Trusted Cooperative Caching Scheme in Ad-hoc Networks

Dinesh Singh Baghel
(M.Tech Scholar)
Department Of Computer Science and Engineering
Maulana Azad National Institute of Technology
Bhopal-462051

Sweta Jain
(Assistant Professor) Department Of Computer Science and Engineering
Maulana Azad National Institute of Technology
Bhopal-462051

ABSTRACT
This paper investigates about the cooperative caching scheme which allows sharing their cache memory spaces among different node in ad-hoc network. We have applied trust model in cooperative caching which prevents malicious node from taking part in caching scenario and hence improves the reliability of the network. As routing protocol does not bother about the data accessibility and availability, instead their work mainly focus on finding the optimal route between the source and destination. To make routing more useful and reliable trust parameter can be used and to improve the data availability caching can be used. In this paper we apply the trust relationship among the nodes, and according to this trust relationship we take decision whether to make that node as a caching node or consider that node as a future threat in the network. Using this model, we can take more advantage of caching technique by neglecting the overhead generated by such malicious node.

Keywords
Caching, Cooperative Caching, Trust, Cache Consistency

1. INTRODUCTION
Wireless Ad-hoc Network basically consist of multiple heterogeneous devices also called nodes, these node may consist of different memory capacity, battery power, processing power etc. In ad-hoc network when one node wants to communicate with other node which is not in its direct range, then it will forward it's packet via another node which act as an intermediate node between these nodes.

Most of the research in ad hoc networks is focused mainly on the development of efficient routing protocol that can find optimal routes between two communicating nodes. Although routing is an important issue in ad hoc networks but other issues such as data access are also very important since the ultimate goal of using ad hoc networks is to provide data access to mobile nodes [1]. If one node accesses a data item from the data centre, it may possible that its nearby nodes also access the same data item some time later. Then to access that data item it needs to generate a fresh request again. If this node caches such data then it may make it available to other nearby nodes which may save a large amount of battery power, Bandwidth, and Time [17]. Previously cooperative caching has been used in wired network to improve the performance of network and reduce the access cost. In ad hoc networks nodes have restriction on cache memory and battery power which makes cooperative caching difficult to implement. In our scenario we cache data on some nodes in a cooperative manner through which we can significantly reduce the traffic towards the data base server and improve the performance of the system. In previous years lot of work has been done on trusted routing in ad-hoc networks; few works has been done where trust has been applied to the field of data accessibility in ad-hoc network. When we access data in ad-hoc network we do not bother about which route has been taken by the data packet whether it is trusted route or not; we even do not bother about the node with which we are communicating is trusted or not. If the nodes are malicious or captured by some intruder then this caching overhead cannot be neglected. In our proposed model we apply Trust model on cooperative caching scenario in ad-hoc networks.

2. A SYSTEM MODEL
The idea of Trusted Cooperative Caching can be understood by the given Fig.1; in this figure we have shown a scenario in ad-hoc network, in which we have taken fifteen nodes. In this scenario, one node (node no 13) act as a database node and other act as a client which generate data request as per their need. Each node also consist a trust relationship with its neighboring nodes. In this paper we have taken assumption that database node is fixed node and can also be connect through wired network. Other nodes can be mobile nodes; any node needs data, first check data in its local cache memory if it is not available then generate the request for data from database node by flooding the request. The other intermediate node receive the request first check whether it's come from trusted node or not then check the availability of data in its own cache if it is not found then forward the request towards the data base node by optimal route using any routing algorithm. If it found then generate the reply packet towards the requesting node. For caching there are some technique present[3], data cache, path cache, hybrid cache; In Data cache we cache the data in cache memory of node but this technique is only applicable when the data size is small if the data size is large then we apply path cache technique, in which we cache the path for particular data item when any data request comes it make the route available instead of generating the route request, but this technique also has some draw back as it is only applicable for small network, for large networks we cannot apply this technique. For this hybrid cache technique can be applied. Before caching we take decision whether we cache data or path based on parameter like data size and TTL value [2]. Apart from this in trusted caching technique we always check whether the reply come from trusted node or not. We take caching decision based on the trust value of the node. This provides the real essence of trusted cooperative caching.

