Performance Analysis of Chord Protocol for Peer to Peer Overlay Topology in Wireless Mesh Network

P.Prasanna Murali Krishna
Research scholar
JNTUK,

M.V.Subramanyam, PhD.
Principal,
Santhiram Engineering College,

K.Satyaprasad, PhD.
JNTUK

ABSTRACT

Peer-to-Peer (P2P) applications have been extensively used for file sharing and accounts for more than 70% of the traffic in the internet. The inherent characteristics of P2P make it advantageous in content distribution when compared to client-server architecture. In a P2P network, there is no clear distinction between client and server nodes, as every peer is both a client and a server at the same time. Thus, the time required to distribute data is lesser when compared to the client-server method. Due to the distributed and scalability of P2P, structured P2P approaches have also been proposed for ad-hoc networks. The potential for distributed P2P networking applications using a structured lookup algorithm such as Chord is enormous. However, whilst it has been proven that a structured approach improves lookup performance in a high bandwidth wired network; overlay networks may create unnecessary overhead that could negatively impact performance given an unstable environment such as a Wireless Mesh Network (WMN). In this paper, it is proposed to study existing CHORD protocols on WMN when nodes are static and dynamic.

General Terms

Network, Performance Analysis.

Keywords

Wireless Mesh Networks (WMN), Peer to Peer Resource Sharing, CHORD Protocol.

1. INTRODUCTION

Wireless mesh networks (WMNs) include mesh routers and mesh clients, with the latter having minimal mobility forming the backbone of WMNs. They also ensure network access to mesh and conventional clients. Integrating WMNs with other networks like Internet, cellular, IEEE 802.11, IEEE 802.15, IEEE 802.16, sensor networks, is done through a gateway and bridging processes in mesh routers [1]. Mesh clients are either stationary or mobile, and form a client mesh network among themselves and also with mesh routers. WMNs are currently undergoing rapid progress leading to many deployments. WMNs deliver wireless services for various applications in personal, local, campus, and metropolitan areas. Many research challenges remain in protocol layers despite advances in wireless mesh networking.

Each node operates as both host and router and advances packets for other nodes within direct wireless transmission range of destinations. A WMN is dynamically self-organized and self-configured, with network nodes automatically establishing and maintaining mesh connectivity among themselves (creating an ad hoc network) [2]. This feature has great advantages for WMNs like low up-front cost, easy network maintenance, robustness, and trust worthy service coverage.

Peer-to-Peer (P2P) applications are used widely for file sharing in wired networks accounting for more than 70% of internet traffic. Inherent P2P characteristics have advantages in content distribution as compared to client-server architecture [3, 4]. P2P networks do not have the distinction between client and server nodes, as all peers are both clients and servers simultaneously. Hence, lesser time is needed to distribute data in comparison to client-server method. Peer-to-peer networks which created a lot of interest started in a wired domain being implemented initially for targeted file-sharing and distributed computing. The aim was to use distributed storage and computational capacity instead of a central server as this approach makes it less vulnerable to denial of service attacks. It ensures that the networks are robust.

Peer-to-Peer (P2P) systems were derived from file sharing applications and based on present network overlays like Chord [5], Content Addressable Network (CAN) [6], Pastry [7] and Tapestry [8], many research issues like load balancing analysis, query processing and data mining are being investigated. As P2P systems are very large, there is an urgent need for efficient procedures to handle large amount of data distributed to all peers.

As distributed P2P applications have to choose the node which stores data, the Chord protocol solves this in a decentralized way by offering a powerful primitive: provided with a key, it determines which node stores the key’s value efficiently. In the steady state, in an N -node network, every node maintains routing information only for O (log N) other nodes, and solves lookups via O(log N) messages to other nodes. Chord’s attractive features are simplicity, provable correctness and performance even when concurrent node arrive and depart. It functions efficiently albeit at degraded performance when a node has only partially correct information. Chord scales well with many nodes, recovers from numerous simultaneous node failures, join and answers most lookups correctly even when recovering.

The potential for distributed P2P networking applications using a structured lookup algorithm such as Chord is vast. However, whilst it has been proven that a structured approach improves lookup performance in a high bandwidth wired network; overlay networks may create unnecessary overhead that could negatively impact performance given an unstable environment such as a Wireless Mesh Network (WMN). In this paper, it is proposed to study existing CHORD protocols on WMN when nodes are static and dynamic.

