

# Scalable Mobility Management in Large-Scale Wireless Mesh Networks

Shih-Hsien Yang and Lichun Bao

Donald Bren School of ICS, University of California, Irvine, USA

**Abstract**—Wireless mesh networking (WMN) is an economic means to provide the last-mile Internet access service through ad hoc peer-to-peer communication links. However, WMNs suffer from scalability, performance degradation and service disruption issues due to inherent network mobility. In this paper, we present a mobility management protocol, called HDR (Hierarchical Directory Resolution), using a hierarchical Distributed Hash Table (DHT) approach. Different from prior DHT solutions that were based on consistent hashing algorithms, HDR uses a novel NCR (neighbor-aware contention resolution) algorithm to maintain the DHT lookup functions. Simulation results validate the correctness and advantages of our proposed protocol.

## I. INTRODUCTION

Wireless mesh networking (WMN) is an economic and convenient way to provide the last-mile Internet access service through ad hoc peer-to-peer communication links. However, without systematic network management, WMNs suffer from scalability, performance degradation and service disruption issues due to inherent network mobility. In this paper, we discuss the mobility management problems and propose a solution to these problems using Distributed Hash Tables (DHTs).

DHT has been a widely adopted approach to provide scalable building blocks for large-scale wireless network management, especially in P2P distributed applications. Since its inception, many different application-specific architectures have been proposed, including CAN [10], Pastry [11], Chord [13], Kademila [8], Tapestry [18], VRR [3]. Many hierarchical DHT schemes were proposed in the past as well, such as Canon [4], Cyclone [1], MADPastry [17] and HIERAS [16].

Existing DHT applications were all based on the concept of consistent hashing [6], in which participating network nodes maintain a single flat virtual table, and each node is responsible to serve a segment of the table in the directory service. In practice, DHT suffers from uneven load balancing problem, because the segment lengths are different for nodes serving the DHT, which is a relatively static overlay structure. In addition, neighbors on the virtual rings could physically sit across the whole network, thus incurring long delays and large overheads during DHT lookup services. Localized and scalable topology-dependent DHT mechanisms are highly desirable in WMNs for performance reasons.

We propose a new approach, called HDR (Hierarchical Directory Resolution), to the directory lookup and update

services. HDR addresses on the fairness issue by using a new directory mapping algorithm, called NCR (neighbor-aware contention resolution). In HDR, instead of maintaining a single global virtual table for all search items, each search item forms its own table lookup index using the NCR algorithm. The NCR algorithm generates a priority value for each directory server based on a hash function with the lookup key value and the directory server's ID as inputs. The lookup request is sent to the directory server that has the highest priority value among the server set.

Similar to the home agent functions in Mobile IP [9], HDR is used to find the current routing address of the destination node, only in a distributed fashion. In simple terms, mesh nodes in HDR serve as the directory server, and the destination IP address is used as the directory lookup key to find the current routing information of the destination. With the priority-based directory lookup mechanism offered by NCR, we demonstrate that HDR achieves fairness in load balancing the directory lookup, and that HDR can be organized into a hierarchical structure to achieve efficiency and scalability in large-scale WMNs.

The remainder of this paper is organized as follows. We present our system operation model of HDR in Section II, and specify our directory mapping algorithm in Section III. Section IV describes the mobility management mechanisms. Section VI concludes this paper.

## II. SYSTEM OPERATION MODEL

### A. Architecture

In most common deployments, WMNs are utilized to provide WDS (wireless distribution system) to mobile clients for Internet access purposes. Following the terminology defined in IEEE 802.11s, we categorize WMN nodes into the following three architectural components:

- *Mesh point (MP)*, which is a router that provides packet forwarding functions for traffic originated from other mesh nodes and itself.
- *Mesh access point (MAP)*, which is a mesh point that also works as a network access point (AP) for mobile client stations.
- *Mesh portal point (MPP)*, which is a mesh point that also serves as a bridge device between the mesh network and the Internet.

