Appendix A
An Example of Neighbor nMR Information Formation

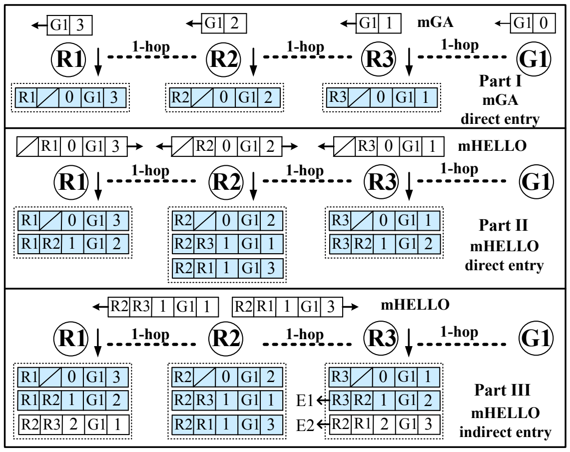
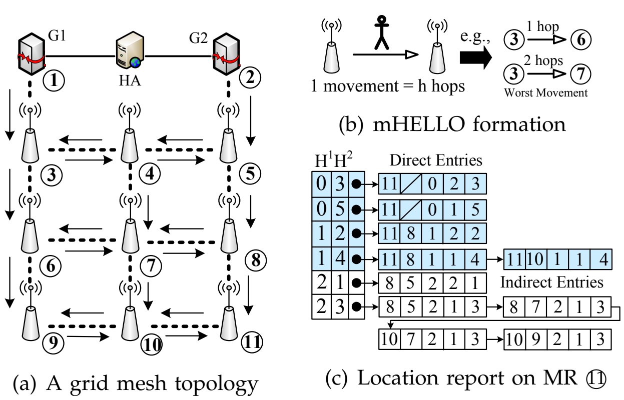


Fig. 11: An example of neighbor MR information formation. Direct mHELLOs are in shaded color to be differentiated from indirect ones.

As shown in Fig. 11 Part I, each MR ($R1$, $R2$, $R3$) initially forms one direct entry after receiving the gateway information sent by $G1$. In Fig. 11 Part II, MR $R2$ receives two mHELLOs from its 1-hop neighbor $R1$ and $R3$ which are direct entries. $R2$ changes the M_{ID} in these entries with its own ID and updates the field of N_{ID} and N_{HOP}. $R1$ and $R3$ also update the received mHELLO direct entries received from their 1-hop neighbor in a similar way. On the other hand, indirect mHELLO entries are the ones in which the node ID (M_{ID}) is a neighbor MR within $\lceil \frac{h}{2} \rceil$ hops. In Fig. 11 Part III, $R3$ can receive its 2-hop neighbor $R1$'s mHELLO forwarded by $R2$, in which the M_{ID} is R2. This indirect mHELLO entry indicates the relations of neighboring MRs ($R2$ and $R1$). In a word, direct entries can be used by sMNs to perform location estimation for the movement within $[1, \lceil \frac{h}{2} \rceil]$ hops while indirect entries are for the movement within $[\lceil \frac{h}{2} \rceil, h]$ hops.



(a) A grid mesh topology

(b) mHELLO formation

(c) Location report on MR ⑪

Fig. 12: A location report formation example in a grid mesh topology.

Algorithm 1: Pseudocode of gateway and neighbor MR information (mGA and mHELLO) formation

Assume that the number of gateways in an IiWMN is num_G; Two temporary tables to store entries are gateway information table (G_info_table) and neighbor MR information table (N_info_table). The hop distance threshold of N_{HOP} is defined as $TH\_h = \lceil \frac{h}{2} \rceil$;

/* During the IiWMN deployment phase, each gateway triggers the mGA procedure */;
Use its IP address as G_{ID} & initialize $G_{HOP} = 0$;
Encapsulate mGA in an IPv6 packet with TTL = 255;
Broadcast the mGA message;

