Volume 4, Issue 10, October 2014 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Caching Strategies Based on Information Solidity Evaluation in Wireless Ad Hoc Networks

A. Praksah, K. B. K. S. Durga, P. Divya, B. Jayanth
Department of Computer Science and Engineering, St.Martin's Engineering College,
R.R Dist, TLG and India

Abstract:Data caching strategy for ad hoc networks whose nodes exchange information items in a peer-to-peer fashion. Data caching is a fully distributed scheme where each node, upon receiving requested information, determines the cache drop time of the information or which content to replace to make room for the newly arrived information. These decisions are made depending on the perceived "presence" of the content in the nodes proximity, whose estimation does not cause any additional overhead to the information sharing system. We devise a strategy where nodes, independent of each other, decide whether to cache some content and for how long. In the case of small-sized caches, we aim to design a content replacement strategy that allows nodes to successfully store newly received information while maintaining the good performance of the content distribution system. Under both conditions, each node takes decisions according to its perception of what nearby users may store in their caches and with the aim of differentiating its own cache content from the other nodes'. The result is the creation of content diversity within the nodes neighborhood so that a requesting user likely finds the desired information nearby. We simulate our caching algorithms in different ad hoc network scenarios and compare them with other caching schemes, showing that our solution succeeds in creating the desired content diversity, thus leading to a resource-efficient information access.

Key Terms:* *Data Caching, Self-Organizing, MANET, Integrated Cache-Routing, Localized Caching Policy, Distributed Caching Algorithm.

I. INTRODUCTION

Ad hoc networks are multi hop wireless networks of small computing devices with wireless interfaces. The computing devices could be conventional computers (for example, PDA, laptop, or PC) or backbone routing platforms or even embedded processors such as sensor nodes. The problem of optimal placement of caches to reduce overall cost of accessing data is motivated by the following two defining characteristics of ad hoc networks. First, the ad hoc networks are multi hop networks without a central base station. Thus, remote access of information typically occurs via multi hop routing, which can greatly benefit from caching to reduce access latency. Second, the network is generally resource constrained in terms of channel bandwidth or battery power in the nodes. Caching helps in reducing communication, this results in savings in bandwidth, as well as battery energy. The problem of cache placement is particularly challenging when each network node has a limited memory to cache data items.

In this paper, our focus is on developing efficient caching techniques in ad hoc networks with memory limitations. Research into data storage, access, and dissemination techniques in ad hoc networks is not new. In particular, these mechanisms have been investigated in connection with sensor networking peer-to-peer networks mesh networks world wide Web and even more general ad hoc networks. However, the presented approaches have so far been somewhat "ad hoc" and empirically based, without any strong analytical foundation. In contrast, the theory literature abounds in analytical studies into the optimality properties of caching and replica allocation problems. However, distributed implementations of these techniques and their performances in complex network settings have not been investigated. It is even unclear whether these techniques are amenable to efficient distributed implementations.

Our goal in this paper is to develop an approach that is both analytically tractable with a provable performance bound in a centralized setting and is also amenable to a natural distributed implementation. In our network model, there are multiple data items; each data item has a server, and a set of clients that wish to access the data item at a given frequency. Each node carefully chooses data items to cache in its limited memory to minimize the overall access cost. Essentially, in this article, we develop efficient strategies to select data items to cache at each node. In particular, we develop two algorithms—a centralized approximation algorithm, which delivers a 4-approximation (2-approximation for uniform size data items) solution, and a localized distributed algorithm, which is based on the approximation algorithm and can handle mobility of nodes and dynamic traffic conditions. Using simulations, we show that the distributed algorithm performs very close to the approximation algorithm. Finally, we show through extensive experiments on ns2 that our proposed distributed algorithm performs much better than a prior approach over a broad range of parameter values. Ours is the first work to present a distributed implementation based on an approximation algorithm for the general problem of cache placement of multiple data items under memory constraint.