### B. Hierarchical Addressing

In the Internet, IP addresses serve both as routing information and transport layer ID. Such violation of layering se-

mantics causes potential network service disruptions in mobile wireless networks, in which mobile stations could constant change their network attachment points. In order to avoid such disruptions, mobile stations have to retain the same IP addresses when moving. This implies WMNs have to maintain per-IP routing information among the mesh points.

However, in large-scale WMNs with potentially large number of mesh points and mobile stations, per-IP routing is not scalable, and could incur frequent routing information updates across the network.

In HDR, we introduce a new addressing method based on MAC addresses, called *VMAC (virtual MAC)*. Similar to IP addresses, VMAC defines a topology decedent subnet addressing scheme that allows us to maintain a compact routing information table for reaching a large number of network nodes.

In VMAC, the 48-bit IEEE 802 MAC address space is organized into hierarchies, in which every 4-bit address space is mapped to one level of hierarchy. Consequently, we could have 12 levels of hierarchies in the address space, in which each level contains 16 different sub-address spaces. However, we exclude all 0's and all 1's addresses in each level as they indicate information other than addressing. All 1's address is reserved for broadcast, and all 0's address indicates that the corresponding hierarchical level does not exist. That is, every layer of addressing hierarchy provides 14 possible addresses.

Using 24-bit MAC address space as a simplified example, address  $0 \times 001234$  means that there are four levels of address hierarchies, and the top two levels do not exist. Note that we do not need to assign VMAC to mobile stations because they can be reached at the last hop using ARP.

In summary, the hierarchical addressing architecture provides twelve addressing levels among the mesh nodes, each with 4-bit addressing space. Therefore, the total addressing capacity in HDR addressing architecture is  $14^{12} = 56$  trillion addresses, large enough for routing purposes in any WMNs in the foreseeable future.

### C. VMAC Routing

With the application of VMAC addresses, we have inserted a virtual shim layer between the physical MAC address space and the network-layer IP address. The benefit of using VMAC is that we no longer need to change the IP address when mobile stations change their network attachment points from time to time, while we are still able to keep the topology dependent scalable routing capability using VMAC.

Now suppose that a MP needs to send a packet to a mobile station with VMAC, the forwarding MPs in the WMN can read down the VMAC from the top level of the addressing hierarchy, and layer-by-layer find the next hop to reach the corresponding MP that is closer to the destination VMAC, until the packet reaches the final MAP associated with the destination mobile station.

However, VMAC also introduces an extra layer of indirectivity to reach the destinations, thus requires address translations. The address translation is carried out by HDR.

### D. VMAC Construction Protocol

The VMAC addressing hierarchy is dynamically constructed using routing control messages.

In general, we initialize the VMAC of each MP to all 0's, and the construction of the VMAC hierarchy starts from the bottom level of the addressing space at each mesh point, *i.e.*, bit 3-0 in IEEE 48-bit MAC address space. Using control messages, we group adjacent MPs into clusters according to the neighbor relations. Each MP in the bottom-level cluster is assigned a unique address in the bit 3-0 space, which provides up to 14 addresses.

If more than one cluster is needed to contain all the MPs, the bottom-level cluster again uses the bit 7-4 address space to form cluster of clusters. Therefore, we can gradually build a hierarchical cluster tree, and eventually result in a single cluster at the top of the tree, at which point the hierarchical address space completes construction.

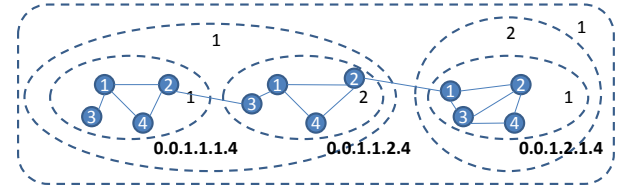


Fig. 1. A results of hierarchical address space construction.

Fig. 1 illustrates an example of address assignment results using a simplified 24-bit address space. As we can see, three MPs with the same label “4” have been assigned different VMAC addresses because of their positions in the address hierarchy. Specifically, there are 4 levels, each of which differentiate nodes between clusters in the same layer, respectively.

