high prioritized node as NCL by cache replacement technique. The cache replacement is based on popularity of the data in the network which can be identified by most user requested and received data.

ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Special Issue 9, May 2016

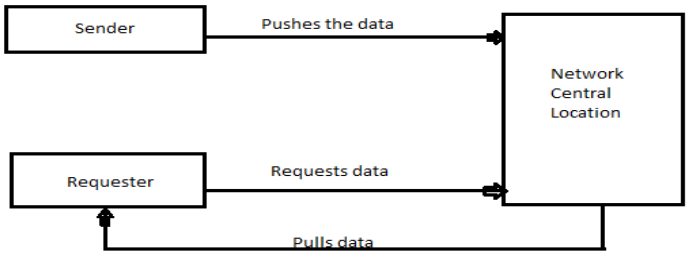

Figure 6: System Model

V. IMPLEMENTATION

An application developed which uses the concept of cooperative caching where we have two users: Sender and Requester. A node, called as NCL is used to cache the data which is implemented as a database. A front end developed using Visual Studio C# 2010, where two users can communicate to send messages, files of any form.
A sender and requester have to authenticate by login with valid username and password. Once it is valid the user can further communicate. The sender composes a new message with an attachment and forwards. This particular generated data is pushed to the NCL. At the requester, it checks for the new message. Further requester queries the NCL for the attachment. A requester pulls the data, and copies of data from NCL are returned to ensure prompt access.
Initially sender sends the data to NCL and then to the requester. Further, if any requester requests for the same file it will be fetched from NCL. The time taken initially will be more compared to time taken for remaining requests. The implementation results are shown in figure 7., where the data transmission rate is reduced when being compared with existing schemes.

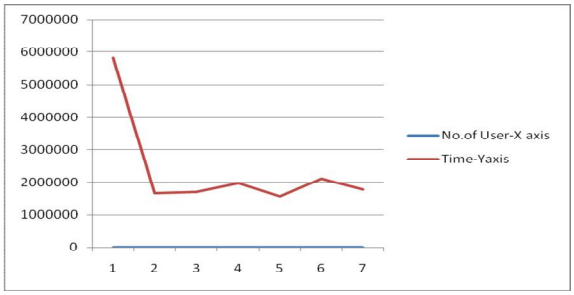

Figure 7: Result Analysis

VI. CONCLUSION

In this paper, we propose a novel scheme to support cooperative caching in networks. Our basic idea is to intentionally cache data at a set of NCLs which can be easily accessed by other nodes. We ensure appropriate NCL selection based on a probabilistic metric; our approach coordinates caching nodes to optimize the trade off between data accessibility and caching overhead.

REFERENCES

[1] Wei Gao, Guohong Cao, Arun Iyengar, Mudhakar Srivatsa. “Cooperative Caching for Efficient Data Access in Disruption Tolerant Networks”, IEEE Transactions on Mobile Computing ,Volume 13,Issue-3, 2014
[2] I. Psaras, L. Wood, and R. Tafazolli. Delay-/ disruption-tolerant networking: State of the art and future challenges. *University of Surrey, Technical Report*, 2010.

ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Special Issue 9, May 2016

[3] W. Gao and G. Cao. User-centric data dissemination in disruption tolerant networks. In *Proceedings of INFOCOM*, 2011.
[4] V. Lenders, G. Karlsson, and M. May. Wireless Ad hoc Podcasting. In *Proceedings of SECON*, pages 273–283, 2007.
[5] C. Boldrini, M. Conti, and A. Passarella. ContentPlace: socialaware data dissemination in opportunistic networks. In *Proceedings of MSWiM*, pages 203–210, 2008.
[6] L. Yin and G. Cao. Supporting Cooperative Caching in Ad Hoc Networks. *IEEE Trans. on Mobile Computing*, 5(1):77–89, 2006.
[7] A. Chaintreau, P. Hui, J. Crowcroft, C. Diot, R. Gass, and J. Scott. Impact of Human Mobility on Opportunistic Forwarding Algorithms. *IEEE Trans. on Mobile Computing*, 6(6):606–620, 2007.
[8] Y. Huang, Y. Gao, K. Nahrstedt, and W. He. Optimizing File Retrieval in Delay-Tolerant Content Distribution Community. In *Proceedings of ICDCS*, pages 308–316, 2009.
[9] Y. Huang, Y. Gao, K. Nahrstedt, and W. He. Optimizing File Retrieval in Delay-Tolerant Content Distribution Community. In *Proceedings of ICDCS*, pages 308–316, 2009.
[10] S. Ioannidis, L. Massoulie, and A. Chaintreau. Distributed caching over heterogeneous mobile networks. In *Proceedings of the ACM SIGMETRICS*, pages 311–322, 2010.
[11] W. Gao, G. Cao, A. Iyengar, and M. Srivatsa. Supporting cooperative caching in disruption tolerant networks. In *Proceedings of Int'l Conf. on Distributed Computing Systems (ICDCS)*, 2011.