ISSN (e): 2250 – 3005 || Vol, 04 || Issue, 5 || May – 2014 ||
International Journal of Computational Engineering Research (IJCER)

Efficient Multicast Algorithms for Wireless Mesh Networks

Dr.Sivaagorasakthivelmurugan[1], P.Ravi Shankar[2]
Associate Professor[1], Assistant Professor[2]
Department of EEE[1], Department of EEE[2],
SreeSakthi Engineering College, Nehru Institute of Engineering and Technology,
Coimbatore

ABSTRACT

Wireless Mesh Networks (WMN) are promising and emerging technology for providing low cost and high quality internet service to the end user. The Lookupalgorithm (LU) in wireless infrastructure enables the possibility of ID assignment of peers and 1-hop broadcast between peers through cross-layering technique. Thus message overhead reduces and increase information retrieval performance. Channel assignment (CA) algorithm builds efficient multilevel trees and reduces the number of relay nodes and hop distances of the trees. The algorithms use dedicated channel assignment strategies to reduce the interference to improve the network capacity. The result of our study enables efficient resource sharing and best throughput is performed in wireless mesh network.

Keywords – *Wireless Mesh Network, Cross-layeringtechnique, Channel Assignment.*

I. INTRODUCTION

Wireless mesh network is the next generation technology for providing low cost and efficient internet access to the end users. Mesh networks are classified by the use of multiple channels and multiple interfaces. Wireless mesh networks come in a range of architectures and functional components. The wireless mesh networks are comprised into, classification of mesh networks, communication technologies for mesh networks and the key differences between mesh networks and traditional wireless multi-hop networks. Mesh networks comprise of three types of nodes: Mesh Routers (or Access Points - APs), Mesh Clients and Gateway Routersgateway routers at the top border provide wired connectivity to the Internet. The mesh routers at the other border act as access points for mesh clients and user networks.Peer-to-peer systems and applications are distributed systems without any centralized control or hierarchical organization, where the software running at each node is equivalent in functionality. A review of the features of recent peer-to-peer applications yields a long list: redundant storage, permanence, selection of nearby servers, anonymity, search, authentication, and hierarchical naming. Despite this rich set of features, the core operation in most peer-to-peer systems is efficient location of data items. The contribution of this paper is a scalable protocol for lookup in a dynamicpeer-to-peer system with frequent node arrivals and departures.

Three features that distinguish Lookup algorithm from many other lookup protocols are its simplicity, provable correctness, and provable performance. Chord is simple, routing a key through a sequence of other nodes toward thedestination. A node requires information about other nodes for efficientrouting, but performance degrades gracefully when that informationis out of date. This is important in practice because nodes willjoin and leave arbitrarily, and consistency of even statemay be hard to maintain. Only one piece information per node needto be correct in order for Chord to guarantee correctrouting of queries; lookup program has a simple algorithm for maintainingthis information in a dynamic environment. Traditional casting protocols for wireless networksassume that each node is equipped with one interface. Amesh network provides the nodes with multiple interfacesthat can be used to improve the throughput substantially.However, channel assignment is subject to the number ofavailable channels and interfaces, the network topology,the communication requests, and other factors. Interferencecannot be completely eliminated due to the limited numberof available channels. An inappropriate channel assignmentstrategy will result in throughput reduction due tothe multichannel hidden terminal problem [3], disconnectionof the topology [4], or unfair bandwidth allocation tovarious users [5].

The rest of this paper is structured as follows. Section 2 describes the design model and its function Section 3 consists of distributed lookup protocol. Section 4 presents the functions of channel assignment protocol, Section5 demonstration aboutalgorithms performance throughsimulation and experiments on a deployed prototype. Finally, we describesoutlinefor future work in Section 6 and summarize our contributionsin Section 7.

II. SYSTEM MODEL

A. Introduction

THE design simplifies the systems based on by addressing many problemsandby introducing basic techniquefor channel assignment algorithms followed by design considerations for CA algorithms in WMNs.

B. Design Consideration

The design simplifies the design of mesh systems andapplications based on it by addressing these difficult problems:

Load balance: Chord acts as a distributed hash function, spreading keys evenly over the nodes; this provides a degreeof natural load balance.

Decentralization: Chord is fully distributed: no node is more important than any other. This improves robustness and makes Chord appropriate for loosely-organized peer-to-peer applications.
Scalability: The cost of a Chord lookup grows as the log of the number of nodes, so even very large systems are feasible. No parameter tuning is required to achieve this scaling.

