Hatch: The Design of a Hybrid Location Tracking Chain in Internet-based Wireless Mesh Networks

Weiyi Zhao and Jiang Xie
Department of Electrical and Computer Engineering
The University of North Carolina at Charlotte
Email: {wzhao2, linda.xie}@uncc.edu

***Abstract*—Location management in wireless networks serves the purpose of tracking mobile nodes and locating them prior to establishing new communications. However, existing location management schemes in cellular and wireless local area networks cannot be directly applied to wireless mesh networks (WMNs) without non-trivial modifications due to the special properties of WMNs. In this paper, we propose a hybrid location tracking chain (*Hatch*) framework which includes a hierarchical architecture design and a hybrid location tracking scheme for location management in WMNs. Under the *Hatch* framework, a location update scheme is developed that takes the number of hops into consideration. Moreover, the proposed location tracking chain is formed by the routing protocol adopted by WMNs and the length of location tracking can be dynamically changed to improve the performance of packet delivery. We evaluate the proposed location management schemes in different case studies using OPNET simulations, the results of which show the efficiency of our proposed location management scheme in WMNs.**

I. Introduction

Wireless mesh networks (WMNs) have recently emerged to be a promising cost-effective solution for providing large-scale wireless Internet access. An infrastructure-based WMN is comprised of a combination of static mesh routers (MRs) and mobile mesh nodes (MNs) [1]. MRs form the wireless mesh backbone infrastructure and provide connectivity for MNs. Some MRs are gateways with a wired interface that can route traffic to-and-from the Internet. A large variety of applications using WMNs demand real-time, always-reachable IP services.

Location management (LM) is the process by which the current location of an MN is determined. It consists of two procedures: location update (LU) and packet delivery (PD). When an MN does not have active communications with a correspondent node (CN), it regularly performs an LU procedure to update its current location to the network, so that during the PD procedure, the network can locate the MN for the delivery of incoming packets. Therefore, efficient LM is a critical design in WMNs which serves the purpose of tracking the MNs that move from place to place in the coverage of the wireless mesh backbone.

Various LM schemes have been proposed for cellular and wireless local area networks (WLANs) in the literature [2] [3]. However, these schemes cannot be directly applied to WMNs, since LM in WMNs may suffer from poor performance and scalability problems due to the existence of multihop wireless links in the mesh backbone [4]. This multihop wireless transmission increases the transmission delay of signaling or data messages for LM. As a result, it increases the delay of LUs, location tracking, and data PD. Therefore, the number of wireless hops a signaling or data message traverses is an important factor that can affect the performance of LM in a multihop WMN.

This work was supported in part by the US National Science Foundation (NSF) under Grant No. CNS-0855200, CNS-0915599, and CNS-0953644.

In this paper, we propose a hybrid location tracking chain (*Hatch*) framework for LM in WMNs which includes a hierarchical architectural design and a hybrid location tracking scheme. Under the *Hatch* framework, firstly, a dynamic LU scheme based on the number of hops is developed, in which a *silently* roaming MN can save battery consumption by performing an LU only when moving a number of hops away from an MR. Secondly, a *hybrid* location tracking scheme is formed by the routing protocol adopted by the WMN to dynamically change the length of location tracking to improve the PD performance. Moreover, with a hierarchical architectural design, the proposed *Hatch* framework can achieve a reasonable tradeoff between location tracking accuracy and signaling overhead in both intra- and inter-subnet roaming scenarios. By integrating IPv6 protocols and address management with the implementation and carrying out simulation studies with different design options, we see that the proposed *Hatch* framework provides a scalable and practical LM solution in Internet-based WMNs.

The rest of the paper is organized as follows. In Section II, the proposed network architecture for LM is described. In Section III, the proposed hop-based LU and *Hatch* scheme are introduced in detail. In Section IV, OPNET simulation results are provided to demonstrate the performance of the proposed LM. Section V concludes our paper.

II. Architecture Overview

IP-based WMNs are different from mobile ad hoc networks (MANETs) with respect to the special properties of WMNs: 1) MRs in WMNs are usually static. Dynamic multihop routing in WMNs should be based on network load and QoS conditions (e.g., the end-to-end PD delay) rather than MRs' mobility; 2) Unlike MANETs where traffic is inside a network, WMNs are primarily used for Internet-based applications [4]. Hence, proper IP addressing is needed for efficient delivering packets to-and-from the Internet and within the mesh backbone. In our proposal, we partition the mesh backbone into a hierarchy of levels and consider both multihop routing and hierarchical addressing based on IPv6 address management.

