IMPLEMENTING P2P RESOURCE SHARING APPLICATIONS IN WIRELESS MESH NETWORKS

Dr.N.Kumareshan[1], Mr.N.Prakash[2]
[2]Assistant Professor (SG), [2] Assistant Professor
[1,2]Department of ECE,
[1]Sri Shakthi Institute of Engineering and Technology, [2] kalaignar karunanidhi institute of technology
Coimbatore, TamilNadu, India
kumareshan.ece@gmail.com

Abstract - Wireless mesh networks are a promising area for the deployment of new wireless communication and networking technologies. In this paper, we present a peer-to-peer file sharing platform combined with Grid topology for wireless devices. Facing the limit of network bandwidth and unstable signal strength in wireless network, wireless peer-to-peer architecture is now widely accepted as a new computing paradigm for wireless devices. On the other hand, the Peer-to Peer (P2P) paradigm emerged as a successful model that achieves scalability in distributed systems. Starting from the well-known Chord protocol for resource sharing in wired networks, we propose a specialization (called MESHCHORD) that accounts for peculiar features of wireless mesh networks: namely, the availability of a wireless infrastructure, and the 1-hop broadcast nature of wireless communication, which bring to the notions of location awareness and MAC layer cross-layering. Through extensive packet level simulations, we investigate the separate effects of location awareness and MAC layer cross-layering, and of their combination, on the performance of the P2P application.MESHCHORD outperforms in the parameters of received packets, packet delivery ratio (PDR), Delay and Energy when compared to Chord protocol.NS2 network simulator is used for simulation and performance metrics of throughput, packet delivery ratio, delay and energy are being discussed.

Keywords – Wireless Mesh networks, chord protocol.

1. INTRODUCTION

Wireless mesh networks are a promising technology for providing low-cost Internet access to wide areas (entire cities or rural areas), and to enable the creation of new type of applications and services for clients accessing the network. Differently from other types of wireless multi-hop networks, wireless mesh networks are composed of two types of nodes: mostly stationary wireless access points (routers), and mobile wireless clients. Routers are connected to each other through wireless links, and provide a wireless access infrastructure to wireless clients. Some of the routers are connected to the Internet via wired links, and act as gateways for the other routers and for the clients.

Wireless community networks can be used to share the cost of broadband Internet access, but also to realize innovative services for the community, such as sharing of community-related resources, live broadcast of local events, distributed backup systems, and so on. As the above mentioned innovative applications suggest, peer-to-peer resource sharing is expected to play an important role in forthcoming wireless networks based on the mesh technology. This project investigates the feasibility of the well-known Chord algorithm for peer-to-peer resource sharing in wired networks in a wireless mesh network environment. MESHCHORD-that accounts for peculiar features of mesh networks: namely, 1) the availability of a wireless infrastructure, which enables location-aware ID assignment to peers and 2) the 1-hop broadcast nature of wireless communications, which is exploited through a cross-layering technique that bridges the MAC to the overlay layer.

We compare the performance of MESHCHORD to a chord. It leads to a significant increase in the successful packet delivery ratio and a clear decrease in the average delivery delay compared to chord.

2. RELATED WORK

Several Distributed Hash Table approaches have been proposed in the literature to address the problem of realizing distributed peer-to-peer resource sharing. The various DHT approaches proposed in the literature mainly differ on the structure imposed to the virtual overlay and on the mechanism used to route search requests in the overlay. Among them, we cite Chord [6] (which we briefly describe in the next section). However, these DHT approaches have been designed and optimized for operation in wired networks, and issues such as limited bandwidth, node mobility, and so on, are not relevant.

Wireless mesh networks present a potential advantage with respects to MANETs for a successful realization of scalable P2P approaches, namely the presence of a stationary, wireless infrastructure that can be used by mobile clients to communicate with each other (or to access the Internet). In [1], the authors evaluate the gain that can be obtained from using network coding when running file sharing applications in wireless mesh networks, and conclude that some gain can actually be achieved, although not as much as in a wired network. In [6], some of the authors of this paper introduced a two-tier architecture for file sharing in wireless mesh networks: the lower tier is composed of mobile mesh clients, which provide the content (files/resources to share) to the P2P overlay; the upper tier is composed of stationary mesh routers, and implements a distributed hash table for locating resources within the network.

