Locality-aware Chord over Mobile Ad Hoc Networks

Sonia Gaied Fantar
Research Unit Prince
ISITC Hammam Sousse
University of Sousse
Tunisia
Email: sonia_f_g@yahoo.fr

Habib Youssef
Research Unit Prince
ISITC Hammam Sousse
University of Sousse
Tunisia
Email: habib.youssef@fsm.rnu.tn

***Abstract*—DHT-based Peer-to-Peer (P2P) systems construct an overlay network over the physical network where neighborhood relations are determined randomly using a hashing scheme. This usually results in a mismatch between the P2P overlay and the physical network, leading to high latencies and communication overheads. In this paper we propose a topology-based nodeId assignment to solve the topology mismatch for Chord. In this work, we change the hash-based overlay construction of Chord with physical location based overlay construction, where locations are determined using the Global Network Positionning (GNP) system. The GNP-based Chord is then deployed over a MANET and compared with original DHT-based Chord Experimental results show that GNP-based Chord improves the traditional DHT-based Chord in terms of average number of overlay hops (18%), the total number of overhead messages exchanged(4,3%), end-to-end path latency, and success rate.**

I. INTRODUCTION

With the increasing number of wireless devices, the importance of mobility management in future mobile networks is growing. Traditional mobility management approaches are based on client/server paradigms, and suffer from their well-known shortcomings (single point of failure, congestion, bottlenecks). With the success of Peer-to-Peer (P2P) for file sharing applications, its benefits can be brought into new mobility management schemes to improve their scalability, robustness, availability, and performance.

Locality-awareness is one of the essential characteristics for P2P systems, especially for Mobile Ad-hoc Networks (MANETs), which build and operate their topology independently of the underlying physical network topology.

None of the popular P2P applications (such as Chord, CAN, Tapestry, and Pastry) takes physical network topology into consideration. The selected neighbor host in a randomly constructed overlay network may actually be on the opposite side of the globe when the same file can be serviced by a nearby peer.

Previous research has focused on presenting locality-aware algorithms, in which locality can be defined by different network metrics. In this paper, we focus on introducing the locality on structured peer-to-peer algorithm in a MANET and study the impact of locality on P2P application via Mobile Environments.

One important issue to consider locality in Chord for MANET is to attribute peer identifiers by choosing physically closer nodes as logical neighbors. Thus, to choose physically closer nodes, we can benefit from predicting internet network distances. Among several categories of approaches that predict internet network distances, the coordinates based approaches may be the most promising. Several approaches have been proposed among which GNP [1] [2].

Specifically, we propose to use coordinates-based approaches for network distance prediction in Chord architecture in a MANET.

The main idea is to ask peers to maintain their coordinates that characterize their locations in the Internet such that network distances can be predicted by evaluating a distance function over hosts coordinates. Coordinates-based approaches fit well with the peer-to-peer architecture because when a peer discovers the identities of other peers in a peer-to-peer application, their pre-computed coordinates can be piggybacked, thus network distances can essentially be computed instantaneously by the peer.

Our contribution is to allow participating mobile peers in Chord to collaboratively construct an overlay based on physical locations. And, to potentially benefit from some level of knowledge about the relative proximity between the peers, the Global Network Positioning approach (GNP) is integrated into Chord to capture physical location information of network peers.

This paper is organized as follows. Section II briefly overviews related work. In section III, we present MANETChordGNP scheme, related concepts and main processes. Simulation results are presented in section IV. We conclude in section V.

II. RELATED WORK

There has been little work on using P2P concepts for mobility management in all-IP networks. However, the issues of P2P and mobility management have been extensively studied separately.

P2P concepts for mobility management in all-IP networks were introduced in [3], where the P2P overlay of Mobility Agents is based on Chord structured topology [4].