/* MRs process the received *mGA* message */;
$G_{HOP} = G_{HOP} + 1$ and get G_{ID};
if *G_{ID} already exists in the G_info_table* **then**
    if *new G_{HOP} < old G_{HOP}* **then**
        Update *mGA* entry in the G_info_table;
    else
        Discard this mGA message;
else /* this is a new mGA entry */
    Add mGA entry to the G_info_table;
if *TTL != 1* **then**
    Encapsulate mGA in an IPv6 packet with TTL = TTL - 1 ;
    Rebroadcast mGA;
else
    Discard this mGA message;

/* After the establishment of G_info_table, each MR triggers the *mHELLO* procedure */;
Use its IP address as N_{ID}, initialize $N_{HOP} = 0$, $M_{ID} = NULL$;
for $i = 0$; $i < num\_G$; $i++$ **do**
    Get the mGA entry in the G_info_table;
Insert each mGA entry to an mHELLO entry;
Encapsulate mHELLO in an IPv6 packet with TTL = 255;
Broadcast the mHELLO message;

/* MRs process the received mHELLO message */;
N_{HOP} = N_{HOP} + 1;
if $N_{HOP} == TH\_h$ ***or*** $N_{HOP} == 2 * TH\_h$ **then**
    Discard this mHELLO message;
else
    Get M_{ID}, N_{ID}, and G_{ID} from mHELLO;
if *N_{ID} already exists in the N_info_table* **then**
    if *new N_{HOP} < old N_{HOP}* **then**
        if *$M_{ID} == NULL$ **and** new $N_{HOP} < TH\_h$* **then**
            Update M_{ID} using its IP address;
        Update mHELLO entry in the N_info_table;
    else /* check whether this mHELLO has a new G_{ID} */
        if *G_{ID} already exists in the G_info_table* **then**
            Discard this mHELLO message;
        else
            if *$M_{ID} == NULL$ **and** new $N_{HOP} < TH\_h$* **then**
                Update M_{ID} using its IP address;
            Update mHELLO entry in the N_info_table;
else /* this is a new mHELLO entry */
    Add mHELLO entry to the N_info_table;
Encapsulate mHELLO in an IPv6 packet with TTL = TTL - 1 ;
Rebroadcast mHELLO;

Appendix B
An Example of Location Report Formation

Let us consider a grid mesh topology with two gateways, as shown in the Fig. 12(a). For simplicity, we assume that a sMN is required to perform an LU whenever making a movement (e.g., visiting a different MR). We see that the *worst* movement of a sMN can cause a maximum 2-hop distance (e.g., moving from MR_3 to MR_7). Thus, mHELLOs only need to be exchanged between 1-hop neighboring MRs. After the propagation of mHELLOs, the corresponding location report can be formed. For example, the shaded items of the first four linked lists in the location report of MR (⑪) as shown in the figure are the direct mHELLO entries, while the rest indirect entries received from the 1-hop neighbors of MR (⑪) are placed underneath the direct ones. All entries under each category are sorted by H^1 and H^2.

Algorithm 2: Location report formation with desired sequence

```
input : An unsorted location report with randomly received
        mHELLO entries
output: A sorted location report with a linked list data structure
        based on two keys
1  Let δ be the desired mHELLO entry index in a location report. ζ
   is the size of an unsorted location report and M is the ID of the
   node itself;
2  for n = 0; n < ζ; n++ do
3      δ = n  /* assume the start entry as desired */;
4      Build_list(mhello_n);
5      /* Search through each entry from n+1 */;
6      for ω = n+1; ω < ζ; ω++ do
7          /* Direct mHELLO entries placed ahead */;
8          if (mhello[ω].M_ID == M) and (mhello[δ].M_ID ≠ M)
           then
9              δ = ω    /* Store the desired index */;
10         else        /* Entries sorted with H^1 & H^2 */
11             if (mhello[ω].M_ID == mhello[δ].M_ID == M) or
               ((mhello[ω].M_ID ≠ M) and
               (mhello[δ].M_ID ≠ M)) then
12                 if mhello[ω].H^1 < mhello[δ].H^1 then
13                     δ = ω ;
14                 else
15                     if mhello[ω].H^1 == mhello[δ].H^1 then
16                         if mhello[ω].H^2 < mhello[δ].H^2 then
17                             δ = ω ;
18                         else
19                             if mhello[ω].H^2 == mhello[δ].H^2
                               then
20                                 Append_list(mhello_ω);
21     /* Swap the entry with the desired one */;
22     Swap(mhello_n, mhello_δ);
23 S_1 = Remove_empty_cells(mhello_ζ);
```