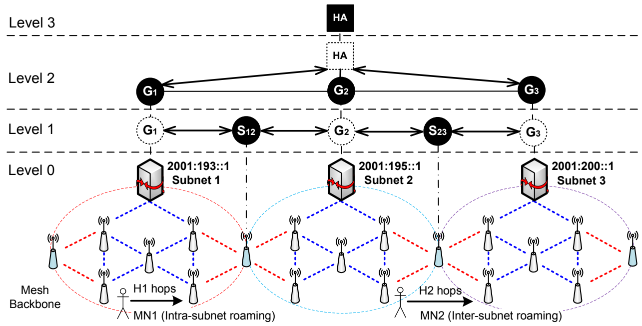


Fig. 1. A hierarchical architecture for LM in WMNs.

Fig. 1 shows our proposed network architecture for LM in WMNs. The WMN is divided into four hierarchical levels. Home agent (HA) is at the highest level in the architecture. The next level includes the set of all gateways in the WMN denoted as $G = \{G_1, G_2, \dots, G_k\}$ (k is a positive integer), each of which belongs to a different subnet[1]. Special mesh routers (SMRs) which have multiple IP addresses are capable of selectively routing traffic across multiple subnets, e.g., a set of SMRs $\{S_{12}, \dots, S_{1k}, S_{23}, \dots, S_{2k}, \dots, S_{k-1,k}\}$, each of which has two IP addresses, can route traffic across two subnets, i.e., the SMR $S_{k-1,k}$ can route traffic between subnet $k-1$ and k. The required number and optimal placement of SMRs that form the mesh architecture are explained in our earlier paper [5]. This feature of our architecture brings two benefits for the design of LM: 1) In an intranet session scenario, i.e., an active session between two MNs inside the WMN, information sharing between different subnets can be implemented with the help of an SMR, e.g., in Fig. 1, traffic routing between MN_1 and MN_2 located in different subnets can be done within the mesh backbone via SMR S_{12} other than routing the traffic to the wired Internet; 2) In an internet session scenario, i.e., an active session between an MN in the WMN and the Internet, an MN roaming to a new subnet can still route traffic to the gateway in the old subnet via an SMR that can route traffic across subnets other than being forced to perform an inter-subnet handoff to a new gateway in the new subnet. Hence, our hierarchical architecture can facilitate efficient LM for both intra- and inter-subnet roaming scenarios.

III. Proposed HATCH Framework for Location Management in Internet-based WMNs

In this section, we take advantages of the proposed hierarchical mesh architecture shown in Fig. 1 and propose a hop-based LU and hybrid location tracking scheme.

A. Proposed Hop-based LU

In our design, in order to save battery consumption, an MN performs an LU only after it has moved h hops away from an originally updated MR (uMR). Note that this cannot be directly interpreted as it has visited h different MRs. As one of the key features of the IPv6 neighbor discovery protocol (NDP) [6], the *Neighbor Unreachability Detection* (NUD) algorithm helps to form and maintain the group of *neighbor mesh routers* (nMRs) for each MR. Basically, an MR actively sends *neighbor solicitation* (NS) messages to all its nMRs to track the reachability "state" of the nMRs. An nMR receiving this message replies with a unicast *neighbor advertisement* (NA) message. We utilize this NUD algorithm to 1) form the nMRs of each MR and 2) record the number of hops an NS message has passed from the uMR.

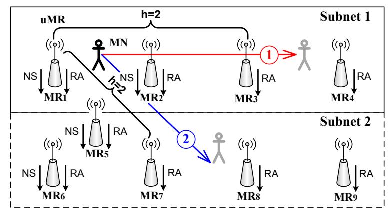


Fig. 2. Two types of movement trajectories of an MN.

