International Journal of Electrical, Electronic and Communication Sciences
ISSN: 2517-9438
Vol:9, No:9, 2015

Investigation of Chord Protocol in Peer to Peer-Wireless Mesh Network with Mobility

P. Prasanna Murali Krishna, M. V. Subramanyam, K. Satya Prasad

***Abstract*—**File sharing in networks is generally achieved using Peer-to-Peer (P2P) applications. Structured P2P approaches are widely used in adhoc networks due to its distributed and scalability features. Efficient mechanisms are required to handle the huge amount of data distributed to all peers. The intrinsic characteristics of P2P system makes for easier content distribution when compared to client-server architecture. All the nodes in a P2P network act as both client and server, thus, distributing data takes lesser time when compared to the client-server method. CHORD protocol is a resource routing based where nodes and data items are structured into a 1-dimensional ring. The structured lookup algorithm of Chord is advantageous for distributed P2P networking applications. However, structured approach improves lookup performance in a high bandwidth wired network it could contribute to unnecessary overhead in overlay networks leading to degradation of network performance. In this paper, the performance of existing CHORD protocol on Wireless Mesh Network (WMN) when nodes are static and dynamic is investigated.

***Keywords*—**Wireless mesh network (WMN), structured P2P networks, peer to peer resource sharing, CHORD protocol, DHT.

I. Introduction

PEER-TO-PEER networks are typically formed by a large number of distributed, heterogeneous, self-configurable and dynamic peers. All the peers cooperate to provide a common service. Applications using peer-to-peer networks use sophisticated discovery mechanisms at each peer to find and communicate with other peers. If the resource discovery mechanism is centralized then it is not scalable, so decentralized discovery mechanisms are required. Peer-to-Peer (P2P) applications have been significantly used for content distribution and sharing in the networks and more than 70% of the traffic are created in the internet by P2P applications.

Major peer-to-peer applications are categorized into three classes. They are distributed file sharing, person-to-person messaging and distributed computing [1]. In Distributed file sharing, each peer shares the file content with other peers in the network based on the requirement of the application. Some examples are Freenet, Lime Wire and Morpheus. In Person-to-person messaging systems, each peer allows to exchange text messages. Some examples are Jabber, Groove, and Yahoo Messenger. Distributed computing systems make the peers to compute some values needed for other peers. Some examples are SETI@home and Entropia.

Some of the advantages of P2P networks are: (1) improvement in scalability because resources are aggregated from peers and reducing the dependency on centralized servers, (2) Reduce the cost by utilizing available resources from other peers and eliminating the need for fixed infrastructure, and (3) load balancing by dividing all the computing at the end systems. In P2P layered architecture a software layer used to construct and maintain the peers in the network is called as P2P substrate. It may be structured or unstructured [2]. Since P2P networks do not have the distinction between client and server nodes as all peers are both clients and servers simultaneously, lesser time is needed to distribute data in comparison to the client-server method [3], [4]. Peer-to-peer networks were started in a wired domain, when being implemented initially for file sharing and distributed computing.

There are two types of overlay networks; Structured and unstructured method [5]. In Structured methods, the P2P overlay network topology is tightly controlled and contents are placed at specified locations to make subsequent data requests more efficient. One of the structured P2P systems use the Distributed Hash Table (DHT), in which data object location information is placed deterministically, at the peers with identifiers corresponding to the data object's unique key. Content Addressable Network (CAN), Tapestry, Chord, Pastry, Kademlia and Viceroy are some of the structured P2P overlay networks.

In unstructured P2P overlay networks, peers are organized in a flat or hierarchical manners and use flooding or expanding-ring and Time-To-Live (TTL) search, to query content stored in the overlay peers. Each peer is visited and evaluated whether content for a query is located locally and will support for complex queries. This is in efficient because queries for content that are not widely replicated must be sent to a large fraction of peers and there is no coupling between topology and content's location. Some of the unstructured P2P overlay networks are Freenet, Gnutella, FastTrack KaZaA and BitTorrent.

In the distributed P2P applications, a node is chosen which stores data. This is solved by the Chord protocol in a decentralized very effectively: provided with a key, it determine which node stores the key's value efficiently. When in steady state, every node maintains routing information only for O (logN) other nodes (for N-node network), and resolves lookups via O (logN) messages to other nodes. Chord's major advantages are simplicity, provable correctness and

P. Prasanna Murali Krishna is with the SGIT, Markapur, India (e-mail: pprasannamurali@gmail.com).
M. V. Subramanyam is with the Santhi Ram Engineering College, ECE Department, India (e-mail: mvsraj@yahoo.com).
Satya Prasad K. is with the JNTUK, ECE Department, India.

performance even while node join or leave network. It also scales well with increase in nodes, pull through from simultaneous node failures, joins and even when recovering answers most lookups appropriately. When a node has only partially correct information, the network performance degrades.

