Two-Tier CHORD for Decentralized Location Management

Waqas A. Imtiaz

ABSTRACT

To conceive the full potential of ubiquitous mobile computing, mobile nodes must be able to roam seamlessly across different networks. However, the location i.e. IP address, of a called mobile nodes must be determined within a certain amount of time, before the services are blocked. With the upcoming 3G, 4G and wireless LAN technologies, it is necessary to have an efficient location management system that adapts to such dynamic environments, and augment problems like center point of failure, delay and lack of reliability. One approach is to provide decentralized location management, using CHORD, whose key vale mapping can efficiently determine the required name to IP mapping. However, a major issue concerning this technique is the presence of unstable chord nodes, which brings instability to the framework. Nodes with minimum computation capabilities can slow down the lookup mechanism, and hence terminates the ongoing session. This paper provides a solution to this problems using well-known two-tier CHORD architecture. Nodes with high computation power and stability will act as Location Servers, which will be responsible for storing and distributing temporary locators i.e. IP addresses of the called mobile node. Suitability of the proposed framework is tested using analytical analysis in comparison with the existing chord mobility framework. Such an approach can bring stability to the framework, with its decentralization feature still intact**.**

Keywords

Two-Tier Model, CHORD, Location Management

1. INTRODUCTION

Mobility is one of the projecting features that describe communication devices today. Mobility in wireless IP networks is becoming increasingly popular with new advancements in technology. However, while being mobile it is important to stay connected to your network at any time. Seamless roaming across different networks is thus an important criterion, which includes the ability to switch network interfaces during an ongoing session (handover management), and maintain reachability for new connections (location management). Numerous solutions provide seamless mobility at different layers; however, the choice of an efficient scheme is still an open issue.

The most known mobility frameworks developed by the IETF are Mobile IPv4 (MIPv4) and Mobile IPv6 (MIPv6). Unfortunately, MIPv6 is not adequate to support fast handover and MIPv4 is not widely enabled between administrative domains even though it was preliminary proposed in 1996. SIP and DDNS, are application layer approaches, which like MIP, rely on centralized servers for the provision of required mobility services. These centralized servers although efficient, introduce an unavoidable delay during handover. Moreover, the network is down if one of such servers stops working. Thus, a solution is required, which can provide seamless mobility at end user, with decentralized servers for location management.

Mobile Stream Control Transmission (mSCTP) can provide the required handover management at user end, which eliminates the issues like dependence on centralized servers. Mobile SCTP however cannot provide the required location management to initiate the communication session from Correspondent Node (CN) towards Mobile Node (MN). Peer-to-Peer (P2P) algorithm like Chord can provide the required location management using its efficient key-value mapping [1][9]. Location management provides the required IP address in order to initiate a communication session. It is important to determine the MN current location i.e. IP address in a certain amount of time before the session expires. Traditional chord location management architecturecan provide the necessary name to IP mappings; however, one problem associated with the framework is that; every node can act as a location server [1]. Such solution however decentralized, brings instability to the framework. Nodes with minimum computation capabilities, acting as location servers, can slow the lookup mechanism, and hence terminates the ongoing session. Moreover, high rate of nodes entering and leaving the network can affect the performance, as high rate of node's departure will result in more configuration and update messages, thus congesting the entire network along with terminating any ongoing sessions.

This paper proposes a decentralized location management framework, with two-tier architecture [2][7][8]to deal with stability and scalability issues in single-tier chord (STC) mobility framework [1]. Rest of the paper consists of the following session: section 2 gives a detailed system model of the proposed two-tier mobility framework, followed by joining and migration of MNs in section 3. Query process of the required TL value is presented in section 4 and section 5 of the paper give analytical analysis of the proposed two-tier mobility framework with single layer framework in [1]. Conclusion is given in last section of the paper.

2. SYSTEM Model

2.1 Addressing Scheme

Two-tier chord (2TC) mobility framework, addressing scheme consists of an *identifier-locator* set where identifier corresponds to the unique identity of the MN [1][3][4], and locator represents the current IP address of MN inside the network.

