ARCHITECTURAL AND MOBILITY MANAGEMENT DESIGNS IN INTERNET-BASED INFRASTRUCTURE WIRELESS MESH NETWORKS

by

Weiyi Zhao

A dissertation submitted to the faculty of
The University of North Carolina at Charlotte
in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in
Electrical Engineering

Charlotte

2011

Approved by:

Dr. Jiang (Linda) Xie

Dr. Teresa Dahlberg

Dr. Ivan Howitt

Dr. Asis Nasipuri

Dr. Yu Wang

ABSTRACT

WEIYI ZHAO. Architectural and mobility management designs in internet-based infrastructure wireless mesh networks.
(Under the direction of DR. JIANG (LINDA) XIE)

Wireless mesh networks (WMNs) have recently emerged to be a cost-effective solution to support large-scale wireless Internet access. They have numerous applications, such as broadband Internet access, building automation, and intelligent transportation systems. One research challenge for Internet-based WMNs is to design efficient mobility management techniques for mobile users to achieve seamless roaming. Mobility management includes handoff management and location management. The objective of this research is to design new handoff and location management techniques for Internet-based infrastructure WMNs.

Handoff management enables a wireless network to maintain active connections as mobile users move into new service areas. Previous solutions on handoff management in infrastructure WMNs mainly focus on intra-gateway mobility. New handoff issues involved in inter-gateway mobility in WMNs have not been properly addressed. Hence, a new architectural design is proposed to facilitate inter-gateway handoff management in infrastructure WMNs. The proposed architecture is designed to specifically address the special handoff design challenges in Internet-based WMNs. It can facilitate parallel executions of handoffs from multiple layers, in conjunction with a data caching mechanism which guarantees minimum packet loss during handoffs. Based on the proposed architecture, a Quality of Service (QoS) handoff mechanism is also proposed to achieve QoS requirements for both handoff and existing traffic before and after handoffs in the inter-gateway WMN environment.

Location management in wireless networks serves the purpose of tracking mobile users and locating them prior to establishing new communications. Existing location management solutions proposed for single-hop wireless networks cannot be directly

applied to Internet-based WMNs. Hence, a dynamic location management framework in Internet-based WMNs is proposed that can guarantee the location management performance and also minimize the protocol overhead. In addition, a novel resilient location area design in Internet-based WMNs is also proposed. The formation of the location areas can adapt to the changes of both paging load and service load so that the tradeoff between paging overhead and mobile device power consumption can be balanced, and at the same time, the required QoS performance of existing traffic is maintained. Therefore, together with the proposed handoff management design, efficient mobility management can be realized in Internet-based infrastructure WMNs.

ACKNOWLEDGMENTS

We at the height are ready to decline.
There is a tide in the affairs of men,
Which, taken at the flood, leads on to fortune;
Omitted, all the voyage of their life
Is bound in shallows and in miseries.
On such a full sea are we now afloat;
And we must take the current when it serves,
Or lose our ventures. - Shakespeare's Julius Caesar

Living in an era with saturated telecom penetration, I am looking back and at the same time very grateful that I had the opportunity and made the right decision to pursue the PhD degree in the field of wireless networking in 2007. When I find myself on the top of tides in an ocean of knowledge enjoying the beautiful scenery that surrounds me today in 2011, I realize that it was, in fact, many people who got me there.

First and foremost, I would like to express my most heartfelt gratitude to my advisor, Professor Jiang (Linda) Xie, who accepted me as her PhD student at UNCC. Over the past four years, she has constantly kept teaching me and transforming my interests into something concrete and fundamentally important in the field of technology innovation. Her guidance has greatly improved and refined my research work, but more importantly, she has been training me to have a balanced development of morality, intelligence, physique, social skills, and esthetics. I could not have imagined having a better advisor and mentor for my PhD study. Furthermore, the forever passion and enthusiasm she has for her research was contagious and motivational for me, especially during tough and painful times in my PhD pursuit. I am deeply admiring for the excellent example she has provided as a successful woman teacher and researcher.

I would also like to thank my thesis committees: Dr. Asis Nasipuri, Dr. Ivan Howitt, Dr. Yu Wang, and Dr. Teresa Dahlberg for their support, direction, and invaluable advice along this dissertation.

I thank my colleagues and friends from the Wireless Communications and Signaling Processing Lab: Yi Song, Haopeng Li, Shilpa Bhave, and many others. It's been a wonderful and unforgettable experience to share my doctoral studies and life with them. Furthermore, four years of PhD study would be a hard and lonely journey without the strong support and care from friends in Charlotte. I am deeply grateful to Zhi Li, Yao Hu, Yunfeng Sui, Zhe Dang, Guangyi Cao, Cheng Li, Rong Sun, Lance Newby, and many others. I also thank my close friends in domestic China, Lu Tian, Yinghuan Zhou, Yuming Sun, Jing Du, and Jia Yi who have stood by me in times of difficulties.

Last, but by no means least, I would not have achieved half of what I have without the constant love and support of my parents, Zhongqiu Zhao, Jiawen Shen and my grandparents Shiqing Zhao, Shaolan Qi, Mingxia You, and Chunquan Shen. They have taught me and let me believe that a genius is 10% of inspiration and 90% perspiration. This has been encouraging me to work hard so far and will continue to inspire me to be ready for the new expedition, to float out and sail as far as I could in an ocean of knowledge.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

WLAN	Wireless Local Area Network
MANET	Mobile Ad Hoc Network
WMN	Wireless Mesh Network
IiWMN	Internet-based Infrastructure WMN
MN	Mobile Node
sMN	Silently Roaming MNs
aMN	Active Roaming MNs
AP	Access Point
MR	Mesh Router
Xcast	Explicit Multicast
XGR	Xcast-based MR
uMR	Originally Updated MR
vMR	Visited MR
cMR	Currently Resided MR
HA	Home Agent
FA	Foreign Agent
CN	Correspondent Node
IPv6	IP Address Version 6
MIP	Mobile IPv4
MIPv6	Mobile IPv6
HMIPv6	Hierarchical MIPv6
DHMIP	Dynamic HMIP
F-HMIPv6	Fast HMIPv6
DAD	Duplicate Address Detection
CoA	Care-of-address

L2	Link-layer
L3	Network-layer
L5	Application-layer
PP	Path Preparation
AU	Address Update
AA	Address Acknowledgement
SR	Session Redirection
LU	Location Update
PD	Packet Delivery
LA	Location Area
RA	Router Advertisment
NS	Neighbor Solicitation
NA	Neighbor Acknowledgement
RSSI	Received Signal Strength Indication
SLA	Service Level Agreement
QoS	Quality of Service
ETE	End-to-end
SIP	Session Initiation Protocol
AODV	Ad hoc On-Demand Distance Vector Routing
OLSR	Optimized Link State Routing
OSPF	Open Shortest Path First
SLA	Service Level Agreement
QoS	Quality of Service
DES	Discrete-event-driven Simulations
VoIP	Voice over IP

CHAPTER 1: INTRODUCTION

The wireless mesh network (WMN) technology has recently emerged as a promising solution for providing large-scale wireless Internet access [1, 2]. It has numerous applications, such as broadband Internet access, building automation, and intelligent transportation systems. One important component of realizing large-scale WMNs in order to provide broadband cost-effective Internet access is mobility management.

Currently, Internet Engineering Task Force (IETF) has proposed Mobile IP (MIP) [3] and Mobile IPv6 (MIPv6) [4] as the main IPv4 and IPv6 solution for mobility management at the IP layer. However, both of them have some well-known drawbacks such as long handoff delay, especially when the home agent (HA) or the correspondent node (CN) is located far away from the mobile node (MN). In this case, the delay for binding updates becomes very high, which may result in long handoff delay and high packet loss rate, thereby causing user-perceptible deterioration of real-time traffic. Although several extensions of MIP such as Fast Handovers for MIPv6 [5] and Hierarchical MIPv6 [6] have been proposed to enhance the performance of MIPv6, none of these solutions consider the special design issues in Internet-based WMNs and hence they are not suitable to be directly applied to WMNs without non-trivial modifications.

Aiming to provide efficient mobility management for Internet-based infrastructure WMNs, in this research, a new scalable mobility management architecture considering special design issues in WMNs is proposed first. Secondly, new cross-layer handoff designs along with a data caching mechanism are proposed. Thirdly, a quality-of-service (QoS) handoff mechanism is proposed based on the scalable architecture. Fourthly, location management mechanisms for WMNs are proposed which include

two novel designs. In the first design, a dynamic location management solution in WMNs is proposed. In the second design, resilient location area design in WMNs is proposed.

1.1 Mobility Management in IP-based Wireless Networks

Mobility management enables a system to maintain connections as a mobile user moves into a new service area, i.e., handoff management, and to locate roaming users for packet delivery, i.e., location management. Many solutions have been proposed to provide IP mobility. They are briefly presented in this section.

A handover or handoff is caused by the movement of an MN between two attachment points, i.e., the process of terminating existing connectivity and obtaining new connectivity. Handoffs in IP-based wireless networks may involve the changes of the access point at the link layer and the changes of the IP address and routing at the network layer. Efficient handoff mechanisms ensure minimal handoff latency, signaling overhead, packet loss, and handoff failures.

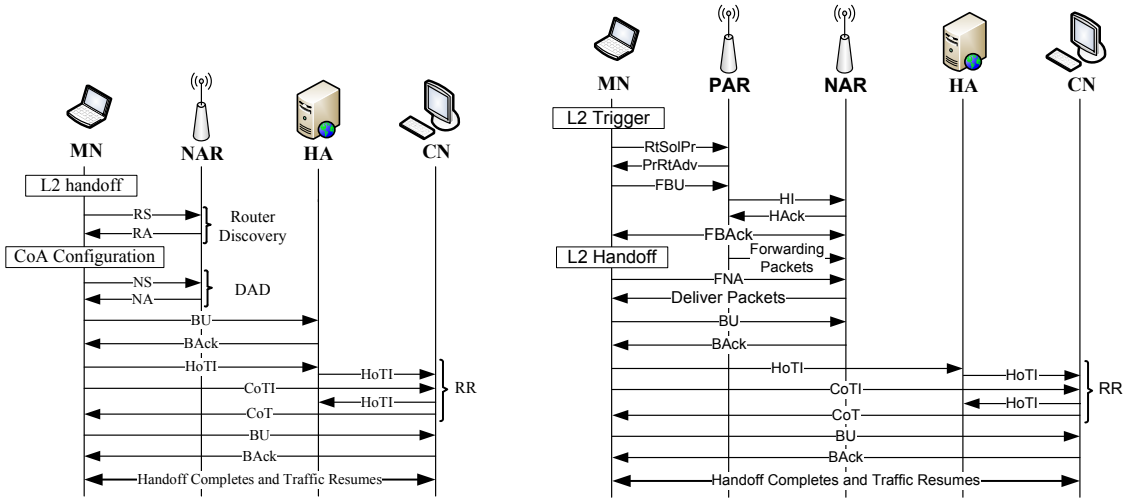
The handoff delay is the time interval in which an MN does not receive any packets from the network during a movement. It can include the link layer (L2) and network-layer (L3) handoff delays. The link-layer handoff takes care of the switch of the communication channel, while the network-layer handoff takes care of the change of the IP address and/or routing path.

