International Journal of Engineering Research and Development
***e**-ISSN: 2278-067X, **p**-ISSN: 2278-800X, www.ijerd.com*
Volume 2, Issue 9 (August 2012), PP. 28-32

Secure Cooperative Caching In Wireless P2P Networks

Mr. Girish M, Mr. Sundeep Kumar K, Mr. Manoj Challa, Dr. Jitendranath Mungara

Abstract—Secure cooperative caching , that permits the distributing and synchronization of data among manifold peers, could be employed to augment the functioning competency of the data access in wireless P2P networks. As caching is used, data from the base station is duplicated on the caching peer. In view of the fact that a wireless node might revisit the cached data, or change the direction and move up a request to a sharing peer, it is incredibly imperative that the wireless peers do not spitefully change data, drop or move up the request to the erroneous destination. Here in this project we use the strong encryption standard of Triple DES, where data is first encrypted and then gets transmitted so, that the privacy of data is maintained from the hackers and only the genuine nodes would be eligible to decrypt the encrypted file.

Keywords: Mutual sharing, security, wireless P2P networks.

I. INTRODUCTION

Wireless ad hoc networks, wireless mesh networks and wireless sensor networks have been the focus of recent research due to their potential applications in civilian and military environments such as disaster recovery, battlefield, group conference, transportation and intelligent. In such type of networks, nodes will communicate with each other using multiple hop wireless links. And due to lack of any infrastructure support, each node acts as a relay, forwarding data packets for others. Although there are differences (e.g., data transmission speed, link and physical layer characteristics) among these networks,and they have many common features: and also they are based on wireless communication, and these packets are transmitted through the multi-hop relay; also, applications in these networks are typically peer-to-peer rather than client- server communication. There fore,due to these common features, we can refer them as wireless P2P networks.Most of the research in earlier, P2P wireless networks focuses on the development of dynamic routing protocols that can efficiently find routes between two communicating nodes. And although routing is an important issue, other issues such as information (data) access are also very important since the ultimate goal of using wireless P2P networks is to provide information access to the users.

Actually, secure cooperative caching, which allows the sharing and coordination of cached data among multi- nodes, and has been used to increase the Web performance in wired networks. Although this mutual caching and proxy techniques have been studied in wired networks, some part has been done to apply this technique to P2P wireless networks. Due to the special characteristics of P2P wireless networks, techniques designed for wired network may not be applicable. For example, most implementations on secure mutual caching in the Web environment are at the system or application level. As a result, none of them deals with the multiple routing hop problem, and not able to address the on-demand nature of the wireless P2P routing protocols, which will significantly affect the system performance. Further, most research on secure mutual caching in the Web environment assumes a fixed topology, which may not be the case in P2P wireless networks (e.g., wireless P2P networks) due to t h e mobility. Since the expensive of the wireless link is different from the wired link, the decision regarding where to share the data and how to get the cached data may be different.

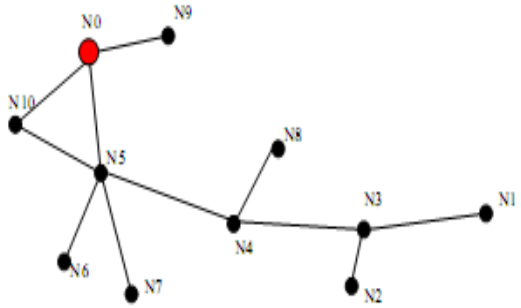

Fig.1 A wireless P2P Network

The paper is organized as follows. In Section 2 we focus on related work, In Section 3, we present our secure mutual share schemes based data access. In Section4, we conclude the paper.

II. RELATED WORK

Mutual caching has been studied in the web environment [4], but little work has been done to efficiently manage the cache in ad hoc networks. Due to movable and constrained resources (i.e., bandwidth, battery power and computational capacity) in wireless networks, mutual cache management techniques designed for wired networks may not be applicable to ad hoc networks.

M. Kodialam and T. Nandagopal [6] proposed several replica allocation methods to increase data accessibility and tolerate network partitions in wireless P2P networks. Although replication can improve data accessibility, the overhead for rechanging the location of replicas periodically is significantly high.Due to updates at the server, the cost of maintaining the consistent copy of replicas is quite high. D. Johnson and D. Maltz [5] suggested the 7DS architecture, in which a set of protocols are defined to cache and disseminate information among users. Depending on the collaborative behavior, a P2P and the server to client mode are used. Unlike our approach, this strategy mainly looks on data dissemination, and thus the management of cache including cache admission control and replacement is not well explored. Sailhann and Issarny [7] proposed a mutual cooperative caching scheme to increase data accessibility by wireless P2P networks, when they are out of bound of a fixed infrastructure. However, the wireless P2P users' location, data popularity and the network density will often change in a real wireless environment, so the static broadcast scheme is hard to adapt to real wireless applications. L. Yin and G. Cao, J. Zhao, P. Zhang [8][9] have proposed mutual caching schemes in wireless P2P networks environments.