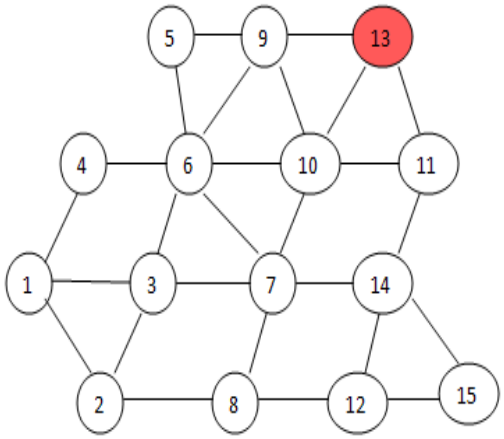

Fig 1: An Ad-hoc network

3. CACHE CONSISTENCY

There is a cache consistency issue When we dealing with caching. In caching it is compulsory that client receive correct data always, for there are two consistency model already present [2]. one is weak consistency model, in which we can use one parameter like validation period field with that data, that make this data validate effectively, the other model is strong consistency model, in which we do not believe on any intermediate node and each time we receive the reply we first confirm that reply from the database node by generating a request when it finds correct then accept that data otherwise reject that data. As we see by strong consistency model we increase the overhead by accessing the database each time. And also we lost the benefits of caching as each time we go to the database node [13]. Due to bandwidth and power constraints in ad hoc networks, it is too expensive to maintain strong cache consistency; the weak consistency model is more attractive. A simple weak consistency model can be based on the Time-To-Live (TTL) mechanism, in which a node considers a cached copy up-to date if it's TTL has not expired, if the TTL expires it will remove corresponding entry from cache table, as a result future requests for this data will be forwarded to the data centre.

4. CACHE MANAGEMENT

The cached data should be updated dynamically to ensure the consistency between the cached data and the original data in the database, for this we have provide the new mechanism to maintain the consistency. The steps are as follows-

- When any node cache the data then it will sent this information to the data base.
- Data base node will maintain the table of which node cache witch data
- Whenever any change in the data then database sent this updated information to the that caching node
- Caching node will update this information by replace that data with new arrival one

5. POSSIBLE MALICIOUS ACTIVITIES OF NODE IN CACHING MODULE

- Stop forwarding the data_request packet and data_reply packet.
- Cache consistency attack.
- Dropping data packets randomly lead to their unnecessary retransmissions and possibly even new route discovery procedures, Modifying data contents as well as its checksum before forwarding it to next-hop node. This activity will go undetected by the next-hop node, since at the TCP layer it will only recalculate the data checksum and verify with the checksum present in the received data packet, and find it equal.

In our scheme, we handle these malicious activities as described in this section. Before forwarding the data packet; the node buffers its sequence no. and checksum so that it can use it for comparison later. For this we have use the concept of promiscuous mode, while in promiscuous listening mode, the node will listen to all the transmissions of its next-hop neighbor. For each data packet it listens, it checks the packet sequence no. and the checksum against those buffered earlier for this packet. If they match, it means that "correct" data packet was forwarded.

In cache consistency attack any node may turn out to be a malicious and generate fake reply or wrong reply packet for each data request by which it will increase the hit ratio as in previous caching technique we use weak consistency model, by this attack we also increase the processing overhead of fake reply packets.

6. PROPOSED CACHING ALGORITHM

We have proposed a hybrid caching algorithm witch process any data_request and data_reply packet based on their trust value. In our algorithm each node maintains a data structure of its neighboring node; in that data structure it will maintain the TrustA value and TrustB value and also number of reply generated by that node. Based on these values, we evaluate the Trust factors and according to that we will decide whether the packet should be accepted or not. With use of concept of trust model we can also isolate the malicious node from the network.

/*in handle_data_request when request arrive from node x */

```
If (TrustA(x)>ThresoldA)
 {
  If there is a valid copy in the cache
  Then send di to the requester;
   Else
   Forward the request to the data centre;
  }
Else
  Drop the packet and stop the retransmission of the packet;
```

/*in handle_data_reply when reply arrives from node x*/

```
If (TrustB(x)>ThresoldB)
 {
  If (di is the requested data by the node)
  Then cache di;
   Else
     If (there is a copy of di in the cache)
       Then update the cached copy;
}
Else
 Drop the packet;
```

6.1 Trust A Calculation Based on Packet Forwarding

- Before forwarding the packet, the node buffers its sequence no. and checksum.
- In Promiscuous listening mode, the node will listen to all the transmissions of its next-hop neighbor.