978-1-4244-4624-7/09/$25.00 ©2009 IEEE

Chord was chosen because of its simplicity, provable correctness, and performance. However, several modifications were introduced to the original Chord so that it becomes suitable for P2P mobility management in all-IP networks. This new Chord approach, referred to as m-Chord, assigns two identifiers for each mobile node (MN): a permanent identifier and a temporary identifier. The Temporary Identifier is related to the temporary IP address that a mobile node (MN) acquires in a foreign network when moving. The Permanent Identifier does not depend on the location of the MN at a particular point in time. The temporary identifier is obtained by hashing the temporary IP address obtained by the MN when moving to a new foreign network. The hashing is largely oblivious to the actual physical topology so that two neighbor nodes can be physically located arbitrarily far from each other. This can lead to a large overlay stretch as subsequent overlay hops can literally crisscross the physical network. MA-Chord [5] is a new approach for MANET with DHT based unicast scheme. It combines Ad hoc On-Demand Distance Vector (AODV) routing [6][**?**] and Chord overlay routing at the network layer to provide an efficient primitive for key-based routing in MANETs.

To exploit physical locality in its overlay, MA-Chord uses Random Landmarking [7]. Instead of having fixed land mark nodes, which simply are not available in MANETs, fixed land mark keys are used. Clustering methods reduce the relaying communication overhead in MANETs and provide for a more efficient hierarchical network topology.

In [8], Farha et al. performed an extensive comparative analysis of all-IP mobility mechanisms, and evaluated the benefits of each proposed solution. Currently, the most popular solutions are network layer approaches built around Mobile IP [9], and application layer approaches built around SIP Mobility [10].

Ekta [11] is a popular approach that proposes the integration of a conventional DHT with an Ad hoc routing protocol to provide indirect routing in MANETs. Ekta, unlike MA-Chord, is based on Pastry [12], and it also uses DSR [13] route discovery phase.

III. MANETCHORDGNP ARCHITETCURE

In recent years, research on locality-awareness in mobile P2P systems has received much attention. Using the Chord lookup protocol, the peers are assumed to be uniformly distributed on the ring. In particular, there is a base hash function which maps peers, based on their IP adresses, to points on the circle, i.e. that it maps identifiers to essentially random locations on the ring.

This Section introduces MANETChordGNP, a Chord architecture designed for the use in MANETs. MANETChordGNP considers physical locality and integrates the functionality of a DHT and AODV routing protocol.

Network coordinate systems can be used to define locality, which maps high-dimensional network measurements into a location in a geometric Euclidean space by associating each node with a virtual coordinate in that space. Among the recently described Internet coordinate systems is GNP [1][2]. In our approach, nodes are sorted on the ring on the basis of their coordinates given by GNP.

General Network Positioning (GNP) picks particular nodes to act as landmarks which serve as points of reference for other nodes that wish to embed themselves into the coordinate space. This approach is implemented using the application layer DHT Chord [4] in which AODV routing protocol designed for ad hoc mobile networks is used. The MANETChordGNP peers join and leave frequently and unpredictably. In fact, Another specific characteristic of MANETChordGNP system concern peer participation that is each peer joins and leaves the system at any arbitrary time.

A. SYSTEM DESIGN

Due to node mobility and the lack of a central infrastructure, conventional routing protocols in MANETs have to resort to flooding packets during their route discovery process at one time or another. However, these route discovery/maintenance broadcasts create an immense overhead and, thus, constitute a key scalability bottleneck. For this reason, our approach was explicitly designed to avoid broadcasts whenever and wherever possible.

MANETChordGNP integrates the reactive ad hoc routing protocol AODV [6] and the application layer DHT Chord [4] to provide light-weight and scalable routing functionality.

In standard DHTs, two overlay neighbours can be located arbitrarily far from each other in terms of the underlying physical network. This can lead to a large ratio between the physical route length travelled during an overlay key lookup compared to the direct physical path from the source to the eventual target node as subsequent overlay hops can literally crisscross the physical network. Therefore, it is essential for any DHT in the context of a MANET to consider physical locality [14].

In Chord, each peer has a finger table, which implements searches in O(log n) query forwards [4]. Peers are arranged on a ring, where their positions are determined by the hash of their IP addresses. Usually, only one query message is necessary, since it is redirected to the correct peer using the finger tables. However, this message is frequently lost on MANETs, due to the instability of the wireless medium. Also, the consistency of the ring on MANETs is hard to maintain, due to node failure and mobility.