Wireless Mesh Networks (WMNs) are fast emerging wireless networks inspiring numerous applications such as community scale, Peer-to-Peer networking. WMNs include mesh clients and mesh routers, with the latter having minimal mobility forming the spine of the networks [6]. These networks are self-configuring and self-organizing with the nodes establishing an ad hoc network. Communications take place through multi-hop routing. Gateway and bridging processes in mesh routers integrate WMNs with other networks like Internet, IEEE 802.15, IEEE 802.16, cellular, IEEE 802.11, and sensor networks [7]. WMNs have advantages such as low cost, easy maintenance, reliable coverage, robustness when compared to other ad hoc networks. WMNs deliver wireless services for several applications in campus, metropolitan areas, local, and personal. Many research challenges remain in protocol layers even when there are advances in wireless mesh networking.

Mesh clients are either static or mobile, and they form a client mesh network amid themselves and with mesh routers. In WMN, all nodes serve as both host and router. The packets are advanced through multi-hop communication within the network. The characteristics of WMN are ad hoc networking, multi-hop communication with a wireless backbone provided by a mesh routers, mobility of nodes supported by the mesh routers and ability of mesh routers to integrate with heterogeneous networks necessitate new design principle for efficient performance of the network. Some of the factors, which influence the performance of WMN, are radio techniques, scalability, mesh connectivity, compatibility, and inter-operability.

The possibility for distributed P2P networking applications is vast using a structured lookup algorithm such as Chord. However, a structured approach improves lookup performance in a high bandwidth wired network is proved; the overlay networks create unnecessary overhead that negatively impacts performance in an unstable environment such as a WMN. This study investigates CHORD protocols on WMN when the nodes are static and dynamic.

II. Chord Protocol

Stoica et al. [8] presented a scalable peer-to-peer lookup protocol-chord: for internet applications. The Chord protocol used Distributed Hash table (DHT). All the nodes were organized in an identifier circle based on the node identifiers as shown in Fig. 1. Keys were assigned to their successor node in the identifier circle. Hash function was used for even distribution of nodes and the keys on the circle. In the ring, all the peers and the resources were assigned to a unique identifier. Each peer stored its identity in the format of (key, value).

For efficient routing each Chord node stored the details about O (log N) other nodes. If these stored details were out of date then the performance degradation occurred. It was difficult to maintain consistent details of O (log N) nodes because any node might join or leave at any time. So chord node should maintain the dynamic information about other nodes. Whenever a node left from the network, it was considered as node failure.

When requesting content or a resource, each node performed one operation: For the given a key, it mapped the key onto a node. Depending on the application using Chord, that node might be responsible for storing a value associated with the key. Chord used consistent hashing methods to assign keys to Chord nodes. Consistent hashing was used for load balancing, since each node had to store the some number of key, value pairs approximately, and required relatively little movement of keys whenever any node joined or left in the network.

CHORD protocol used Consistent hashing like SHA-1 to generate corresponding identifier. It also generates key, which is the basis for a node and data item location of nodes on a circular overlay. Node identifiers are generated by hashing an IP address, while hashing its unique name produced data item's key. Regular hashing mapped keys to nodes as follows: Nodes were ordered on to an identifier circle. Then key k was assigned to that node. Then this node follows it directly in the identifier space and was designated the successor node of k. Chord uses virtual nodes to maintain uniform key distribution in nodes. In such cases, multiple Ids are assigned to every physical node in the same Chord system ensuring uniform key distribution for nodes [9]. The peer ID could be obtained by hashing the IP address of a particular peer; and data value are hashed to obtain resource ID. Resource ID was kept in the first peer, whose ID >= Resource ID. Finger table are kept by each peer, which stores routing information records.

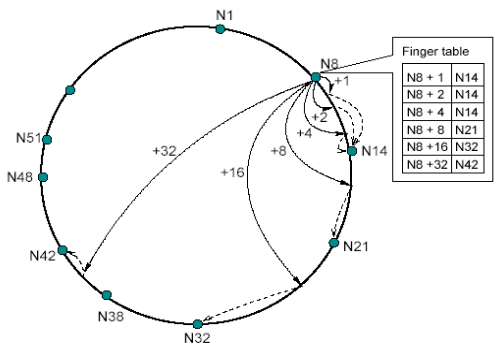

Fig. 1 The CHORD Protocol Ring

Each node had a finger table to record log N successors where N is the number of peers. Every peer checked its successors for the updating the Finger table. It contacted a peer predecessor, as it was useful when a peer leaves the ring and asks previous peer to bring the finger table up to date.