The address assignment protocol is driven by messages. When a unique cluster is formed, all members of the cluster know each other to maintain uniqueness of their addresses in the corresponding level of hierarchy. In addition, they can reach each other due to network connectivity offered by the tree structure.

We define three types of messages to communicate between different clusters: *Hello*, *JoinRequest*, and *JoinReply*, and each message contains the following essential information:

- The VMAC address of the clusterheads of all parent clusters, denoted by *CHMAC*.
- The height of the tree that the node is currently belonged to, denoted by *height*.
- The VMAC address of the node, denoted by *vmac*.

The *Hello* message is broadcast periodically by the MPs. Any new MP which just entered the WMN can gather neighbor information by listening to this message, and choose a nearby cluster to join. If it does not receive any *Hello* message for a certain period, it turns into a clusterhead and sends out this message to its neighbors. The *JoinRequest* message is sent when a non-clustered node tries to join a nearby cluster, or when a cluster tree tries to merge into another. The node which receives the *JoinRequest* message will reply with the *JoinReply* message to indicate the request is accepted or rejected.

If a node is elected as the clusterhead for a cluster, it maintains some additional information listed below:

- The number of members in the cluster, denoted by *member*.
- Current available VMAC addresses which can be assigned to new members.
- The mapping between IEEE 802 MAC and VMAC addresses for all cluster members.

The details of reactions of each node during the message exchange processes are described below:

a) *On receiving Hello*: Suppose a node  $v$  receives a Hello message from its neighbor  $u$ , node  $v$  first checks if they are in the same cluster tree by comparing the MAC address of the root nodes. If their root nodes are the same, node  $v$  will simply discard this message. If not, we will try to combine the two cluster trees into one.

If node  $u$ 's tree has more levels than node  $v$ 's tree, node  $v$  will send a *JoinRequest* message to node  $u$  to start the merging, and vice versa. If the height of both tree are the same, we decide which node should initiate the merging by comparing the MAC address of the root node.

b) *On receiving JoinRequest*: If a node  $v$  receives a *JoinRequest* message from a neighbor  $u$ , it indicates that node  $u$ 's cluster tree wants to merge into node  $v$ 's cluster tree. Node  $v$  then forwards this message to its parent clusterhead  $CHMAC_h$ , in which  $h$  is the height of node  $u$ 's tree. The join request will only be granted if the number of members of the parent cluster is fewer than 14, which is the max limit of the cluster members.

If the request is accepted, the parent clusterhead sends a *JoinReply* message to node  $v$  containing an available VMAC address drawn from the 4-bit address space. Node  $v$  helps forward this message to node  $u$ . If the request is rejected, the *JoinReply* message is sent to node  $u$ , containing an invalid VMAC address filled with all '0's.

c) *On receiving JoinReply*: If node  $u$  receives the *JoinReply* message from node  $v$  containing a valid VMAC address, node  $u$  forwards this message to the root node of node  $u$ 's cluster tree. The root node replaces the original VMAC address with the assigned address, and update the VMAC address to the leaf nodes. If the *JoinReply* message indicates that the request is rejected, node  $u$  will start scanning the Hello message from nearby neighbors again, and try to find another candidate for merging.

### E. Proxy ARP

After the hierarchical address assignment is done, we obtain a VMAC address for each mesh point in the WMN. On the other hand, in practical network administrations, we need an ARP entry that maps the mobile station's IP address to its MAC address in order to deliver packets to the mobile station.

Therefore, in order to deliver packets to the mobile stations in WMNs, we use the VMAC address of the corresponding MAPs as the destination MAC addresses in the packets. When the packets arrive at the MAPs, MAPs in turn replace the VMAC with the actual MAC address of the mobile station, and send the packets to it. Such mechanisms are the well-known

"Proxy ARP" protocol. In the reverse direction, the packet forwarding works similarly. Because each mobile station is associated with an MAP, the MAC address translation works for "Proxy ARP" with little extra effort.