MANETChordGNP utilizes the concept of coordinates based approaches [1] [2] to predict internet network distances. Thus, two nodes that are physically close to each other are also likely to be ”close” to each other in the overlay.

GNP gives best results for sparse landmarks (short distances or for large numbers of nodes and few landmarks).

MANETChordGNP is proposed to enable the scalable computation of geometric host coordinates in the Internet. In the first part, a small distributed set of hosts called Landmarks first compute their own coordinates in a chosen geometric space. The Landmarks coordinates serve as a frame of reference and are disseminated to any host who wants to participate. But since the Landmarks coordinates are only used as a frame of reference in GNP, only their relative locations are important, hence any solution will suffice. When a re-computation of Landmarks coordinates is needed over time, we can ensure the coordinates are not drastically changed if we simply input the old coordinates instead of random numbers as the start state of the minimization problem. Once the Landmarks coordinates, are computed, they are disseminated, to any ordinary host that wants to participate in MANETChordGNP.

Therefore, when one of the current landmark nodes fails or moves, another node (whose overlay id is now closest to the landmark key) will automatically assume its role. In the second part, equipped with the Landmarks coordinates, any end host can compute its own coordinates relative to those of the Landmarks.

In a mobile network, nodes can join, move, and leave at any time. The main challenge in implementing these operations is preserving the ability to locate every key in the network. To achieve this goal, node joining and leaving in MANETChordGNP is handled like Chord does. The difference is that in Chord, A node's identifier is chosen by hashing the node's IP address, but MANETChordGNP provides temporary location-aware identifier assignment function. This function assigns temporary peer's identifier designating its position in the virtual ring. Each node in a MANETChordGNP network assigns itself a temporary unique id (as a function of its geometric coordinates in the Internet), which defines its provisional logical position on the virtual overlay ring.

B. Routing Tables

MANETChordGNP maintains two different routing tables: a standard AODV routing table for physical routes from a node to specific target nodes, as well as a Chord routing table.

1) Chord Routing Table: Each node n, maintains a routing table with 160 entries, called the finger table. The ith entry in the table at node n contains a reference to the first node, s, that succeeds n by at least $2^{(i-1)}$ on the identifier circle, i.e., $s = successor(n + 2^{(i-1)}), where 1 \leq i \leq 160$ (and all arithmetic is modulo 2^{160}). The node s is called the ith finger of node n, and denoted by n.finger[i].node. The first finger of n is its immediate successor on the circle.

2) AODV Routing Table: To carry out a concrete overlay hop, a MANETChordGNP node also maintains a standard AODV routing table. It includes for specific physical destinations the next (physical) hop address as well as, for each such route a sequence number.

C. Joining the system

To allow MANETChordGNP to grow incrementally, a new node that joins the system must derive its own coordinates that characterize its location in the Internet to be allocated in the ring. This is done by the GNP coordinates based approache. In GNP, the Internet is modelled as a D-dimensional geometric space. Peers maintain absolute coordinates in this geometric space to characterize their locations on the Internet. Network distances are predicted by evaluating a distance function over peers' coordinates. A small distributed set of peers known as landmarks provide a set of reference coordinates. Peers measure their latencies to a fixed set of landmark nodes in order to compute their coordinates. While the absolute coordinates provide a scalable mechanism to exchange location information in a peer-to-peer environment, the GNP scheme uses distance measurements to a fixed set of landmarks to build the geometric model.

The process takes five steps:

1) First the new Node n must find a Node n0 already in the MANETChordGNP.
2) Next, using the Chord routing mechanisms, Node n0 uses its finger table to find the successor of n.
3) n sets its successor to the value returned by n0.
4) Each node in the system periodically performs stabilization. It is during this process that newly joined nodes are recognized. This process basically involves each node checking that its successors and predecessors have not changed. If a newly joined node lies between two nodes, they each update their successor values to reflect the change.
5) Finally, each node in the system is also responsible for maintaining its finger table by performing stabilization on each node in the table. This is also how a new node establishes its finger table.