Data caching strategy for ad hoc networks whose nodes exchange information items in a peer-to-peer fashion. Data caching is a fully distributed scheme where each node, upon receiving requested information, determines the cache drop time of the information or which content to replace to make room for the newly arrived information. These decisions are made depending on the perceived "presence" of the content in the nodes proximity, whose estimation does not cause any additional overhead to the information sharing system. We devise a strategy where nodes, independent of each other, decide whether to cache some content and for how long. In the case of small-sized caches, we aim to design a content replacement strategy that allows nodes to successfully store newly received information while maintaining the good performance of the content distribution system. Under both conditions, each node takes decisions according to its perception of what nearby users may store in their caches and with the aim of differentiating its own cache content from the other nodes'. The result is the creation of content diversity within the nodes neighborhood so that a requesting user likely finds the desired information nearby. We simulate our caching algorithms in different ad hoc network scenarios and compare them with other caching schemes, showing that our solution succeeds in creating the desired content diversity, thus leading to a resource-efficient information access.

Large-sized caches. In this case, nodes can potentially store a large portion (i.e., up to 50%) of the available information items. Reduced memory usage is a desirable (if not required) condition, because the same memory may be shared by different services and applications that run at nodes. In such a scenario, a caching decision consists of computing for how long a given content should be stored by a node that has previously requested it, with the goal of minimizing the memory usage without affecting the overall information retrieval performance; Small-sized caches. In this case, nodes have a dedicated but limited amount of memory where to store a small percentage (i.e., up to 10%) of the data that they retrieve. The caching decision translates into a cache replacement strategy that selects the information items to be dropped among the information items just received and the information items that already fill up the dedicated memory.

II. RELATED WORK

Hamlet is a fully distributed caching strategy for wireless ad hoc networks whose nodes exchange information items in a peer-to-peer fashion. In particular, we address a mobile ad hoc network whose nodes may be resource-constrained devices, pedestrian users, or vehicles on city roads. Each node runs an application to request and, possibly, cache desired information items. Nodes in the network retrieve information items from other users that temporarily cache (part of) the requested items or from one or more gateway nodes, which can store content or quickly fetch it from the Internet. We assume a content distribution system where the following assumptions hold: 1) A number I of information items is available to the users, with each item divided into a number C of chunks; 2) user nodes can overhear queries for content and relative responses within their radio proximity by exploiting the broadcast nature of the wireless medium; and 3) user nodes can estimate their distance in hops from the query source and the responding node due to a hop-count field in the messages. Although Hamlet can work with any system that satisfies the aforementioned three generic assumptions, for concreteness, we detail the features of the specific content retrieval system that we will consider in the remainder of this paper. The reference system that we assume allows user applications to request an information item i ($1 \le i \le I$) that is not in their cache. Upon a request generation, the node broadcasts a query message for the C chunks of the information item. Queries for still missing chunks are periodically issued until
either the information item is fully retrieved or a timeout expires. If a node receives a fresh query that contains a request for information i's chunks and it caches a copy of one or more of the requested chunks, it sends them back to the requesting node through information messages. If the node does not cache (all of) the requested chunks, it can rebroadcast a query for the missing chunks, thus acting as a forwarder.

Problem Definition

Mobile environment can be classified as message-based, directory-based, hash based, or router-based data source is, Hybrid Cache might not be a good option. The cache array routing protocol is the most notable hash based cooperative caching protocol.

Existing System

Existing cooperative caching schemes for the Web environment can be classified as message-based, directory-based, hash based, or router-based. The cache array routing protocol is the most notable hash based cooperative caching protocol. The rationale behind CARP constitutes load distribution by hash routing among Web proxy cache arrays. A mobile node doesn't know whether the data source or some other nodes serve its request. If multiple data sources exist, or if the mobile node doesn't know where the data source is, Hybrid Cache might not be a good option. In addition, caching nodes outside the path between the requesting node and the data source might not be able to share cache information with the requesting node.

Limitations of Existing System

Existing cooperative caching schemes for the Web environment can be classified as message-based, directory-based, hash based, or router-based. The cache array routing protocol is the most notable hash based cooperative caching protocol. The rationale behind CARP constitutes load distribution by hash routing among Web proxy cache arrays.