The following are examples of applications for lookup provide a good foundation.
Cooperative Mirroring, as outlined in a recent proposal [6].Imagine a set of software developers, each of whom wishes to publish a distribution. Demand for each distribution might vary dramatically, from very popular just after a new release to relatively unpopular between releases. An efficient approach for this would be for the developers to cooperativelymirror each others' distributions. Ideally, the mirroring system would balance the load across all servers, replicate and cache the data, and ensure authenticity. Such a system should be fully decentralized in the interests of reliability, and because there is no natural central administration.
Time-Shared Storage for nodes with intermittent connectivity. If a person wishes some data to be always available, but is only occasionally available, they can offer to store others' data while they are up, in return for having their data stored elsewhere when they are down. The data's name can serve as a key to identify the (live) Chord node responsible for storing the data item at any given time. Many of the same issues arise as in the Cooperative Mirroring application, though the focus here is on availability rather than load balance.To improve the throughput of WMNs, many studies have been conducted on how to assign orthogonal channels to adjacent wireless links to minimize interference. It is known that 802:11b=g and 802:11a provide 3 and 12 nonoverlapping channels, respectively. Although 802:11a provides more nonoverlapping channels than 802:11b=g, it hasseveral drawbacks. Because 802:11a works on a higherfrequency spectrum (5 GHz) than 802:11b=g (2 GHz), it ismore difficult to penetrate walls and other obstructions, andthus 802:11a has a shorter range.Through experiments, we observe that the interferencebetween two links depends on both their physical distanceand channel separation [6]. Unlike the traditional interferencemodel, the interference range is no longer a constant.Instead, it varies with the channel separation. Let Ic be theinterference range of two links with channel separation c.That means, when the channel separation of two links is c,they will interfere with each other if their distance is less thanIc, and otherwise not. For example, I0 ¼ 2R, which means thesame channel can be used on two links without anyinterference only when they are over twice the transmissionrange away.

III. LOOKUP ALGORITHM

The Lookup protocol specifies how to find the locations of keys,new nodes join the system, and how to recover from the failure (or planned departure) of existing nodes. This section describes simplified version of the protocol that does not handle concurrent joins or failures. Section 5 describes enhancements to the base protocol to handle concurrent joins and failures.

A. Hashing Function

Consistent hashing assigns keys to nodes as follows. Identifiersare ordered in an *identifier circle with m* modulo Key is assigned to the first node whose identifier is equal to or follows in the identifier space. This node is called the *successor node* of key denoted by *successor* If identifiers are represented as a circle of number is the first node clockwise from Figure 2 shows an identifier circle with the circle has three nodes: 0, 1, and 3. The successor of identifier 1 is node 1, so key 1 would be located at node 1. Similarly, key 2 would be located at node 3, and key 6 at node 0. Consistent hashing is designed to let nodes enter and leave the network with minimal disruption. To maintain the consistent hashing mapping when a nodejoins the network, certain keys previously assigned to successor now become assigned to leaves the network, all of its assigned keys are reassigned to successor. No other changes in assignment of keys to nodesneed occur. In the example above, if a node were to join with identifier7, it would capture the key with identifier 6 from the nodewith identifier 0.The following results are proven in the papers that introducedconsistent hashing [11, 13].

Figure 1: an identifier circle consisting of the three nodes 0, 1, and 3

A very small amount of routing information suffices to implementconsistent hashing in a distributed environment. Each node need only be aware of its successor node on the circle. Queries for a given identifier can be passed around the circle via these successor pointers until they first encounter a node that succeeds the identifier; this is the node the query maps to. A portion of the Chord protocol maintains these successor pointers, thus ensuring that all lookups are resolved correctly. However, this resolution scheme is inefficient: it may require traversing all N nodes to find the appropriate mapping. To accelerate this process, Chord maintainsadditional routing information. This additional information is notessential for correctness, which is achieved as long as the successorinformation is maintained correctly. As before, let m be the number of bits in the key/node identifiers. Each node, maintains a routing table with mentries, called the *finger table*.