However, an unsolved puzzle in the above mechanisms is where to store the ARP table that maps mobile stations' IP to their VMAC addresses. The traditional ARP protocol works by subnet broadcast, which would incur excessive traffic in large-scale WMNs. A DHT based directory service that fulfills the ARP lookup has been extensively applied in recent years to solve such scalability problems, such as SEATTLE [7], VL2 [5], MADPastry [17] and HIERAS [16]. We apply similar ideas, but use different algorithm for the directory server mapping mechanisms.

### III. HIERARCHICAL DIRECTORY RESOLUTION (HDR)

Our directory server mapping algorithm is based on the NCR (neighborhood-aware contention resolution) algorithm [2], and is closely related with the aforementioned hierarchical addressing mechanisms.

The NCR algorithm was originally designed to solve the node election problem for channel access purposes in ad hoc networks using the TDMA scheme. In NCR, each node with a distinct ID are assigned with a priority, derived using Eq. (1).

$$i.prio = \text{Hash}(i \oplus t) \oplus i, \quad (1)$$

in which the node ID is denoted by  $i$ , its priority denoted by  $i.prio$ ,  $t$  denotes the context parameter, such as a time slot number, and the sign ' $\oplus$ ' represents the concatenation operation on its operands. Function  $\text{Hash}(x)$  is a fast message digest generator that returns a random integer on input value  $x$ , and is similar to that of consistent hashing [6]).

In HDR, we find the directory server with the ARP mapping information by searching down the VMAC addressing hierarchy. At each level of the hierarchy, we determine the clusterhead that will be serving the ARP mapping information using NCR. That is, we substitute  $i$  with the IP address of the mobile station, and  $t$  with the level-corresponding 4-bit VMAC address of each clusterhead, and generate the priority for each clusterhead using Eq. (1). The cluster members under the clusterhead that has the highest priority would be serving the ARP mapping. In order to finalize the MP that stores the mapping information, we repeat above process until we get to the VMAC bottom level.

Because of the hierarchical directory server mapping mechanisms, we have designated the 4-bit address space limit so that each address hierarchy level would contain manageable number of cluster members for priority comparisons using NCR.

Note that the directory server finding mechanism always starts from the top-level VMAC hierarchy. Thus, any ARP request that is originated from an MAP will be first forwarded to the top level cluster along parent nodes in different levels, then find the MP that server the ARP mapping.

Figure 2 illustrates an example of how ARP request is routed in the mesh network in HDR.

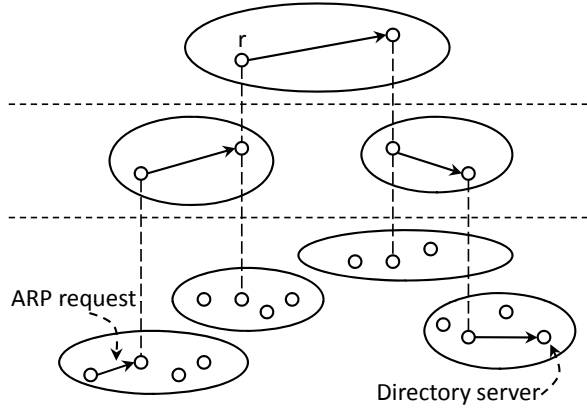


Fig. 2. An example of the path of an ARP request.

After receiving an ARP request message, the directory server generates an ARP reply message, which is directly sent to the requesting MP using VMAC routing mechanisms.

NCR-based directory server mapping is different from consistent hashing based approaches in that it does not maintain a ring-like structure among the directory servers, and results in a more evenly distributed probability for each directory server to provide the directory information in each search item. Thus, HDR provides better load balancing than prior approaches.

