IOSR Journal Of Computer Engineering (IOSRJCE)
ISSN: 2278-0661 Volume 6, Issue 4 (Sep.-Oct. 2012), PP 11-19
Ww.Iosrjournals.Org

Architecture and Evaluation on Cooperative Caching In Wireless P2P

[1]B. Sateesh, [2]V. Satish, [3]D. Sujatha

***Abstract**— All researches shown that cooperative cache can improve the system performance in wireless P2P networks such as ad hoc networks and mesh networks. And all these studies have been done up to some level only.So for that purpose now iam using an protocol technique called Dynamic Source routing (DSR). And previous studies have been done mainly by using an AODV routing technique and these are all at high level and leaving many design and implementation issues unanswered. In this paper, I am mainly presenting my design and implementation of cooperative cache in wireless P2P networks, and propose solutions to find the best place to cache the data. For that one I propose an novel asymmetric cooperative cache approach by using an Dynamic Source Routing Protocol. By using these technique I can say that packet routing to be trivially loop-free, avoids the need for up-to-date routing information in the intermediate nodes through which packets are forwarded, and allows nodes forwarding or overhearing packets to cache the routing information in them for their own future use. All aspects of the protocol operate entirely on-demand, allowing the routing packet overhead of DSR to scale automatically to only that needed to react to changes in the routes currently in use. The data has been accessed easily by the user. And also by using an data pipelines we can reduce the end-to-end delays between the server and client. My results show that the Dynamic Source Routing out performs the AODV in wireless P2P networks.*
***Key Words—** wireless networks, caching, protocols, DSR.*

I. INTRODUCTION

Wireless networks are ad hoc network, mesh networks, and sensor networks. Ad hoc networks are ideal in situations where installing an infrastructure is not possible because the infrastructure is too expensive or too vulnerable. Due to lack of infrastructure support, each node in the network acts as a router, forwarding data packets for other nodes. Most of the previous researches [1], [2], [3] in ad hoc networks focus on the development of dynamic routing protocols that can efficiently find routes between two communicating nodes. Although routing is an important issue in ad hoc networks, other issues such as information (data) access are also very important since the ultimate goal of using ad hoc networks is to provide information access to mobile nodes. Wireless P2P networks, have received considerable attention due to their potential applications in civilian and military environments those examples are given below.

1.1 Examples

i. In a battlefield, a wireless P2P network may consist of several commanding officers and a group of soldiers. Each officer has a relatively powerful data center, and the soldiers need to access the data centers to get various data such as the detailed geographic information, enemy information, and new commands. The neighboring soldiers tend to have similar missions and thus share common interests. If one soldier has accessed a data item from the data center, it is quite possible that nearby soldiers access the same data some time later. It will save a large amount of battery power, bandwidth, and time if later accesses to the same data are served by the nearby soldier who has the data instead of the far away data center.

ii. Recently, many mobile infostation systems have been deployed to provide information for mobile users. Forexample, infostations deployed by tourist information center may provide maps, pictures, history of attractive sites. Infostation deployed by a restaurant may provide menus. Due to limited radio range, an infostation can only cover a limited geographical area. If a mobile user, Pravallika moves out of the infostation range, she will not be able to access the data provided by the infostation. However, if mobile users are able to form an ad hoc network, they can still access the information. In such an environment, when Pravallika's request is forwarded to the infostation by other mobile users, it is very likely that one of the nodes along the path has already cached the requested data. Then, this node can send the data back to Pravallika to save time and bandwidth.

iii. People in the same residential area may access the Internet through a wireless P2P network, e.g., the Roofnet. After one node downloads a MP3 audio or video file, other people can get the file from this node instead of the far away Web server.

By the above examples I can say that wireless networks are very useful for humans. Therefore, if nodes are able to collaborate with each other, bandwidth and power can be saved, and delay can be reduced. Cooperative caching[2],[3],[10], which allows the sharing and coordination of cached data among multiple nodes. And here I am mainly concentrating on how to perform these operations easily and how to send data with out any delays. In [5] Prof E.Royer and C.Perkins suggested modifications to the wireless peer 2 peer networks through the existing kernel code to implement AODV. By extending ARP, Desilva and Das[10] presented another kernel implementation of AODV. Dynamic Source Routing (DSR)[15] has been implemented by the Monarch project in FreeBSD. This implementation was entirely in kernel and made extensive modifications in the kernel IP stack. And all authors explored several system issues regarding the design and implementation of routing protocols for ad hoc networks. And all use an AODV routing only so here we are facing an problem. So for that purpose I am proposed an DSR routing. However, none of them has looked into cooperative caching in wireless P2P networks.