The pseudo code that implements the search process is shown inFigure 2. The notation *n.foo()* stands for the function *foo()* beinginvoked at and executed on node n Remote calls and variable references are preceded by the remote node identifier, while local variable references and procedure calls omit the local node. Thus *n.foo()* denotes a remote procedure call on node, while *n.bar*, without parentheses, is an RPC to lookup a variable *bar* on node *find successor* works by finding the immediate predecessor node of the desired identifier; the successor of that node must be the successor of the identifier. We implement *find predecessor* explicitly,because it is used later to implement the joint operation. In a dynamic network, nodes can join (and leave) at any time.The main challenge in implementing these operations is preservingthe ability to locate every key in the network. To achieve this goal,Chord needs to preserve two invariants:

1. Each node's successor is correctly maintained. For every key K, node successor(k) is responsible for k. In order for lookups to be fast, it is also desirable for the fingertables to be correct To preserve the invariants stated above, Chord must perform three tasks when a node joins the network:

1. Initialize the predecessor and fingers of node n.
2. Update the fingers and predecessors of existing nodes to reflectthe addition of n.
3. Notify the higher layer software so that it can transfer state associated with keys that nodeis now responsible.

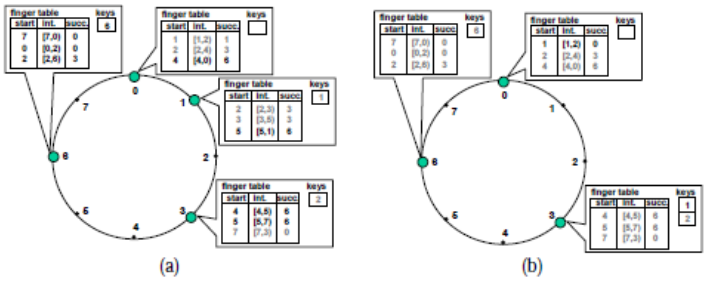

Figure 2: (a) Finger tables and key locations after node 6 joins. (b) Finger tables and key locations after node 3 leaves.

IV. CHANNEL ASSIGNMENT ALGORITHM

Routing protocols do exist to offer efficient multicasting service for conventional multihop wireless networks, such as MANETs and wireless sensor networks. Since the nodes become increasingly mobile or the networks only have scarce resources such as power constraints and limited computing ability, most previous work pays much attention to energy efficiency and how to build the multicast structure without knowing the global topology. As a result, the multicast structure should be distributable constructed, energy efficient, and should take care of the topology change as well as group member management, which may conflict with maximizing the throughput of the network to some extent.However, since mesh networks are deployed to provide last-mile Internet access for enterprises or communities, the throughput and the network capacity are the

major concerns. Deployed at fixed locations, mesh routers have limited mobility. Furthermore, they are computationally powerful and do not rely on battery power compared with their counterparts in MANETs or sensor networks, which help to achieve sufficient network capacity to meet the requirement of applications such as audio or video sharing among end users. Thus, we need to create a multicast structure that aims to deliver the packets rapidly to the multireceivers(multireceivers are defined as the multicast group members except for the source node) without worrying about the energy consumption and topology changes.

A common method for multicast is to build a multicast tree,where the source node is usually the gateway. In this paper, we first propose the LCA algorithm, which can be achieved by the following steps. First, the nodes obtain their level information. The BFS is used to traverse the whole network. All the nodes are partitioned into different levels according to the hop count distances between the source and the nodes.

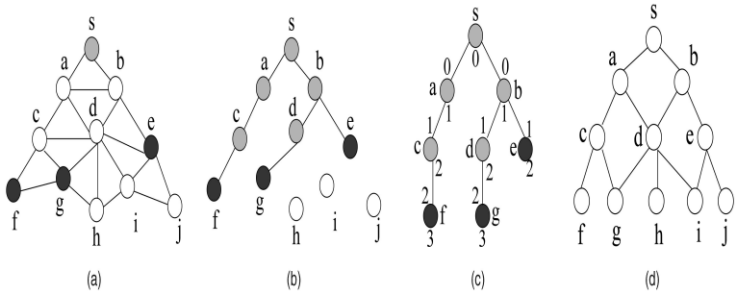

Figure 3: An example for CA and tree mesh. (a) Network topology, (b) multicast tree, (c) channel assignment, and (d) tree mesh.

Figs. 3a and 3d give an example of the original network channel at the same level. For example, in Fig. 3c, since g is in the transmission range of both c and d, there will be interference when c and d use the same channel. Second, when the number of available channels is more than that of the levels, some channels will not be utilized, which is a waste of channel diversity. Third, the channel assignment does not take the overlap property of the two adjacent channels into account. As we know channel i and channel i+1 are adjacent in frequency, so they partially interfere with each other. Thus, the channel i for level i still has someinference effect with the channel i+1 for level i+1.

V. SIMULATION RESULTS

A. Delay Comparison

Total number of network-level packets exchange by no. of nodes to maintain the overlay, and to resolve the queries. the delay is the average time it takes for a packet to reach the destination after it leaves the source.