As shown in Fig. 2, by receiving an NS message, nMRs that are less than h hops from the uMR are required to rebroadcast to MNs who are currently residing under them. Therefore, an MN which silently roams to a new MR (called current MR, cMR) can be aware of how many hops the cMR is from the uMR. h can be dynamically adjusted for each uMR to balance the signaling overhead and PD delay. When the MN visits a cMR, it listens to NS messages broadcasted on the link of the cMR. If it receives an NS message from the uMR, the MN is still less than h hops away from the uMR and no LU is needed. Once an MN does not receive NS messages from the uMR, it indicates two cases. One is that the MN has moved h or more hops away from the uMR and the other is that the MN is visiting an MR that is not an nMR of the uMR. In either case, the MN performs an LU. To which network entity the LU message is sent depends on the way the location tracking chain is set up, which is explained in detail in the next section. Moreover, based upon *router advertisement* (RAs) messages broadcasted from each MR, MNs can be aware of the change of subnets (RAs can indicate whether a change of the subnet occurs). MNs can have two types of movement trajectories, as shown in Fig. 2, the intra-subnet (movement ①) and inter-subnet movement (movement ②).

B. Proposed Location Tracking Chain Setup

1) LU Message: During each LU, an LU message is sent to update the MN's current IP address. Similar to the hierarchical mobile IPv6 (HMIPv6) [7] design, there are several care-of-addresses (CoAs) defined in our design. Each MN has an on-link care-of-address (LCoA) attributed to the MN's interface and the prefix information advertised by its current cMR. However, only the uMR's address (UCoA) needs to be registered with CNs and the HA. In addition, a new feature, cross-subnet CoA (SCoA) can be added to hide the inter-subnet movement of the MN from the uMR when cross-subnet location tracking is set up for such roaming MN. To realize

[1]In this paper, we assume that different gateways are located in different subnets in the Internet. However, multiple gateways can coexist in a subnet for load balancing.

This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE Globecom 2011 proceedings.

LUs based on SCoA, we propose to add a new S flag using one reserved bit of the *binding update* message in HMIPv6.

2) Default LTC based on Movement (LTC-M): A straightforward LTC based on movement (LTC-M), which is similar to the LU scheme proposed for MIP networks in [3], can be formed as shown in Fig. 3. Here, MNs adopt the same h for triggering each LU procedure. In Fig. 3, an MN without any active session initially residing under its uMR follows a movement trajectory passing each cMR (e.g., MR_1, MR_2, S_{12}, S_{2k}) that is h hops from each other and finally reaches MR_3. During the movement, the MN is required to perform an LU via the cMR it currently resides under to update to the MN's previously visited cMR (e.g., MR_1 to uMR, MR_2 to MR_1, etc.) with the MN's current LCoA. An inter-subnet movement occurs when the MN moves to S_{12} that has two global IPv6 addresses and can route traffic between subnet 1 and 2. Hence, a *cross-subnet* LTC that combines both the intra- and inter-subnet LU scenarios within the wireless mesh backbone is formed after a 5th LU sent from MR_3 to S_{2k}.

Since the LTC-M scheme only requires a local LU to the previously updated MR, it incurs low LU overhead on the architecture side. However, the formation of the LTC under the LTC-M scheme ($uMR \rightarrow \ldots \rightarrow S_{2k} \rightarrow MR_3$) has the potential of causing longer PD delay for the MN, e.g., 1) MR_2 is on the LTC based on the MN's movement. However, the *best* routing path for PD between the uMR and the latest updated MR_3 may not pass through MR_2; and 2) MR_2 can only be aware of the path length to MR_1 and S_{12}. However, the MN cannot detect *how far* it is away from the uMR, hence it cannot determine when to trigger an LU to the HA to stop forming the chain in order to prevent the potential long PD delay.

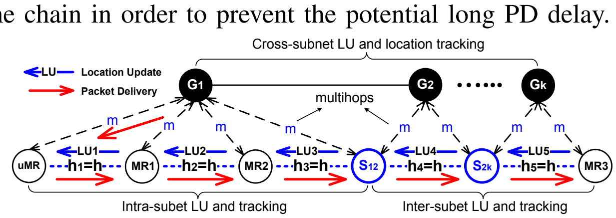
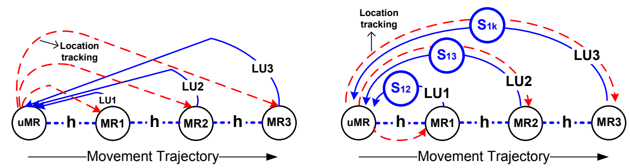


Fig. 3. Location update and tracking under the LTC-M scheme.