Existing cache network although cooperative cache has been implemented by many researchers, these implementations are in the Web environment, and all these implementations are at the system level. As a result, none of them deals with the multiple hop routing problems and cannot address the on-demand nature of the ad hoc routing protocols. To realize the benefit of cooperative cache, intermediate nodes along the routing path need to check every passing-by packet to see if the cached data match the data request. This certainly cannot be satisfied by the existing ad hoc routing protocols.

In this paper, we present our design and implementation of cooperative cache in wireless P2P networks. Through real implementations, we identify important design issues and propose an Dynamic Source Routing with asymmetric approach to reduce the overhead of copying data between the user space and the kernel space, and hence to reduce the data processing delay.

II. Architecture And Evaluation of Cooperative Caching

IN this section iam mainly going to discuss about cooperative caching. This cooperative caching mainly have 3 schemas. They are CachePath,CacheData and Hybrid Cache.

2.1 Caching Schemas

The below fig. mainly going to demonstrate about the CachePath and Cache Data concepts. Suppose if node N1 requests a data item from N0. Thus N3 forwards to N1;N3 knows that N1 has a copy of the data. Later, if N2 requests N3 knows that the data source N0 is three hops away whereas N1 is only one hop away. Thus, N3 forwards the request to N1 instead of N4. Many routing algorithms (such as AODV and DSR) provide the hop count information between the source and destination. Caching the data path for each data item reduces bandwidth and power consumption because nodes can obtain the data using fewer hops. However, mapping data items and caching nodes increase routing overhead, and the following techniques are used to improve CachePath's performance.

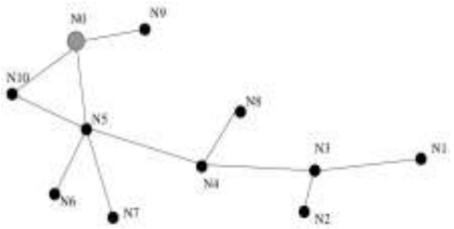

Figure 1:- A WIRELESS P2P NETWORKS

CachePath :

In CachePath, a node need not record the path information of all passing data. Rather, it only records the data path when it is closer to the caching node than the data source. For example, when N0 forwards di to the destination node N1 along the path N5 _ N4 _ N3; N4 and N5 won't cache path information because they are closer to the data source than the caching node N1. In general, a node caches the data path only when the caching node is very close. The closeness can be defined as a function of the node's distance to the data source, its distance to the caching node, route stability, and the data update rate. Intuitively, if the network is relatively stable, the data update rate is low, and its distance to the caching node is much shorter than its distance to the data source, then the routing node should cache the data path.

CacheData

In CacheData, the intermediate node caches the data instead of the path when it finds that the data item is frequently accessed. For example, in Fig. 1, if both N6 and N7 request di through N5;N5 may think that di is popular and cache it locally. N5 can then serve N4's future requests directly. Because the CacheData approach needs extra space to save the data, it should be used prudently. Suppose N3 forwards several requests for di to N0. The nodes along the path N3;N4, and N5 may want to cache di as a frequently accessed item. However, they will waste a large amount of cache space if they all cache di. To avoid this, CacheData enforces another rule: A node does not cache the data if all requests for the data are from the same node. In this example, all the requests N5 received are from N4, and these requests in turn come from N3. With the new rule,N4 and N5 won't cache di. If N3 receives requests from different nodes, for example, N1 and N2, it caches the data. Certainly, if N5 later receives requests for di from N6 and N7, it can also cache the data.

Hybrid Cache

Analytical results show that CachePath performs better when the cache is small or the data update rate is low, while CacheData performs better in other situations. To further improve performance, we can use Hybrid Cache, a hybrid scheme that exploits the strengths of CacheData and CachePath while avoiding their weaknesses. Specifically, when a node forwards a data item, it caches the data or path based on several criteria

2.2 Design Issues And Asymmetric Cooperative Cache

In all previous studies the all researches proposed an two design's for design issues. Those are Integrated Design and Layered Design but in these studies the data has request's has been traveled to each node from source to destination. And same like data reply's have been transferred from node to node but some node's cache that data by using an caching technique. To realize the benefit of cooperative cache, intermediate nodes along the routing path need to check every passing-by packet to see if the cached data match the data request. This certainly cannot be satisfied by the existing ad hoc routing protocols. So now I am mainly using an Dynamic Source Routing with the asymmetric cooperative cache approach.