Chord routed the message by forwarding to a successor near to destination identifier. First peer checked in finger table records; chose a successor near the destination, and then forwarded a request to it. The peer on receiving requests would also do some checking and forward a message to a successor. The entire cost was not more than log N hops and ½ log N in average where N is the number of overlay peers [10]. Chord defined advertisement function about joining/leaving procedure for peers.

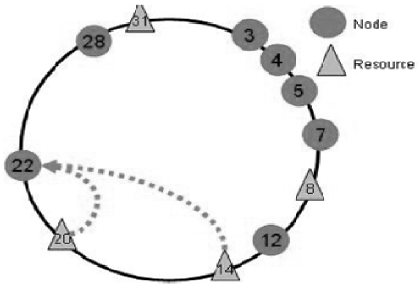

Fig. 2 Chord Overlay with Peers and Resources

A. System Model

Chord streamlines peer-to-peer systems design by addressing some difficult issues.

- **Load balance**: Due to distributed hash function, the keys are spread evenly over nodes providing some natural load balance in Chord.
- **Decentralization**: Chord improves robustness as it is fully distributed. This makes Chord suitable for loose peer-to-peer applications.
- **Scalability**: As the cost of a Chord lookup is proportional to the log of number of nodes, it is feasible for large systems.
- **Availability**: Chord automatically updates internal tables to reveal every new node and node failures, so that the node responsible for a key is always found.

The Chord protocol is implemented either in an iterative or recursive style. In iterative method, lookups are initiated by communicating queries to a series of nodes for information from their finger tables, moving closer to a successor on the Chord ring. Whereas in recursive method, every intermediate node forwards a request to the next node until the successor is reached. The simulations conducted in this study implements the Chord protocol in an iterative style.

Node updates its successor during stabilization process, and one other entry in its successor list/finger table. Therefore, all the unique entries in the node's successor list/finger table is refreshed once during stabilization rounds.

Many modifications for Chord have been proposed in the literature. Karger and Ruhl [11] proposed a modified chord protocol where only a subset of nodes offers a service without forming their own chord ring. The creations of the subgroups carry out the functions and the routing functionality of the existing chord ring with less resource. The proposed diminished Chord requires only O (k) storage per k-sized subgroup compared of O (k log k) from creating a new Chord ring.

Zöls et al. [12] presented the Hybrid Chord Protocol (HCP) to solve the problem of frequent joins and leaves of nodes. The proposed HCP supports the grouping of shared objects in interest groups for efficient search. Information profiles of the shared objects help in automatically establishing context space for assigning and transferring. It also significantly reduces the traffic load.

Burresi et al. [13] adapted the Chord for WMN, by exploiting location awareness and 1-hop broadcast of WMN. The proposed MeshChord reduces message overhead and improves the information retrieval performance. Novak and Zezula [14] proposed M-Chord, a distributed data structure for metric-based similarity search, which also distributes the storage space and executes similarity queries in parallel. Thaalbi et al. [15] proposed a new distributed lookup protocol for mobile P2P networks with the aim to reduce the overhead traffic and the lookup delay. Simulation results demonstrate the efficiency of the proposed scheme when compared to Chord protocol and Backtracking Chord protocol.

III. Methodology

In Chord, every node has a finger table, which implements searches in the forward direction. Nodes based on the position of join are arranged to form a ring. The positions are determined using hello messages with neighbor nodes. In normal P2P network, one query message is sufficient to identify the correct node using the finger table. However, in the case of ad hoc networks, these queries are lost consistently due to the dynamic nature of the nodes and instability of the wireless medium. Similarly, the consistency of the ring for the nodes in the network is hard to maintain, due to node failure and mobility. Thus, when a node joins the network, the steps involved in the finger table updating/stabilization is given by

1) The node joining must first find a node already in the network.
2) The node already in the network uses the CHORD mechanism and its finger table to find a successor for the new node.
3) Since the system is dynamic, finger table updating operations are carried out periodically such that each node sees that its successor and predecessor have not changed.
4) When there is a change in the successor or predecessor, the node in the network assumes that a new node has joined the network, or its successor or predecessor has left the network.
5) If a newly joined node lies amid two nodes, they update their successor values to reflect change.

The Chord protocol maintains just one operation: it maps the given key details into a node. Depending on the application, the node might be accountable for storing a value connected with the key. In this work, the IP address generated dynamically are taken as keys. The pseudo code is shown in Fig. 3.