2. CHORD PROTOCOL

CHORD protocol is a resource routing based on Distributed Hash Table (DHT). Nodes and data items are organized into a 1-dimensional circular space (ring) by the Chord protocol as seen in Figure 1. Consistent hashing like SHA-1 generates corresponding identifier and key which is the basis for node and data item location of nodes on a circular overlay. Hashing an IP address generates node identifier, while data item's key is produced by hashing its unique name. Regular hashing maps keys to nodes as follows: first nodes are ordered on to an identifier circle of module 2m. Then key k is assigned to that node which follows it directly in the identifier space and is termed the successor node of k, denoted by successor (k). Chord controls uniform key distribution of nodes through virtual nodes. In such cases, every physical node has multiple virtual images with different Ids in the same Chord system ensuring uniform key distribution more system nodes [9].

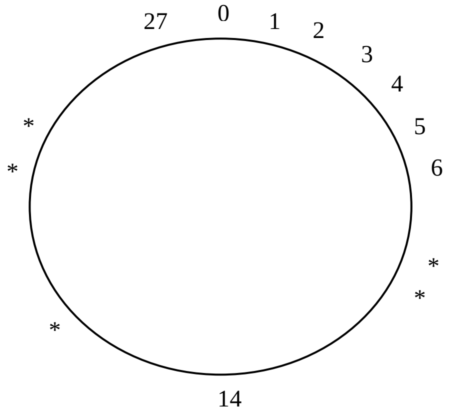

Figure 1: The CHORD Protocol Ring

In Chord overlay, peers and resources form a ring, as shown in Figure 2. Within this ring, the peers and resources are assigned an integer Node ID/Resource ID. The peers store ID in <id, value> pairs, where id is the peer/resource ID, value is the peer address information or data storage. Consistent hashing assigns Peer/resource ID, e.g.SHA-1 algorithm. For example peer ID can be got through hashing the IP address of a particular peer; and resource ID is done through hashing data value. Resource ID is kept in the first peer, whose ID>= Resource ID .Each peer includes a routing table, known as Finger table, to store routing information records. The Finger table records logN successors where N is the number of peers in the overlay.

Every peer contacts successors to update the Finger table. It contacts a peer predecessor as it is useful when a peer leaves the ring and asks previous peer to bring the finger table upto date.

Chord routes message by forwarding messages to a successor near destination identifier. Peer will first check finger table records; choose a successor near the destination, and then forward a request to it. The peer on receiving request would also do some checking and forward a message to a successor. The entire cost is not more than logN hops and ½log N in average where N is the number of overlay peers [10]. Chord defines advertisement function about joining/leaving procedure for peers. It is this which orders a corresponding successor and predecessor to update their finger tables.

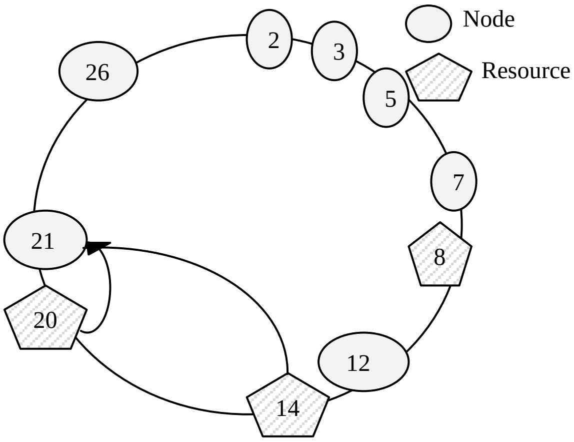

Figure 2: Chord Overlay with Peers and Resources

Chord's routing procedure can be considered a one-dimensional analogue of a Grid location system (GLS) [11] which relies on real-world geographic location information to route questions; Chord maps nodes to an artificial one-dimensional space and routing is carried on inside this by an algorithm similar to Grid's. Napster [12] and Gnutella [13] provide a lookup operation to locate data in a distributed set of peers, searching based on user-supplied keywords, as Chord searches data with unique identifiers. Use of keyword search leads to problems in both systems. While Napster relies on a central index, leading to a single point of failure, Gnutella floods all queries over the entire system, leading to high communication and processing costs in huge systems. Chord was the basis for many research projects subsequently.