The Asymmetric Cooperative Cache Approach

To address the problem of the layered design, we propose an asymmetric approach. We first give the basic idea and then present the details of the scheme. The Basic Idea In our solution, data requests and data replies are treated differently. The request message still follows the path shown in Fig. 2a; however, the reply message follows a different path. If no intermediate node needs to cache the data, N0 sends the data directly to N5 without going up to the cache layer. Suppose N3 needs to cache the data based on the cooperative cache protocol, as shown in Fig. 3. After N3 receives the request message, it modifies the message and notifies N0 that the data should be sent to N3. As a result, the data are sent from N0 to N3 through the cache layer, and then sent to N5. Note that the data will not go to the cache layer in intermediate nodes such as N1;N2, and N4 in this example. In this way, the data only reach the routing layer for most intermediate nodes, and go up to the cache layer only when the intermediate node needs to cache the data. Although the request message always needs to goup to the cache layer, it has a small size, and the added overhead is limited.

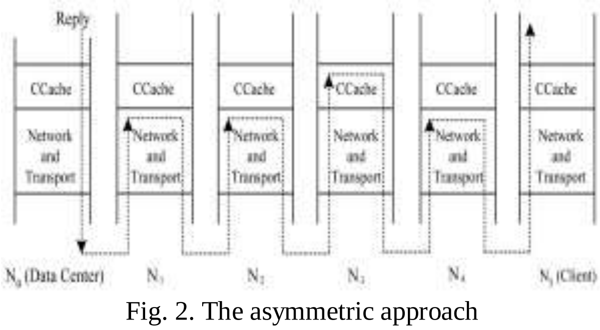

Fig. 2. The asymmetric approach

If the requested data item is large, this asymmetric approach allows data pipeline between two caching nodes, and hence reduces the end-to-end delay. The cache layer processing overhead, especially data copying between kernel and user spaces, is also minimized because the data item is not delivered to the cache layer at the nodes that are unlikely to cache the data. The Asymmetric Approach has been already clearly discussed by Jing Zhao and Ping Zhang and Guohong cao[1],[9] in On Cooperative Caching in Wireless P2P Networks.

III. Dynamic Source Routing Protocol

The Dynamic Source Routing protocol (DSR) is a simple and efficient routing protocol designed specifically for use in multi-hop wireless ad hoc networks of mobile nodes. DSR allows the network to be completely self-organizing and self-configuring, without the need for any existing network infrastructure or administration.DSR has been implemented by numerous groups, and deployed on several test beds. Networks using the DSR protocol have been connected to the Internet. DSR can interoperate with Mobile IP, and nodes using Mobile IP and DSR have seamlessly migrated between WLANs, cellular data services, and DSR mobile ad hoc networks. The DSR protocol is composed of two mechanisms that work together to allow the discovery and maintenance of source routes in the ad hoc network:

Route Discovery is the mechanism by which a node N0 wishing to send a packet to a destination node N1 obtains a source route to by using an cache path. Route Discovery is used only when N1 attempts to send a packet to and does not already know a route to N0. *Route Maintenance* is the mechanism by which node is able to detect, while using a source route N0 to , if the network topology has changed such that it can no longer use its route to N1 because a link along the route no longer works. When Route Maintenance indicates a source route is broken, source node can attempt to use any other route it happens to know to destination Node, or can invoke Route Discovery again to find a new route. Route Maintenance is used only when **S** is actually sending packets to destination Node.

Route Discovery and Route Maintenance each operate entirely *on demand*. In particular, unlike other protocols, DSR requires *no* periodic packets of *any kind* at *any level* within the network. For example, DSR does not use any periodic routing advertisement, link status sensing, or neighbor detection packets, and does not rely on these functions from any underlying protocols in the network. This entirely on-demand behavior and lack of periodic activity allows the number of overhead packets caused by DSR to scale all the way down to *zero*, when all nodes are approximately stationary with respect to each other and all routes needed for current communication have already been discovered. As nodes begin to move more or as communication patterns change, the routing packet overhead of DSR *automatically* scales to only that needed to track the routes currently in use.