- For each data packet it listens and compares the sequence no. and checksum against the buffered one. If they match, it means that "correct packet" was forwarded and TrustA will be incremented otherwise TrustA will be decremented.

6.2 TrustB Calculation Based on Reply Packet

The TrustB can be calculated based on number of reply packet generated be some neighbor node x. we maintain the trust table on node and we update the table by calling cache check_consistency procedure by the node. The check_consistency procedure will generate a request packed for database, for that packet it will only accept the reply generated by database node only and check whether the value of dataid di return by database is equal to the value for which we have checked, and update the corresponding entry in the table. If this procedure returns that the node is truth then only we increment the value of TrustB otherwise we decrement the TrustB and also discard the reply packet.

```
/* check_consistency procedure */
If (value_id==value_database)
        Truth=1;
Else
        Truth=0;
If (truth)
        TrustB++;
Else
        TrustB--;
```

7. RELATED WORK

Cooperative Caching is very helpful to reduce the network bandwidth uses and reduce the access time also, cooperative caching also very useful to reduce the traffic load towards the database node. Previously cooperative caching has been applied in the wired network so it is quit new in ad-hoc because of some resource constraints. for this some work has been done toward the decision taking about where to cache the data and how to deal with Cache Replacement issue and Cache Updating issue, also one of the important issue we have to deal with, apart from this the major issue is cache consistency issue.

Yin and Cao propose three schemes: CachePath, CacheData, and HybridCache[3]. Chiu and young proposed IXP and DPIP cooperative caching scheme [1], Index Push (IXP) is push based scheme, a mobile node broadcasts an index packet in its zone to advertise a caching event. The Data Pull/Index Push (DPIP) is a pull based scheme in which an implicit index pushes property by exploiting in-zone request broadcasts. Chow and Leong have proposed CoCa, a Cooperative Caching protocol [5]. In this protocol, mobile nodes share their cache contents with each other to reduce both the number of server requests and the number of access misses. In the improvement of CoCa, a Group-Based Cooperative Caching scheme, called GroCoCa, has been proposed in [6], in which a centralized incremental clustering algorithm is adopted. GroCoCa improves system performance but also cost extra battery consumption, as in GroCoCa author also consider mobility and data access pattern. Han and Stephan have proposed FCCS called Fuzzy Cooperative Caching scheme [16]. FCCS addresses two basic problems for cooperative caching in MANETs, Cache Resolution and Cache Management. For cache resolution, FCCS tries to discover a data source, which induces less communication cost by utilizing historical profiles and forwarding nodes. For cache management, FCCS minimizes caching duplication between neighbor nodes and uses fuzzy cooperative caches based on data item utility and access similarity to improve the overall performance. To deal with the security issue, very little work has been done in the field of cooperative caching in ad-hoc networks. But in Ad-hoc networks Hubaux, Butyaan and Capkun addressed the issue of distributed public keys, by proposing to let nodes issue certificates for each other based on their personal experience for that node [8]. Zhou and Haas proposed a solution based on threshold cryptography [11]. Based on a trusted certificate authority, nodes get certificates from the CA to identify them to avoid spoofing and malicious route updates. To address the high overhead associated with obtaining and verifying the digital certificates. Hu, Perrig and Johnson proposed a protocol [7] to secure on-demand routing protocols based on TESLA [12], an efficient broadcast authentication scheme that requires loose time synchronization. In [10], the authors proposed to use a watchdog entity to identify misbehaving nodes and a pathrater mechanism to route the packet only through trusted nodes. When a node forwards a packet, the node's watchdog verifies that the next node in the path also forwards the packet. This can be accomplished by listening promiscuously to the next node's transmissions. If the next node does not forward the packet, it is misbehaving. The pathrater uses this knowledge of misbehaving nodes to choose the network path that is most likely to deliver packets.

Zang, Yin and Chao proposed a secure cooperative caching scheme [9], In this scheme author has been identify possible security attacks on cache consistency and propose a randomized grouping based schemes for intrusion detection, damage recovery and intruder identification with the use of Massage Authentication Code(MAC) which also has significant overhead. In our paper we have reduce such overhead by using trust mechanism.

8. CONCLUSION

We have proposed a trusted cooperative caching scheme which provides a solution to the malicious activity present in any caching scenario with the help of trust relationship between the nodes. Trust has been applied on mobile nodes according to their operational behavior so that their activity can be monitored efficiently. Each node will maintain the trust value of its neighbor's node. By applying Trust model we can isolate the malicious node and their activities such as fake reply packet generation in the network. Trusted cooperative caching will be beneficial for MANETs as here nodes have limited cache space. This scheme will be helpful in efficient and correct utilization of cache space as data request generated by any node will be replied by a trusted node with minimum cost and minimum overhead even in the presence of the malicious node.