Each MN inside the network has UID (Unique Identifier), which correspond to the key in the key-value pair. UID consists of three basic components, *name: device:ID*. Name contains the owner's name; and it can be selected in any desired form, i.e. surname or name initials. Device refers to the type of the device, e.g. laptop, mobile, and PDA etc. ID is a unique identity that can be user's mobile number, email address or NIC number, e.g. xyz: laptop: 17301xxxx. Any naming scheme can be adapted as chord provides flexible naming mechanism.

Locator refers to the TL (Temporary Locator), which represents the value in the key-value pair. It is identical to the current IP address of the MN. TL has the ability to update itself as soon as MN enters a new network and attains a new IP address. TLs can also determine the position of a MN on the identifier circle. As position of a MN on the identifier circle primarily depends on its Node ID, which obtained by hashing its IP address i.e. TL value and port number. However, such values can lead to a random placement of the node in the overlay network. Therefore, it is a better to place the neighboring nodes in a single network, as close as possible. Adding a network prefix to the node id such that each node in same network will have the same prefix can provide the required node placement.

2.2 Framework Components

2TC mobility framework contains heterogeneous wired and wireless networks connected via high-speed wired backbone network. Each network contains with a set of location servers (LSs), which are distributed and mobile. These location servers form an overlay network and publish location information to each other for storage, and resolving CNs queries regarding MNs TL value. Based on the capacity and availability of MNs in the overlay network, they can be classified into two categories: Ordinary Chord Nodes and Location Servers.

2.2.1 Ordinary Chord Nodes (OCNs)

Ordinary chord nodes are the members of the lower layer hierarchy as shown in figure 1, and form a single-connection structure with LS in the upper layer as shown in figure. OCNs do not store and participate in TL value queries, so they can frequently join and leave the network without affecting the location management process. Whenever an OCN wants to start a communication session, it forwards the query containing MN TL value, to its associated LS. LS perform the required operation and forward the required TL value to the querying OCN. After getting the TL value of the MN, OCN can initiate its communication session.

In the Single-connection structure, every OCN uses PING/PONG algorithm periodically to check its connectivity with LS. The OCN sends a PING messages to its associated LS, and the LS responds with a PONG message. This ensures the availability of LS in the network.

Every OCN maintains the following variables:

- *Successor list*: It contains a list of successor nodes.
- *Predecessor list*: It contains a list of predecessor nodes present in the identifier circle.
- *Finger Table*: It gives information about the 2^{ith}successor node [2].
- Two pointers are also maintained at every OCN, LS pointer and a pointer to the predecessor of LS in the upper layer [2].

2.2.2 Location Servers (LSs)

MNs with higher availability, processing, storage capacity, and which are available in future, for a long time are place in the upper layer of the overlay network. Location servers form a circle where peers are connected to one another, similar to that of a chord network. Moreover, LS is the gateway or path to communicate among the OCNs in the lower layer i.e. OCN cannot initiate a communication session unless LS provide the TL value of the MN. In the chord ring, each location server applies the stabilization protocol to update its successor / predecessor pointer in the event of MN failure or arrival from the lower layer.

When location servers fail or leave the overlay network, OCN previously marked as candidate LS will take its place. A random OCN is selected on the basis of capacity, firewall support and availability, if the system is no able to locate any candidate OCN.

Every LS maintains the following variables:

- *Successor list*: It contains a list of successor nodes.
- *Predecessor list*: It contains a list of predecessor nodes present in the upper layer.
- *Finger Table*: It gives information about the 2^{ith}successor node.
- *Index Pointers*: It contains information about the UID and TL values of all the OCNs associated with it.
- *Successor Pointers*: It contains information about the UIDs present in its successor nodes.
- *LSs list*: It contains a list of all successor LSs present in the upper layer of the overlay network.
- *OCNs list*: It contains priority wise information regarding all the potential OCNs, which are willing to become LS.