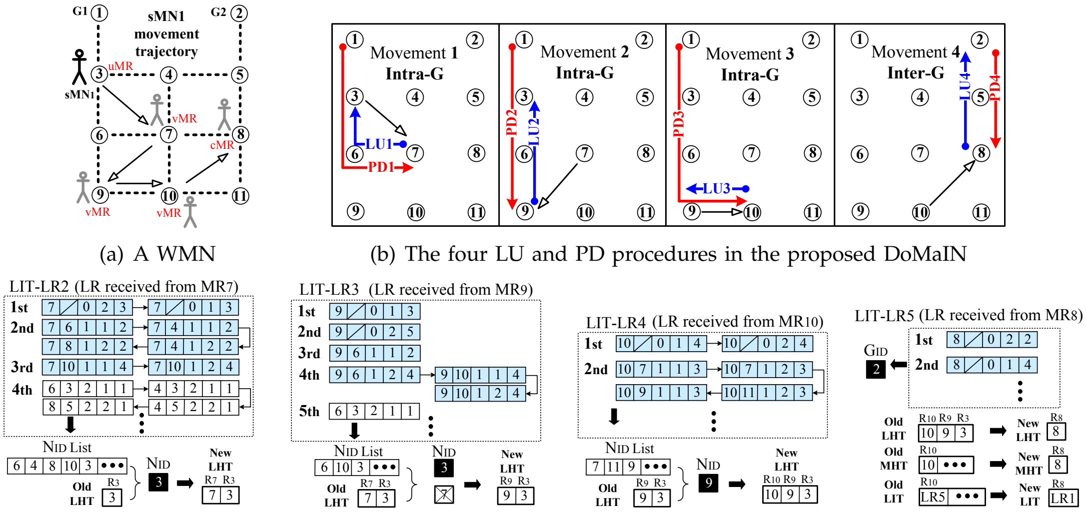
APPENDIX C
AN EXAMPLE OF LOCATION ESTIMATION

Fig. 13(a) shows an example of a sMN's movement trajectory ($MR_3 \rightarrow MR_7 \rightarrow MR_9 \rightarrow MR_{10} \rightarrow MR_8$.) in a 3×3 grid mesh backbone. We assume that the sMN initially resides under the uMR (MR_3) and G_1 is chosen as its default gateway for Internet access. Assume that the sMN is required to perform an LU when visiting a different MR. It receives a location report from each MR it visits and adds it to the LIT. Fig. 13(c)-(f) shows the sMN's LU cases corresponding to the four movements. In movement 1 ($MR_3 \rightarrow MR_7$), location

Algorithm 3: Location estimation on the sMN side

```
input : Location database (LHT and LIT)
output: The desired gateway ID (G_ID) for the inter-gateway LU
        or the desired MR ID (N_ID) for the intra-gateway LU,
        and the updated LHT.
1  Assume l is the length of the LHT;
2  if LIT[m,1].G_ID != LIT[1,1].G_ID then
3      for m' = m−1; m' > (m − num_G); m' −− do
4          if LIT[m',1].G_HOP > LIT[m,1].G_HOP then
5              G_ID = LIT[m,1].G_ID;
6              sMN performs an LU to the HA via the new
               gateway G_ID;
7              Reset MHT, LHT, and LIT;
8              Break     /* Inter-gateway LU done */;
9          else
10             if LIT[m',1].G_ID == LIT[1,1].G_ID then
11                 Intra-gateway LU case and goto line 12;
12 else
13     for i = m*S_2; i > 0; i −− do
14         n= List_length(mhello_i);
15         for j = 0; j < n; j++ do
16             mHello_j = Get_jth_list(mhello_i, j);
17             if mHello[j].G_ID==G_ID then
18                 for p = l; p > 0; p −− do
19                     if mHello[j].N_ID == R[p].N_ID then
20                         Remove (R_l, ..., R_{p+1}) entries;
21                         Add R_m to the LHT;
22                         Perform an LU to N_ID;
23                         Break  /* Intra-gateway LU done
                           */;
```