Proposed System

We simulate our caching algorithms in different ad hoc network scenarios and compare them with other caching schemes, showing that our solution succeeds in creating the desired content diversity, thus leading to a resource-efficient information access. Data caching strategy for ad hoc networks whose nodes exchange information items in a peer-to-peer fashion. These decisions are made depending on the perceived "presence" of the content in the nodes proximity, whose estimation does not cause any additional overhead to the information sharing system. However, the solution that was proposed is based on the formation of an overlay network composed of "mediator" nodes, and it is only fitted to static connected networks with stable links among nodes. We proposes a complete framework for information retrieval and caching in mobile ad hoc networks, and it is built on an underlying routing protocol and requires the manual setting of a network wide "Caching zone" These assumptions, along with the significant communication overhead needed to elect "mediator" nodes, make this scheme unsuitable for the mobile environments. The advantages of proposed system are data caching can significantly improve the efficiency of information access in a wireless ad hoc network by reducing the access latency and bandwidth usage. Data for information retrieval and caching in mobile ad hoc networks, and it is built on an underlying routing protocol and requires the manual setting of a network wide "Caching zone".

III. IMPLEMENTATION

Literature survey is the most important step in software development process. Before developing the tool it is necessary to determine the time factor, economy n company strength. Once these things r satisfied, ten next steps are to determine which operating system and language can be used for developing the tool. Once the programmers start building the tool the programmers need lot of external support. This support can be obtained from senior programmers, from book or from websites. Before building the system the above consideration r taken into account for developing the proposed system. We propose, called Hamlet, aims at creating content diversity within the node neighborhood so that users likely find a copy of the different information items nearby (Regardless of the content popularity level) and avoid flooding the network with query messages. Although a similar concept has been put forward in the novelty in our proposal resides in the probabilistic estimate, run by each node, of the information presence (i.e., of the cached content) in the node proximity.

The Hamlet approach applies to the following cases

- Large-sized caches. In this case, nodes can potentially store a large portion (i.e., up to 50%) of the available information items. Reduced memory usage is a desirable (if not required) condition, because the same memory may be shared by different services and applications that run at nodes. In such a scenario, a caching decision consists of computing for how long a given content should be stored by a node that has previously requested it, with the goal of minimizing the memory usage without affecting the overall information retrieval performance;
- Small-sized caches. In this case, nodes have a dedicated but limited amount of memory where to store a small percentage (i.e., up to 10%) of the data that they retrieve. The caching decision translates into a cache replacement strategy that selects the information items to be dropped among the information items just received and the information items that already fill up the dedicated memory. We evaluate the performance of Hamlet in different mobile network scenarios, where nodes communicate through ad hoc connectivity. The results show that our solution ensures a high query resolution ratio while maintaining the traffic load very low, even for scarcely popular content, and consistently along different network connectivity and mobility scenarios.

Modules Analyzed

According to the analysis three modules has been traced out in the design of work. The modules are Self-Organizing, Self-Addressing (Manet), Integrated Cache-Routing, Localized Caching Policy, Distributed Caching Algorithm.

Module Description

Self-Organizing

Multiple Data's for cache placement of multiple data items in ad hoc networks. In the first approach, each node caches the items most frequently accessed by itself; the second approach eliminates replications among neighboring nodes introduced by the first approach; the third approach require one or more "central" nodes to gather neighborhood information and determine caching placements. The first two approaches are largely localized, and hence, would fare very badly when the percentage of client nodes in the network is low, or the access frequencies are uniform. In other related works, discuss placement of "transparent" caches in tree networks.

Self-Addressing

A multi-hop ad hoc network can be represented as an undirected the nodes in the network and E are the set of weighted edges in the graph. Two network nodes that can communicate directly with each other are connected by an edge in the graph. The edge weight may represent a link metric such as loss rate, delay, or transmission power. For the cache placement problem addressed in this article, there are multiple data items and each data item is served by its server (a network node may act as a server for more than one data items). Each network node has limited memory and can cache multiple data items subject to its memory capacity limitation. The objective of our cache placement problem is to minimize the overall access cost. Below, we give a formal definition of the cache placement problem addressed in this system.