V. Kawadia, Y. Zhang. [10] Proposed a mutual caching architecture for supporting continuous media caching proxy. They introduced an application manager to transparently perform data location and session migration of continuous media streams among all proxy caches. L. Yin and G. Cao [2] proposed a secure mutual caching scheme for wireless P2P networks. A broadcast based simple search scheme is proposed to establish cooperation among all wireless nodes in the network to share data items which are cached. Although the data search broadcasting based scheme can locate the nearest required data item, the bandwidth and energy cost of the flooding search is significantly high for Wireless P2P networks. H. Eriksson. [14] Proposed a broadcast based secure mutual caching scheme for hybrid networks where a client shares the caches of clients lying in its proximity. Energy and bandwidth and consumption to locate a client having cached the requested data ,which is very high due to flooding of the messages.

M. Cieslak, D. Foster S, Desilva and S. Das [11][12] design and evaluate three caching algorithms to efficiently support data access in Wireless P2P networks. These algorithms mainly focus on the problem of choosing data item or data path for caching in the limited cache space of wireless nodes. Zang et al. [13] talk of security concerns for mutual caching in wireless P2P networks.

III. SECURE COOPERATIVE CACHE

Caching nodes can replicate data from the server. Because of security concerns, the owners of some sensitive data might want to restrict access to the data, preventing its duplication. We define different levels of data security with regard to duplication and storage in cache node.The data source/server can able specify the level of security for each data item. Depending on the security level, the server can prevent nodes from caching some data or limit the number of nodes that can cache it. For most sensitive data, the data server sends the encrypted version to a few trusted nodes, which decrypt the data using a shared key. With cooperative caching, mobile nodes can return the cached data or modify the route and forward the request to the caching node; hence, the mobile nodes should not be able to modify the data maliciously. With data authentication, a receiver can ensure that the received data is authentic—that is, it originated from the source and was un- modified on the way even of when noneof the other data receivers is trusted. Authenticating the data source is more complicated and overhead. Appending each packet with a message authentication code calculated using a shared key does not work because any receiver with the shared key can forge the data and impersonate the sender. Consequently, we use solutions namely digital signature schemes based on asymmetric encryption and decryption. After the data source signs with its private key of the data, mobile nodes can verify the data's integrity using the data source's public key. Because digital signatures have high overhead in terms of both time to sign and verify and bandwidth, we focus on reducing overhead authentication. For example, if the data nodes have gone through with good reputations, the receiver might not need to verify the signature. This trades some security strength for system performance. Periodically, the receiver might want to verify the signature and change a node's credit rating based on the verification results. A mobile node can verify the signature if the data is important or the node has enough computation power. This allows the user to choose the proper tradeoffs between security and performance—for example, trading security strength for performance if the data is not very important and requires less computation power.

IV. CACHE PLACEMENT ALGORITHMS

Optimal cache placement : We first define a new term called aggregate of delay, which includes the time to service the current client request and the delay to serve future data requests coming from the sameline of path . We can also able to assign different weights to these two parts of the delay based on the objective optimization, e.g., giving less weight to the future share delay if the current request has strict requirement delay. For simplicity, we are going to assign equal weight for the current and future access delay in this paper. Below, we formally define the optimal cache placement problem.

A greedy cache placement algorithm.
To get the optimal placement cache, the data server needs to compute the aggregate delay for every possible cache placement set. Since there are 2n (n is he length of the forwarding path) possible ways to choose placement cache set, which is too costly. Therefore, we propose a Heuristic greedy to efficiently compute the optimal placement cache. Let P*(k) be the optimal cache placement for a forwarding path when only the nearest k hops from the data server are considered as possible cache nodes. With the same condition, let L*(k) be the aggregate delay of the optimal n placement P*(k), and p*(k) be the hop distance of the farthest cache node from the data server in P*(k). When k =0, no cache node is between the data server and the client, and then the N0 data server transmits the data directly to the client Nn without reassembling at any of the intermediate node. All future requests need to get data from the data server from Ni.

Therefore P*(o)=0,p*(k)=0 And

$$L^*(0) = d_{0,n}(S_D) + \sum_{i=1}^{n-1} f_i \cdot (d_{0,i}(S_D) + d_{0,i}(S_R)) \cdot \Delta t.$$

Given l*(k),p*(k)and p*(k), we check whether to select the intermediate node Nk+1 as a cache node.