IV. Experimental Setup

The performance of WMN using CHORD protocol when the nodes are static and dynamic is evaluated through simulations. A test bed network is constructed in the simulator, which consists of 17 nodes spread over an area of 4000m by 4000m. The wireless links communicate with a bandwidth of 2 Mbps. The simulations are run initially for static scenario and later for dynamic nodes where the node is moving with a speed of 60 kmph. The simulations are run for 240 sec. Table I gives the specifications used in the simulations.

TABLE I
Simulation Specifications

Number of Nodes	17
Bandwidth	2 Mbps
Area	4 km X 4 km
Routing Protocol	CHORD
Mobility Speed	60 kmph
Simulation Time	240 Sec

```
//ask node n to find the successor of id
n.find _ successor (id)
if (id ∈ (n, successor))
return successor;
else
// forward the query around the circle
n0 = closest _ preceding _ node(id);
return n0.find _ successor (id);
// search the local table for the highest
//Predecessor of id //
n.closest _ preceding _ node(id)
for i = m down to 1
if (finger[i] ∈ (n,id))
return finger[i];
return n;
```

Fig. 3 Pseudocode

V. Results

Figs. 4-6 show the simulation results for average query response time, number of hops to look up and data exchanged. Simulations were also conducted for varying number of nodes to evaluate average path length and average search time as shown in Figs. 7 and 8.

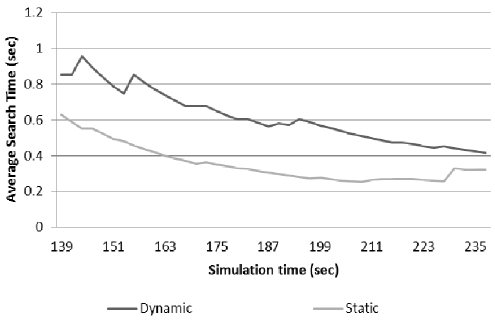

Fig. 4 Average Query response time

It is observed from Fig. 4 that the average query response time for the static nodes is considerably lower than the dynamic node scenario.

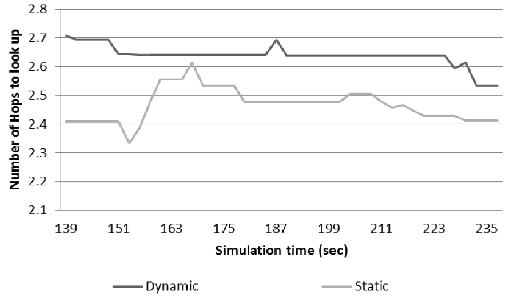

Fig. 5 Number of hops to look up

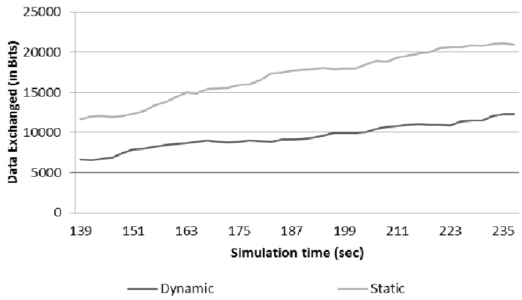

Fig. 6 Data Exchanged in number of bits

From Fig. 5, it is observed that the lookup time increases owing to the dynamic nature of nodes. As seen from Fig. 6, the data exchanged increases considerably for static network when it is compared to the dynamic network.

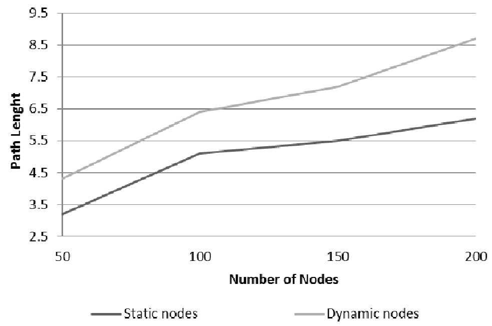

Fig. 7 Average Path Length

Simulation was conducted for varying number of nodes to evaluate average path length. From Fig. 7 it is observed that the Average path length for static node is significantly lower than the dynamic node by an average of 28.07%.

It is observed from Fig. 8 that the average search time increases for dynamic node when compared to static node scenario. On an average, the dynamic nature of nodes requires 33.36% more average search time than static nodes.