1.1.1 Mobile IP (MIP)

Mobile IP (MIP) [3] is proposed as a network layer solution for the mobility support in the global Internet. MIP allows a node to change its current point of attachment (PoA) in the Internet from one subnet to another. MIP introduces three functional units: (i). mobile node (MN): a node that can change its PoA in the Internet; (ii). home agent (HA): a mobility agent located in the home network of an MN; and (iii) foreign agent (FA): a mobility agent located in each visited subnet of an MN. Each MN has two IP addresses: a permanent home address and a temporary

care-of-address (CoA) used to identify the MN in a visited subnet. The home address of the MN never changes. The CoA changes each time the MN changes its PoA to a new subnet. Packets sent from a correspondent node (CN) to an MN are sent to the home address of the MN. Hence, the packets are routed to the home network of the MN first. The HA in the home network intercepts these packets and tunnels them to the CoA of the MN. A tunnel is a path followed by a packet when it is encapsulated within the payload of another packet. The HA tunnels packets destined to the MN through the FA in the visited network. When an MN wants to send packets to a CN, packets are routed to the CN following the direct path. This routing is known as triangular routing.

1.1.2 Mobile IPv6 (MIPv6)



(a) MIPv6 (b) FMIPv6

Figure 1.1: Signaling procedures for: (a) MIPv6; (b) FMIPv6.

The most significant difference between MIPv4 and MIPv6 [4] is that MIPv6 is integrated into the base IPv6 protocol and is not an add-on feature, as in the case of IPv4 and MIPv4. Similar to MIPv4, under MIPv6, each MN is identified by its home address (HoA). While away from its home network, an MN is also associated with a care-of address (CoA), which provides information about the MN's current

location. Discovery of new access router (NAR) is performed through the *Router Solicitation/Advertisement* (RS/RA) message exchange. Furthermore, to ensure that a configured CoA (through stateless or stateful mode [7]) is likely to be unique on the new link, the Duplicate Address Detection (DAD) procedure [7] is performed by exchanging Neighbor Solicitation/Advertisement (NS/NA) messages. After acquiring a CoA, an MN performs binding update to the home agent (HA) through the *Binding Update* (BU) and *Binding Acknowledgment* (BAck) messages exchange. To enable route optimization, the binding update procedure is also performed to all active CNs. However, the return routability (RR) procedure must be performed before executing a binding update process at a CN in order to insure that the BU message is authentic and does not originate from a malicious MN. The return routability procedure is based on the home address test, i.e., the *Home Test Init* (HoTI) and *Home Test* (HoT) message exchange, and the care-of address test, i.e., the exchange of *Care-of Test Init* (CoTI) and *Care-of Test* (CoT) messages. Although the RR procedure helps to avoid session hijacking, it increases the overall delay. Figure 1.1(a) represents the sequence of the message flow used in MIPv6 based on stateless address autoconfiguration. Analysis of MIPv6 shows that it has some well-known disadvantages such as high packet loss rate and handoff latency, thereby causing user perceptible deterioration of real-time traffic. Furthermore, scalability problems arise with MIPv6 since it handles MN local mobility in the same way as global mobility. Simultaneous mobility is another problem MIPv6 faces due to route optimization, which can occur when two communicating MNs have ongoing sessions and they both move simultaneously [8]. These weaknesses have led to the investigation of other solutions to enhance MIPv6 performance.

1.1.3 Fast Handovers for Mobile IPv6 (FMIPv6)

FMIPv6 [5] was proposed to reduce the handoff latency and minimize service disruption during handoffs pertaining to MIPv6. The link-layer information (L2 trigger)

is used either to predict or rapidly respond to handoff events. When an MN detects its movement toward a NAR by using the L2 trigger, it exchanges *Router Solicitation for Proxy* (RtSolPr) and *Proxy Router Advertisement* (PrRtAdv) messages with the previous access router (PAR) in order to obtain information about the NAR and to configure a new CoA (NCoA). Then, the MN sends a *Fast Binding Update* (FBU) to the PAR in order to associate the previous CoA (PCoA) with the NCoA. A bi-directional tunnel between the PAR and NAR is established to prevent routing failure with *Handover Initiate* (HI) and *Handover Acknowledgment* (HAck) message exchanges.

The *Fast Binding Acknowledgment* (FBAck) message is used to report the validation status of the pre-configured NCoA and tunnel establishment to the MN. Moreover, the PAR establishes a binding between the PCoA and NCoA and tunnels any packets addressed to the PCoA towards the NCoA through NAR's link. The NAR buffers these forwarded packets until the MN attaches to NAR's link. The MN announces its presence on the new link by sending the *Router Solicitation* (RS) message with the *Fast Neighbor Advertisement* (FNA) option to the NAR. Then, the NAR delivers the buffered packets to the MN. The sequence of the messages used in FMIPv6 is illustrated in Figure 2(b) for MN-initiated handoffs with the predictive mode. A counterpart to the predictive mode of FMIPv6 is the reactive mode. This mode refers to the case where an MN does not receive the FBack on the previous link since either the MN did not send the FBU or the MN has left the link after sending the FBU but before receiving a FBack. In the latter case, since an MN cannot ascertain whether the PAR has successfully processed the FBU, it forwards a FBU, encapsulated in the FNA, as soon as it attaches to the NAR. If the NAR detects that the NCoA is in use (i.e., address collision) when processing the FNA, it must discard the inner FBU packet and send a *Router Advertisement* (RA) message with the *Neighbor Advertisement Acknowledge* (NAACK) option in which the NAR may

include an alternate IP address for the MN to use. Otherwise, the NAR forwards the FBU to the PAR which responds with a FBack. At this time, the PAR can start tunneling any packets addressed to the PCoA towards the NCoA through NAR's link. Then, the NAR delivers these packets to the MN.

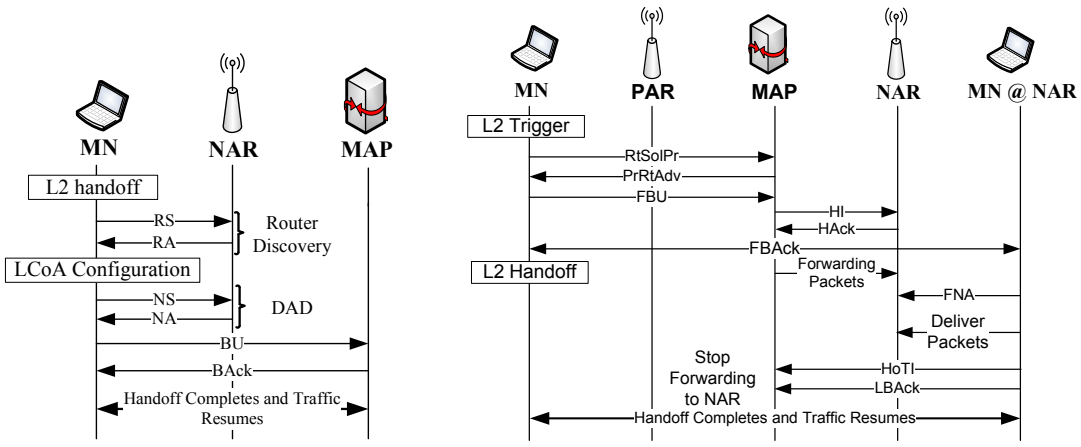
1.1.4 Hierarchical Mobile IPv6 (HMIPv6)

With MIPv6, an MN performs binding updates to the HA/CNs regardless of its movements to other subnets. This induces unnecessary signaling overhead and latency. To address this problem, HMIPv6 [6] was proposed to handle handoff locally through a special node called Mobility Anchor Point (MAP). The MAP, acting as a local HA in a visited network, limits the amount of MIPv6 signaling outside its domain and reduces the location update delay. An MN residing in a MAP's domain is configured with two temporary IP addresses: a regional care-of address (RCoA) on the MAP's subnet and an on-link care-of address (LCoA) that corresponds to the current location of the MN.

As long as an MN moves within the MAP's domain, it does not need to transmit BU messages to the HA/CNs, but only to the MAP when its LCoA changes. Hence, the movement of an MN within a MAP domain is hidden from the HA/CNs. For inter-MAP domain roaming, MIPv6 is used rather than HMIPv6. When an MN crosses a new MAP's domain, in addition to registering with new MAP, BU messages need to be sent by the MN to its HA/CNs to notify them of its new virtual location. Figure 1.2(a) presents the sequence of message flows used in HMIPv6 with the assumption that an MN has entered into a new MAP domain and the MIPv6 registration procedure was already completed.

1.1.5 Fast Handover for HMIPv6 (F-HMIPv6)

Combination of HMIPv6 and FMIPv6 motivates the design of Fast Handover for Hierarchical Mobile IPv6 (F-HMIPv6) [9]. Like FMIPv6, F-HMIPv6 aims to reduce the handoff latency and packet loss. In F-HMIPv6, the bi-directional tunnel



(a) HMIPv6 (b) F-HMIPv6

Figure 1.2: Signaling procedures for: (a) HMIPv6; (b) F-HMIPv6.

is established between the MAP and the NAR, rather than between the PAR and the NAR as in FMIPv6. After signaling message exchanges between an MN and the MAP based on FMIPv6, an MN follows the normal HMIPv6 operations by sending a local BU (LBU) message to the MAP. When the MAP receives the LBU with the new LCoA (NLCoA), it stops packet forwarding to the NAR and then clears the established tunnel.

In response to the LBU, the MAP sends a local BAck (LBAck) message to the MN and the remaining procedure follows the operations of HMIPv6. In the original F-HMIPv6 proposal, when handoff anticipation cannot be supported, regular operations of HMIPv6 are used [9]. Hence, HMIPv6 corresponds to the reactive mode of F-HMIPv6. Figure 1.2(b) illustrates the sequence of messages used in F-HMIPv6 when an MN moves from the PAR to the NAR within the MAP's domain and the MAP already knows the adequate information on the link-layer address and network prefix of each AR. This illustration is based on the assumption that an MN has entered into a new MAP domain and that MIPv6/HMIPv6 registration procedures were already completed.