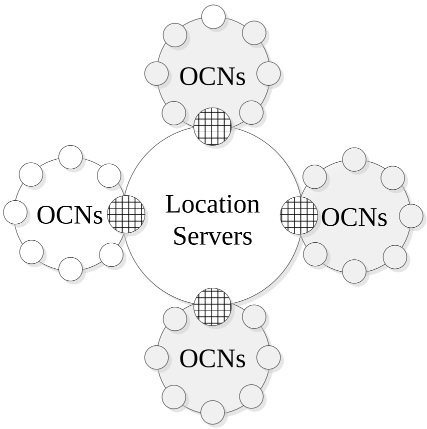

Figure 1: Two-Tier Chord Mobility Framework

3. JOINING AND MIGRATION

Every MN that wants to join the network will enter as OCN and cannot connect as LS directly [2]. Every node joining the overlay network, knows one of the existing LS, and queries it to find the point of attachment. Location server on receiving the joining request examines the hashed UID for MN and finds out whether it is to be attached in its OCN layer or with other LS. Accordingly, the joining MN is attached in the OCN layer under its root LS, before its successor by using Chord's find successor process. MN after joining the overlay network, shares its UID and TL value with its root LS. LS update its index pointers, and publish successor pointers towards its successor nodes (successor location servers) containing the UID value of MN (M). It shows that the root LS has the required UID TL pair for node M. Successor pointers can help in reducing the number of queries during a conventional chord lookup process [1].

As an illustration consider the example in figure 2, when the MN (M) wants to join the network. It sends a registration request on LS (D) TL value, using its cache. LS (D) on receiving the registration request, determine if it is to be attached in its OCN layer, or shall it locate another location server, which will act as root node for joining mobile node. Accordingly, node D locates a root node for mobile node M using the chord lookup process. Root LS will publish its successor pointers towards its successor LSs.

OCNs, willing to become location servers send an upgrade level message to their respective LS. The current LS form a list of requesting OCNs, which may become location servers. The current LS, shares the list of such OCNs, with nodes in its lower layer. Now when the current LS fails or leaves the network, OCN with highest priority is promoted to its place. The selected OCN waits for some time before migrating to the upper layer, so that other OCNs can update their LS pointers and fix their fingers [2].

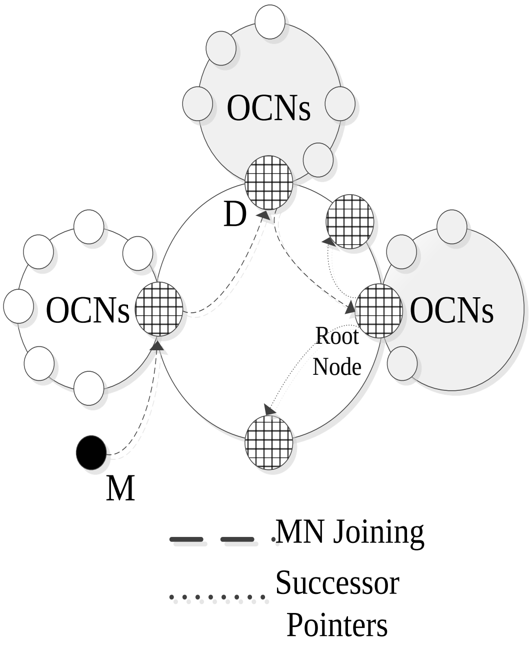

Figure 2: Node joining process for 2TC mobility Framework

4. LOOKUP PROCESS

Lookup process for the proposed 2TC mobility framework is shown in figure 3. When correspondent node acting as an OCN, wants to start a communication session with the mobile node, it creates a query message containing MN's UID address. Query message from OCN is forwarded to its associated LS. Afterreceiving thequeryrequest, LS checks its index pointers against the provided UID. Iftherequired UID isfoundinthe index pointers, lookupprocessisdone, and LS replies to the querying OCN with the required TL value.

Ifthe required TL for the provided UID is not found, location serverforwards thequerytoitssuccessor LS,basedonthefingertablesimilartochord ring. This process continues until the LS containing the required UID and TL pair receives the query. The number of queries required to find the require LS are reduced if the query reaches a LS, with successor points. On receiving the query packet, LS forwards the required TL value to the querying node.