9. REFERENCES

[1]. Ge-Ming Chiu and Cheng-Ru Young, "Exploiting In-Zone Broadcasts for Cache Sharing in Mobile Ad Hoc Networks", IEEE TRANSACTIONS ON MOBILE COMPUTING, Volume:8, NO.3 MARCH 2009.

[2]. L. Yin and G. Cao, ""Supporting Cooperative Caching in Ad Hoc Networks," Proc. IEEE INFOCOM 2004, pp. 2537-2547, 2004.

[3]. L. Yin and G. Cao, "Supporting Cooperative Caching in Ad Hoc Networks," TRANSACTIONS ON MOBILE COMPUTING, Volume: 5, no. 1, pp. 77-89, Jan. 2006.

[4]. C.-Y. Chow, H.V. Leong, and A. Chan, "Peer-to-Peer Cooperative Caching in Mobile Environments", Proc. 24th

Int'l Conf. Distributed Computing Systems Workshops (ICDCSW '04), pp. 528-533, 2004.

[5]. Chow, H.V. Leong, and A. Chan, "Cache Signatures for Peer to-Peer Cooperative Caching in Mobile Environments," Proc. 18th Int'l Conf. Advanced Information Networking and Applications (AINA '04), pp. 96-101, 2004.

[6]. Chi-Yin Chow; Hong Va Leong; A.T.S Chan, "GroCoca: Group-Based Peer-to-Peer Cooperative Caching in Mobile Environment" IEEE TRANSACTIONS Volume: 25, issue 1, pp 179-191, 2007.

[7]. Y. Hu, A. Perrig, and D. Johnson, "Packet Leashes: A Defense against Wormhole Attacks in Wireless Ad Hoc Networks," IEEE Infocom, April 2003.

[8]. H. Hubaux, L. Buttyan and Capkun, "The Quest for Security in Mobile Ad Hoc Networks," ACM MobiHoc, pp. 146–155, 2001.

[9]. W. Zhang, L. Yin and G. Cao, "Secure Cooperative Cache Based Data Access in Ad Hoc Networks," NSF International Workshop on Theoretical and Algorithmic Aspects of Wireless Ad Hoc, Sensor, and Peer-to-Peer Networks, June 2004.

[10]. S. Marti, T. Giuli, K. Lai, and M. Baker, "Mitigating Routing Misbehavior in Mobile Ad Hoc Networks," ACM MobiCom, Aug. 2000.

[11]. L. Zhou and Z. Haas, "Securing Ad-Hoc Networks," IEEE Network, Volume: 13, no. 6, pp. 24–30, November/December 1999.

[12]. A. Perrig, R. Canetti, J.D. Tyger and D. Song, "The TESLA Broadcast Authentication Protocol," In CryptoBytes, Volume: 5, No. 2, Summer/Fall 2002, pp. 2-13.

[13]. P. Cao and C. Liu, "Maintaining Strong Cache Consistency in the World-Wide Web," Proceedings of the 17th International Conference on Distributed Computing Systems, Page(s): 12 – 21, 1997.

[14]. Naveen Chauhan, L. K. Awasthi, N. Chand, "Cooperative Data Caching with Prefetching in Mobile Ad-hoc Network", AH-ICI 2009 First Asian Himalayas International Conference, 2009.

[15]. G. Cao, L. Yin, and C. Das, "Cooperative Cache-Based Data Access in Ad-Hoc Networks," IEEE TRANSACTIONS, Volume: 37, Issue: 2, pp. 32–39, Feb. 2004.

[16]. Ihn-Han Bae,Stephan Olariu, "Design and Evaluation of a Fuzzy Cooperative Caching Scheme for MANETs", 6th International Conference on Wireless Communications Networking and Mobile Computing (WiCOM), 2010, Page(s): 1 - 5.

[17]. P.Kuppusamy, Dr. K. Thirunavukkarasu and Dr. B. Kalaavathi, "A Review of Cooperative Caching Strategies in Mobile Ad Hoc Networks", International Journal of Computer Applications (0975 – 8887) Volume: 29, No.11, September 2011.