1.2 Wireless Mesh Networks (WMNs)

A generic WMN is comprised of a combination of static *mesh routers* and mobile *mesh nodes* (MNs). Mesh routers form a wireless multihop backbone network. Some mesh routers function as the gateways and are connected via wired links to the Internet. Mesh routers are dedicated nodes for routing wireless traffic either from MNs to the wired Internet or between MNs. MNs access the network via a mesh router which serves as the access point (AP). With the help of multihop connectivity among mesh routers, the number of required Internet entry points can be reduced. Therefore, WMNs may cover a large area with low deployment cost.

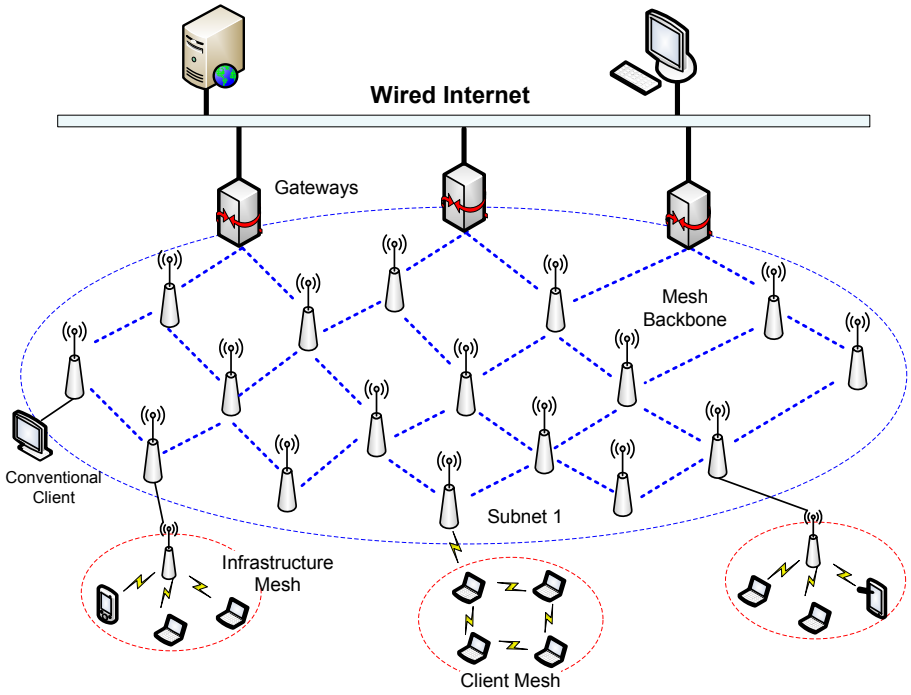


Figure 1.3: A hybrid wireless mesh network.

Depending on their architecture and deployment configuration, WMNs can be broadly categorized into three main types: infrastructure mesh, client mesh, and hybrid mesh networks [1].

- Infrastructure mesh: This type of WMNs includes mesh routers forming a multihop wireless infrastructure for clients. The multihop mesh infrastructure back-

bone can be built using various types of radio technologies, e.g., IEEE 802.11 and IEEE 802.16 technologies. Typically, two types of radios are used in each mesh router, one for backbone communications and one for user communications. With the gateway functionality, mesh routers can be connected to the Internet via a single hop or multiple hops. Generally, clients are connected to mesh routers via a single wireless hop. Infrastructure WMNs are the most commonly used type.

- Client mesh: Client meshing provides mutlihop networking among client devices. In this type of WMNs, only client nodes constitute the actual network. They perform routing and configuration functionalities to provide end user applications. A packet destined to a node in the network hops through multiple nodes to reach the destination. Hence, a mesh router is not required for client mesh networks. Client WMNs are usually formed using one type of radios on devices. Moreover, the requirement on client devices is increased as compared to infrastructure meshing, since in client WMNs, the client nodes must perform additional functions such as routing and self-configuration.
- Hybrid mesh: As illustrated in Figure 1.3, a hybrid mesh network is the most generic type of WMNs, combining the concepts of infrastructure and client mesh networks. A hybrid WMN consists of relatively static mesh routers which form the multihop wireless backbone. Mesh routers are dedicated nodes for routing wireless traffic either from mobile client nodes to the wired Internet backbone or between mobile client nodes. Mobile clients can act as a dynamic extension of the static infrastructure part by implementing routing and packet forwarding functionality. A client network can be a Wi-Fi network, a cellular network, a sensor network, etc. At least one node inside the client network is connected to a mesh router in the wireless backbone.

1.3 Issues of Mobility Management in Wireless Mesh Networks

Mobility management in Internet-based WMNs is not a simple extension of traditional mobility management schemes to multihop wireless networks. The performance of the mobility management schemes designed for cellular and Mobile IP networks is based on the good performance of mobility-related signaling traffic delivery in the wired infrastructure network. However, when these schemes are applied to Internet-based WMNs, the good performance of signaling traffic delivery is no longer guaranteed.

First, in WMNs, signaling messages for mobility management must go through multiple wireless hops from a mesh client to its mesh router and then multiple wireless hops again among mesh routers to reach the wired backbone. It is well-known that throughput degrades quickly when the number of hops along wireless connectivity increases [10][11], due to the delay of medium access, route discovery, route recovery, and so on. Hence, this multihop wireless transmission increases the transmission delay of signaling messages, packet loss probability, number of retransmissions, signaling overhead, and so on. As a result, it increases the delay and failure rate of location update, paging, and handoff.

Second, although extensive research on routing and medium access control (MAC) has been conducted to address the scalability issue in multihop wireless networks, these protocols are designed with the goal of improving data throughput but at the cost of generating more signaling overhead messages. These additional signaling messages compete for scarce wireless resources with the signaling messages required for mobility management, which aggravates the performance of mobility-related signaling-traffic delivery. Moreover, even low-rate signaling traffic can produce detrimental effects on the performance of WMNs [12].

Regarding handoff management, new handoff design issues which were not problems in traditional handoff management will arise in Internet-based WMNs, such as

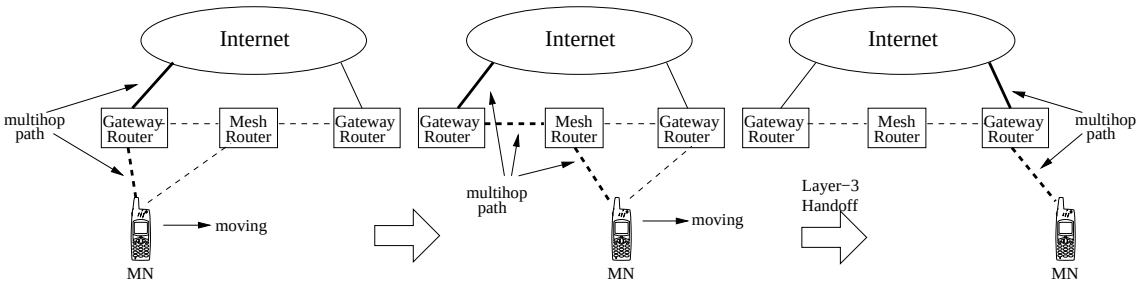


Figure 1.4: A roaming scenario requiring a L3 handoff in an Internet-based WMN.

in the scenario of a L3 handoff. A L3 handoff is triggered when an MN changes its connectivity to the Internet from one subnet to another. Without loss of generality, we assume that each gateway is connected to a different subnet in the Internet. Hence, in this case, a L3 handoff is triggered when an MN's movement causes its Internet connection change from one gateway to another, as shown in Figure 1.4. The new design challenge here is that how an MN knows that it has moved from one gateway to another when the MN is connected to the Internet via multiple mesh routers, i.e., a new design challenge for L3 handoff detection.

Regarding location management, one of the core issues in location management is how often a location update (LU) is needed so that an MN does not consume excessive battery on location update and the network can deliver packets to the MN efficiently with short delay. As shown in Fig. 1.3, when an MN silently roams from one subnet to another, it has visited a number of foreign agents (FAs) before an LU action is triggered. The impact of packet traversal (i.e., data packets for packet delivery (PD) from the HA to the MN or control packets for LU from the MN to the desired location entity) between the FAs in the shaded area on the performance has not been addressed in traditional location management, since all FAs are wired connected. The packet traversal delay is negligible as compared to the wireless counterpart and there is no interference caused by the transmission among FAs. However, if the infrastructure network is replaced by wirelessly connected MRs, this issue can no longer be ignored. Therefore, the number of wireless hops a data/signaling packet traverses

is an important factor that can affect the performance of location management in a multihop wireless mesh backbone and should be considered when addressing 1) the path setup for PD from the traffic initiator to the MN which is important to provide efficient location management for each MN in terms of the PD delay and 2) the path setup for LU packet traversal from the MN to the desired location entity which is important to provide robust location management in terms of LU overhead in the mesh backbone when the number of MNs increases.

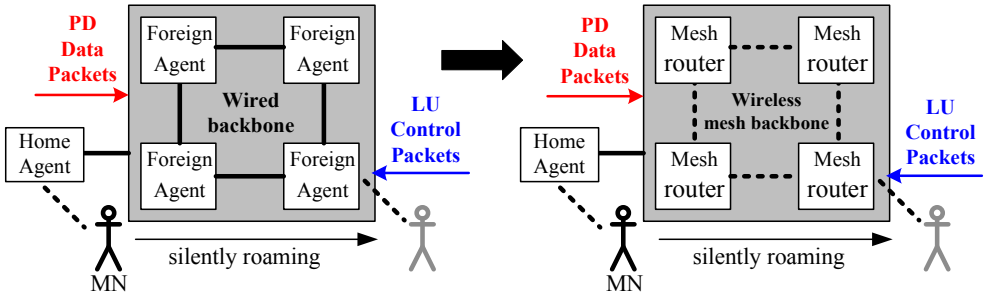


Figure 1.5: New challenges for location management arise when wired-connected FAs become wireless-connected MRs.