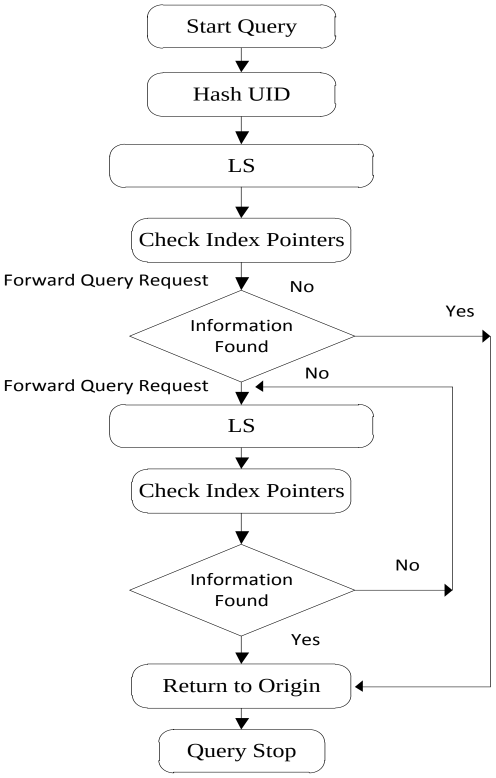

Figure 3: Lookup Process

5. ANALYTICAL ANALYSIS

This section presents the cost analysis of the proposed 2TC mobility framework and its comparison with STC mobility framework. These two frameworks are compared on the basis of lookup time and number of hops required to perform a TL query. For this purpose, following assumptions are made for the proposed 2TC mobility framework:

- Total number of MN in the overlay network = N
- Total number of Location servers = L
- Total number of OCNs = L – N
- Number of entries in Finger table of LSs = $O(\log L)$
- Probability of finding TL value using successor pointer = μ
- Processing time of OCN = t_o
- Processing time of LS = t_l

The lookup process takes $(\frac{1}{2}\log N)$ number of hops on an average to perform a lookup process [5]. Here the number of nodes of location servers is L, so $(\frac{1}{2}\log L)$ hops on an average are required to find the root LS. As OCN forwards a query to its associate LS [6], so a single node is traversed before the query reaches LS. Thus the number of hops required to perform the query becomes$(\frac{1}{2}(\log L+1))$.

If probability of finding TL value at location servers using successor pointers is (μ), then the average number of hops and lookup latency for 2-tier chord mobility framework becomes:

$hopcount_{2TC} =$
$(1-\mu)\times\frac{1}{2}(\log L+1)(5.1)lookup\ latency_{2TC} = (t_l \times$

$$(1-\mu)\times\frac{1}{2}(\log(L+(t_o\times 1)))$$
(5.2)

Average number of hops required to perform a query in STC as location manager becomes:

$$hopcount_{TC}=(1-\mu)\times\frac{1}{2}(\log N) \qquad (5.3)$$

$$lookup\ latency_{TC}=(t_o\times(1-\mu)\times\frac{1}{2}(\log N))(5.4)$$

Analytical analysis is performed for 50, 100, 250, 500 and 1000 nodes, and their results are shown in figure 4 to 7. Location server to OCN ratio is selected as 1:10 for these analyses. Further probability is set equal to 0 in equation (5.1) and (5.3) for figure 4 and 6, in order to compare the basic system models. Processing time of OCN is set as 1 sec. Location servers are more stable and contain high computation capability than OCNs that is why their processing time is set equal to 0.5 s. In figure 4 and 5, using equation (5.1) and (5.3), number of hops is plotted, as a function of number of nodes for both STC and 2TC mobility frameworks.

Figure 4 shows that as we increase the number of nodes, the hop count required for query process also increases. However, our proposed 2TC mobility framework, results in less hop count as compare to STC. Further, it is observed that better lookup efficiency is achieved by introducing the probability of finding the successor pointers at location servers. It can be seen in figure 5, that as we introduce the probability, the amount of hops required to perform a query significantly decreases. 2-Tier chord mobility framework is still efficient for performing the required query with less number of hops as compared to STC mobility framework.

Lookup latency is calculated using equations (5.2) and (5.4). $t_o$is set equal to 1 sec for simplicity, and the value of $t_l$is taken as 0.5 sec. These values are selected on the basis of computation capabilities of OCNs and LS. Probability is set as 0 for figure 6, and 10 % probability of finding successor pointers in included in figure 7. Results show that the amount of time required to perform a query process is less for the proposed 2TC mobility framework. Even when the number of nodes increases, the time taken to perform a query process increases by a very small amount.