When a node joins the MANETChordGNP network it will be placed in the ring by choosing physically closer nodes as logical neighbors. The successor pointers of some nodes will have to change. It is important that the successor pointers are always up to date otherwise the correctness of lookups is not guaranteed. The MANETChordGNP protocol uses a stabilization protocol running periodically in the background to update the successor pointers and the entries in the finger table.

D. Routing

MANETChordGNP routes packets based on an overlay id but the final (physical) target node is usually unknown. It does so by integrating overlay and physical routing. Therefore, when a MANETChordGNP node wants to send a packet to a specific key, it consults its Chord routing table to determine the closest prefix match, as stipulated by standard Chord. Next, it consults its AODV routing table for the physical route to execute this overlay hop. Intermediate nodes on the physical path of an overlay hop consult their AODV table for the corresponding next physical hop. This process continues until the packet reaches the eventual target node that is responsible for the packet key, i.e. whose overlay id is the numerically closest to the packet key.

MANETChordGNP integrates overlay and physical routing. Therefore, when a node receives a packet, it can principally be due to the following two situations:

1) The node could be the target (i.e. the physical destination) of an overlay hop. In this case, the node needs to determine the next overlay hop using standard Chord routing. Again, it will then consult its AODV routing

table to determine the physical route to the destination node of the next overlay hop.

2) The node could be an intermediate node on the physical path of an overlay hop that is being carried out. Now, the node will behave like a regular AODV node. It will consult its AODV routing table to determine the next physical hop on the path toward the destination of this overlay hop and then forward the packet to it.

This process continues at each intermediate node until the packet eventually arrives at the node that is currently responsible for the packet key.

Due to the dynamic overlay IDs in MANETChordGNP networks, another special routing situation could occur. Some node A might change its overlay ID because it has joined a new position. In this case, to leave the former position in the ring, a node can give its keys to its successor and then inform its predecessor. The successor and predecessor nodes then update their finger tables and successors lists.

MANETChordGNP ensures also that each node successor list is up-to-date. It does this using a "stabilization" protocol that each node runs periodically in the background and which updates Chord finger tables and successor pointers. Chord protocol relies on the fact that each node maintains a successor list of size r, containing the node's first r successors. Even MANETChordGNP, when the successor node does not respond or fails, the node simply contacts the next node on its successor list. Assuming that each node fails with a probability p, the probability that every node on the successor list fails is pr. Increasing r makes the system more robust. By tuning this parameter, any degree of robustness can be achieved.

E. Mobility

When using AODV, the node's IP address remains unchanged even though the node moves in the network. That means node mobility is transparent to the upper layers and no mobility management is needed. In extreme cases node movement can however cause loss of connectivity. To the overlay layer this will appear like some nodes have left the network or the network has split. These are recoverable errors that the Chord protocol can handle on its own.

IV. EXPERIMENTAL RESULTS

In this section, we present results from a series of simulations, in which we compare the conventional Chord protocol and MANETChordGNP in MANET environnement.

A. Simulation setup

To evaluate the performance of MANETChordGNP, we implemented MANETChordGNP using OverSim [15]. We run both Chord and MANETChordGNP overlays in ad-hoc scenarios employing AODV [6] as a routing protocol. The OverSim instances communicate with each other over the 802.11 interfaces to build a logical Chord ring.

We implemented Chord complete set of functionalities, including the protocols necessary for predicting internet coordinates and mobile routing. Regarding Chord simulation parameters, the finger table is updated every 5s and stabilize runs every 10 s. PING messages (used for GNP) are sent every 10 s. The nodes move at a speed of 9 m/s. Peer locations are determinated using Global Network Positioning (GNP) system.

The original Chord supports only SHA-1 hashing algorithm to assign identifiers to peers. In this paper, the Chord code was extended to assign identifiers based on GNP coordinates. Our modifications to Chord are essentially as described in Section 3. We changed Chord identifier assignment function to implement coordinates computation engine that takes pre-measured distance matrices as inputs and produces the GNP coordinates and predicted distances for the peers in the distance matrices.