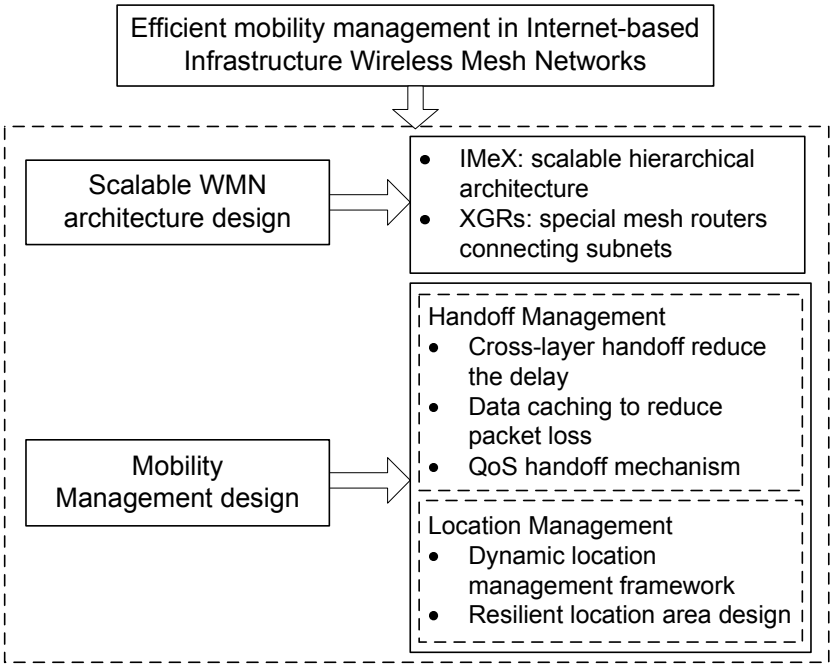


Figure 1.6: Overview of proposed architecture and mobility management designs in Internet-based infrastructure WMNs.

1.4 Overview of Proposed Architectural and Mobility Management Designs

Unlike the traditional mobility management design in which mobility management and architectural designs are considered independently, in this research, integrated design of architecture and mobility management for Internet-based infrastructure WMNs is proposed. An overview of the proposed designs is shown in Figure 1.6.

Based on the above, this thesis is divided into the following chapters:

- Chapter 1 is the introduction to the whole thesis.
- Chapter 2 briefly introduces related work on mobility management techniques in wireless networks. Their limitations if applied to infrastructure WMNs are pointed out. In addition, a useful simulation tool, OPNET[13], which is beneficial for modeling the WMN architecture and mobility management is introduced.
- Chapter 3 describes the proposed handoff management architecture for WMNs, called IMeX, to specifically address the special handoff design challenges in Internet-based WMNs. It can facilitate parallel executions of handoffs from multiple layers.
- Chapter 4 depicts the proposed data caching mechanism on top of the proposed cross-layer scheme in Chapter 3, which guarantees minimum packet loss during handoffs. In addition, the optimal number of mesh routers and their placement to form the proposed IMeX handoff architecture are determined.
- Chapter 5 uses the proposed IMeX as the handoff architecture to facilitate the proposed quality-of-service (QoS) handoff mechanism.
- Chapter 6 presents a dynamic location management solution (DoMaIN), which addresses the new location management challenges in Internet-based WMNs. In addition, the proposed DoMaIN framework facilitates a new dynamic location update triggering method which is suitable for the multihop wireless mesh backbone.

- Chapter 7 introduces the proposed resilient location area design (ReLoAD) which can achieve a balanced tradeoff between signaling overhead caused by the paging procedure and MN power consumption caused by the LU procedure for both intra- and internet sessions, while simultaneously preserve the QoS performance of existing traffic of active MNs.
- Chapter 8 concludes the whole thesis. It summarizes the work, highlights the contributions of this thesis, and presents some possible future work based on this thesis.

CHAPTER 2: BACKGROUND

Mobility management is vital for realizing large-scale wireless mesh networks to provide cost-effective broadband Internet access. Although a considerable amount of research on mobility management for cellular, Mobile IP, and mobile ad hoc networks has been proposed, mobility management for IP-based WMNs, including the mobility support from both the network and link layers, remains largely unexplored.

2.1 Existing Handoff Management Schemes

A complete handoff procedure for mobile multimedia applications in Internet-based WMNs requires the mobility support from the link-layer, network-layer, and application-layer [14]. The L2 handoff process in IEEE 802.11-based wireless networks can be divided into three steps: scanning, authentication, and reassociation [15]. The L3 handoff process includes the IP address and routing path update. In addition, if the application-layer mobility support is provided based on the Session Initiation Protocol (SIP) [16], which is an application-layer mobility solution adopted by 3GPP [17] for IP-based streaming multimedia services, the L5 handoff process includes session redirection steps.

2.1.1 Existing L2 Handoff Schemes

The scanning delay occupies the largest proportion of the entire link-layer handoff latency (more than 90% [18]). It involves the delay in order to help an MN find potential APs to reassociate with. There are two types of scanning in the IEEE 802.11 standard: passive and active. Both scanning modes require a full scan to probe all channels. Since the time associated to a full scan is very long, many researchers propose different approaches to selectively scan the most possible channels [15]. In [19], the concept of *neighbor graph* is proposed which captures the mobility topology of

each MN. APs in the neighbor graph of an MN are candidate handoff APs. Partial scan is conducted to probe channels only in the neighbor graph. In the *selective scanning* algorithm proposed in [20], a selected subset of all available channels is probed. Channel selection is performed by means of a channel mask which is formed by the most frequent channels used by all APs. Another fast handoff scheme, *SyncScan* [21], tries to avoid the slow scanning process by requiring all APs to be synchronized in sending beacons. During the full pre-scan process, an MN switches to each channel at pre-determined moments. In such a way, a complete picture of all nearby APs can be observed in advance and thus no scanning is needed to find the best AP during a handoff. Similar idea of pre-scan is also used in the *Proactive Scan* [22] and *MultiScan* [23] fast handoff schemes.

2.1.2 Existing L3 Handoff Schemes

Mobile IPv4 [3] and Mobile IPv6 [4] are the main mobility solutions at the network layer. A significant amount of research has been conducted on reducing the L3 handoff latency [24]. They can be broadly classified into two categories. The first category aims to reduce the address update time by using a hierarchical network architecture to limit address registrations within a domain for intra-domain mobility, such as *Hierarchical Mobile IPv6 Mobility Management* (HMIPv6) [6], *Intradomain Mobility Management Protocol* (IDMP) [25], and *Dynamic HMIP* (DHMIP) [26]. The second category uses link-layer event triggers to anticipate the handoff initiation time and prepare for the network-layer handoff in advance [5]. In addition, host-specific-routing-based protocols adopt new routing schemes to support intradomain mobility, and thus reduce the L3 handoff delay. In these protocols, such as Cellular IP [27] and HAWAII [28], standard IP routing is not used for intradomain mobility management.

2.1.3 Existing L5 Handoff Schemes

Mobility support at the application layer has also been attempted by SIP [16] and MOBIKE [29]. The basic idea of handoffs using SIP involves an MN sending

a RE-INVITE message to the correspondent node (CN) to update the application session. This message informs the CN about the MN's new address. MOBIKE allows both the MN and CN to have several IP addresses. When the MN changes its IP address, it sends a notification to the CN about the new address.

2.1.4 Existing Handoff Schemes in WMNs

A number of work has been conducted to provide the network-layer handoff support in WMNs. Existing work supports the mobility by either managing node address changes [30, 31] or modifying the multihop routing protocol to facilitate handoffs [32, 33, 34, 35, 36]. In addition, multicast routing is proposed in *SMesh* [36] wireless mesh system. *SMesh* reduces the handoff delay by assuming that all MNs work in the ad hoc mode, which is not always true in real systems. However, none of the existing WMN handoff schemes specifically address the new L3 handoff detection issue in WMNs, as explained previously using Figure 1.4. In addition, none of them adopt a cross-layer approach and attempt to reduce the total handoff delay caused from multiple layers.

2.1.5 Data Caching for Mobility Management

The caching scheme has been adopted in mobility management in wireless networks. To reduce the L2 handoff delay in WLANs, a cache mechanism is introduced in [20]. The information of the APs involved in an MN's recent handoff history is maintained in a cache at the MN. When an L2 handoff is needed, if the new AP has a matched cache entry, the MN can associate to the AP without any further probing procedures. A proactive neighbor caching (PNC) scheme is proposed in [37] to reduce the reassociation delay of L2 handoffs. The PNC scheme uses a neighbor graph to dynamically cache the required authentication information needed at an MN's neighboring APs for the purpose of pre-positioning the MN's mobility context and reducing the authentication delay involved in L2 handoffs. For location management, a location cache in WMNs is proposed in [38] to cache mobile stations' location

information so that the network can efficiently route packets to mobile users. To the best of our knowledge, no existing work has applied data caching mechanisms for handoff management in WMNs.

2.1.6 Existing QoS Schemes in WMNs

In the literature, a number of mechanisms are proposed addressing the issue of gateway placement and load balancing in WMNs [39, 40, 41]. These papers propose gateway load balancing schemes under the assumption that mesh routers or gateways can reach one another as if they belong to the same IP subnet. None of them consider the routing problem among gateways that belong to different subnets. Our proposed gateway selection algorithm aims to addressing this missing part and facilitating QoS handoffs across domains. On the other hand, considerable amount of work addressing the QoS routing issue in WMNs is proposed [42, 43, 44]. However, none of these proposed schemes consider the interaction with existing protocols in WMNs. Our proposed QoS traffic forwarding scheme leverages the Neighbor Discovery Protocol (NDP) [7] in IPv6 to provide the up-to-date QoS information for routing decisions while preserves the precious wireless bandwidth.

2.2 Existing Location Management Schemes

Location management enables a system to track the locations of mobile terminals between consecutive communications. It includes two major tasks. The first is location registration or location update, where the mobile terminal periodically informs the system to update relevant location databases with its up-to-date location information. The second is packet delivery, where the system determines the current location of the mobile terminal based on the information available at the system databases when a communication for the mobile terminal is initiated.

2.2.1 Location Management in Cellular and WLANs

Various location management schemes have been proposed for cellular and WLANs in the literature [26][45][46]. Centralized location caching and paging schemes for MNs

in the idle mode are not suitable in a WMN environment due to the scalability issue.

Many dynamic location management schemes [47, 48, 49, 50, 51] have been proposed for traditional single-hop wireless networks such as time-based, movement-based, and distance-based schemes. Numerical results in [52] show that the distance-based LU scheme has a better performance in terms of a lower overall cost for LU and paging when compared to the time- and movement-based schemes. However, the distance-based scheme cannot be directly applied to Internet-based infrastructure WMNs (IiWMNs), since it does not consider the impact of multihop wireless transmissions of signaling or data packets in the location management design.

Moreover, a dynamic hierarchical mobility management strategy (DHMIP) for MIP is proposed in [26], in which different hierarchies are dynamically set up for different users and the signaling burden is evenly distributed among the network. Thus, signaling overhead in DHMIP can be greatly reduced compared to that in HMIP. The concept of pointer forwarding to dynamically decide the optimal threshold of the forwarding chain is incorporated in [45][53]. In the pointer forwarding scheme, certain percentage of all location updates are accomplished by the mobility agent instead of always carrying out location updates to the home location register (HLR) which is considered to be the traffic bottleneck. However, these pointer forwarding schemes cannot be directly applied to WMNs without considering multihop routing in WMNs.