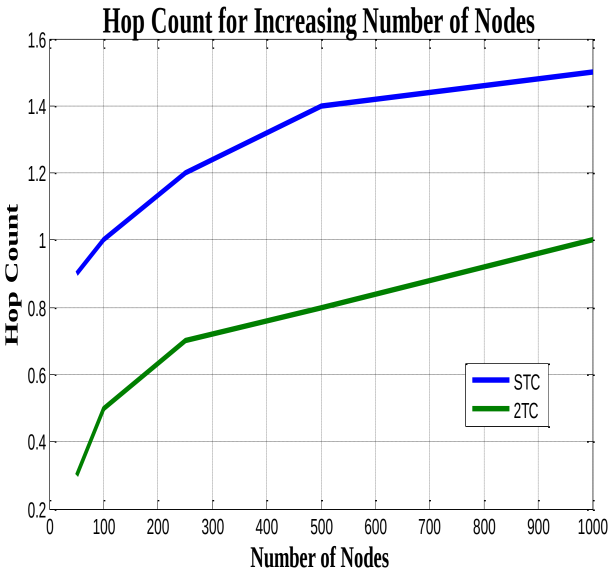

Figure 4: Hop count

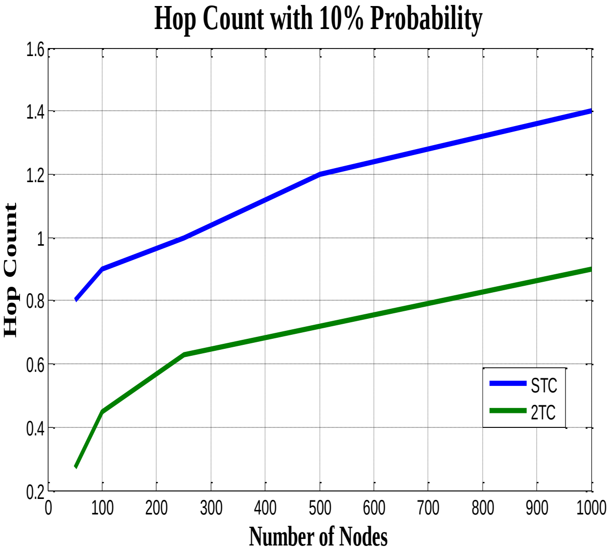

Figure 5: Hop count with 10 % probability

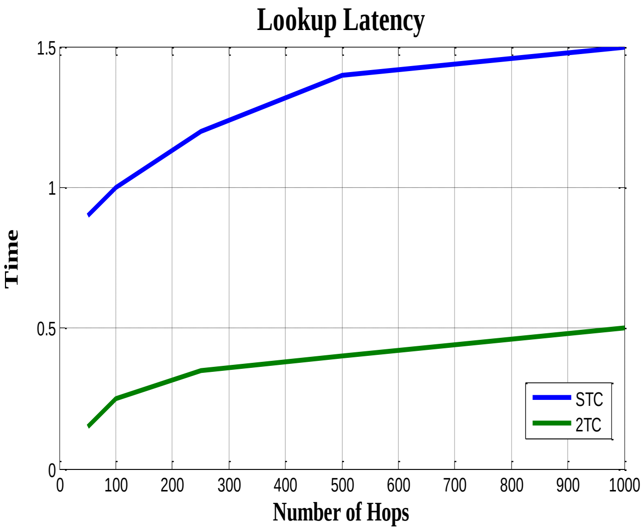

Figure 6: Lookup latency

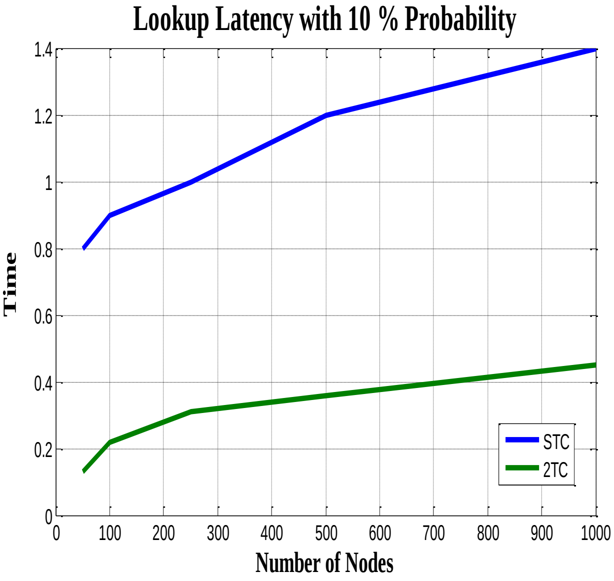

Figure 7: lookup latency with 10% probability

6. CONCLUSION

Efficient location management is important in seamless mobility; because the required IP address i.e. TL value of the called MN must be determined before the session expires. Different solutions can provide LM; however, an efficient decentralized LM scheme is necessary to remove central point of failure, which introduces unavoidable delay and dependence on central point of failure. Two-tier Chord can efficiently full fill the purpose using its efficient key-value mapping to provide the required TL value. This paper presents 2TC mobility framework to compensate the issues relating STC. Simulation result shows that two-tier chord can perform efficiently than STC mobility framework. Even after significant increase in the number of nodes, 2TC is still able to find the required TL value in minimum possible time. Moreover, introduction of stable MNs as LSs can eliminate the problem of unstable nodes. Two-tier chord mobility framework is an efficient approach to provide location management, which can cope with all issues regarding unstable nodes, and nodes with minimum computation capabilities.

7. REFERENCES

[1] Waqas Ahmed, M. Jashim, "An SCTP Based Decentralized Mobility Framework", Master's Thesis, Department of Electrical Engineering, Blekinge Institute of Technology, Karlskrona, Sweden, July 2010.

[2] MayanakPandey, Syed Mushtaq Ahmed, BanshiDhar, "2T-DHT: A Two Tier for Implementing Publish/Subscribe", *International Conference on Computational Science and Engineering, 2009.*

[3] K. Sethom, H. Afifi, G. Pujolle, "PALMA: A P2P Based Architecture for Location Management", IFIP *International Conference on Mobile and Wireless Communications Networks,* 2005.