Notations

- P: cache placement set.
- L: aggregate delay.
- pos: hop distance of the farthest cache node from the data server.
- $f[i]$: Excluded data access frequency on N_i.
- $d_D[i,j]$: delay of forwarding the data item from N_i to N_j.
- $d_R[i,j]$: delay of forwarding the data request from N_j to N_i.
- S_D: size of the data item.
- S_R: size of the data request.
- $R[i]$: link throughput between nodes N_i and N_{i+1}.
- $T[i,j]$: link throughput between nodes N_i and N_j.
- $l[i]$: channel used for the link between nodes N_i and N_{i+1}.
- $h(S)$: cache processing cost for the data size of S.
- Δt: the expiration time of the data item.

Greedy Cache Placement Algorithm
Get frequency data access on nodes—f[]
Get the distance between node I and node i+1 store It intoR [] (Same Cell)
Get the node distance between node I and node j store it into L[] (other Cell)
Get data Expiration time stored into t
Get the size of data request stored into Sr (number of nodes in the cell)
Get size of the data item stored into Sd (size of the node)
Begin
For i=0 to number of nodes (n) -1 do
Distance between nodes in the same cell are stored into t[I,i+1]
For i=0 to n do
For j=i+1 to n do

Calculate time delays from node I to j, calculate delay time from node j to I stored into dD[I,j],dS[I,j] respectively, and store the distance between nodes from node I to j stored into t[I,j]
End for
End for
Set delay time (L) as first cell nth node data forwarding time
Set cache placement as null
Set pos as 0
For i=0 to n-1 do
Calculate the time of delay and add it into L
End for
For k=0 to n-2 do

Calculate the delay aggregate time of transferring the data to another node and calculate the number of nodes the data will be moved and that mean value is stored into L' If l' value is less than l then l' is stored into l Pos value as k+1;
End if
End for
End

V. MUTUAL CACHE BASED DATA ACCESS SCHEMES

5.1 Secure data access: The goal of this research is to provide a secure data access framework for P2P wireless networks. In a wireless P2P network, nodes may be isolated or connected to the wired network through some special nodes that have wireless interfaces to access the wireless infrastructure. Each node can work as a router based on the protocol routing , on which we provide software support to allow them to share their cached (or replicated) data. To allow secure mutual caching, nodes need to decide whether to cache the data locally or cache the path to the data to save space cache. This decision is made off schemes like, such as CachePath, CacheData and HybridCache [1, 2, and 3]. In case of a partition of network , the isolated nodes can only access the cached data item. To increase the accessibility of data ,nodes should cache different data items from their neighbors. However, this may increase the query in delay, since the nodes may have to get access some data from their neighbors instead of locally cache, but accessibility of data will increase. Thus, there will be a tradeoffs between the accessibility of data and query in delay, and we are going to address these tradeoffs by studying plausible cache management techniques.

VI. EXPERIMENTAL RESULTS

Comparing the performance of CachePath, CacheData, HybridCache, and traditional caching,in which a mobile node only caches the data it requests from the data center.Figure 2 shows the impact of cache size on aver-age query delay. When the cache is small,CachePathperforms better thanCacheData. Whenthe cache is more than 800 Kbytes, CacheData performs better because it uses more cache spaceto save passing data.HybridCache performs better than either approach because it applies either CacheData or CachePath to different data items. HybridCachedynamically switches between CacheData and CachePath at the data item level, not at the data-base level.

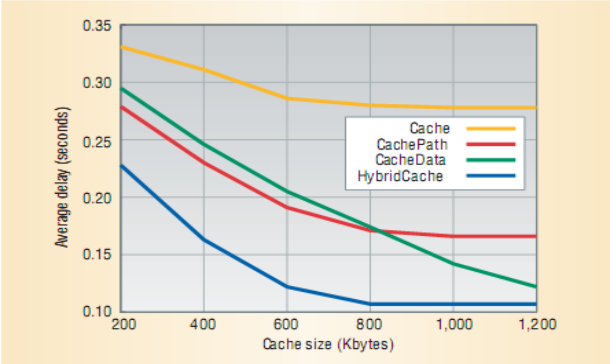

***Figure 2*. The average query delay as afunction of the cache size. The Hybrid Cache performs better than both CacheData and CachePathbecause itcombineesthestrengths of the twoapproaches.**

VII. CONCLUSION AND FUTURE WORK

As wireless nodes in wireless P2P networks may perform similar tasks using common data sets, mutual caching, which allows sharing and coordination of cached data among the multiple nodes, that can be used to reduce the p o w e r a n d bandwidth consumption. We presented a mutual cache based the data access framework, where the nodes may cache the data or the path to the data. To secure the mutual sharing, we proposed solutions to reduce the data authentication overhead and identified possible attacks on cache consistency. Ad hoc networks have been attracted many researchers, the most previous research in this area focuses on routing or how to secure routing, and not much work on data access. The present w o r k on cooperative cache based data access and its security issues will stimulate further research along these directions. Future research should focus on mechanisms that let the data owner control the caching scope without undermining its flexibility. Such mechanisms should maintain a balance between security strength and system performance because encrypting or limiting the distribution of the most sensitive data. To improve further the accessibility of data and reduce the query delay, data can be actively replicated instead of passively cached. Further, we can also identify possible security threats to data consistency and propose viable mechanisms to defend against such attacks.