2.2.2 Location Management in MANETs and WMNs

The comparisons of the performance of various scalable location services for MANETs are studied in [54]. However, none of these schemes can be directly applied to infrastructure WMNs, as they are designed in consideration of those characteristics unique to MANETs, e.g., they focus on the study of node mobility impact on location services and only consider traffic in Intranet sessions where traffic is inside the network. According to our best knowledge, [38] is the first published work that addresses loca-

tion management design in WMNs. In [38], a distributed location cache scheme for WMNs is proposed that caches each MN's location information in mesh routers while routing the data for the MN. However, this scheme only considers location updates when an MN initiates an active data session, but does not consider the case if an MN only receives data packets but not send, or the MN silently moves with no active data sessions. In addition, the requirement to ensure the time synchronization of all mesh routers to share the freshest location information can be an implementation burden in real systems. Therefore, such flat location management architecture can cause scalable problems in WMNs. Therefore, hierarchical and efficient location management in WMNs considering special design challenges of WMNs is needed to minimize the signaling overhead on the MN side while still tracking MNs efficiently.

2.3 OPNET Modeler for Modeling WMNs

One way to evaluate networking architecture and protocols is using simulations. As one of the leading simulators for network research and development, OPNET [13] provides powerful simulation capability for the study of network architectures and protocols. It is widely used in both industry and academia. Compared to another well-known simulator NS-2[55], OPNET has a well-engineered user-interface using mainstream software and operating system which are attractive to network operators. Another reason to choose OPNET is the fact that it contains a vast amount of models for commercially available network elements and has various real-life network configuration capabilities, which makes the simulation of real-life networks close to reality. OPNET is built on top of a discrete event system which simulates the system behavior by modeling each event happening in the system and processes it by user-defined processes. It uses a hierarchical strategy to organize all the models to build a whole network. Other features of OPNET include GUI interface, comprehensive library of network protocols and models, source code for all models, graphical results and statistics, etc.

2.3.1 Network Deployment and Planning in OPNET

OPNET Modeler can provide different levels of modeling depending on the necessities and requirements of the simulation. OPNET simulations are discrete-event-driven simulations (DES). A simulation in OPNET is divided into three-tired structures, namely network model, node model, and process model. Generally, the simulator comes with a huge library of pre-defined models for various simulations and has the facility for users to define custom models. The GUI feature in OPNET helps to establish an overall environment called a Project. From that Project, the operator can develop several network scenarios in order to evaluate and analyze the performance of that network in different "what-if" circumstances.

Fig. 2.1 shows the workflow for planning and analyzing a WMN in OPNET. The configuration of a customized WMN network model can be produced by utilizing and interacting two basic deployments in OPNET: network and traffic deployment, each of which follows different implementation procedures. In addition, by defining certain performance metrics and running the DES simulation, the initial global and local trail data can be collected for further parameter tuning to help regenerate an revised and strengthened network model.

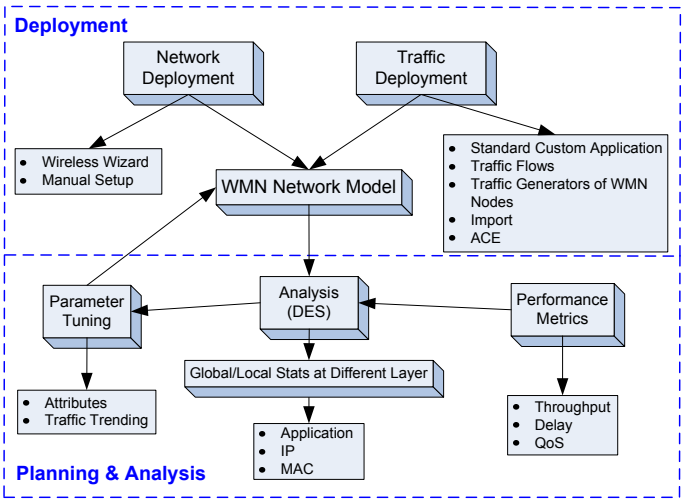


Figure 2.1: Workflow for planning & analyzing WMN networks in OPNET.

2.3.2 An Internet-based Infrastructure WMN Architecture for Handoffs

Based on the procedures of network deployment in OPNET as shown in Fig. 2.1, we set up a GUI-based project to form the WMN architecture. A scenario of an IiWMN architecture with end-to-end applications is shown in Fig. 2.2. The WMN project includes several functional entities: gateway mesh routers with an additional wired interface that allows traffic to-and-from the Internet; common mesh routers which have multiple wireless interfaces: one interface for forming the mesh backbone and another interface for end users to allow stations (STA) or mobile nodes (MNs) to communicate to various correspondent nodes (CNs) located in the Internet via the home agent (HA). When a roaming MN follows certain movement trajectory, it first disconnects with the current AP, then associates to a new one and resets up a new multihop path to reach the Internet.

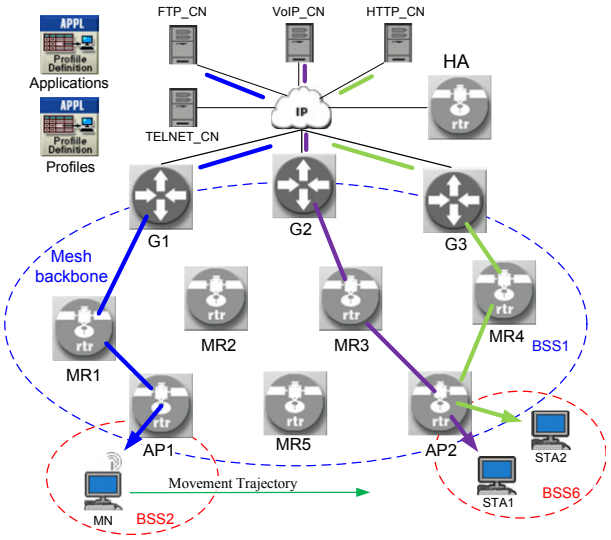


Figure 2.2: A scenario of an IiWMN architecture in OPNET.

In this study, we assume the wireless interface of all mesh routers is based on the IEEE 802.11b standard. If the transmission and reception threshold is set to be 0.05W, the default transmission range is less than 300m in OPNET. In order to set up an exact multihop routing within the mesh backbone, the end-to-end IP traffic demand feature in OPNET Modeler is utilized which can give detailed packet

traversal information (e.g., the number of hops), as shown in Fig. 2.3.

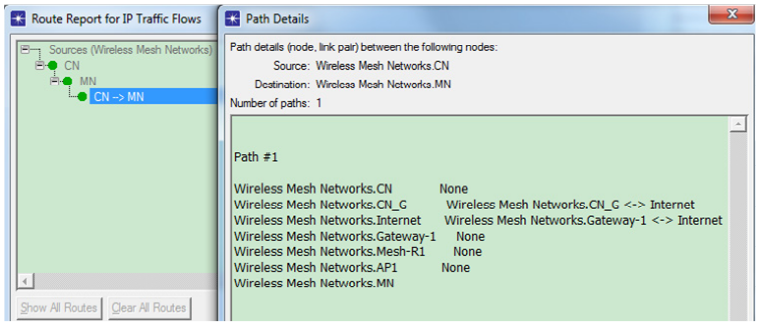


Figure 2.3: Determine a multihop path in OPNET.

2.3.3 Node Models Used for Simulation in OPNET

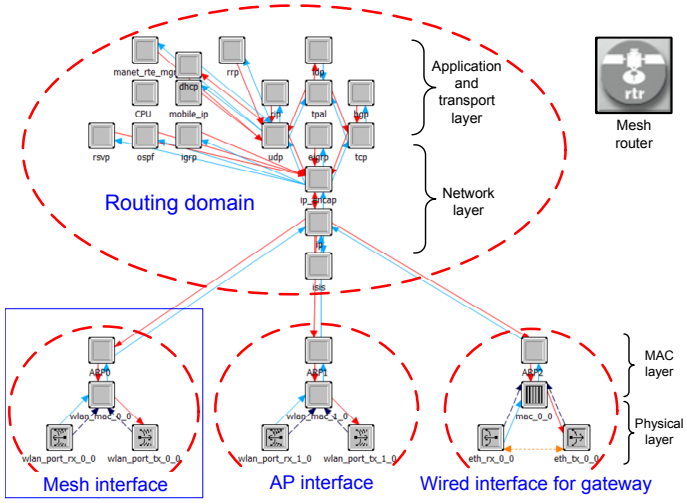


Figure 2.4: Node topology of a gateway mesh router in WMNs.

Several necessary node models in OPNET are used in this research:

1. Gateway Mesh Router: Based on the existing mobile ad hoc network (MANET) gateway node in OPNET Modeler library which provides the connectivity for a multihop network to the Internet, we customize an additional wireless interface to provide the mesh backbone connectivity, as shown in Fig. 2.4. A gateway mesh router has three separate interfaces for providing the wired connectivity to the Internet, the wireless connectivity for forming the mesh backbone, and the wireless connectivity with AP functions.

2. Common Mesh Router: Based on the IEEE 802.11b standard, the common mesh router differs from the gateway mesh router in the number of interfaces available. The common mesh router has only two interfaces and both interfaces are for the wireless connectivity. These interfaces support the operation of a router in two separate channels.
3. End Users: The static or mobile stations are the end user products which has only one wireless interface. In addition, the movement trajectory of an end user can be defined in order to enable the roaming capability.
4. Application and profile node modules are capable of defining various end-to-end applications such as HTTP, FTP, and other services or delay sensitive traffic such as Voice over IP (VoIP) and video conferencing.

Since IPv6 provides more resilient features than IPv4, the IiWMN is deployed based on IPv6 in OPNET. Each interface of mesh routers is assigned with at least one unique global IPv6 address for IP-based traffic. The link-local IPv6 addresses of mesh routers are used for control message exchanges between mesh routers. We assume that mesh routers in the wireless mesh backbone can find the best multihop route to a gateway using any multihop routing protocol [56, 57, 58].

2.4 Conclusion

In this chapter, the mobility management issues in Internet-based infrastructure WMNs are described and a comprehensive literature review of existing mobility management methods is presented. Previous mobility management schemes proposed for cellular and wireless local area networks (WLANs) cannot be directly applied to WMNs due to the multihop wireless links in WMNs. Moreover, since the multihop wireless links increase the end-to-end packet delivery delay of both signaling messages and data packets, new challenges of mobility management arise in WMNs. Finally, a useful simulation tool, OPNET, which is beneficial for modeling the WMN architecture and mobility management is introduced.