3) Proposed Hybrid LTC based on Routing: To overcome the shortcomings of the LTC-M scheme, we propose a new scheme. Fig. 4(a) shows the LU procedure in an intra-subnet movement, where cMRs (e.g., MR_1, MR_2, MR_3) which are h hops from each other are located in the same subnet of the uMR. The MN performs an LU directly to the uMR using its current LCoA. Therefore, a new intra-subnet LTC can be formed based on the routing path between the uMR and cMR. In this way, the MN can be aware of how many hops the current cMR is away from the uMR and perform a full LU to the HA if the growing chain no longer meets the performance criteria. After a full LU to the HA, the current MR becomes the new uMR of this MN. On the other hand, the LU procedure in an inter-subnet movement scenario is depicted in Fig. 4(b), where cMRs (e.g., MR_1, MR_2, MR_3) are located in different subnets. The MN needs to send an LU message containing its current LCoA to an SMR that can route traffic between subnets, e.g., when the MN reaches MR_3, it sends an LU message to SMR S_{13} and S_{13} updates the uMR with its SCoA. Hence, a new cross-subnet LTC can be formed. Under our proposed scheme, the MN can know *how far* (H hops) it is away from the uMR via the routing protocol adopted. Hence, the MN can perform a full LU to the HA after the MN detects that H exceeds a certain threshold. The details of the proposed LU procedures are shown in Algorithm 1.



(a) The intra-subnet LU (b) The inter-subnet LU

Fig. 4. Location update and tracking under the proposed hybrid scheme.

Algorithm 1: LU Procedure of the Proposed Hybrid Scheme

```
N is a predefined number of hops for triggering a full LU to the HA;
if the MN receives an NS message from the currently resided cMR, then
    The MN moves less than h hops and no LU is needed;
else
    if cMR is in the same subnet, then
        The MN acquires the LCoA from the cMR ;
        The MN performs an LU to the uMR ;
        Set up intra-subnet location tracking /* Movement ① */;
    else        /* cMR is located in a new subnet */
        The MN acquires the LCoA from the cMR ;
        The MN performs an LU to the uMR via an SMR;
        The SMR updates to the uMR with its SCoA ;
        Set up inter-subnet location tracking; /* Movement ② */
    The MN can obtain H via the routing protocol adopted;
    if H ≥ N, then
        The MN performs a full LU to the HA;
        cMR becomes the uMR and H = 0;
```

4) Self-Configured Feature in the Proposed Hybrid Scheme: Under our proposed *Hatch* framework, the formation of the LTC based on routing brings the following benefits: 1) the LTC can adapt to varying QoS conditions of the network, i.e., each node on the LTC in our proposed hybrid scheme can make routing decisions based on locally available information and dynamically adjust the current LTC to a *new* and *better* path according to the predicted QoS of available paths, and 2) the MN can choose to perform a full LU to the HA if it detects the length of the chain (e.g., the number of hops from the uMR to its latest updated cMR) can cause performance degradation of PD due to longer delays.

We utilize existing mechanisms to illustrate the self-configurable feature of the proposed *Hatch*. In [8], ICMPv6 [9] signaling messages can be exploited to obtain the end-to-end (ETE) delay q_i of each available path p_i. For example, an SMR $S_{k,k+1}$ on the LTC originally adopts the default LTC as shown in Fig. 5(a). When the default path is no longer qualified for achieving the QoS requirement of PD, and if a better path through a different gateway is available among alternative ones (e.g., paths in case 2 and 3), $S_{k,k+1}$ triggers an LU to G_k and sets up a new LTC along with a deregistration message to the original uMR as shown in Fig. 5(b). Moreover, $S_{k,k+1}$ can perform an LU to the HA and sets up a new LTC via the new

This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE Globecom 2011 proceedings.

gateway G_{k+1} as shown in Fig. 5(c). In case 2 and 3, $S_{k,k+1}$ becomes MN's new uMR.