The investigation presented in this paper complements our work [6] under many respects. While the emphasis in [6] was on the procedures for dealing with client mobility at the lower tier of the architecture, in this paper we are concerned with the efficiency of DHT implementation in the upper tier of the architecture. Another major difference with respect to [6] is that we consider Chord instead of Viceroy for implementing the DHT, and that we perform packet-level simulations to investigate the performance of Chord and of our proposed specialization MESHCHORD in realistic wireless mesh network scenarios. The simulations carried out in the present paper account also for the (considerable) message exchange needed to maintain the Chord overlay, and for dynamic join/leave of nodes at the upper tier of the architecture, performed assuming static topology of the upper tier (no mesh router joining/leaving the network during the simulated time interval), which ignored maintenance overhead. As the results presented in this paper show, the overhead for overlay maintenance is considerable and, in some cases, can lead to network congestion. Hence, results obtained from high level simulations that ignore maintenance overhead and do not implement the MAC layer can be very inaccurate (this applies not only to the results presented in [6], but also to most of simulation based results presented in the literature). Finally, differently from [6], we investigate also the performance of information retrieval in terms of percentage of successful queries and query response time.

Finally we consider three different routing protocols, and show that Chord performance is only marginally influenced by the choice of the routing protocol. The main finding of [3] is that node mobility considerably impairs Chord consistency, with a dramatic effect on information retrieval performance: in presence of even moderate mobility, the percentage of successful queries can drop below 10%.

The main difference between our study and the one reported in [3] is that we target mesh networks instead of MANETs, and that we also study a location-aware, cross layer specialization of Chord for this type of networks. Contrary to what reported in [3], our analysis shows that Chord, and in particular our proposed specialization MESHCHORD, can successfully be applied in wireless mesh network scenarios.

3. MESHCHORD

In this section, shortly describe the two-tier architecture used in our design, and the basic Chord operations. Then describe in details MESHCHORD, our proposed specialization of Chord for wireless mesh networks.

3.1 Network architecture

In this project assuming a two-tier architecture for file/resource sharing: the lower tier of the architecture is composed of (possibly) mobile mesh clients (clients for short), which provide the content to be shared in the P2P system; the upper tier of the architecture is composed of stationary mesh routers (routers for short), which implement a DHT used to locate file/resources within the network. Unless otherwise stated, in the following using the term peer to refer to a router forming the DHT at the upper tier of the architecture. Assume routers are stationary, but they can be switched on/off during network lifetime. This is to account for situations that might arise in some mesh network application scenarios (e.g., community networking), in which routers might be managed by users and occasionally shut down. Also, changes in the upper tier topology might be caused by failures of some router.

International Journal of Scientific Engineering and Applied Science (IJSEAS) – Volume-3, Issue-8, August 2017
ISSN: 2395-3470
www.ijseas.com

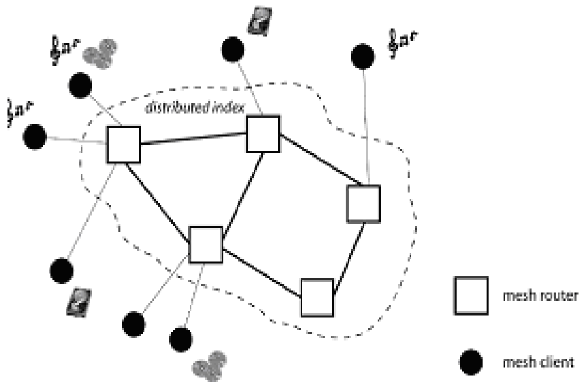

Fig. 1. Two-tier architecture assumed in our design

Table 1:Simulation Parameters	
Parameter	Value
Network Size	1000 by 1000
Nodes	18
Protocol	MESHCHORD
Propagation Model	Two Ray Ground model
Link data rate	11 Mbps
Radio technology	802.11
Application layer	CBR
Transport layer	UDP/TCP
Simulation time	30 secs
Initial energy	100 Joules