	AODV	**DSR**
Update Routing Table Periodically	Send hello	No
Support One Way Link	No	Yes
Mechanism Of Routing	Next Hop	Source Routing
The Metrics Of Routing	Shortest Path	Shortest Path
Need Other Protocol Support	No	No

The above table shows that why I choose DSR routing instead of AODV Routing and also the shortcoming of AODV is that it does not support one-way link. To support any cast, DSR is selected to extend, although the cost of DSR routing is greater than AODV as for carrying path information in each packet. The essential reason is its mechanism of routing – source routing which helps to resolve the maintaining state-based connection difficulty in any casting routing.

3.1 Routing Discovery

The route discovery process of DSR is by a source node broadcasts a RREQ packet to all its neighbors; the RREQ packet appends the each node's id to its route record when it is forwarded by these nodes. In this way, the RREQ is flooded throughout the network, and it includes all the nodes' information of the path when it is received by the destination node. Then, the destination node sends a RREP return to the source after choosing the optimum route from the numerous RREQs. Source node would append the route record which is in RREP to the routing table and all data packets which would be sent to destination node will include the route record in it. Intermediate nodes just check the route record of packets and forward them as the route. The route discovery process diagram as follows:

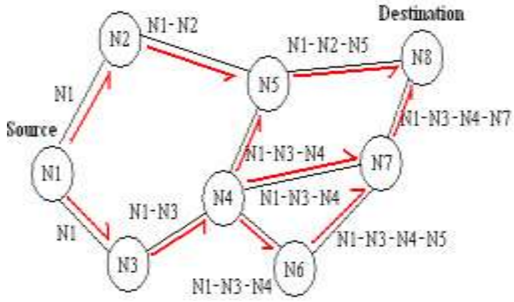

Fig 3. DSR routing discovery I, RREQ

See figure 3, a source node N1 wants to send data packet to destination node N8, it broadcasts a RREQ message first. Then, the intermediate nodes receive the packet check whether they are the destination node. If they are not, they would add their own address information to the route record of the packet and forward the packet along its outgoing links until the destination node N8 received the RREQ.

Then, a RREP packet is generated when the RREQ reaches to the destination N8 (see figure 3). Afterwards it will be sent back to the source node N1. On the receiving RREP, source node records the route with destination node as the data delivery route.

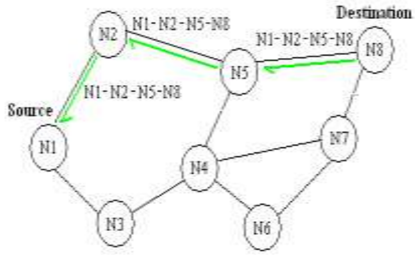

Fig 4. DSR routing discovery II, RREP

3.2 Routing Maintenance

The meaning of Routing maintenance refers to each DSR node maintains a route cache; it records the route information of hop-by-hop which can reach to the other nodes. Otherwise, every node can snoop from the data packet which is transmitting by the neighbor. The process of the snooping can be used to analysis the route information which is recorded in the front of data packet, the node records route information to its route cache if the route is a new one. Thus, more and more route information would be record to the route cache by the node and reduce the time of Flooding to broadcast RREQ. Meanwhile, the bandwidth of each node can also be saved. The processing of routing maintenance detects the changing of network topology, and it knows whether the route is still available or not. When an intermediate node removes from the range of wireless transmission or it is shutdown, the route is no longer to use. When the upstream node detects the route is failure by MAC layer protocol, it sends a RERR message to its upstream and source node. On the receiving RERR, source deletes all route information which includes the failure route from its route cache. If necessary, source node reinitiates a route discovery process in order to establish a new route to destination node. DSR can maintain numerous routes for one destination node. If the main route is failure, a backup route can be used to transfer data. Thus, this mechanism avoids DSR initiates flooding of RREQ frequently.

3.3 Architecture Overview

Above Fig. shows the architecture of our cooperative cache middleware, which consists of three parts: Cooperative Cache Supporting Library (CCSL), Cooperative Cache Daemon (CCD), and Cooperative Cache Agent (CCA). CCSL is the core component to provide primitive operations of the cooperative cache, e.g., checking passing by packets, recording data access history, and cache read/write/replacement primitives A data cache buffer is maintained at every node to store the cached data items. There is an interface between CCSL and the routing daemon, from which CCSL obtains the routing distance to a certain node. All these primitive cache operations are enclosed as CCSL API to provide a uniform interface to the upper layer.CCD is the component that implements different cooperative cache mechanisms, namely, CacheData, CachePath, and HybirdCache. There is one cache daemon for each cooperative cache scheme. It extends the basic CCSL API to accomplish the characteristic of each scheme.CCA is the module that maps application protocol messages to corresponding cooperative cache layer messages. It is a connector between CCD and user applications. There is one CCA for each user application.