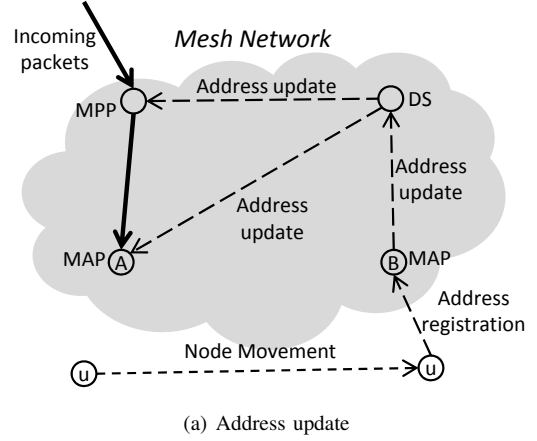
In fact, we could use HDR to resolve other resource names to VMAC addresses, such as 1) the IP address of the Internet gateway, 2) the IP address of the DHCP server, 3) the IP address of the DNS server, or application service names, therefore eliminating the flooding operations in WMNs.

#### IV. MOBILITY MANAGEMENT

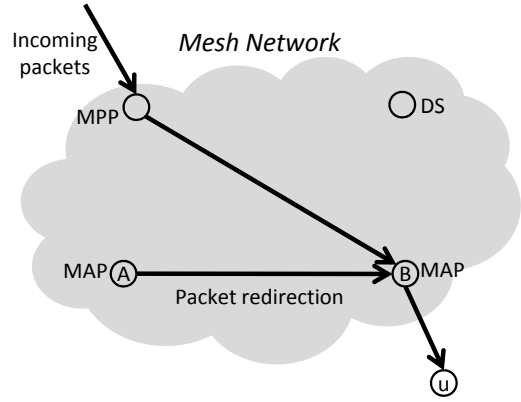
When a mobile station newly joins the mesh network, it registers at its closest MAP with its IP and MAC addresses in order to access the network. Upon receiving the registration request, the MAP starts the HDR protocol to update the mobile station's IP to VMAC mapping at the corresponding directory server. The MAPs will act as proxy ARP servers for the mobile nodes for traffic going through the MAP.

When a mobile station changes its associated MAPs, the ARP directory needs to be updated with a new mapping between the mobile station IP and its currently associated MAP's VMAC address.

Fig. 3 shows an example to explain the handoff mechanism in HDR. In Fig. 3 (a), an existing mobile station  $u$  roams away from its old associated MAP  $A$ , and re-associates with MAP  $B$ . Once MAP  $B$  associates the station  $u$ , it updates the ARP mapping entry of node  $u$  at the directory server  $DS$ . Once the directory server  $DS$  receives the update, it has to check whether node  $u$  was previously associated with any other MAPs. If so, it notifies the previously associated MAPs of the current ARP mapping, so that the ongoing traffic destined to mobile station  $u$  can be redirected the correct MAP. In addition,  $DS$  will also update the ARP mapping at the MPP so that future traffic goes directly to node  $u$ . As shown in



(a) Address update



(b) Packet redirection

Fig. 3. Node handoff example.

Fig. 3 (b), both new packets coming from the MPP and the in-flight packets are delivered the roaming station  $u$ .

#### V. EVALUATIONS

We evaluate HDR performance using simulations. Four sets of simulation scenarios were carried out to gather the performance metrics about the load balancing feature, directory lookup path length, handoff latency, and packet success rate. Specifically, we compare our results of HDR with three other protocols, Chord [12], VRR [3], and Mobile IP [9], respectively under different metrics. We have implemented or utilized different simulation tools to collect the metrics.

##### A. Load Balancing and Scalability

In order to evaluate the address lookup schemes in HDR, we implemented our own custom simulator to evaluate the distribution of address mapping entries in a large-scale WMN using the NCR algorithm. For Chord, we use the simulator provided in [15] to run our simulation scenarios.

To compare the load balancing feature between HDR and Chord, we set up a WMN with  $10^4$  nodes working as directory servers. Then, we insert a specific amount of directory entries, ranging from  $10^5$  to  $10^6$ , at  $10^5$  increments in each round, using HDR and Chord, respectively. In each round, the simulations are repeated 100 times so that we can derive the