When a client wants to find a certain resource, it sends to its responsible router (a mesh router within its transmission range) a Find Key message, containing the key (unique ID) of the resource to find (see next section for details on key assignment to node/resources). The responsible router forwards the resource request in the DHT overlay according to the rules the rules specified by the Chord protocol, until the resource query can be answered. In case of successful query resolution, a message containing the IP address of the client holding the requested file/resource is returned to client through its responsible router.

3.2 Basic Chord operations

The DHT approach investigated in this project is Chord [6].

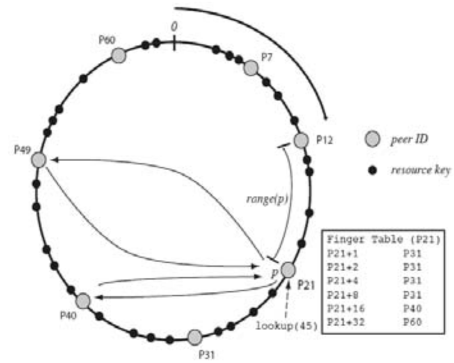

Fig. 2. Basic Chord operations. m is set to 6

Chord is based on the idea of mapping both peer (mesh router) IDs and resource IDs (keys) into the same ID space, namely the unit ring. Each key resides on the peer with the smallest ID larger than the key, i.e., peer p manages keys comprised between its own ID and the ID of the predecessor of p in the unit ring (denoted range (p)). Chord maps peer and resource IDs into the unit ring using a hashing function, which has the property of uniformly distributing IDs in [0, 1]. Indeed, IDs in Chord are represented through m-bit numbers, i.e., at most 2m distinct (peer or resource) IDs are present in the Chord system.

When a lookup (k) operation is invoked at peer p and the operation cannot be resolved locally (because k is not within range (p)), a message is sent to the peer p0 with largest ID < k in the finger table of node p. If p0 cannot resolve the lookup operation, it replies to peer p with a message containing the ID of the peer p00 with largest ID < k in its own finger table. Peer p then forwards the request to peer p00, and so on, until the lookup operation can be resolved (in at most steps) 3. A lookup operation for key 45 issued at node P21 is first forwarded to node P40, and then to node P49, which is responsible for the key and can resolve the lookup.

To deal with dynamic join/leaves of peers in the systems, the following procedures are implemented. When a new peer p joins the network, it first needs to initialize its to any peer currently joining the predecessor and finger table. This is done by sending requests network peer p is aware of (called hook peer). Then, the finger tables and predecessor pointers of currently active peers must be updated to account for the new peer joining the network. Finally, peer p must contact its successor s in the ring so that the key range previously managed by s can be split with p. In case no (active) hook peer can be found, the join operation fails, and the peer cannot take join the Chord overlay. When an existing peer p leaves the network, it first informs its predecessor and successor in the ring about its intention of leaving the network, so that they can change their finger tables and predecessor pointers accordingly; then, peer p transfers to its successor the key range it is responsible for. Finally, mention that, in order to deal with dynamic network conditions, each active peer in the network periodically performs a Stabilize operation, which verifies and possibly updates the content of the finger table and predecessor pointer. The period between consecutive Stabilize operations is a critical parameter in the Chord design: if the period is relatively short, the network is more reactive, but a higher message overhead is generated; on the other hand, a longer stabilize period reduces message overhead, at the expense of having a less reactive network.

4. SIMULATION SETUP

The simulations were conducted in Network simulator NS2 with version 2.31, with a deployment area of 1000m by 1000m. In this area, 18 nodes are placed as shown in Fig.3, with the separation between neighboring nodes along both axes being 100 m. Note that we use grids as deployments in this project to emulate uniformly dense deployments and such grids are not a requirement of our algorithms. As long as the neighborhood relationships are similar, the results will not differ significantly from those presented in this project.