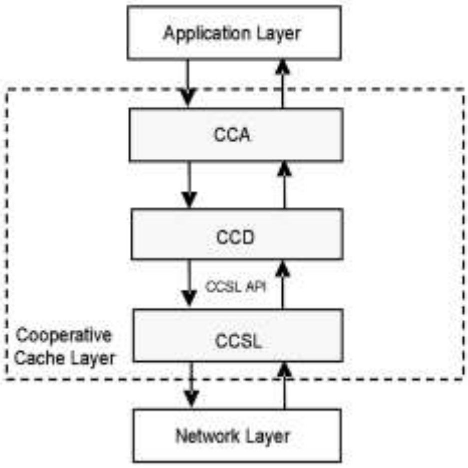

Fig. 5. System architecture.

IV. Experimental Results And Evaluation

The keep the experiments on cooperative cache implementation we did some good work an the wireless networks and these results have been shown in below. manageable and repeatable they were run in a static (non-mobile) configuration. In some experiments route breaks are emulated by artificially purging the routing table (ARP table) entries using an independent daemon program to force new route discoveries. In others, route loss was a part of the natural dynamics of the network because of loss of hello messages.

This section summarizes some of our experiences evaluating DSR through detailed studies using discrete event simulation, and through implementation and actual operation and experience with the protocol in an ad hoc networking test bed environment. Our simulation environment consists of a set of wireless and mobile networking extensions that we have created a*ns-2* network simulator from these extensions we provide a detailed model of the physical and link layer behavior of a wireless network and allow arbitrary movement of nodes within the network. At the physical layer, we provide realistic modeling of factors such as free space and ground reflection propagation, transmission power, antenna gain, receiver sensitivity, propagation delay, carrier sense, and capture effect. At the link layer, we model the complete Distributed Coordination Function (DCF) Media Access Control (MAC) protocol of the IEEE 802.11 wireless LAN protocol standard, along with the standard Internet Address Resolution Protocol (ARP). These wireless and mobile networking extensions are available from a version of them have also now been adopted as a part of the standard VINT release of *ns-2*. We have done a number of different simulation studies with this environment, analyzing the behavior and performance of DSR and comparing it to other proposed routing protocols for ad hoc networks. Here we summarize only some of the basic results that indicate DSR's excellent performance. In the results presented here, all simulations were run in ad hoc networks. We have found that this model can produce large amounts of relative node movement and network topology change, and thus provides a good movement model with which to stress DSR or other ad hoc network routing protocols.

4.1 Simulation Model

We design our simulation test with ns-2[16].All simulations were in an ad hoc network consisting of 50 nodes spread uniformly through a 1000×1000 meter square area. Nodes are equipped with an IEEE 802.11 radio network interface, operating at 2Mbps data rate with a 250m transmission range. Nodes move according to the Random Waypoint mobility model with speed uniformly distributed between 0 and 10m/s and pause time between 0 and 500 ms. Simulations run for 1200 seconds and there are 20 sources and one any cast group with 5 members in the network. *Max D* of the network is defined as 40 hops. Three protocols are analyzed in the simulation: Dynamic Source Routing (DSR), Packet Delivery Ratio which is the ratio of the data packets reached the destination to those originated by the source, Mean Message Hop Count which is the average count of hop s from source to destination and Routing Overhead which is measured by radio of request and answering packets to the sum of packets.

The node movement model: We model a group of nodes moving in a 1000×500m rectangle area, which is similar to the model used in [20]. The moving pattern follows the random way point movement model [22]. Initially, nodes are placed randomly in the area. Each node selects a random destination and moves toward the destination with a speed selected randomly from (0 m/s, *vmax* m/s). After the node reaches its destination, it pauses for a period of time and repeats this movement pattern. Two *vmax* values, 2 m/s and 20 m/s, are studied in the simulation.

The client query model: The client query model is similar to what have been used in previous studies [8], [13]. Each node generates a single stream of read-only queries. The query generate time follows exponential distribution with mean value *Tquery*. After a query is sent out, the node does not generate new query until the query is served. The access pattern is based on $Zipf - like$ distribution [17], which has been frequently used [7] to model non-uniform distribution. In the Zipf-like distribution, the access probability of the *ith* ($1 \leq i \leq n$) data item is represented as follows.

$$P_{a_i} = \frac{1}{i^\theta \sum_{k=1}^{n} \frac{1}{k^\theta}}$$

where $0 \leq \theta \leq 1$. When $\theta = 1$,it follows the strict Zipf distribution. When $\theta = 0$, it follows the uniform distribution. Larger θ results in more "skewed" access distribution. We choose θ to be 0.8 according to studies on real web traces.