estimation finds G_1 and G_2 in the first two mHELLO entries in the first linked list in the LIT having the same G_{HOP}, thus it is an intra-gateway LU case. The lowest number of hops between G_1 and MR_7 can be obtained from the entry $(7, NULL, 0, 1, 3)$, which is 3. Next, the location estimation starts from the first entry of the second linked list where $N_{HOP} = 1$ in the LIT. The N_{ID} of each entry in the LIT is compared with the entries in the LHT in a reverse order and $N_{ID} = 3$ is obtained in the LIT indicating R_3, as shown in Fig. 13(c), which is the desired MR for the sMN's intra-gateway LU. Then, R_7 is added to the sMN's LHT and the sMN performs an intra-gateway LU to MR_3 updating the IP address of MR_7. In movement 2 ($MR_7 \rightarrow MR_9$), the sMN's location estimation also obtains MR_3, as shown in Fig. 13(d). Entry R_7 is removed from the LHT before the new R_9 is added. In this case, sMN performs an intra-gateway LU to MR_3 updating the IP address of MR_9. Similarly, in movement 3 ($MR_9 \rightarrow MR_{10}$), the sMN obtains $N_{ID} = 9$ (MR_9), as shown in Fig. 13(e) and performs the third intra-gateway LU to MR_9. In movement 4 ($MR_{10} \rightarrow MR_8$), sMN's location estimation obtains a new gateway G_2 with $G_{HOP} = 2$ lower than $G_{HOP} = 4$ for G_1. Hence, the sMN performs an inter-gateway LU and resets its MHT, LHT, and LIT, as shown in Fig. 13(f). The corresponding paths for the LU and PD procedures in each step of the location estimation are shown in Fig. 13(b).



(a) A WMN

(b) The four LU and PD procedures in the proposed DoMaIN

(c) Movement 1

(d) Movement 2

(e) Movement 3

(f) Movement 4

Fig. 13: An example of location report formation in grid mesh topology with two gateways.

Appendix D
DoMaIN Implementation Issues

In our proposed *DoMaIN* scheme, similar to RA messages, the formed location report needs to be periodically broadcasted by MRs. There is an alternative method that such overhead on the network side can be reduced by allowing MNs to solicit the location report only when needed. However, if the power preservation on the MN side is considered as the major objective and exploiting sMN's "*listen*" capability is preferred, the overhead burden should be placed on the network side. Moreover, during sMN's intra-gateway movements, a sMN needs to keep the location reports it receives to perform location estimations. We argue that a sMN equipped with a large memory is quite common nowadays. Among the three tables, entries from the MHT/LHT can be directly obtained from the location report. In addition, the MHT may not be necessary if the movement-based LU triggering method as explained in Section 4.3 is not adopted.

Additional, we consider an infrastructure WMN in this paper in which MRs are static or mostly static. If this is not the case, then regenerating location information for keeping location reports fresh is necessary to avoid errors in location reports. Moreover, we can use hop distance h' as the threshold for each MR to set up the neighbor information table for the LIT. The larger the h', the more topology information should be included in one single LIT generated by an MR, but less robust this LIT will be. On the other hand, the minimum topology information an MR needs to have is $\lceil \frac{h}{2} \rceil$, where h is the largest hop distance when an MN makes one random movement. Therefore the range of the threshold h' is: $\lceil \frac{h}{2} \rceil < h' < \infty$ (i.e., one LIT includes the topology information of the entire network). Hence, if the robustness is an issue due to some MR movements in the mesh backbone, we can reduce the LIT formation threshold h' and make it close to $\lceil \frac{h}{2} \rceil$; if the robustness is satisfactory and energy saving on the sMN side is the primary goal, we can make an MR's LIT formation threshold h' large enough to cover sufficient information an LIT can have. Under this condition, a sMN can reduce its frequency of listening to new LITs from other MRs, thus it may enter into a sleep period without receiving new LITs and save energy. Providing robust information for sMNs and dealing with errors in location updates from sMNs whose energy saving is the primary goal are non-trivial issues. To address these issues, a delicate tradeoff analysis and design are needed.