CHAPTER 3: INTER-GATEWAY CROSS-LAYER HANDOFFS IN INFRASTRUCTURE WMNS

Existing handoff management schemes for wireless networks, are designed independent of the underlying network architecture design, hence, there are inherent limitations in those schemes. A new WMN architectural design is proposed to position and configure mesh routers in order to form a scalable wireless mesh backbone for mobility assistance. The benefit of this approach is that the protocols used for address management and handoffs can be streamlined to take advantages of the resulting network architecture.

3.1 Problem Description

A new handoff design issue in Internet-based WMNs is the L3 handoff detection issue, that is, how an MN knows that it has moved from one gateway to another when the MN is not directly connected to the gateway in WMNs. Since different subnets have different address prefix, an MN can tell whether it has moved into a new subnet (i.e., a new gateway) from the address it may obtain after its movement. Depending on how an MN obtains a new address from the new subnet, there are two possible handoff designs based on the conventional Mobile IP scheme with extensions to specifically address the layer-3 handoff detection issue, as described in the following.

3.1.1 Default-based Handoff Design

In this design, the foreign agent (FA) functionality in Mobile IP is implemented in each AP, that is, each AP is responsible for assigning a new care-of-address (CoA) to each MN. Hence, during the network deployment phase, each AP should be assigned to a specific gateway for Internet access and allocated available CoAs that correspond to the subnet represented by the assigned gateway. The complete handoff procedure

(layer-2, layer-3, and layer-5 handoffs) is shown in Figure 3.1 and we call this design the *default-based* handoff design.

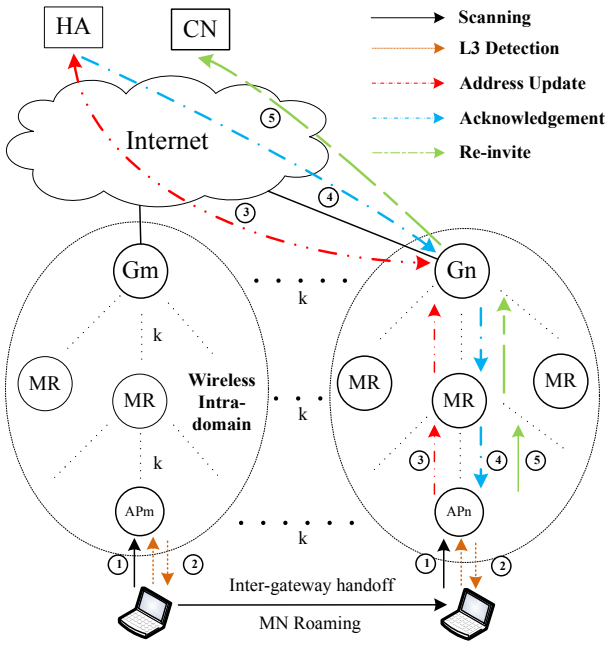


Figure 3.1: A default-based handoff design in Internet-based WMNs.

The handoff procedure starts from scanning for the channel of the new AP (*step (1)*). After the layer-2 handoff is completed and the MN is associated to a new AP, the MN obtains a CoA through the received *Agent Advertisement* message broadcast from the new AP (*step (2)*). From the prefix of the received CoA, the MN knows that whether its Internet connection has changed to a new gateway or not. If yes, the MN sends a *Binding Update* message to its home agent (HA) to update its new CoA (*step (3)*) and the HA replies with a *Binding Acknowledgement* message (*step (4)*). The MN also needs to send a message to its CN to update their multimedia communication session (*step (5)*). Note that before sending a *Binding Update* message, if the multihop routing protocol adopted by the wireless mesh backbone is a reactive routing protocol (e.g., the ad hoc on-demand distance vector routing (AODV) [59] protocol), the MN initiates a route discovery process to find a path to the new gateway first.

3.1.2 Gateway-based Handoff Design

In the second design, the FA functionality in Mobile IP is implemented in the gateways, that is, only the gateways can assign CoAs to MNs. The advantage of this design is that mesh routers are not pre-assigned to specific gateways during the deployment phase and they can use dynamic routing to balance the traffic load passing through each gateway [40]. The complete handoff procedure of this *gateway-based* handoff design is shown in Figure 3.2.

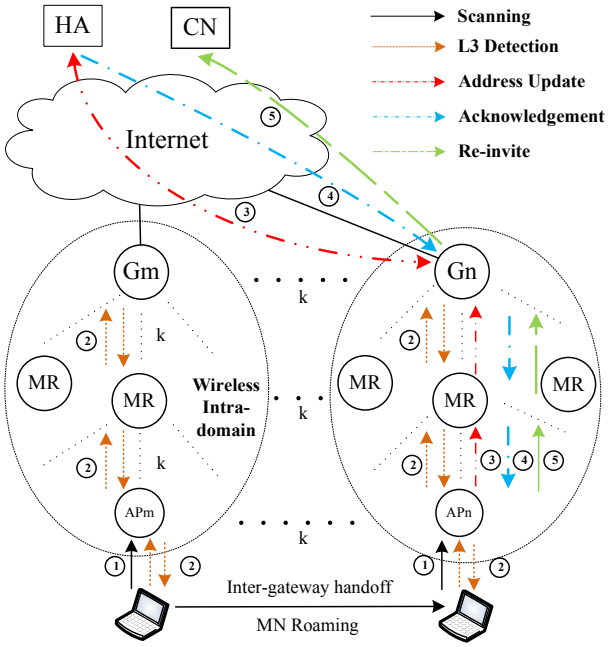


Figure 3.2: A gateway-based handoff design in Internet-based WMNs.

After the L2 handoff is completed (*step (1)*), the MN sends a *Gateway Request* message to request a CoA (*step (2)*). This message is forwarded by the associated mesh router of the MN to a gateway based on the adopted multihop routing protocol. The gateway replies with a *Gateway Reply* message which contains a CoA. From the received CoA, the MN can tell whether it needs a L3 handoff or not. Therefore, we call the delay of completing *step (2)* layer-3 handoff detection delay. If the MN is connected to the Internet via a different gateway, the MN sends a *Binding Update* message to its HA (*step (3)*) and the HA replies with a *Binding Acknowledgement*

message (*step (4)*). The MN also needs to send a session redirection message to its CN (*step (5)*).

3.1.3 Summary

From the above descriptions, it can be seen that the default-based handoff design has shorter layer-3 handoff detection delay, but the gateway-based handoff design has the flexibility of using dynamic routing for load balancing among gateways. For both designs, the total handoff delay, including the delays incurred in L2, L3, and L5, is summarized in Figure 3.3.

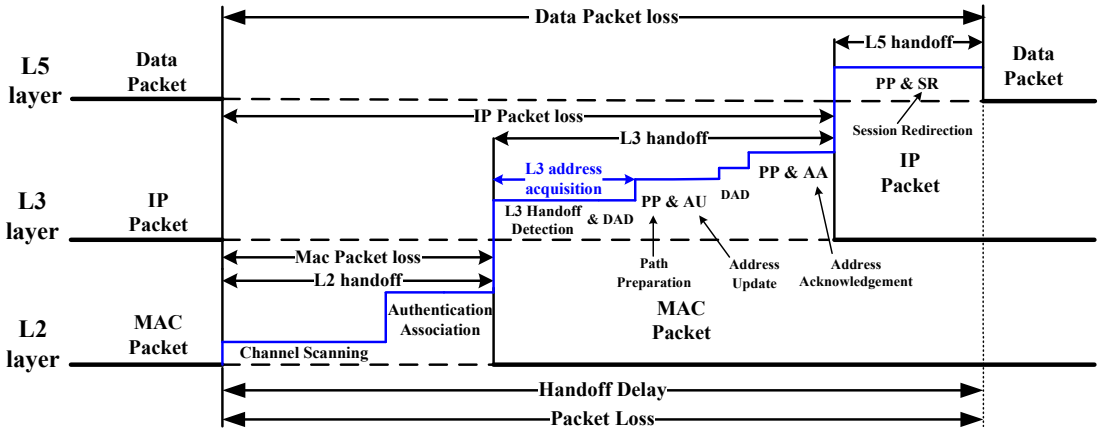


Figure 3.3: Handoff delays using conventional multiple layer handoff design.

From Figure 3.3, it can be concluded two possible design strategies to reduce the total handoff delay and packet loss in Internet-based WMNs:

1. Efficient architectural design to facilitate cross-layer handoffs so that some delays shown in Figure 3.3 can be eliminated, if L3 and L5 handoffs can be prepared in advance during the process of L2 handoffs;
2. Data caching in handoff candidate (cAPs) of an MN so that packet loss can be minimized during an MN's handoff (explained in detail in Chapter 4).

3.2 Proposed Approach

A new architectural design is proposed to position and configure mesh routers in order to form a scalable wireless mesh backbone for mobility assistance. Then, under

the proposed architecture, a cross-layer handoff scheme to reduce the handoff delay and a data caching mechanism (described in detail in Chapter 4) to minimize packet loss are proposed.

3.2.1 Proposed Architecture Design

Under the proposed WMN architecture, mesh routers are grouped into *connected* groups rooted at each gateway mesh router. Each group corresponds to a different subnet and mesh routers belonging to different groups have different IP address prefix. Special mesh routers, namely Xcast-based Group Routers (*XGRs*), are equipped with multiple IP addresses with each address corresponding to a different subnet. Hence, a *XGR* belongs to more than one groups. Note that a mesh router can use the Neighbor Discovery Protocol (NDP) [7] in IPv6 to map different IP addresses to the MAC address of the router. *XGRs* are the bridging nodes connecting different groups, as shown in Figure 3.4. They can facilitate information exchange between different groups during inter-gateway handoffs.