4.2 Simulation Results

Packet Delivery Ratio Status

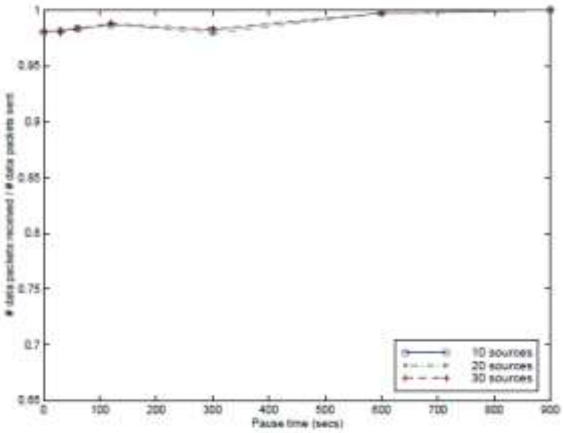

Fig.6 Packet Delivery Ratio in DSR

In Packet delivery ratio status we consider pass time and an other side we are sending an data packets and also data packets have been receiving has been shown in the above fig 6.
Figure 6 shows the fraction of the originated application data packets each protocol was able to deliver, as a function of both node mobility rate (pause time) and network load (number of sources). For DSR and AODV-LL, packet delivery ratio is independent of offered traffic load, with both protocols delivering between 95% and 100% of the packets in all cases.

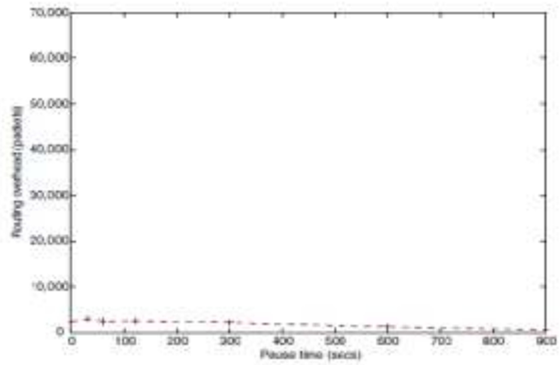

Fig 7 Routing Over Head in DSR(1 m/s)

Fig 7.shows how the data has been transmitted from ! node to the other nodes.To allow multiple types of DSR information to be combined together in a single packet, and to allow DSR information to be piggybacked on existing packets, we used a packet format for DSR modeled after the *extension header* and *option* format used by IPv6 [Deering 1998, Hinden 1996]. In particular, ROUTE REQUESTs, ROUTE REPLYs, and ROUTE ERRORs are each encoded as an option within either a Hop-by-Hop or End-to-End extension header, and a DSR source route on a packet is encoded as a separate extension header.