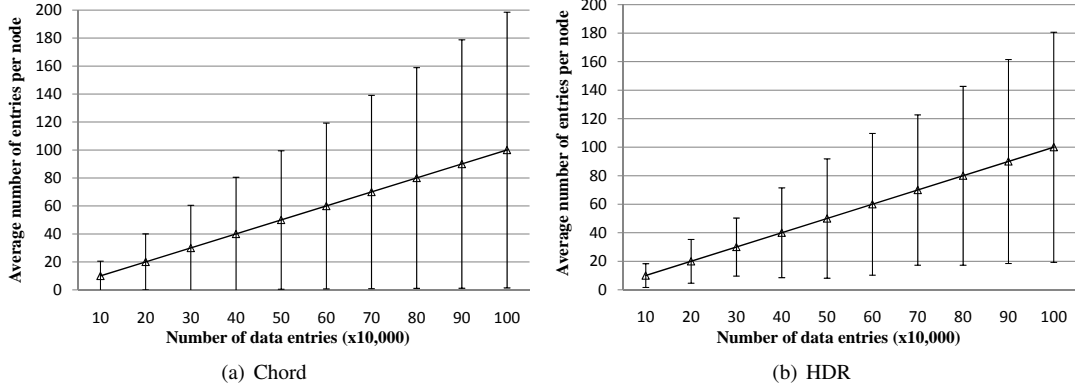


Fig. 4. The mean and standard deviation of the number of entries stored per node in a  $10^4$  node DHT.

average number of entries allocated per directory server, and the standard deviation.

The simulation results for HDR and Chord are shown in Fig. 4. In all cases, the average standard deviation of HDR is about 80% of Chord. The reason that Chord has such scattered distribution is that the directory servers cover fixed section of the DHT each time, as a result, some directory server may store much less entry even when the total number of entries is  $10^6$ . In contrast, the distribution of the address mapping entries in HDR is much even in all cases because of the use of the NCR algorithm.

In the next set of simulations, we compare the lookup path length between HDR, Chord, and VRR with different network sizes. We set the number of DHT entries  $10^6$ , and we vary the number of nodes in the network from  $10^2$  to  $10^5$ .

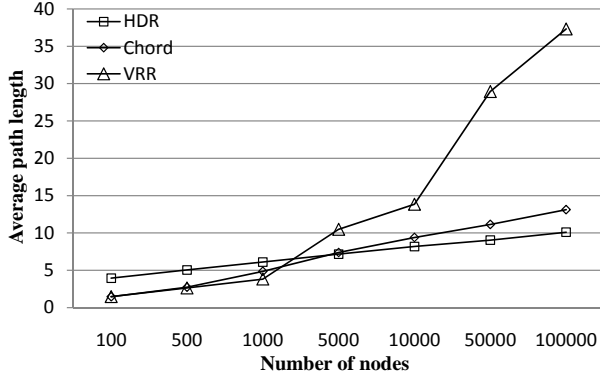


Fig. 5. The directory lookup path length comparison.

Fig. 5 shows the lookup path length feature of Chord, VRR and HDR. Chord and VRR perform better than HDR in small-scale networks. However, when the number of network nodes exceeds 5000, HDR outperforms both of them. The reason for HDR to have a longer path in small-scale networks is because all lookup requests in HDR have to route through the root of the cluster hierarchy. Therefore it takes some additional hops to reach the destination. However, because each cluster in HDR can contain up to 14 members, the height of the cluster tree grows very slowly as the number of nodes increases, thus results in a shorter path in larger scale networks.

### B. Mobility Management

In order to examine the protocol operations of HDR in a more realistic network environment, we implemented our HDR mobility management scheme using the network simulator QualNet 4.5 [14], to compare against the performance of Mobile IP in terms of handoff latency and packet success rate.