The Chord File System (CFS) stores files in a peer-to-peer system and the storage blocks are located using Chord. New analysis techniques revealed that Chord's stabilization algorithms continue with a good lookup performance inspite of continued failure and node joining [14]. Chord following an evaluation has been selected as a tool to serve DNS and maintain a distributed public key database for secure name resolution.

Chord simplifies peer-to-peer systems design and applications by tackling some difficult issues.

• Load balance: Chord is a distributed hash function, spreading keys evenly over nodes providing some natural load balance.

• Decentralization: Chord is fully distributed; one node is not more important than others improving robustness and making Chord appropriate for loose peer-to-peer applications.

• Scalability: The cost of a Chord lookup is proportional to the log of nodes number making large systems feasible. No parameter tuning is needed to achieve this.

• Availability: Chord automatically adjusts internal tables to reflect every new node as also node failures so that - barring major failures - the node responsible for a key can be found always. This is true even if the system changes continuously.

The Chord protocol can be implemented in an iterative/ recursive style. In the former, a node resolving a lookup initiates communication querying a series of nodes for information from finger tables, inching closer to a successor on the Chord ring. In the latter style, every intermediate node forwards a request to the next node till the successor is reached. The simulator implements the Chord protocol in an iterative style.

During a stabilization process, a node updates its successor and one other entry in its successor list/finger table. Hence, if a node's successor list/finger table contains many unique entries, each is refreshed once during stabilization rounds. Unless otherwise indicated, the successor's list size is one. In other words a node only knows its immediate successor.

3. EXPERIMENTAL SETUP AND RESULTS

Simulations are conducted to study existing CHORD protocols on WMN when nodes are static and dynamic. The testbed network consists of 17 nodes spread over an area of 4000m by 4000m with bandwidth of 2 mbps. The simulations are run for 240 sec. The simulations are run twice; initial simulation with the static nodes and later for dynamic nodes. For dynamic scenario, a speed of 60 kmph is used. Figures 3 to 6 show the simulation results for average search time, average query response time, number of hops to lookup and data exchanged. Red is dynamic nodes and blue is static nodes.

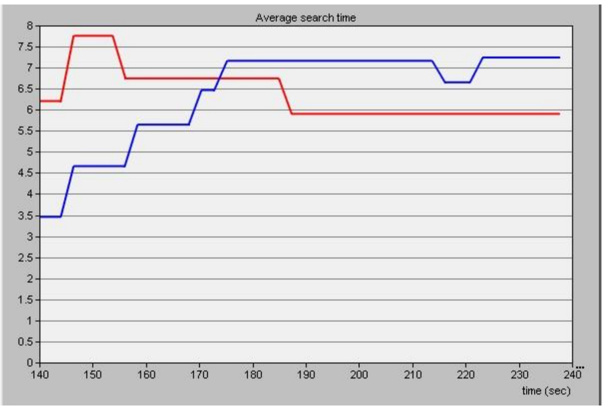

Figure 3: Average search time (Red represents dynamic nodes and blue represents static nodes)

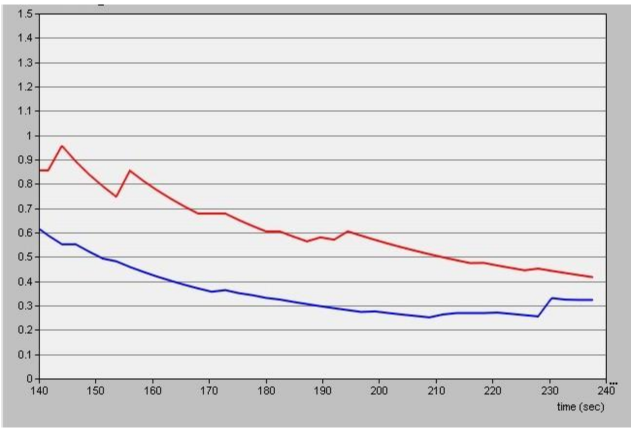

Figure 4: Average Query response time (Red represents dynamic nodes and blue represents static nodes)

It is observed from the Figures 3 and 4 that the average search time increases for the static node scenario after the initial time elapses but average query response time for static node is significantly lower than the dynamic node scenario.