TABLE 2: Simulation Parameters

IiWMN Parameters	
MR transmit power (W)	0.05
Packet reception-power threshold (dbm)	-95
AODV active route timeout (sec)	3.0
IPv6 *Router Advertisement* interval (sec)	constant (20)
Internet Traversal Delay (sec)	0.05
sMNs' Random Waypoint Parameters	
Mobility Domain Name	mesh backbone
Speed (meters/sec)	uniform_int(0,10)
Pause Time (sec)	constant (10)
Start Time (sec)	10
Internet Session Packet Arrival Rate for sMNs	
Start time (sec)	10
Frame interarrival time (sec)	constant (10)
Light Video Application for aMNs	
Frame size (bytes)	172
Frame interarrival time (sec)	constant (0.5)

Appendix E
Number of One-hop Transmissions for Each LU and PD Procedures

Under a certain dynamic LU trigger with its threshold (e.g., $h = 1$) and the *Random Waypoint Mobility* model, we examine the impact of each LU procedure triggered by a sMN and each PD procedure triggered by a CN on the mesh backbone under different location management schemes. That is, the average number of one-hop transmissions for each LU and PD procedures under LTC-H, LTC-M, LTC-R and *DoMaIN*. Note that

TABLE 1: Comparison of *DoMaIN* and other location management solutions for IiWMNs. LM and $R(MT)^2$ stand for location management and the relation between movement trends and mesh topology, respectively.

	LTC-M	LTC-R	LTC-H	DoMaIN
LU Triggers	time-based, movement-based, hop-based, and hybrid above	time-based, movement-based, hop-based, and hybrid above	time-based, movement-based, hop-based, and hybrid above	time-based, movement-based, hop-based, and hybrid above
$R(MT)^2$	movement towards father topological distance to updated MRs	movement towards closer topological distance to uMR	movement towards the same topological distance to different gateways	arbitrary movements random topology
LU Entity	previously updated MR	Static (uMR)	Static (HA)	Dynamic
LM Performance	Static (Good or Bad)	Static (Good or Bad)	Static (Good)	Dynamic (Good)
LM Overhead	Low & Medium	Low & Medium	High & Medium	Low

the average number of one-hop transmissions of LU and PD procedures in the mesh backbone depends on the number of MRs involved in carrying and relaying each LU and data packet generated by the sMN or the CN.

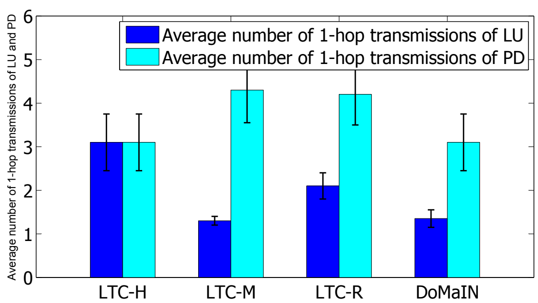
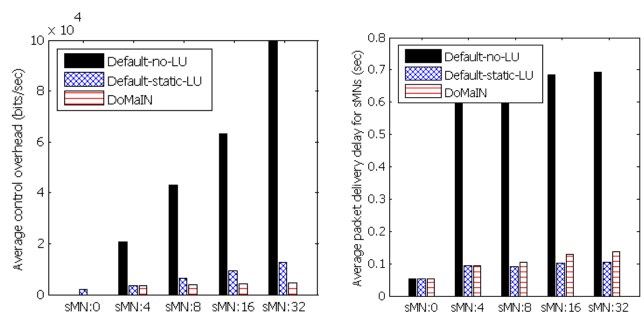