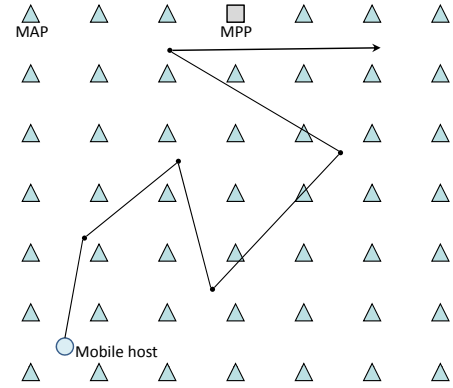
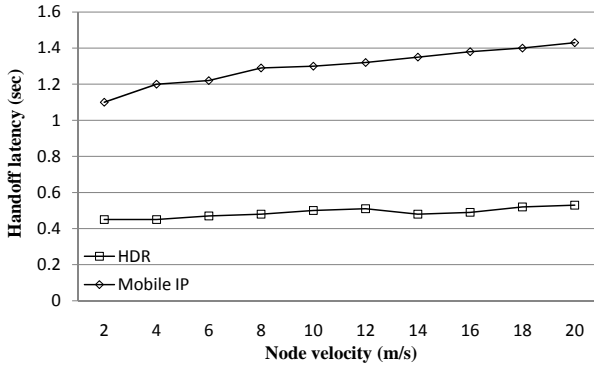


Fig. 6. The grid topology with random waypoint moving pattern.

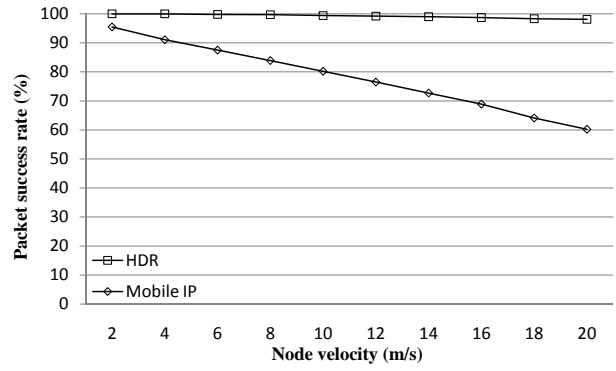
Fig. 6 shows the grid-like WMN network topology in our simulations. The distance between adjacent MPs is 200m, and the transmission range of each wireless station is 250m. The mobile host (MH) follows the random waypoint mobility model as shown in the figure. The simulation ends when the MH moved 3000m from its starting position. A static host outside the mesh network sends CBR (constant bit rate) traffic to the MH, and the CBR flow rate is set to 80Kbps. We collect the performance metrics at various speed of the MH, from 2m/s to 20m/s with 2m/s increments.

In the simulations for evaluating HDR, we let each MP, MAP or MPP in the network store the address mapping entries. We also assume that the routing is in a steady state before the CBR flow starts. With regard to Mobile IP evaluations, we set up a home agent (HA) outside the mesh network, and the currently associated MAP is selected to work as the foreign agent (FA) for the MH.

Fig. 7 shows the performance of HDR and Mobile IP in terms of handoff latency and packet success rate. Handoff latency is defined for a receiving mobile host as the time



(a) Handoff latency



(b) Packet success rate

Fig. 7. Mobility management comparison

that elapses between the last packet received via the old route and the arrival of the first packet along the new route after a handoff. Packet success rate is defined for a receiving mobile host as the percentage of the number of packets that are successfully received comparing to the total packets that are transmitted during the time period.

As we can see in Fig. 7(a), the handoff latency of HDR remains constant as the node velocity increases. The reason is that the address lookup message for HDR always traverse through the same amount of hops according to the hierarchical tree structure. In addition, since the location server also resides in the mesh network, the time to locate the MH by querying the server is relatively short when compared to Mobile IP, which needs to contact the HA through the Internet in order to update the new location information for the MH.

With regard to packet success rate, HDR achieves almost 100% success rate even in high-speed situations. This is because of the packet redirection feature which helps forward the buffered packets during handoff to the new location of the MH. In contrast, Mobile IP only achieves 60% packet success rate in high velocity scenario since all the packets are lost during the handoff period.

## VI. CONCLUSION

We have presented HDR, a novel routing and mobility management protocol in large-scale WMNs, that provides fair DHT service using the NCR algorithm, and demonstrates scalable routing in large network settings. We described the protocol mechanisms in details, and evaluated HDR in the context of mesh networks of different sizes. The simulation results demonstrate that HDR is fair and efficient in directory lookup and routing services. We believe that HDR is a good alternative routing candidate to compose a mesh network in metropolitan areas, beside other DHT-based directory lookup services.