We conduct OPNET simulations to compare the performance of the three cases shown in Fig. 5(a)(b)(c). Fig. 5(d) shows the ETE delay with service level agreement (SLA) for the three cases. Under our definition of the SLA, the performance is in compliance with the SLA if the ETE delay of PD is below 0.1 second 95 percent of each 3 seconds (in simulation time). As seen in Fig. 5(d), flows on the default path from the MR to $S_{k,k+1}$ conform to the SLA requirement till 113s. Then, $S_{k,k+1}$ should send a deregistration message to the uMR and perform an LU to G_k to choose the path from G_k to $S_{k,k+1}$ till 126s. After that, $S_{k,k+1}$ performs an LU to the HA via the new gateway G_{k+1} and chooses the path from G_{k+1} to $S_{k,k+1}$. A new LTC for the MN is formed after each LU. Hence, the proposed *Hatch* scheme can pick up the path that can best achieve the QoS requirement of PD.

C. Packet Delivery and Query Procedure

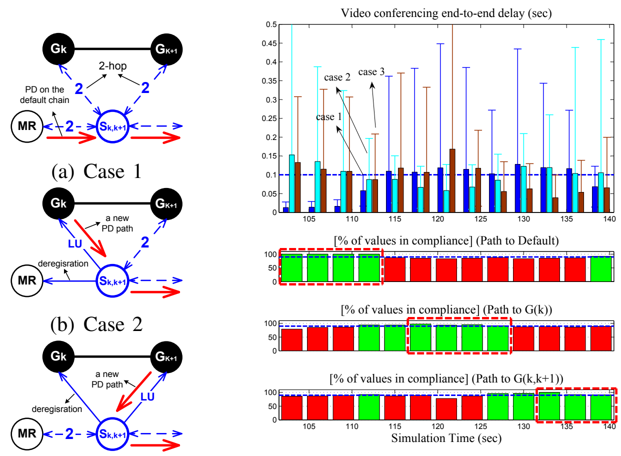
Under both the LTC-M and proposed hybrid schemes, every IP packet destined for an MN is first intercepted by the HA and then forwarded to the uMR of the MN. For the LTC-M scheme, the uMR continues to forward the packets to the next cMR which then follows the same procedure until the packets reach the cMR the MN most recently sent an LU message to. In the hybrid scheme, the uMR can directly forward the packets to the latest cMR that the MN registered to. For both methods, a query procedure to broadcast signaling messages to find the MN in the WMN is needed when the MN is less than h hops from the latest updated cMR. Without introducing new control messages, the implementation of the query procedure for locating the MN follows a similar way to the *duplicate address detection* (DAD) procedure of the NDP protocol. The detailed PD procedure corresponding to both the LTC-M and proposed hybrid schemes is illustrated in Algorithm 2.

Algorithm 2: PD Procedures for LTC-M and Proposed Hybrid Schemes

```
Assume that the roaming MN is less than h hops from a cMR when the
  incoming packets are intercepted by the HA and forwarded to the uMR;
The uMR forwards packets to the cMR;
The cMR initiates a DAD message on its link;
if an NA message is received from the MN, then
    break;                     /* MN resides under the cMR */
else
    The cMR performs a query procedure among its nMRs ;
    The cMR receives an NA message reply from the current MR the
      MN residing under ;
The cMR sends buffered packets to the MN;
```

IV. Performance Analysis

In this section, we evaluate the performance of our proposed hybrid LM using OPNET [10].



(a) Case 1 (b) Case 2 (c) Case 3 (d) A simulation study for different cases.

Fig. 5. Cases for self-configured location tracking and simulation study for different cases. A green bar indicates conformance to the defined SLA and a red bar shows a violation.

A. Simulation Scenarios and Assumptions

Fig. 6 shows a simulation scenario that includes both intra- and inter-subnet roaming cases. As an MN moves, it needs to perform an LU when it reaches MR_2 (when the value of h is 3), where the value of H is 4 (movement of an MN from the uMR to S_{23}). By varying h, the signaling overhead of LUs on the architecture and on the MN side under the default HA (i.e., perform an LU to the HA after each movement), LTC-M, and proposed hybrid schemes are studied. The performance of the three schemes in terms of LU signaling overhead and PD delay are evaluated. In addition, design guidelines to achieve a reasonable tradeoff between the location accuracy and signaling overhead in our proposed hybrid scheme are provided. In our simulation, signaling overhead of the default HA scheme comprises of LU messages and the AODV [11] messages generated for delivering relevant LU messages (no routing overhead generated for delivering LU messages in the OLSR [12] proactive routing case). Both the LTC-M and proposed hybrid schemes also include the overhead of the extra NS messages used to determine the number of hops from the uMR when $h \geq 2$. The parameters used in our simulations are shown in Table I.