Integrated Cache-Routing

Nearest-caching tables can be used in conjunction with any underlying routing protocol to reach the nearest cache node, as long as the distances to other nodes are maintained by the routing protocol. However, note that maintaining cache-routing tables instead of nearest-cache tables and routing tables doesn't offer any clear advantage in terms of number of messages transmissions. We could maintain the integrated cache-routing tables in the similar vein as routing tables are maintained in mobile ad hoc networks. Alternatively, we could have the servers periodically broadcast the latest cache lists. In our simulations, we adopted the latter strategy, since it precludes the need to broadcast Add Cache and Delete Cache messages to some extent.

Localized Caching Policy

The caching policy of DCA is as follows. Each node computes benefit of data items based on its "local traffic" observed for a sufficiently long time. The local traffic of a node i includes its own local data requests, nonlocal data requests to data items cached at i, and the traffic that the node i is forwarding to other nodes in the network. A node decides to cache the most beneficial (in terms of local benefit per unit size of data item) data items that can fit in its local memory. When the local cache memory of a node is full, the following cache replacement policy is used. In particular, a data item is newly cached only if its local benefit is higher than the benefit threshold, and a data item replaces a set of cached data items only if the difference in their local benefits is greater than the benefit threshold.

Distributed Greedy Algorithm (DCA)

In this subsection, we describe a localized distributed implementation of DCA. We refer to the designed distributed implementation as DCA (Distributed Caching Algorithm). The advantage of DCA is that it adapts to dynamic traffic conditions, and can be easily implemented in an operational (possibly, mobile) network with low communication overheads. While we cannot prove any bound on the quality of the solution produced by DCA, we show through extensive simulations that the performance of the DCA.

IV. CONCLUSION

Data caching strategy for ad hoc networks whose nodes exchange information items in a peer-to-peer fashion. Data caching is a fully distributed scheme where each node, upon receiving requested information, determines the cache drop time of the information or which content to replace for the newly arrived information. We have developed a paradigm of data caching techniques to support effective data access in ad hoc networks. In particular, we have considered memory capacity constraint of the network nodes. We have developed efficient data caching algorithms to determine near optimal cache placements to maximize reduction in overall access cost. Reduction in access cost leads to communication cost savings and hence, better bandwidth usage and energy savings. However, our simulations over a wide range of network and application parameters show that the performance of the caching algorithms. Presents a distributed implementation based on an approximation algorithm for the problem of cache placement of multiple data items under memory constraint. The result is the creation of content diversity within the nodes neighborhood so that a requesting user likely finds the desired information nearby. We simulate our caching algorithms in different ad hoc network scenarios and compare them with other caching schemes, showing that our solution succeeds in creating the desired content diversity, thus leading to a resource-efficient information access.

V. FUTURE ENHANCEMENTS

Hamlet is caching self contained and is designed to self adapt to network environments with different mobility and connectivity features.

We assume a content distribution system where the following assumptions hold:

1) A number I of information items is available to the users, with each item divided into a number C of chunks; 2) User nodes can overhear queries for content and relative responses within their radio proximity by exploiting the broadcast nature of the wireless medium; and 3) User nodes can estimate their distance in hops from the query source and the responding node due to a hop-count field in the messages. Although Hamlet can work with any system that satisfies the aforementioned three generic assumptions, for concreteness, we detail the features of the specific content retrieval system that we will consider in the remainder of this paper.

REFERENCE

Good Teachers are worth more than thousand books, we have them in Our Department.

[1] B.-J. Ko and D. Rubenstein, "Distributed self-stabilizing placement of replicated resources in emerging networks," *IEEE/ACM Trans. Netw.*, vol. 13, no. 3, pp. 476–487, Jun. 2005.

[2] G. Cao, L. Yin, and C. R. Das, "Cooperative cache-based data access in ad hoc networks," *Computer*, vol. 37, no. 2, pp. 32–39, Feb. 2004.

[3] C.-Y. Chow, H. V. Leong, and A. T. S. Chan, "GroCoca: Group-based peer-to-peer cooperative caching in mobile environment," *IEEE J. Sel. Areas Commun.*, vol. 25, no. 1, pp. 179–191, Jan. 2007.

[4] T. Hara, "Cooperative caching by mobile clients in push-based information systems," in *Proc. CIKM*, 2002, pp. 186–193.