REFERENCES

[1]. Cao.G , Yin.L , and Das.C, "Mutual Cache Based Data Access in Ad Hoc Networks," IEEE Computer, , February. 2005
[2]. Yin.L and Cao.G, "Supporting Mutual Caching in Ad Hoc Networks," IEEE Transactions on Mobile Computing, vol. 5, no. 1, pp. 77–89, Jan 2006.
[3]. [3] Zhao.J , Zhang.P and Cao.G, "On Mutual Caching in Wireless P2P Networks," IEEE Int'l Conf. on Distributed Computing Systems (ICDCS),-2008
[4]. B. Tang, H. Gupta, and S. Das, "Benefit-Based Data Caching in Wireless P2P Networks," IEEEE Trans. Wireless Computing, vol. 7, no. 3, pp. 289-304, Mar. 2008.
[5]. Johnson .D and Maltz,.D "Dynamic Source Routing in Wireless P2P Wireless Network," Wireless Computing, pp. 153-181, Kluwer Academic Publishers, 1997.
[6]. Kodialam .M and Nandagopal.T "Characterizing the Capacity Region in Multi-Radio, Multi-Channel Wireless Mesh Networks," Proc. ACM MobiCom,2006.

[7]. Perkins.C, Belding-Royer. E, and Chakeres.I, "Wireless P2P on Demand Distance Vector (AODV) Routing," IETF Internet Draft, draft in perkins of manet- aodvbis-00.txt, Oct. 2004. [8] Yin.L and Cao.G, "Supporting Mutual Caching in Wireless P2P Networks," IEEE Trans. Wireless Computing, vol. 5, no. 1, pp. 77-89, Jan. 2007.
[8]. Zhao.J, Zhang.P, and Cao.G, "On Mutual Caching in Wireless P2P Networks," Proc. IEEE Int'l Conf. Distributed i n Computing Systems (ICDCS), pp.731-739, June 2009.
[9]. Kawadia.V , Zhang.Y , and Gupta.B, "System Services for Ad-Hoc Routing: Architecture, Implementation and Experiences," Proc. Int'l Conf.
[10]. Wireless Systems, Applications and Services (MobiSys), 2004.
[11]. Cieslak.M, Foster.D, Tiwana.G, and Wilson.R, "Web Cache Coordination Protocol v2.0," IETF Internet Draft, 2001.
[12]. Desilva.S and. Das.S, "Experimental Evaluation of a Wireless Wireless P2P Network," Proc. Ninth Int'l Conf. Computer Comm. and Networks, 2002.
[13]. Fu .Z, Zerfos.P, Luo.H , Lu.S , Zhang.L, and Gerla.M, "The Impact of Multihop Wireless Channel on TCP Throughput and Loss," Proc. IEEE INFOCOM, 2004
[14]. Eriksson.H, "MBONE: The Multicast Backbone," Comm. ACM, vol. 37, no. 8, pp. 54-60, 1996.

Authors

Mr Girish M is presently doing Master of Technology in Computer Networks and Engineering in CMR Institute of Technology, Bangalore Karnataka. He obtained Bachelor of Engineering degree in Information science and Engineering from S.J.C Institute of Technology, Chickballapur, Karnataka, India ,in the year 2010.

Mr. Sundeep Kumar K received the M.Tech (IT) from Punjabi University in 2003, ME (CSE) from Anna University in 2009 and pursuing Ph. D (CSE) from JNTUA. He is with the department of Computer Science & Engineering and working as an Associate Professor, CMR Institute of Technology, Bangalore. He presented more than 10 papers in International and national Conferences. His research interests include OOMD, Software Engineering and Data Warehousing. He is a life member in ISTE.

Mr. Manoj Challa is pursuing Ph.D. in S.V.University, Tirupati, India. He completed his M.E(CSE) from Hindustan College of Engineering, Tamil Nadu in 2003. He is presently working as Associate Professor, CMR Institute of Technology, Bangalore. He presented nearly 10 papers in national and international conferences. His research areas include Artificial intelligence and computer networks

Dr.M.Jitendranath is double doctorate in Electronics and Computer Science Engineering and working as Professor & Dean of Research in Computer Science Engineering department in CMRIT, Bangalore. He has published 35 papers in the area of Mobile ad hoc Networks international journals