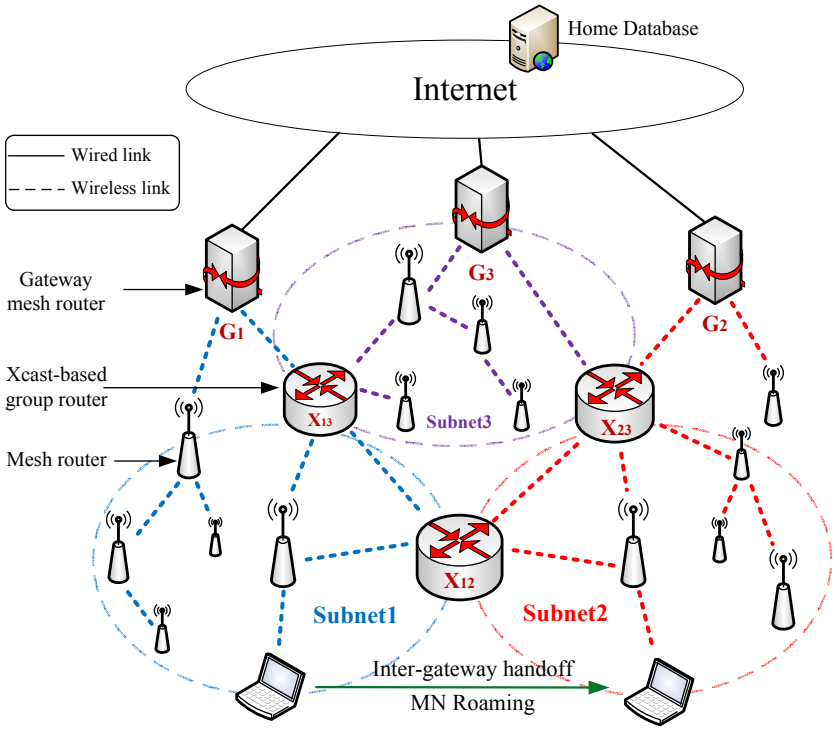


Figure 3.4: An IMeX architecture with three gateways.

The proposed IMeX architecture has the following advantages for handoff management in WMNs:

- By planning groups during the deployment, each mesh router knows that which subnet it belongs to in advance. This design makes it straightforward for address management and L3 handoff detection. In addition, load balancing among gateways is also possible since *XGRs* can direct traffic between different groups. Therefore, our XMesh design combines the advantages of both the default-based and gateway-based designs.
- Information sharing for network management purposes for intra-subnet roaming will be restrained to within a group, instead of broadcasting to the whole mesh backbone, which saves signaling overhead. Information sharing between groups (inter-subnet) is implemented with the help of *XGRs*.
- Xcast-based data caching mechanism based on the planned groups can facilitate some steps of the handoff procedure processed (explained in detail in Chapter 4) without affecting end-to-end applications and thus, guarantees a minimum packet loss during inter-gateway handoffs.
- Since the IMeX architecture can facilitate the cross-layer protocol design and *XGRs* are able to exchange handoff information and cache data packets between different subnets, both intra- and inter-gateway mobility can be supported and improved.

3.2.1.1 Proposed Cross-layer Handoff Designs

A cross-layer handoff scheme under the IMeX architecture is proposed. The basic idea behind this design is to take advantages of the planned group-based IMeX architecture and utilize the information obtained from the L2 to predict the L3 and L5 handoffs in advance so that the L3 handoff detection delay can be eliminated and some of the L3 and L5 handoff steps can be carried out before an MN completes an L2 handoff. Fig. 4.1 shows the sequence of the handoff delays involved in the

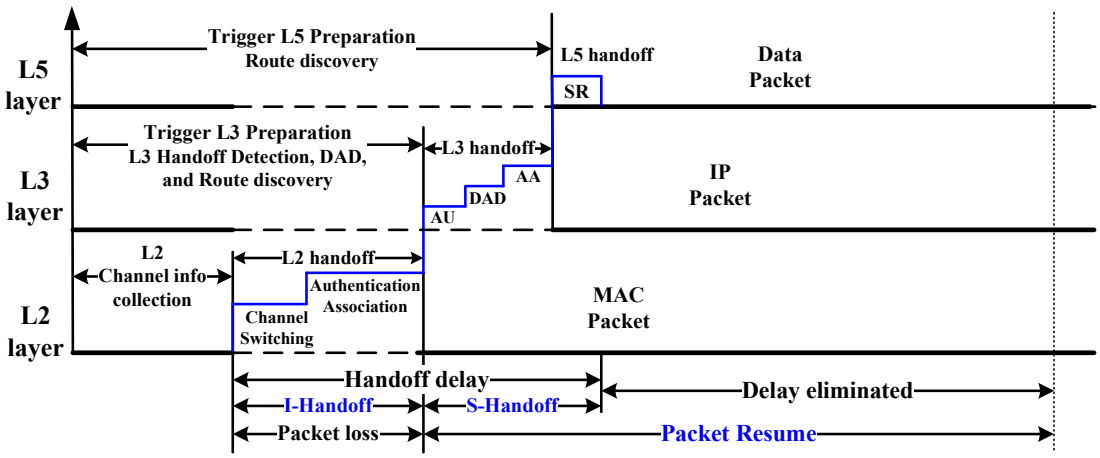


Figure 3.5: Handoff delays based on IMeX architecture.

L2, L3, and L5 of the proposed cross-layer handoff scheme and the complete handoff procedure is shown in Algorithm 1. The notations we use throughout the rest of the paper are listed in Table 3.1.

Table 3.1: Notations Used for Cross-layer Handoffs

Notation	Definition
N	The number of subnets/gateways
M	The number of IP addresses an XGR has
oAP	Old AP before a handoff
nAP	Neighboring AP
cAP	Candidate AP for a handoff
$RSSI_{cur}$	Current received signal strength indication
$RSSI_{L2}$	RSSI threshold of L2 handoff preparation
HO_{TH}	RSSI threshold of L2 handoff

3.2.2 L2 Handoff Preparation

The main purpose in L2 handoff preparation is to obtain the channel information and network ID of an MN's handoff cAP in the new subnet. The network ID of the cAP is needed to locate the XGR that connects the subnets of the oAP and cAP for L3 and L5 handoff preparations. Since there are numerous proposals for obtaining the channel information of nAPs in advance in L2 handoff designs based on mobility prediction [60, 15], we incorporate any of the existing 802.11 fast L2 handoff

schemes in our design to obtain the channel and network ID information of nAPs in advance. Our goal here is not to propose another mobility prediction or fast L2 handoff scheme. Instead, we utilize the existing fast L2 handoff schemes to obtain the necessary information needed for L3 and L5 handoff preparations.

In our design, when an MN's current received signal strength indication ($RSSI_{cur}$) decreases to the threshold of L2 handoff preparation ($RSSI_{L2}$), the MN is triggered to use the adopted fast L2 handoff scheme to obtain the channel and network ID (IPv6 address prefix) information of nAPs. The MN sorts nAPs and obtains the preferred cAP for the handoff. Then, the MN notifies the oAP which sends a group message containing the preferred cAP's network ID to its group to locate the *XGR* that connects the groups of the oAP and cAP. After this, the L2 handoff preparation is done. The contribution in this stage is to select the preferred cAP for L2 handoff before the RSSI from the oAP drops to the L2 handoff threshold (HO_{TH}) and also locate the *XGR* for L3 handoff preparation. In this way, the L2 scanning delay can be reduced to one channel switching delay when the L2 handoff starts.

3.2.3 L3 Handoff Preparation

The L3 handoff preparation starts when the MN triggers the oAP to notify the *XGR*. The *XGR* first checks whether the cAP is located in the current subnet of the MN or not. For the intra-gateway case, the IP address of the MN does not need to change. The *XGR* prepares the routing path between the cAP and *XGR* as well as the routing path between the *XGR* and the gateway for the MN in advance. For the inter-gateway case, the corresponding *XGR* which belongs to both the old and new subnets first formulates an IP address for the MN by using the cAP's network ID and MN's interface ID. This IP address is stored in the *XGR* and cAP's routing table before MN's L3 handoff starts. Furthermore, the *XGR* performs the first DAD procedure for the MN. After that, the *XGR* prepares for the routing path to the cAP and the path to the new gateway. By doing so, the routing path for both the binding

Algorithm 1: Cross-layer handoff algorithm

```
1  while true do
2      if RSSI_cur ≤ RSSI_L2 then
3          MN obtains the channel and network ID information of nAPs;
4          MN sorts nAPs and obtains the cAP for the handoff;
5          MN sends a handoff message which contains the preferred cAP's
           network ID to its oAP;
6          oAP sends a group message to locate the XGR;
7                  /* The XGR starts handoff preparations */;
8          if cAP belongs to another subnet then
9              XGR formulates an address for the MN and performs DAD for the
               new address;
10             XGR prepares the routing path from the XGR to the new gateway,
               to the cAP, and to the CN;
11         else
12             XGR prepares the routing path from the XGR to the old gateway
               and to the cAP;
13     if RSSI_cur ≤ HO_TH then
14         if subnet changes then
15             MN associates to the cAP;
16             MN obtains a new IP address and uses the obtained routing path
               for address binding with the HA;
17             HA performs the DAD and sends back the acknowledgement
               message;
18             MN resumes the multimedia session on L5;
19         else
20             MN associates to the cAP;
21             MN uses the obtained routing path for resuming the multimedia
               session on L5;
```

update to the HA and binding acknowledgement from the HA are prepared for the MN in advance. Nevertheless, since the *XGR* could be multiple hops away from the cAP and the gateway, the path preparation time increases with the number of hops. After the MN finishes the L2 handoff, the cAP sends the MN the new IP address formulated for the MN. The contribution in this stage is to eliminate the L3 handoff detection delay, the first DAD delay, and the routing path discovery delay, which are significant handoff delays in the L3 handoff.

3.2.4 L5 Handoff Preparation

After the *XGR* and cAP finish the MN's L3 handoff preparation, the *XGR* starts a new routing path discovery to the CN. As soon as the L3 handoff is completed, the MN can notify the CN about the new address by sending a session redirection message and resume the L5 session with the CN by using the new IP address. The contribution in this stage is to eliminate the path discovery delay over the wireless mesh backbone for session redirection which is the major delay in the L5 handoff.

3.3 Performance Evaluation

To demonstrate the advantages of the proposed IMeX architecture, we conduct OPNET [13] simulations to evaluate the performance of the proposed cross-layer handoff scheme. We implement new OPNET models for MRs with both MIPv6 and wireless multihop routing functionalities activated so as to realize the handoff support in IP-based infrastructure WMNs. The main implementations in OPNET are made on the L3 and L5 which account for the majority handoff delay in an inter-gateway roaming scenario. Only light L2 modifications are introduced in OPNET by allowing an AP to add its network ID to the L2 *Probe Response* packet. Hence, an MN can obtain the cAP's channel and network ID information during the L2 handoff preparation period which is used for proactive L3 and L5 handoff preparation. In our simulation, both the default and gateway-based WMNs adopt the passive scanning during the L2 handoff period.