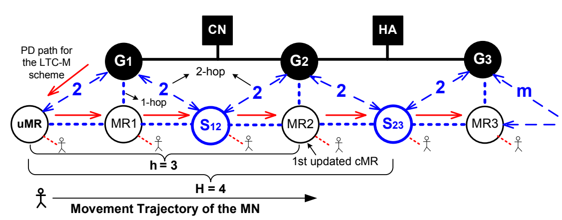


Fig. 6. A simulation scenario in OPNET.

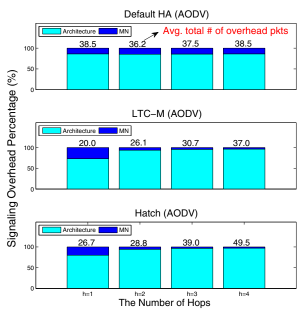
B. Result Analysis

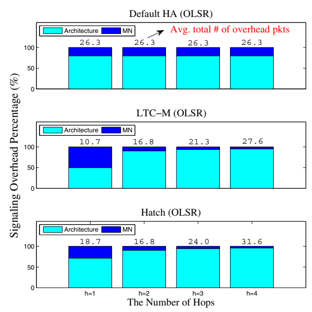
1) Tradeoff Study: Fig. 7 shows comparisons of signaling overhead percentage caused by LUs on the architecture and MN side under the three schemes. Compared to the default HA scheme, the overhead percentage on the MN side in both LTC-M and hybrid schemes decreases as h increases. We observe that when $h \leq 2$ in Fig. 7(a) and $h \leq 3$ in Fig. 7(b), the

TABLE I
SIMULATION PARAMETERS

MR transmit power (W)	0.05
Packet reception-power threshold (dbm)	-95
Multihop routing protocol	AODV/ OLSR
IPv6 *Router Advertisement* interval (sec)	uniform (0.5, 1)
NDPv6 messages interval (sec)	uniform (1, 2)
AODV active route timeout (sec)	3.0
OLSR *HELLO* message interval (sec)	2.0
OLSR *Topology Control* message interval (sec)	5.0

proposed hybrid scheme outperforms the default HA scheme with respect to lower signaling overhead on the MN side and lower total overhead generated. In addition, the proposed hybrid scheme under $h = 3, 4$ in Fig. 7(a) and $h = 4$ in Fig. 7(b) are still preferred if less signaling overhead on the MN side is demanded, even though the total overhead is higher than those in the default HA scheme. Therefore, the design of the value h depends on the requirements and preferences of the corresponding service providers. On the other hand, though the LTC-M scheme incurs the lowest total signaling overhead, it has the potential of causing longer PD delay when the length of LTC grows, which is illustrated in the following study.

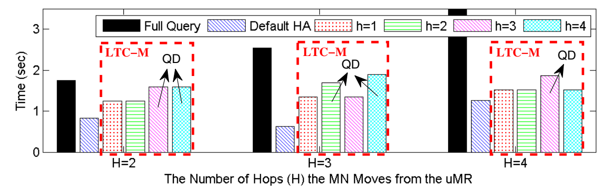




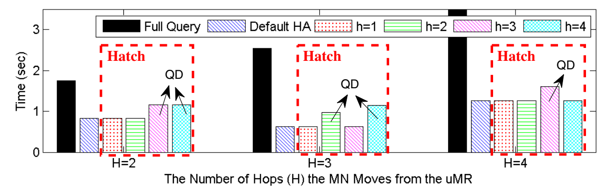
(a) Overhead percentage (AODV) (b) Overhead percentage (OLSR)

Fig. 7. Signaling overhead percentage between the architecture and MN.