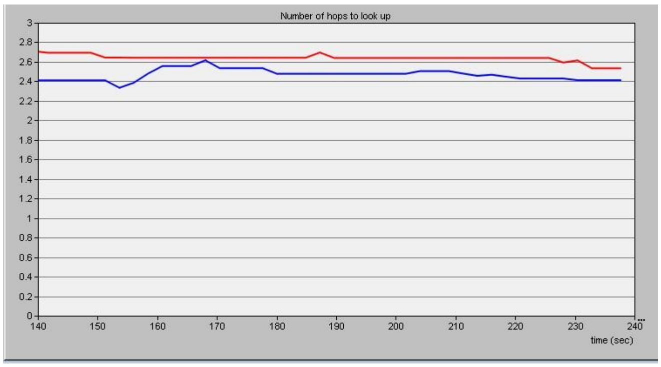

Figure 5: Number of hops to look up (Red represents dynamic nodes and blue represents static nodes)

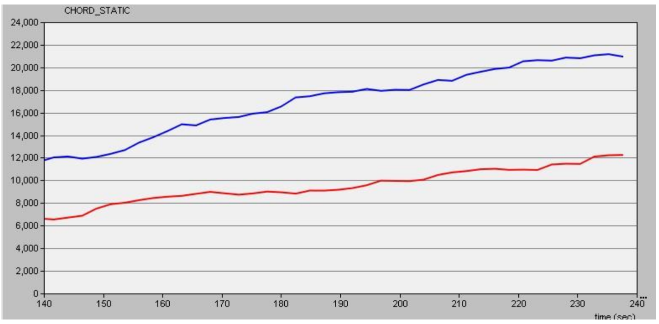

Figure 6: Data Exchanged (Red represents dynamic nodes and blue represents static nodes)

From Figure 5, it is seen that the lookup time increases with due to the dynamic nature of the nodes. The data exchanged increases substantially for static network when compared to the dynamic network.

4. CONCLUSION

The potential for distributed P2P networking applications using a structured lookup algorithm such as Chord is vast. However, it has been proven that a structured approach improves lookup performance in a high bandwidth wired network; overlay networks may create unnecessary overhead that could negatively impact performance given an unstable environment such as a Wireless Mesh Network (WMN). In this paper, it is proposed to study existing CHORD protocol on WMN when nodes are static and dynamic. Simulation results show that the performance of the network in term of average search time, average query response time, number of hops to lookup and data exchanged degrades with dynamic nature of the nodes.

5. REFERENCES

[1] Ian F. Akyildiz, Xudong Wang, Weilin Wang, Wireless mesh networks: a survey, Computer Networks 47 (2005) 445–487.

[2] J. Jun, M.L. Sichitiu, The nominal capacity of wireless mesh networks, IEEE Wireless Communications 10 (5) (2003) 8–14.

[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 [J]. ACM Transactions on Autonomous and Adaptive Systems (TAAS), 2007, 2(1):1-34.

[5] Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan,“Chord: A scalable peer-to-peer lookup protocol for internet applications,” IEEE/ACM Transactions on Networking, Vol. 11, No. 1, pp. 17–32, 2003.

[6] Sylvia Ratnasamy, Paul Francis, Mark Handley, Richard Karp, and Scott Schenker. A scalable contentaddressable network. In SIGCOMM ’01: Proceedings of the 2001 conference on Applications, technologies, architectures, and protocols for computer communications, pages 161–172, New York, NY, USA, 2001. ACM Press.

[7] Rowstron and P. Druschel, “Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems,” Proceedings of IFIP/ACM International Conference on Distributed Systems Platforms (Middleware), Heidelberg, Germany, pp. 329-350,November 12-16, 2001.

[8] Zhao, J. Kubiatowicz, and A. Joseph, “Tapestry: An infrastructure for fault-tolerant wide-area location and routing,” Comput. Sci. Div., Univ. California, Berkeley, Tech. Rep. UCB/CSD-01-1141, 2001.

[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] J. Li, J. Jannotti, D. De Couto, D. R. Karger, and R. Morris, “A scalable location service for geographic ad hoc routing,” in Proc. 6th ACM Int. Conf. Mobile Computing and Networking, Boston, MA, Aug. 2000, pp. 120–130.

[12] Napster.[Online].Available: http://www.napster.com/

[13] Gnutella.[Online].Available: http://gnutella.wego.com/

[14] D. Liben-Nowell, H. Balakrishnan, and D. R. Karger, “Analysis of the evolution of peer-to-peer systems,” in Proc. 21st ACM Symp. Principles of Distributed Computing (PODC), Monterey, CA, July 2002, pp. 233–242.