The GNP program expects to find two distance data files in the execution directory matrix and combined. The file matrix must contain a symmetric distance matrix. Assuming that there are N peers, with IDs 0,1,...,N-1, then the first line has distances 0 to 0, 0 to 1, 0 to 2, etc; the second line has distances 1 to 0, 1 to 1, 1 to 2, etc. The file combined has N+1 columns, each line corresponds to one target peer. The first column is the IP address of the target. The following N columns are the distances between the target and the N measurement peers in matrix, from ID 0 to N-1.

After running GNP, many output data files will be generated. Among them, a file that contains information about the target peers in combined. At the end of the file, the IP addresses and coordinates of the target peers are printed. Next, we changed the type ID which represents the identifier of a node in Chord, by GNP-ID, extracted from the file generated by GNP.

To compare the performance of both Chord and MANETChordGNP, 10 objects have been published randomly. Each node publishes 2 random objects from the set of all published objects. Each node then locates 2 objects, chosen randomly from the set of all published objects. In experiments, nodes are sorted on the ring on the basis of their coordinates given by GNP.

B. Simulation results

1) Overlay hop number: We produce several object publishing scenarios. In this series of simulation, we analyze the performance of MANETChordGNP when the network size varies from 50 to 200 nodes. The main purpose of locality-awareness overlay construction in Mobile Ad Hoc Networks is to take physical network topology into consideration. It shows that the number of overlay hops in MANETChordGNP gets reduced (see Table I). By increasing the number of nodes, the hop count of Chord and MANETChordGNP increases.

As Table I shows, MANETChordGNP resulted in lower average hop count (reduction by 24%) compared to Chord. In addition, even for the average number of hops per query message, MANETChordGNP results in an average reduction by as much as 13% (Table II). Table II shows the average hop length by each protocol in query message routing. From the Table II, we can see that the MANETChordGNP have

Number of nodes	Average Hop Number Chord	Average Hop Number MANETChordGNP	Average Hop length (Km) Chord	Average Hop length (Km) MANETChordGNP
50	4,3796	3,9438	28,7	25,2
100	7,3212	7,1478	24,5	14,1
150	11,3326	10,2119	13,3	8,5
200	15,1045	14,5434	9,3	8,8

TABLE I
AVERAGE NUMBER OF OVERLAY HOPS PER OBJECT PUBLISHING IN A MANET WITH VARYING NETWORK SIZES

smaller hop number than the Chord. It also indicates that MANETChordGNP is much more efficient in terms of Hop length. In fact, on the Chord overlay, for each hop peers route a message to the next intermediate peer that can be located very far away with regard to physical topology of the underlying network. This can result in a high network delay and unnecessary long-distance network traffics for a very short logical path, which are much longer in terms of Hop length than MANETChordGNP.

Number of nodes	Average Hop Number Chord	Average Hop Number MANETChordGNP	Average Hop length (Km) Chord	Average Hop length (Km) MANETChordGNP
50	5,1231	4,9922	25,3	23,2
100	10,5545	10,0876	16,2	13,1
150	14,8936	13,8439	10,5	9,8
200	20,1615	19,1365	4,2	3,5

TABLE II
AVERAGE NUMBER OF OVERLAY HOPS PER QUERY MESSAGE ROUTING IN A MANET WITH VARYING NETWORK SIZES.

In summary, MANETChordGNP is superior to Chord in query message routing and in objects publishing. Since objects are chosen randomly, hop count is high unless choosing physically closer nodes as logical neighbors, but the Hop length. This is because MANETChordGNP uses short and recently updated routes, and, therefore, it achieves much better overlay hop compared to the Chord protocol.
This suggests that the implementing a DHT in MANETs by integrating prediction of internet coordinates is a correct way.

2) Total number of messages: Figure 1 depicts the total number of messages that have to been exchanged among all nodes during the object publishing process in both Chord and MANETChordGNP.

As can be seen, Chord object publishing introduces a significant overhead. With the incorporation of peer location during the construction of the overlay P2P network over the MANET, the number of overhead messages exchanged has been decreased compared to DHT-based Chord. In Chord, the selected peer neighbor may actually be on the opposite side of the globe when the same file can be serviced by a peer a few hops away.