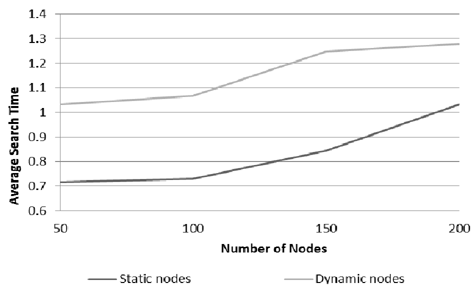

Fig. 8 Average Search Time

VI. Conclusion

Chord algorithm is an efficient lookup algorithm for the distributed P2P networking applications. A structured approach enhances the lookup performance in a high bandwidth wired network; though unnecessary overhead is generated in overlay networks that impact performance negatively in Wireless Mesh Network (WMN). It has been proved that a structured approach increases lookup performance for a high bandwidth wired network; also, overlay networks may create pointless overhead that could adversely impact performance in an unstable environment such as a WMN. This study investigates existing CHORD protocol on WMN when nodes are static and dynamic. The performance of CHORD protocol for metrics like average search time, response time, is evaluated for WMN. Simulation results demonstrate that the dynamic nodes affect the performance of the WMN negatively when compared to the static nodes.

References

[1] M. Kelaskar, V. Matossian, P. Mehra, D. Paul, and M. Parashar, "A Study of Discovery Mechanisms for Peer-to-Peer Applications.
[2] Mohamed Hefeeda, "Peer-to-Peer Systems". School of Computing Science, Simon Fraser University, Canada.
[3] Stratis Ioannidis, Peter Marbach. "On the Design of Hybrid Peer-to-Peer Systems" (J). ACM SIGMETRICS Performance Evaluation Review, 2008, 36(1): 157-168.
[4] Bartosz Biskupski, Jim Dowling, "Jan Sacha. Properties and Mechanisms of Self-Organizing MANET and P2P Systems". ACM Transactions on Autonomous and Adaptive Systems (TAAS), 2007, 2(1): 1-34.
[5] Eng Keong Lua, Jon Crowcroft, Marcelo Pias, Ravi Sharma and Steven Lim, "A Survey and Comparison of Peer-to-Peer Overlay Network Schemes, IEEE Communications Survey and Tutorial", March 2004.
[6] Ian F. Akyildiz, Xudong Wang, Weilin Wang, "Wireless Mesh Networks: A Survey, Computer Networks" 47 (2005) 445–487.
[7] J. Jun, M.L. Sichitiu, "The Nominal Capacity of Wireless Mesh Networks, IEEE Wireless Communications" 10 (5) (2003) 8–14.
[8] Ion Stoica, Robert Morris, David Liben-Nowell, David R. Karger, M. Frans Kaashoek, Frank Dabek, Hari Balakrishnan, "Chord: A Scalable Peer-to-peer Lookup Protocol for Internet Applications University of California".
[9] Xianghan Zheng and Vladimir Oleshchuk, "Improvement of Chord Overlay for P2Psip-Based Communication Systems, International Journal of Computer Networks & Communications (IJCNC)", Vol. 1, No.3, October 2009.
[10] Jani, H. and C. Gonzalo, "Evaluation of DHTs from the Viewpoint of Interpersonal Communications, in Proceedings of the 6th International Conference on Mobile and Ubiquitous Multimedia." 2007, ACM: Oulu, Finland.
[11] Karger, D. R., & Ruhl, M. (2005) "Diminished Chord: A Protocol for Heterogeneous Subgroup Formation in Peer-to-Peer Networks. In Peer-to-Peer Systems III" (pp. 288-297). Springer Berlin Heidelberg.
[12] Zöls, S., Schollmeier, R., Kellerer, W., & Tarlano, A. (2005). "The Hybrid Chord Protocol: A Peer-to-Peer Lookup Service for Context-Aware Mobile Applications. In Networking-ICN" 2005 (pp. 781-792). Springer Berlin Heidelberg.
[13] Burresi, S., Canali, C., Renda, M. E., & Santi, P. (2008, March). "MeshChord: a Location-Aware, Cross-Layer Specialization of Chord for Wireless Mesh Networks (Concise Contribution). In Pervasive Computing and Communications", 2008. PerCom 2008. Sixth Annual IEEE International Conference on (pp. 206-212). IEEE.
[14] Novak, D. & Zezula, P. (2006, May). "M-Chord: A Scalable Distributed Similarity Search Structure. In Proceedings of the 1st International Conference on Scalable Information Systems" (p. 19). ACM.
[15] Thaalbi, M., Meddahi, A., Bejaoui, T. & Tabbane, N. (2011, October). "An Enhanced Chord-Based P2P Lookup Protocol for Mobile Ad Hoc Networks. In Wireless Days (WD)", 2011 IFIP (pp. 1-5). IEEE.