## REFERENCES

- [1] M.S. Artigas, P.G. Lopez, J.P. Ahullo, and A.F.G. Skarmeta. Cyclone: a novel design schema for hierarchical DHTs. In *P2P '05: Proceedings of the Fifth IEEE International Conference on Peer-to-Peer Computing*, pages 49–56, 2005.
- [2] L. Bao and J.J. Garcia-Luna-Aceves. A New Approach to Channel Access Scheduling for Ad Hoc Networks. In *Proc. ACM Seventh Annual International Conference on Mobile Computing and networking*, Rome, Italy, Jul. 16–21 2001.
- [3] M. Caesar, M. Castro, E. Nightingale, G. OShea, and A. Rowstron. Virtual Ring Routing: Network routing inspired by DHTs. In *Proceedings of the conference on Applications, Technologies, Architectures, and Protocols for Computer Communications (SIGCOMM)*, 2006.
- [4] P. Ganesan, K. Gummadi, and H. Garcia-Molina. Canon in G Major: Designing DHTs with Hierarchical Structure. In *Proc. International Conference on Distributed Computing Systems (ICDCS)*, 2004.
- [5] A. Greenberg, J. Hamilton, N. Jain, S. Kandula, C. Kim, P. Lahiri, D. Maltz, P. Patel, and S. Sengupta. VL2: A Scalable and Flexible Data Center Network. In *Proc. of the ACM SIGCOMM*, 2009.
- [6] D. Karger, E. Lehman, T. Leighton, R. Panigrahy, M. Levine, and D. Lewin. Consistent hashing and random trees: distributed caching protocols for relieving hot spots on the world wide web. In *Proc. of the twenty-ninth annual ACM symposium on Theory of computing (STOC)*, pages 654–663, New York, NY, USA, 1997. ACM Press.
- [7] C. Kim, M. Caesar, and J. Rexford. Floodless in seattle: a scalable ethernet architecture for large enterprises. In *Proc. of the ACM SIGCOMM*, pages 3–14, Seattle, WA, USA, 2008.
- [8] P. Maymounkov and D. Mazières. Kademlia: A Peer-to-peer Information System Based on the XOR Metric. In *Proc. of the 1st International Workshop on Peer-to-Peer Systems*, 2002.
- [9] Charles Perkins. RFC 3344 - IP Mobility Support for IPv4. Technical report, IETF, Jan. 2002.
- [10] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Schenker. A scalable content addressable network. In *Proc. of SIGCOMM*, pages 161–172. ACM Press, 2001.
- [11] A. Rowstron and P. Druschel. Pastry: Scalable, decentralized object location and routing for largescale peer-to-peer systems. In *IFIP/ACM International Conference on Distributed Systems Platforms*, pages 329–350, Nov. 2001.
- [12] A. Rowstron and P. Druschel. Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems. In *Proceedings of IFIP/ACM Middleware*, 2001.
- [13] I. Stoica, R. Morris, D. Karger, M. Kaashoek, and H. Balakrishnan. Chord: A Scalable Peer-to-peer Lookup Service for Internet Applications. In *Proc. ACM SIGCOMM*, San Diego, CA, Aug. 2001.
- [14] Scalable Network Technologies. <http://www.scalable-networks.com>. QualNet Network Simulator.
- [15] The Chord/DHash Project. <http://pdos.csail.mit.edu/chord/>.
- [16] Z. Xu, R. Min, and Y. Hu. HIERAS: A DHT Based Hierarchical P2P Routing Algorithm. In *Proc. of International Conference on Parallel Processing (ICPP)*, 2003.
- [17] T. Zahn and J. Schiller. DHT-based Unicast for Mobile Ad Hoc Networks. In *Proc. of MP2P*, Mar. 2006.
- [18] B. Y. Zhao, L. Huang, J. Stribling, S. C. Rhea, A. D. Joseph, and J. D. Kubiatowicz. Tapestry: A resilient global-scale overlay for service deployment. *IEEE JSAC*, Nov. 2003.