Fig. 14: Average number of one-hop transmissions for each LU and PD procedures under LTC-H, LTC-M, LTC-R, and DoMaIN

As seen in Fig. 14, LTC-H incurs the highest average number of one-hop LU transmissions, which induces the highest total LU overhead cost on the mesh backbone among all the four schemes. The lowest average number of one-hop LU transmissions can be seen under LTC-M and *DoMaIN*. Secondly, PD procedures under LTC-H and *DoMaIN* always select the shortest path in terms of the number of hops, thus they incur the lowest average number of one-hop PD transmissions as compared to the other two. On the other hand, the formed location tracking chain can induce longer hop distance for PD under both LTC-M and LTC-R schemes. Hence, we conclude that the proposed *DoMaIN* provides the best location management performance with respect to the lowest average number of one-hop LU and PD transmissions among all the four schemes.

APPENDIX F
CONTROL OVERHEAD STUDY OF *DoMaIN*

Based on the *Random Waypoint Mobility* model, we study the control overhead of the proposed *DoMaIN*. Three schemes are developed and examined: 1) Default-no-LU scheme (i.e., no LUs are required on the sMN side, but the WMN performs the paging procedure upon the arrival of data packets), 2) Default-static-LU scheme (i.e., a sMN performs an LU to the HA every time it moves to a new MR and no paging procedure is needed), and 3) *DoMaIN* (dynamic hop-based LU trigger: 2-hop). In IiWMNs, aMNs are static and have light end-to-end video conferencing traffic. The corresponding control overhead and PD performance under different schemes are studied. Control message overhead of the Default-no-LU scheme includes paging request broadcast messages and the paging reply message. Control overhead of the Default-static-LU scheme includes periodic RA broadcast messages and the corresponding LU messages triggered at sMNs. Control overhead of *DoMaIN* includes LIT formation overhead during the WMN deployment, periodic LIT broadcast messages, paging request/reply, and LU messages (paging is needed when a sMN moves between two LU triggers upon the arrival of a data packet).



(a) Average control overhead (b) Average PD delay

Fig. 15: Control overhead and corresponding PD performance.

Fig. 15(a) shows the time averaged control overhead under different schemes. Initially, when all MNs are in the active mode (i.e., the number of sMNs is zero), control overhead of both Default-no-LU and *DoMaIN* is zero. For *DoMaIN*, since all MNs are in the active mode, there is no need to generate *DoMaIN* control information for sMNs. As the number of sMNs increases, control overhead of the Default-no-LU scheme increases much faster than the other two schemes, where the overhead caused by the paging procedure accounts for a significant portion of the total control overhead. In Default-static-LU, sMNs are forced to perform LUs to the HA every time they visit a different MR. The corresponding LU overhead incurred increases as the number of sMNs increases. Compared to the two default schemes, we see that *DoMaIN* effectively reduces the overhead caused either by the paging procedure in Default-no-LU or by the periodic LU messages in Default-static-LU.

Correspondingly, Fig. 15(b) shows the average PD delay for sMNs. Default-no-LU has the worst PD performance because of its paging delay among the three schemes. In contrast, *DoMaIN* allows sMNs to perform LUs only after their movement exceeds the LU triggering threshold and performs LUs to the closest MR in terms of the number of hops, which greatly reduces the paging delay compared to the Default-no-LU scheme. Default-static-LU has the best performance in terms of the lowest average PD delay among the three schemes, since the network always has sMNs' explicit locations. However, battery energy on sMNs can diminish fast under the Default-static-LU scheme if sMNs perform frequent LUs.

Hence, we conclude that the Default-no-LU scheme is not a scalable location management solution, while Default-static-LU does not consider the power efficiency on mobile devices. The proposed *DoMaIN* presents a dynamic location management design and balances the tradeoff between the two targets in order to retain sMNs' battery energy as much as possible.