[4] Waqas A. Imtiaz, Hilal Ahmed, "DHT CHORD as Location Manager", 2nd*International Conference on Aerospace and engineering,* ICASE-2011.

[5] I.Stocia, R. Morris, D. Karger, "Chord: A Schalable peer-to-peer lookup service for internet applications", *ACM SIGCOMM Computer Communication,* vol. *31,* pp.149-160, 2001.

[6] Shimul Shil, A.K.M. Mahfuzur Rahman, "A Decentralized Key Database for Overlay Identification", Master's Thesis, School of Electrical Engineering, Blekinge Institute of Technology, Karlskrona, Sweden.

[7] Hofstatter Q., Zols S., Michel M., Despotovic Z., and Kellerer W., "Chordella – A Hierarchical Peer-to-Peer Overlay Implementation for Heterogeneous, Mobile Environments", *8th International Conference on Peer-to-Peer Computing,* Inst. Of Communication Networks, Munich University of Technology, Sep 2006.

[8] Z. Xu, Y. Hu, "SBARC: A Superpeer Based Peer-to-Peer File Sharing System", Computer and Communications, Eight IEEE International Symposium on Digital Object Identifier, 2003.

[9] R. Farha, K. Khavari, N. Abji and A. Leaon-Garcia, "Peer-to-Peer Mobility Management for all-IP Networks", IEEE International Conference on Communications, pp. 1946-11952, June 2006.

[10] P. Zhuo, D. Zhenhua, Q. Jian-jun, C. Yang and L. Ertao, "HP2P: A hybrid Hierarchical P2P Network", *1st IEEE International Conference on Digital Society,* pp. 18-18, 2007.

[11] K.Sethom, H. Afifi, Guy Pujolle, "A Distributed Architecture for Location Management in Next Generation Networks", IEEE Wireless communication 2005.

[12] C. Perkins, "IP Mobility Support", RFC 2002, Internet Engineering Task Force, October 1996.

AUTHOR'S PROFILE

Engr. Waqas Ahmed Imtiazis a Pakistani citizen by origin. He got his MSc in Electrical Engineering from Blekinge Tekniska Hogskola, Sweden in 2010, and BSc Electrical Engineering from the University of Engineering and Technology Peshawar, Pakistan in 2007. He is currently working as a Lecturer at IQRA National University Peshawar. His research interest includes Seamless Mobility, Optical Communication, Radio on Fibre and Renewable energy for Telecommunication Systems.