In this simulations verified that, in case of grid topology, setting the transmission range to 200m (i.e., twice the node separation) is the best choice for reducing Chord overhead. For uniformity, we have used the same transmission range also in the case of random uniform topology. In our project we are considering 18 nodes for peer to peer resource sharing in wireless mesh networks, remaining 9 nodes acts as a client and other 9 nodes acts as a peer.

In the following, we present the results of four different sets of simulations, focusing on the effect of

1. Increasing the number n of peers,
2. changing the number of join/leave events in the simulated time interval,
3. Changing the query rate, and
4. Different types of background traffic, on MESHCHORD Performance.

For all the sets of experiments, the simulated time interval was 3,800 seconds, where the first 200 seconds were used to incrementally add peers to Chord, and to stabilize the overlay; all the simulation results presented in the following refer to data gathered in the last 3,600 seconds of the simulation, and are averaged over 10 (50) runs in case of grid (uniform random) topology. The largest sample set in case of random topologies was needed to account for the higher degree of randomness (which also plays a role in determining the network topology) in this setting. The simulation parameters are summarized in Table 1.

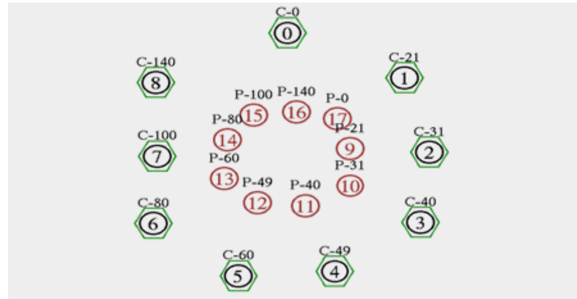

Fig.3.Simulation scenario

In this project comparing MESHCHORD to an enhanced version of chord that we implemented, that is, mesh chord. This fig 3 shows the simulation scenario for mesh chord wireless mesh networks.

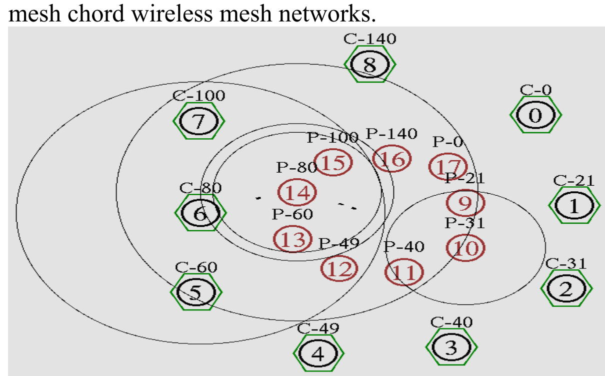

Fig.4.MESHCHORD Routing View in NS2 Simulator Environment

The MESHCHORD Routing View is shown in the above figure as in NS2 environments as we considering 18 nodes.

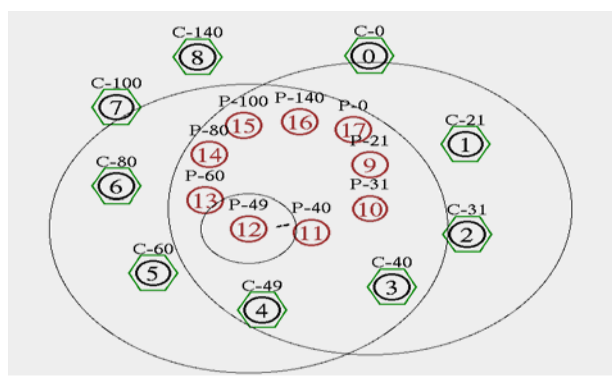

Figure.5.Resource sharing from p-p

The above fig shows about peer to peer resource sharing in Wireless mesh networks. Here peer value 31 sending files/resource to peer 49.