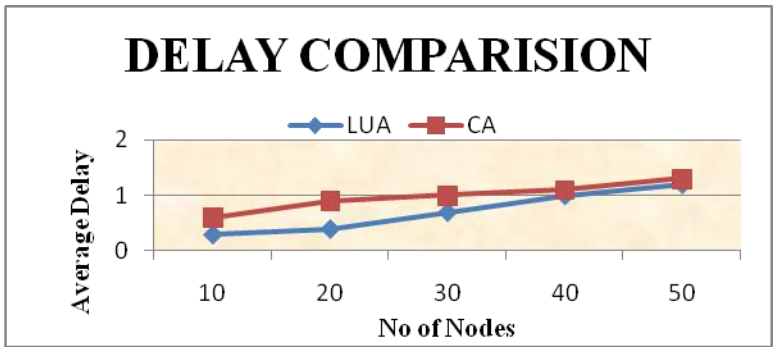

Figure 4: Delay Comparison

From figure 4 shows the comparison of average delay between the number of nodes and the performance of lookup algorithm and channel assignment algorithm where the delayin channel assignment is large compare to the lookup algorithm is less.

B. Average Response Time

Percentage of queries which are successfully resolved; a query on key k is successfully resolved if the ip address of the peer responsible for key k is returned to the peer which issued the query.

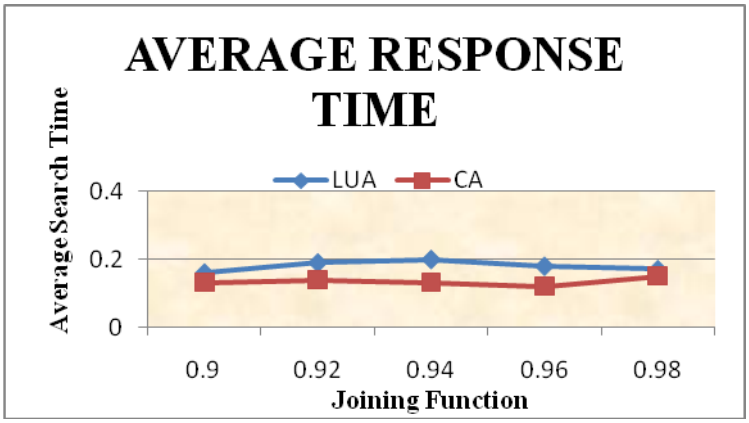

Figure 5: Average Response Time

The above Figure 5 shows the comparison between average search time and joining function of the nodes. It compares the successful queries of the nodes between the LUA algorithm and CA algorithm lookup algorithm will outperform more loss when compare to the channel assignment algorithm.

C. Throughput Comparison

The throughput is the average number of packets each multireceiver receives during a time unit.

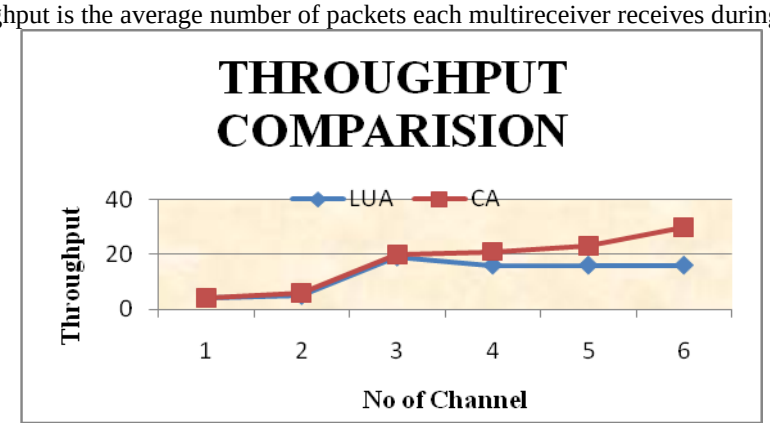

Figure 6: Throughput Comparison

The above figure6 shows the comparison of throughput between the number of channels from the comparison it is clearly knows that the channel increases from the number of nodes channel assignment algorithm performs high throughput when compare to lookup algorithm.

D. Average Query Response Time

For successful queries, the time elapsed between the instant the query is issued by node n and the instant answer is received at the node n.