2) Performance of PD: Fig. 8(a) shows the average PD delay corresponding to six LU schemes (four LTC-M schemes are shown in the dashed rectangular area). Assuming the roaming MN is H hops away from the uMR when the incoming packets from a CN are intercepted by the HA and forwarded to the uMR. As expected, the full query scheme has the worst PD delay in all cases since it queries all MRs within H hops to the uMR. The larger the H, the higher overhead it incurs. On the other hand, both the default HA and LTC-M schemes when $h = 1$ always have the precise location of the MN. Thus, no query procedure is needed and the PD delay relies on the route discovery delay for delivering the data packets. In addition, the lowest PD delay is shown in the default HA scheme since the MN always perform LUs directly to the HA. When h varies from 2 to 4, packets can be forwarded to the latest registered cMR by the uMR. The query procedure at the latest updated cMR is performed to locate the MN and querying delay (QD) is added to the whole PD delay. (e.g., when the value of (H, h) is $(2, 3), (2, 4), (3, 2), (3, 4), (4, 3)$). Otherwise, the delay of PD depends on the length of the LTC from the uMR to the latest registered cMR. For comparison, in our proposed hybrid scheme, each node on the LTC of the MN can change the path based on the QoS conditions, so that the PD delay between the HA and the latest registered cMR can be shortened, as shown in Fig. 8(b).



(a) Average PD delay under LTC-M (AODV)



(b) Average PD delay under Hatch framework

Fig. 8. The delay of PD under the LTC-M and Hatch framework.

V. CONCLUSIONS

In this paper, we proposed a *Hatch* framework for LM in Internet-based WMNs. Under the *Hatch* framework, a dynamic hop-based LU is developed with a hybrid location tracking scheme formed based on the routing protocol adopted by the WMN. Moreover, the length of the location tracking chain formed by our proposed *Hatch* framework can be dynamically adjusted to improve the performance of PD for MNs. With a hierarchical architectural design, the proposed *Hatch* framework can achieve a reasonable tradeoff between location tracking accuracy and signaling overhead in both intra- and inter-subnet roaming scenarios. Through OPNET simulations, we demonstrated that our proposed *Hatch* framework offers scalable and efficient LM in Internet-based WMNs.

REFERENCES

[1] I. F. Akyildiz, X. Wang, and W. Wang, “Wireless mesh networks: a survey,” *Computer Networks*, vol. 47, no. 4, pp. 445–487, March 2005.
[2] J. Xie and I. F. Akyildiz, “A novel distributed dynamic location management scheme for minimizing signaling costs in Mobile IP,” *IEEE Trans. Mobile Computing*, vol. 1, no. 3, pp. 163–175, July 2002.
[3] W. Ma and Y. Fang, “Dynamic hierarchical mobility management strategy for Mobile IP networks,” *IEEE Journal on Selected Areas in Communications*, vol. 22, no. 4, pp. 664–676, May 2004.
[4] J. Xie and X. Wang, “A survey of mobility management in hybrid wireless mesh networks,” *IEEE Network*, vol. 22, no. 6, 2008.
[5] W. Zhao and J. Xie, “A novel Xcast-based caching architecture for inter-gateway handoffs in infrastructure wireless mesh networks,” in *Proc. IEEE INFOCOM*, 2010, pp. 2766–2774.
[6] T. Narten, E. Nordmark, W. Simpson, and H. Soliman, “Neighbor discovery for IP version 6 (IPv6),” RFC 4861, IETF, Janruay 2007.
[7] H. Soliman, C. Castelluccia, K. E. Malki, and L. Bellier, “Hierarchical MIPv6 mobility management (HMIPv6),” RFC 4140, Auguest 2005.
[8] W. Zhao and J. Xie, “NETF: An integrated design for inter-gateway QoS handoffs in infrastructure wireless mesh networks,” in *Proc. IEEE GLOBECOM*, December 2010.
[9] A. Conta, S. Deering, and M. Gupta, “Internet control message protocol ICMPv6 for the (IPv6) specification,” RFC 4443, IETF, March 2006.
[10] OPNET Technologies, Inc., http://www.opnet.com/.
[11] C. Perkins, E. Belding-Royer, and S. Das, “Ad hoc on-demand distance vector (AODV) routing,” RFC 3561, IETF, July 2003.
[12] T. Clausen and P. Jacquet, “Optimized link state routing protocol (OLSR),” RFC 3626, IETF, October 2003.