4.1 Experimental Results

This section presents latency measurements obtained from a prototype implementation of Mesh Chord.

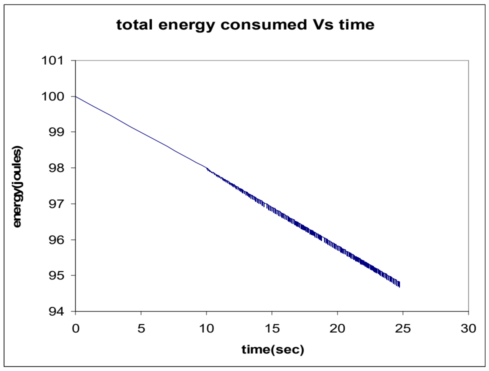

Fig.6.Total energy consumed

According to simulation the nodes consumes different energies, in this scenario the graph is plotted between nodes versus total energy consumed by each nodes. Here mesh chord simulation is considered for calculating total energy consumed for various nodes.

According to simulation the MESHCHORD Routing took lesser amount of energy in transmitting packets compared to chord. This QOS parameter is taken from the energies consumed by each node.

The Energy QOS parameter is being considered for the entire network also this QOS is being calculated based upon the energy consumption by each node. So that it does not consider about the prioritization of network in energy QOS parameter.

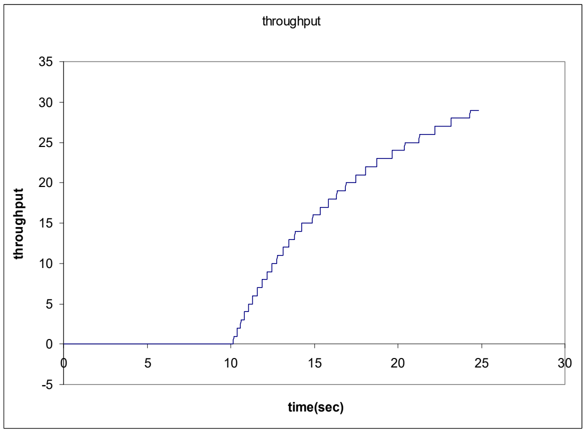

Fig.7.Throughput Performance

Throughput is being plotted between time Versus received packets at receiver.MESHCHORD performance gain with respect to Chord is much more evident in presence of background traffic: contrary to chord, MESHCHORD is able to provide satisfactory performance of the P2P overlay also in presence of background traffic, while in turn increasing the performance of the applications contributing to the background traffic (e.g., increasing the average throughput of TCP flows). In general, MESHCHORD performance appears to be relatively resilient to the presence of background traffic (as long as the network is not congested), which is not the case for Chord.

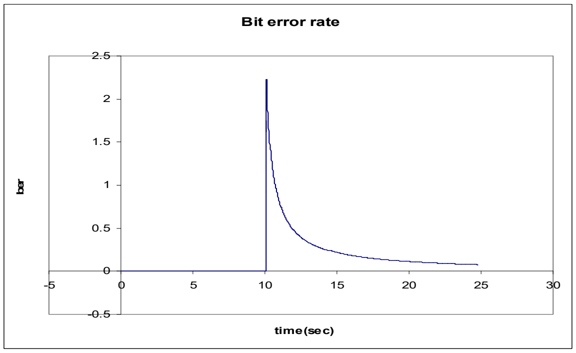

Fig.8.Bit error rate

Here the bit error rate is calculated for mesh chord routing protocol, the graph is plotted between times versus bit error rate.