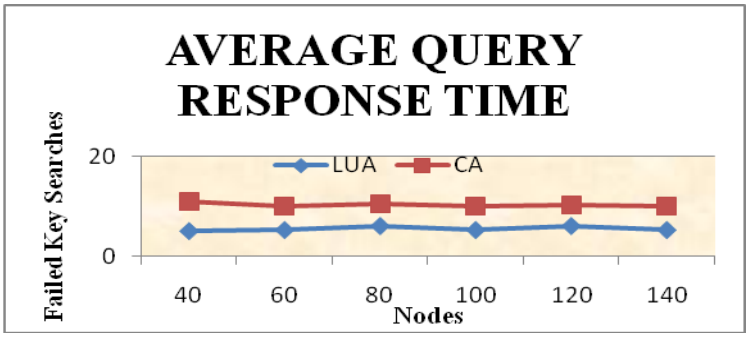

Figure 7: Average Query Response Time

Figure7 we consider number of nodes and its successful query rates of the nodes during the joining functions in the channel assignment between more number of nodes.

VI. CONCLUSION

In this paper, we carefully evaluated the performance of lookup algorithm and channel assignment algorithm through NS2(network simulator-2). The proposed algorithm utilized for implementing file/resource sharing application and the dedicated channel assignment helps to further reduce the interference in wireless mesh networks. Although our performance evaluation shows the outperform of overlay maintenance and single channel assignment in terms of throughput delay.

In future further improve the message overhead and to improve throughput and reduce delay in the multi channel multi interfaces.

REFERENCES

[1] Al Hamra, C. Barakat, and T. Turletti, "Network Coding forWireless Mesh Networks: A Case Study," Proc. IEEE Int'l Symp.World of Wireless, Mobile and Multimedia (WoWMoM), 2006.346 IEEE TRANSACTIONS ON MOBILE COMPUTING, VOL. 9, NO. 3, MARCH 2010

[2] M. Caesar, M. Castro, E.B. Nightingale, G. O'Shea, and A.Rowstron, "Virtual Ring Routing: Network Routing Inspired byDHTs," Proc. ACM SIGCOMM, pp. 351-362, 2006.

[3] Canali, M.E. Renda, and P. Santi, "Evaluating Load Balancingin Peer-to-Peer Resource Sharing Algorithms for Wireless MeshNetworks," Proc. IEEE Workshop Enabling Technologies and Standardsfor Wireless Mesh Networksing (MeshTech), pp. 603-609, 2008.

[4] M. Conti, E. Gregori, and G. Turi, "A Cross-Layer Optimization ofGnutella for Mobile Ad Hoc Networks," Proc. ACM MobiHoc, May2005.

[5] Cramer and T. Fuhrmann, "Performance Evaluation of Chordin Mobile Ad Hoc Networks," Proc. ACM Int'l WorkshopDecentralized Resource Sharing in Mobile Computing and Networking(MobiShare), pp. 48-53, 2006.

[6] M. Denny, M. Franklin, P. Castro, and A. Purakayastha, "Mobiscope:A Scalable Spatial Discovery Service for Mobile NetworkResources," Proc. Int'l Conf. Mobile Data Management (MDM), 2003.

[7] P. Desnoyers, D. Ganesan, and P. Shenoy, "TSAR: A Two TierSensor Storage Architecture Using Interval Skip Graphs," Proc.ACM Conf. Embedded Networked Sensor Systems (SenSys), Nov. 2005.

[8] T. Fuhrmann, "Scalable Routing for Networked Sensors andActuators," Proc. IEEE Comm. Soc. Conf. Sensor and Ad Hoc Comm.Networks (SECON), pp. 240-251, 2005.

[9] K. Ramachandran, E.M. Belding, K. Almeroth, and M. Buddhiko,"Interference-Aware Channel Assignment in Multi-Radio WirelessMesh Networks," Proc. IEEE INFOCOM, 2006.

[10] J. Tang, G. Xue, and W. Zhang, "Maximum Throughput and FairBandwidth Allocation in Multi-Channel Wireless Mesh Networks,"Proc. IEEE INFOCOM, 2006.

[11] Mishra, V. Shrivastava, and S. Banerjee, "Partially OverlappedChannels Not Considered Harmful," Proc. ACM SIGMETRICS/Performance, 2006.

[12] P. Li, N. Scalabrino, Y. Fang, E. Gregori, and I. Chlamtac,"Channel Interference in IEEE 802.11b Systems," Proc. IEEE GlobalTelecomm. Conf. (GLOBECOM), 2007.

[13] T.H. Cormen, C.E. Leiserson, R.L. Rivest, and C. Stein, Instroductionto Algorithms. The MIT Press, 2001.

[14] http://www.isi.edu/nsnam/ns/index.html, 2009.

[15] Royer and C. Perkins, "Multicast Operation of the Ad-Hoc onDemand Distance Vector Routing Protocol," Proc. ACM MobiCom,1999.