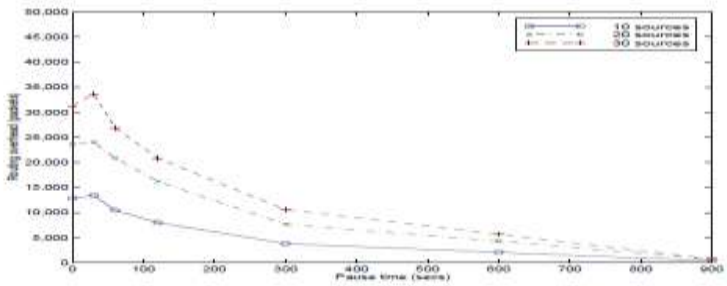

Fig 8. Routing overhead for DSR at pause time

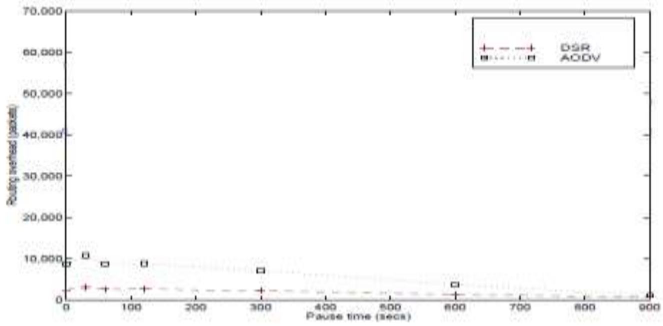

Fig 9. Comparison of AODV and DSR in Routing Over Head

Figure 9 shows the number of routing protocol packets sent by each protocol in obtaining the delivery ratios shown in Figure 4. DSR are plotted as scale , and AODV plotted on different scales to best show the effect of pause time and offered load on overhead. DSR, and AODV are on-demand routing protocols, so as the number of sources increases, we expect the number of routing packets sent to increase because there are more destinations to which the network must maintain working routes. DSR and AODV which use *only* on-demand packets and very similar basic mechanisms, have almost identically shaped curves. Both protocols exhibit the desirable property that the incremental cost of additional sources decreases as sources are added, since the protocol can use information learned from one route discovery to complete a subsequent route discovery. However, the absolute overhead required by DSR and AODV are very different, with AODV requiring about 5 times the overhead of DSR when there is constant node motion (pause time 0). This dramatic increase in AODVs overhead occurs because each of its route discoveries typically propagates to every node in the ad hoc network.

4.3 Performance Evaluation

From more elaborated simulation tests designed on DSR, we come to the following conclusions.

1. With the number of members in any cast group increasing, packet delivery ratio seems to increase.
2. As more number of source nodes means more routing discovery requests, the more source nodes requesting any cast services, the more routing overhead there will be in mobile ad hoc network. However, increasing the number of any cast group members is found to reduce the routing overhead significantly.
3. Increasing the number of any cast group member also can reduce the mean message hop count. So the greater the number of any cast group members, the better this protocol performs.

V. Conclusion

In this paper, we presented our design and implementation of cooperative cache in wireless P2P networks, and proposed solutions to find the best place to cache the data. For that one I used an DSR(Dynamic Source Routing).By using these technique I can say that packet routing to be trivially loop-free, avoids the need for up-to-date routing information in the intermediate nodes through which packets are forwarded, and allows nodes forwarding or overhearing packets to cache the routing information in them for their own future use. All aspects of the protocol operate entirely *on-demand*, allowing the routing packet overhead of DSR to scale *automatically* to only that needed to react to changes in the routes currently in use. The data has been accessed easily by the user. And also by using an data pipelines we can reduce the end-to-end delays between the server and client. My results show that the Dynamic Source Routing out performs the AODV in wireless P2P networks. Data request packets are transmitted to the cache layer on every node; however, the data reply packets are only transmitted to the cache layer on the intermediate nodes which need to cache the data. This solution not only reduces the overhead of copying data between the user space and the kernel space, but also allows data pipeline to reduce the end-to-end delay. we evaluate our design for a large-scale network through simulations developed a prototype to demonstrate the advantage of the asymmetric approach. Since our prototype is at a small scale, we scale network through simulations. Our simulation results show that the With the number of members in any cast group increasing, packet delivery ratio seems to increase. As more number of source nodes means more routing discovery requests, the more source nodes requesting any cast services, the more routing overhead there will be in mobile ad hoc network. However, increasing the number of any cast group members is found to reduce the routing overhead significantly.

Acknowledgement

We would to thank the anonymous referee for helpful comments.