Figure 2 shows the total number of overhead messages generated during 1h of simulation. GNPChord generated lower overhead than Chord for all network sizes. A lower overhead means also an improvement of routing efficiency.

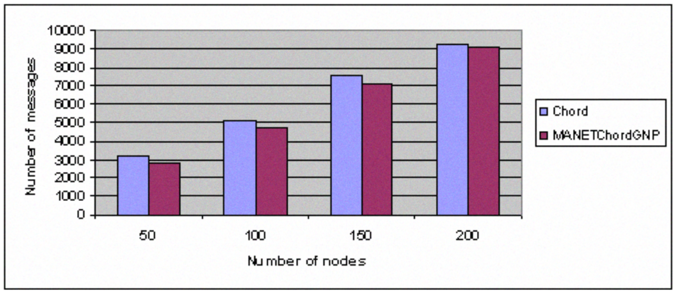

Fig. 1. Total number of messages exchanged

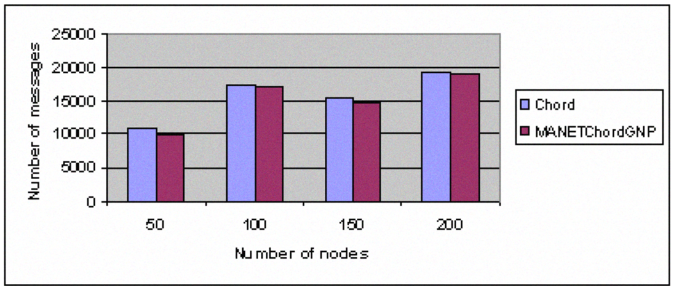

Fig. 2. Total number of messages exchanged during 1h simulation

The simulation pf networks of varying sizes (50, 100, 150 and 200 nodes) showed an improvement of 6,5% on average of the total number of messages exchanged during 1h simulation. The lower number of messages generated by MANETChordGNP comes as no surprise since, with GNP-based overal design two nodes that are physically close to each other are also likely to be close to each other in the overlay.

3) End-to-End Path Latency: This section presents latency measurements obtained from implementations of Chord and MANETChordGNP. We produce several lookup scenarios with the number of nodes increased from 50 to 200 nodes. Then, we generate a simulation topology for these scenarios and evaluate the end-to-end path lookup latencies. Figure 3 shows that the query latency in MANETChordGNP is comparable to original Chord. But, the latency with MANETChordGNP is independent of network size. In Fact, latency decreases when the number of nodes is raised. This is because physical locality is considered. The experiment shows that only when the number of nodes is small, the latency in both systems is the same. In addition, the margin between the lookup latency of Chord and MANETChordGNP gets larger with increasing network sizes. Further, the reader can see, that with network sizes larger than 100 nodes, the latency of MANETChordGNP remains low and almost insensitive to network size, which confirms our prediction.

4) Success rate: The success rate represents the number of data packets received by the destination nodes divided by the number of data packets transmitted by the source nodes. As shown in Figure 4, the Chord success rate of is very low for

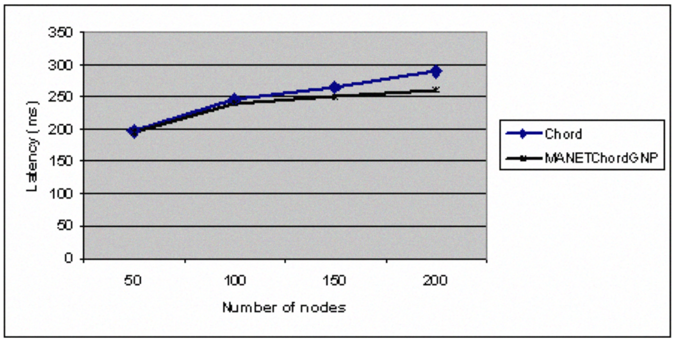

Fig. 3. End-to-end routing latency with the number nodes increased from 50 to 200.