[5] L. Yin and G. Cao, "Supporting cooperative caching in ad hoc networks," *IEEE Trans. Mobile Comput.*, vol. 5, no. 1, pp. 77–89, Jan. 2006.

[6] N. Dimokas, D. Katsaros, and Y. Manolopoulos, "Cooperative caching in wireless multimedia sensor networks," *ACM Mobile Netw. Appl.*, vol. 13, no. 3/4, pp. 337–356, Aug. 2008.

[7] Y. Du, S. K. S. Gupta, and G. Varsamopoulos, "Improving on-demand data access efficiency in MANETs with cooperative caching," *Ad Hoc Netw.*, vol. 7, no. 3, pp. 579–598, May 2009.

[8] W. Li, E. Chan, and D. Chen, "Energy-efficient cache replacement policies for cooperative caching in mobile ad hoc network," in *Proc. IEEE WCNC*, Kowloon, Hong Kong, Mar. 2007, pp. 3347–3352.

[9] M. K. Denko and J. Tian, "Cross-layer design for cooperative caching in mobile ad hoc networks," in *Proc. IEEE CCNC*, Las Vegas, NV, Jan. 2008, pp. 375–380.

[10] H. Chen, Y. Xiao, and X. Shen, "Update-based cache replacement policies in wireless data access," in *Proc. BroadNets*, Boston, MA, Oct. 2005, pp. 797–804.

[11] J. Xu, Q. Hu, W.-C. Lee, and D. L. Lee, "Performance evaluation of an optimal cache replacement policy for wireless data dissemination," *IEEE Trans. Knowl. Data Eng.*, vol. 16, no. 1, pp. 125–139, Jan. 2004.

[12] J. Cao, Y. Zhang, G. Cao, and L. Xie, "Data consistency for cooperative caching in mobile environments," *Computer*, vol. 40, no. 4, pp. 60–66, Apr. 2007.

[13] N. Dimokas, D. Katsaros, and Y. Manolopoulos, "Cache consistency in wireless multimedia sensor networks," *Ad Hoc Netw.*, vol. 8, no. 2, pp. 214–240, Mar. 2010.

AUTHOR'S PROFILE

Mr. A.**PRAKSAH**, Post Graduated in Computer Science & Engineering (**M.Tech**) From **JNTU**, Hyderabad in 2010 and Graduated in Computer Science & Engineering (B.E) from Dr.Baba Saheb Ambedkar Marathwada university, in 2002. He is working as An Associate Professor in the Department of Computer Science & Engineering in **St.Martin's Engineering College**, R.R Dist, TLG and India. He has 11+ years of Teaching Experience. His Research Interests Include Network Security, Distributed and Parallel Computing.

Ms.K.B.K.S.DURGA, Post Graduated in Information Technology (**M.Tech**) From **JNTU**, Hyderabad in 2009 and Graduated in Information Technolgy (B.E) from Osmania university, in 2005. She is working as An Associate Professor in the Department of Computer Science & Engineering in **St.Martin's Engineering College**, R.R Dist, TLG and India. She has 8+ years of Teaching Experience. Her Research Interests Include Software Engineering, Network Security, Computer networks.

Ms.P.DIVYA, Post Graduated in Computer Science & Engineering (**M.Tech**) From **JNTU**, Hyderabad in 2013 and Graduated in Information Technology (B.Tech) from JNTU,Hyderabad in 2010. She is working as An Assistant Professor in the Department of Computer Science & Engineering in **St.Martin's Engineering College**, R.R Dist, TLG and India. He has 4+ years of Teaching Experience. Her Research Interests Include Network Security, Web Technologies.

Mr.B.JAYANTH, Post Graduated in Computer Science & Engineering (**M.Tech**) From **JNTU**, Hyderabad in 2013 and Graduated in Computer Science & Engineering (B.Tech) from JNTU,Hyderabad in 2011. He is working as An Assistant Professor in the Department of Computer Science & Engineering in **St.Martin's Engineering College**, R.R Dist, TLG and India. He has 3+ years of Teaching Experience. His Research Interests Include Network Security & Cryptography, Computer Networks.