References

[1]. R. Agu¨ ero and J.P. Campo, "Adding Multiple Interface Support in NS-2," Jan. 2007.
[2]. G. Cao, L. Yin, and C. Das, "Cooperative Cache-Based Data Access in Ad Hoc Networks," Computer, vol. 37, no. 2, pp. 32-39, Feb. 2004.
[3]. W. Lau, M. Kumar, and S. Venkatesh, "A Cooperative Cache Architecture in Supporting Caching Multimedia Objects in MANETs," Proc. Fifth Int'l Workshop Wireless Mobile Multimedia, 2002.
[4]. B. Barr, J. Bicket, D. Dantas, B. Du, T. Kim, B. Zhou, and E. Sirer, "On the Need for System-Level Suppport for Ad Hoc and Sensor Networks," ACM Operating System Rev., vol. 36, no. 2, pp. 1-5, Apr. 2002.
[5]. E. Royer and C. Perkins, "An Implemenatation Study of the
[6]. AODV Routing Protocol," Proc. IEEE Wireless Comm. and
[7]. Networking Conf., 2000.
[8]. J. Bicket, D. Aguayo, S. Biswas, and R. Morris, "Architecture and Evaluation of an Unplanned 802.11b Mesh Network," Proc. ACM MobiCom, 2005.
[9]. L. Breslau, P. Cao, L. Fan, G. Phillips, and S. Shenker, "Web Caching and Zipf-like Distributions: Evidence and Implications," Proc. IEEE INFOCOM, 1999.
[10]. M. Cieslak, D. Foster, G. Tiwana, and R. Wilson, "Web Cache Coordination Protocol v2.0," IETF Internet Draft, 2000.
[11]. S. Desilva and S. Das, "Experimental Evaluation of a Wireless Ad Hoc Network," Proc. Ninth Int'l Conf. Computer Comm. And Networks, 2000.
[12]. L. Yin and G. Cao, "Supporting Cooperative Caching in Ad Hoc Networks," IEEE Trans. Mobile Computing, vol. 5, no. 1, pp. 77-89, Jan. 2006.
[13]. L. Fan, P. Cao, J. Almeida, and A. Broder, "Summary Cache: A Scalable Wide Area Web CAche Sharing Protocol," Proc. ACM SIGCOMM, pp. 254-265, 1998.
[14]. Z. Fu, P. Zerfos, H. Luo, S. Lu, L. Zhang, and M. Gerla, "The Impact of Multihop Wireless Channel on TCP Throughput and Loss," Proc. IEEE INFOCOM, 2003.
[15]. IEEE, Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Spec, IEEE 802.11 Standard, 1999.
[16]. H. Eriksson, "MBONE: The Multicast Backbone," Comm. ACM, vol. 37, no. 8, pp. 54-60, 1994.
[17]. Lin. C. H and Sheu. T. L, "A Modified AODV with path Traffic Rerouting for TCP Performance Improvement in Ad Hoc Wireless Networks", for the dissertation paper of National
[18]. Sun Yat-sen University, 2004.
[19]. Tamilarasi. M, Palanivelu. T. G, "A strategy to reduce the control packet load of MANETs with bidirectional links using DSR", IEEE International Conference on information
[20]. Technology, ICIT-2006, Orissa, India.
[21]. Charles Perkins. Ad Hoc On Demand Distance Vector (AODV) routing. Internet-Draft, draft-ietf-manet-aodv-00.txt, November 1997. Work in progress.
[22]. Charles E. Perkins and Pravin Bhagwat. Highly dynamic Destination- Sequenced Distance-Vector routing (DSDV) for mobile computers. In *Proceedings of the SIGCOMM '94 Conference on Communications Architectures, Protocols and Applications*, pages 234–244, August 1994. A revised version of the paper is available from
[23]. http://www.cs.umd.edu/projects/mcml/papers/Sigcomm94.ps.
[24]. David C. Plummer. An Ethernet address resolution protocol: Or converting network protocol addresses to 48.bit Ethernet addresses for transmission on Ethernet hardware. RFC 826, November 1982.
[25]. Theodore S. Rappaport. *Wireless Communications: Principles and Practice*. Prentice Hall, New Jersey, 1996.
[26]. Neil Siegel, Dave Hall, Clint Walker, and Rene Rubio. The Tactical Internet Graybeard Panel briefings. U.S. Army Digitization Office. Available at http://www.ado.army.mil/Briefings/Tact%20Internet/index.htm,
[27]. October 1997.
[28]. Bruce Tuch. Development of WaveLAN, an ISM band wireless LAN. *AT&T Technical Journal*, 72(4):27–33, July/August 1993.
[29]. Gary R. Wright and W. Richard Stevens. *TCP/IP Illustrated, Volume 2: The Implementation*. Addison-Wesley, Reading, Massachusetts, 1995.

Short Bio Data About The Author

Mr. B sateesh Kumar Reddy his B.Tech in Computer science and Engineering from Sri Kottam Tulasi Reddy Memorial College Of Eng, JNTU, Hyderabad and Pursuing M.Tech in Computer science (Software Engineering) from Aurora's Technological And Research Institute, JNTU, Hyderabad.

Mr. V. Satish working as Assistant Professor in the Department of Computer Science and Engineering. Aurora's Technological And Research Institute College with a teaching experience of 5years. He has received his M.Tech in Computer science (Software Engineering).His areas of interest includes Distributed Databases, Computer Networks, Mobile Computing, Information Security and Mobile Commerce.

Mrs. D. Sujatha working as Associate Professor and Head of the Department of Computer Science and Engineering at Aurora's Technological And Research institute wih a teaching experience of 12years. She has received her M.Tech in Computer Science and Pursuing his PhD in Data Stream Mining at JNTU, Hyderabad. His areas of interests include data structures, principles of programming languages, algorithm analysis and compiler design