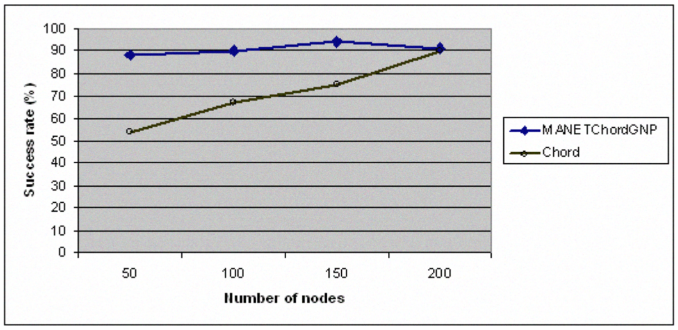

Fig. 4. Success Rate and network size

small network size and increases with larger network size. In contrast, the success rate of MANETChordGNP is high for all network sizes (above 90%). Further, success rate performance of MANETChordGNP seems to be insensitive to network size.

V. CONCLUSIONS

With the increasing proliferation of mobile wireless devices, it is becoming more and more interesting to build efficient distributed network application for MANETs.

In this work we studied (through simulation) the effect of position-based overlay P2P network construction on Chord performance. Experimental results show noticeable performance improvement with respect to the average number of overlay hops, the total number of[width=2.5in] messages exchanged, end-to-end path latency, and success rate.

REFERENCES

[1] T. S. E. Ng and H. Zhang, "Predicting internet network distance with coordinates-based approaches," in *INFOCOM*, 2002.

[2] H. Z. T.S. Eugene Ng, "Towards global network positioning," in *ACM SIGCOMM Internet Measurement Workshop*, San Francisco, CA, Novembre 2001.

[3] N. A. A. L. Ramy Farha, Khashayar Khavari, "Peer-to-peer mobility management for all-ip networks," in *in Proceedings of the IEEE International Conference*, vol. 5, June 2006, pp. 1946–1952.

[4] I. et. al., "Chord:a scalable peer-to-peer lookup service for internet applications." MIT, Tech. Rep. TR-819, 2001.

[5] Q. Meng and H. Ji, "Ma-chord: A new approach for mobile ad hoc network with dht based unicast scheme," in *Wireless Communications, Networking and Mobile Computing*, ser. Issue, I. C. on Volume, Ed., vol. WiCom 2007, 21-25 Sept 2007, pp. 1533 – 1536.

[6] C. E. Perkins and E. M. Royer, "Ad hoc on-demand distance vector routing," in *IEEE WMCSA*, February 1999.

[7] T. Z. R. Winter and J. Schiller, "Random landmarking in mobile, topology-aware peer-to-peer networks," in *FTDCS*, May 2004.

[8] R. Farha and A. Leon-Garcia, "Mobility analysis for all-ip networks," in *in Proceedings of the IEEE Wireless Communications and Networking Conference*, vol. 3, Mar. 2005, pp. 1395–1401.

[9] C. Perkins, "Ip mobility support for ipv4," RFC 3344, Aug. 2002.

[10] H. Schulzrinne and E. Widlund, "Application-layer mobility using sip," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 4, pp. 47–57, July 2000.

[11] S. M. D. H. Pucha and Y. C. Hu, "Ekta: An efficient dht substrate for distributed applications in mobile ad hoc networks," in *In Proc. of IEEE WMCSA*, December 2004.

[12] A. Rowstron and P. Druschel, "Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems," in *in Proceedings of Middleware*, Heidelberg, Germany, Nov 2001, p. 329350.

[13] D. B. Johnson and D. A. Maltz, "Dynamic source routing in ad hoc wireless networks," in *Mobile Computing*. Kluwer Academic Publishers, 1996, pp. 153–181.

[14] T. Zahn and J. Schiller, "Madpastry: A dht substrate for practicably sized manets," in *In Proc. of the 5th Workshop on Applications and Services in Wireless Networks (ASWN 2005)*, Paris, France, June 2005.

[15] I. Baumgart, B. Heep, and S. Krause, "OverSim: A Flexible Overlay Network Simulation Framework," in *Proceedings of 10th IEEE Global Internet Symposium (GI '07) in conjunction with IEEE INFOCOM 2007, Anchorage, AK, USA*, May 2007.