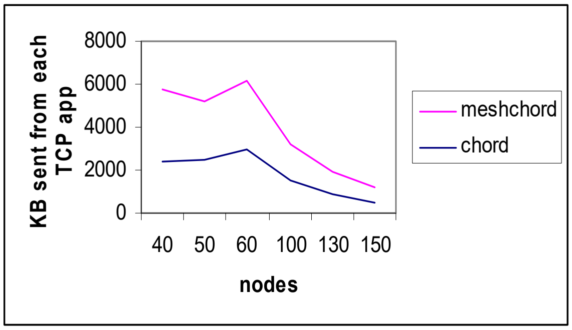

Fig.9.Meshchord Vs chord performance

With respect to the basic Chord design, our proposed MESHCHORD protocol achieves a considerable reduction in message overhead, and a significant improvement in information retrieval performance.

5. CONCLUSIONS

In this paper, we have investigated through packet-level simulation the performance of the Chord approach for peer-to-peer resource sharing in wireless mesh networks. We have also proposed a specialization of the basic Chord approach called MESHCHORD, which exploits peculiar features of wireless mesh networks (location awareness and 1-hop broadcast nature of wireless communications) to improve performance.

The main finding of the study reported in this paper is that, contrary to what happens in MANET environments, the Chord approach can be successfully utilized for implementing file/resource sharing applications in wireless mesh networks. However, the basic Chord design is effective only under relatively static network conditions and in presence of modest background traffic. With respect to the basic Chord design, our proposed MESHCHORD achieves a considerable reduction in message overhead, and a significant improvement in information retrieval performance. This performance improvement allows an effective realization of the P2P overlay also under very dynamic network conditions and in presence of considerable background traffic.

We believe that Mesh chord will be a valuable component for peer to-peer, large-scale distributed applications such as cooperative file sharing, time-shared available storage systems, and distributed indices for document and service discovery, and large-scale distributed computing platforms. Our extensive simulations show that as compared to chord and MESHCHORD.This routing algorithm took lesser energy consumption so that it increases the lifetime of nodes also it took lesser amount of average delay compared to chord. So that MESHCHORD is better suited for static networks with long-duration. Our future work concentrates on implementing split up and add algorithm in wireless mesh networks.

REFERENCES

[1] Amr Al Asaad, Sathish Gopalakrishnan,Victor.peer to-peer file sharing over wireless mesh networks. IEEE 2009.
[2] M. Conti, G. Maselli, G. Turi, and S. Giordano. Cross layering in mobile ad hoc network design. In IEEE Computer, 2004.
[3] G. Ding and B. Bhargava. Peer-to-peer file-sharing over mobile ad hoc networks. In Proc. Of PERCOMW, 2004.
[4] S. Rajagopalan, and C. Shen, a Cross-layer, Decentralized Bit Torrent for Mobile Ad hoc Networks, *Proc.MOBIQUITOUS*, San Jose, CA, 2006.
[5] L. Huiqiong, D. Xuyang, L. Hansheng, and W. Wenmin, “P2P File Sharing in Wireless Mesh Networks,” Proc. Conf. Advanced Parallel Processing and Techniques, pp. 402-413, 2007.
[6] I. Stoica, R. Morris, D. Karger, M. Kaashoek, and H. Balakrishnan, “Chord: A Scalable Peer-to-Peer Lookup Service for InternetApplications,” Proc. ACM SIGCOMM, Aug. 2001.
[7] C. Cramer and T. Fuhrmann, “Performance Evaluation of Chord in Mobile Ad Hoc Networks,” Proc. ACM Int’l Workshop Decentralized Resource Sharing in Mobile Computing and Networking (MobiShare), pp. 48-53, 2006.
[8] ns2: http://www.isi.edu/nsnam/ns/, 2008. Network Simulator.
[9] Simone Burresi, Claudia Canali, Paolo Santi, and M. Elena Renda“MESHCHORD: A Location-Aware, Cross-Layer Specialization of Chord for Wireless Mesh Networks” Proc. IEEE Conf. Peer-to-Peer Computing, 2006.
[10] Azizol Abdullah, Norlida Ramly, Abdullah Muhammed,Mohd Noor Derahman: Performance Comparison Study of Routing Protocols for Mobile Grid Environment, pp 82-88,IJCSNS International Journal of Computer Science and Network Security, VOL.8 No.2, February 2008.