3.3.1 Simulation Setup

We develop handoff scenarios to compare our proposed IMeX-based cross-layer handoff scheme with the two conventional WMN handoff schemes: the default-based handoff scheme which depends on *Router Advertisement (RA)* messages to trigger an MN's L3 handoff, as explained in Section 3.1.1, and the gateway-based handoff scheme under which an MN detects an L3 handoff by receiving a reply message from the gateway, as explained in Section 3.1.1. In our simulation scenario, all MRs and gateways' wireless interfaces use AODV [59] and OLSR [61] as the reactive routing protocol and proactive routing protocol, respectively. Each MR is equipped with two radios: one functions as an AP and the other functions as a relay router. The radio transmission range of each MR partially overlaps. Only *XGRs* have multiple IP addresses with each IPv6 address belonging to a different subnet. In our simulation, the Internet backbone network has a constant latency of 0.1 second. The ratio of the HO_{TH} value (RSSI threshold of L2 handoff) to $RSSI_{L2}$ value (RSSI threshold of L2 handoff preparation) is 33% and the channel switching time is set to be 0.05 second in our simulation. The DAD procedure for a new IP address lasts around 1 second. A detailed list of the parameters used in our simulation is shown in Table 3.2.

As an MN moves at a constant walking speed across the subnets, with light ETE video conferencing traffic starting at 60 second, the total handoff delay and packet loss of different designs are simulated. We also use IPv6 traffic demand between different MRs to model background traffic and simulate the ETE delay and delay jitter with varying background traffic and packet interarrival time.

3.3.2 Simulation Results

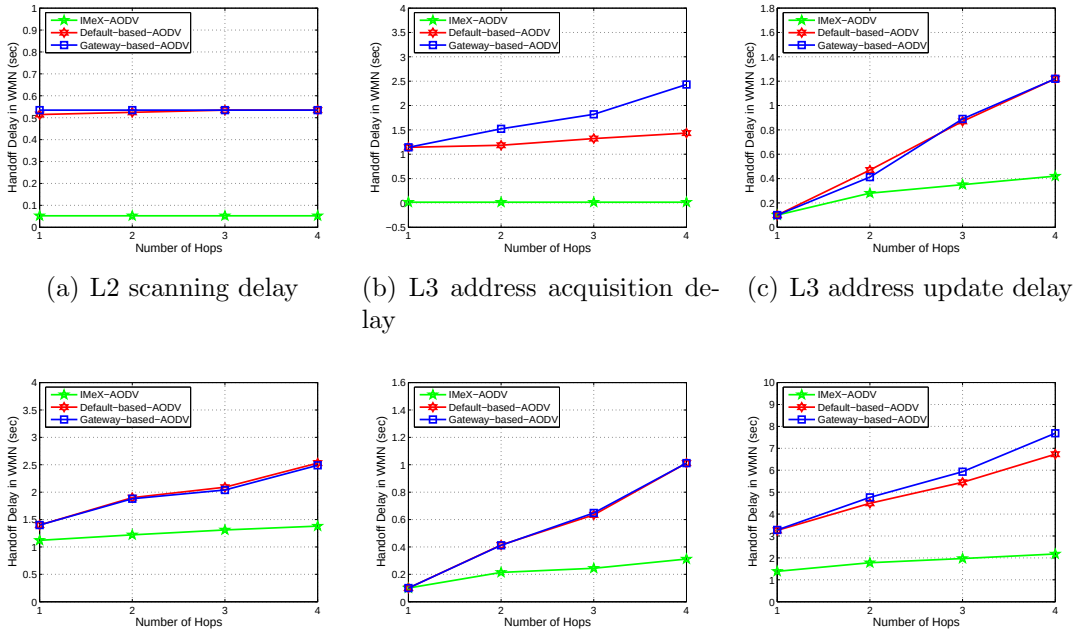
3.3.2.1 Handoff Delays Using AODV Routing Protocol

Fig. 3.6 shows the detailed delay elements incurred in L2, L3, and L5 handoffs versus the number of wireless hops between the MN and its gateway, under the three considered handoff schemes (default-based, gateway-based, and IMeX cross-layer),

Table 3.2: Simulation Parameters

Handoff Common Parameters	
AP transmit power (W)	0.015
Data rate (Mbps)	36
Packet reception-power threshold (dbm)	-95
AP beacon interval (sec)	0.02
Buffer size (bits)	1,024,000
IPv6 interface routing protocol	RIPng+AODV/OLSR
IPv6 *Router Advertisement* interval (sec)	uniform (0.5, 1)
AODV active route timeout (sec)	3.0
OLSR *HELLO* message interval (sec)	2.0
OLSR *Topology Control* message interval (sec)	5.0
MN's ground speed (mi/hr)	3.0
Video Conferencing Parameters	
Start time (sec)	60
Frame size (bytes)	17,280
Frame interarrival time (sec)	constant (0.1)

when the AODV multihop routing protocol is adopted by MRs.



(a) L2 scanning delay (b) L3 address acquisition delay (c) L3 address update delay

(d) L3 address ack. delay (e) L5 session redirection delay (f) Total handoff delay

Figure 3.6: Various handoff delays incurred in L2, L3, and L5 handoffs (using AODV routing protocol).

From Fig. 3.6(a)-(e), it can be seen that our proposed IMeX cross-layer handoff scheme can reduce the delay in every delay component, as compared to the other two schemes. For the L2 handoff delay, under the L2 handoff preparation scheme, the MN obtains the potential channel information before being associated to the cAP. For the L3 address acquisition delay, unlike the case of the default-based and gateway-based handoff schemes in which the MN first needs to wait for either a *Router Advertisement* message or the reply message from a gateway to determine whether it has changed a subnet and then performs the DAD, the MN in our IMeX architecture can start an L3 handoff immediately after an L2 handoff finishes. So the L3 address acquisition delay including the L3 handoff detection delay and a DAD delay in our proposed scheme can be reduced to almost zero. In addition, it is noted that the L3 address acquisition delay in the gateway-based case is larger than that of the default-based case because the gateway-based one has longer L3 handoff detection delay. There is no major difference in the other three delays (L3 address update, L3 address acknowledgment, and L5 session redirection) between the default-based and gateway-based scenarios, since after the MN detects its subnet change, it starts the L3 and L5 handoffs sequentially, which include AODV route discovery, binding update to the HA, binding acknowledgement from the HA, and update to the CN. In our IMeX cross-layer handoff scheme, as the *XGR* triggers the routing path preparation in the target subnet prior to the MN's arrival, the delays of L3 address update, L3 address acknowledgement, and L5 session redirection can be reduced to a value only depending on the multihop signaling message traversal time. In Fig. 3.6(f), the total handoff delay is much lower under our proposed handoff scheme as compared to the other two schemes, because our proposed scheme employs a cross-layer design and eliminates L3 address acquisition and route discovery delay.

Fig. 3.7(a) presents the total handoff delay under different video conferencing packet interarrival time ranging from 0.08 sec to 0.2 sec, when the number of hops

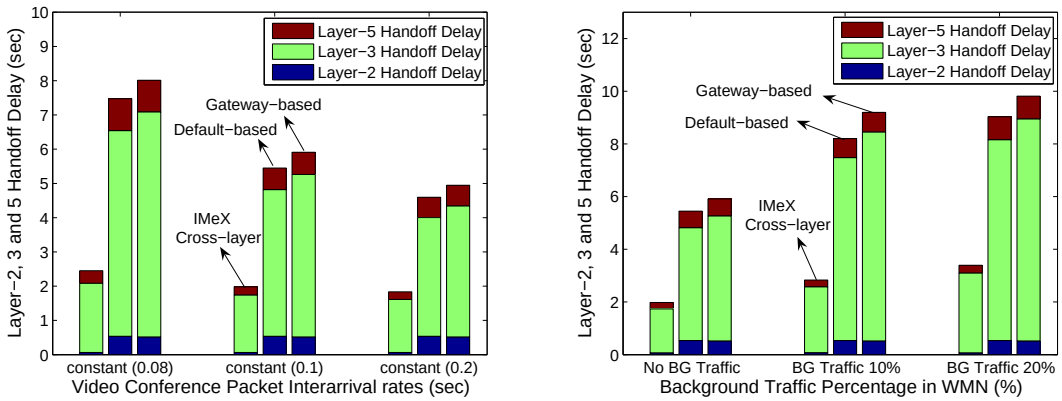
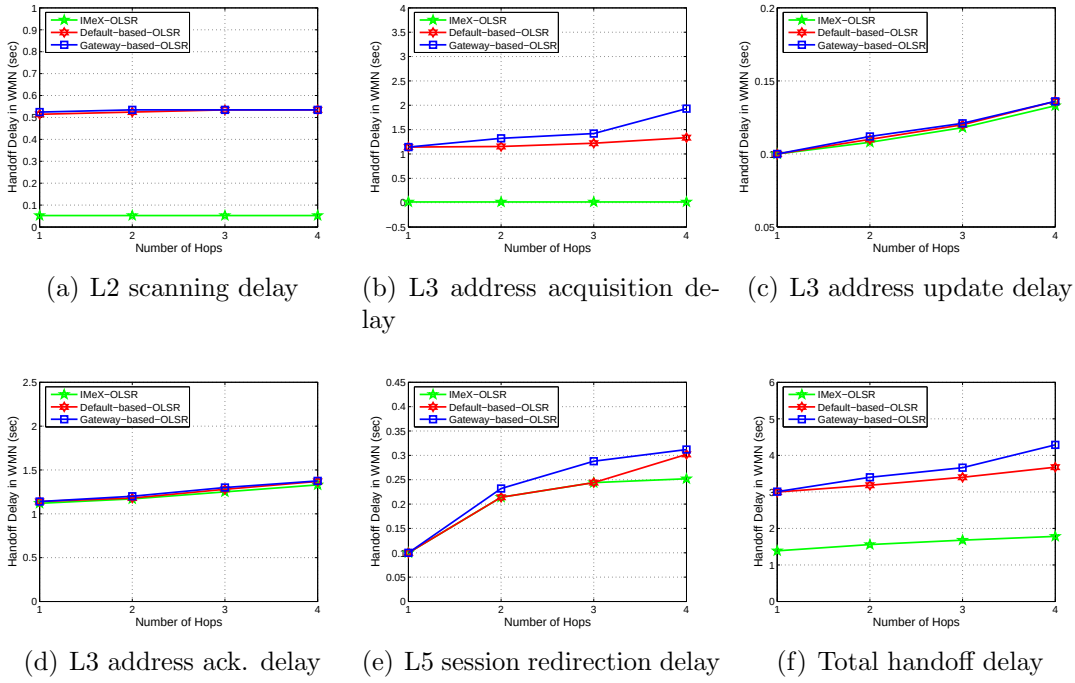


Figure 3.7: Total handoff delay (using AODV routing protocol).

between the MN and the gateway is 3. We see that by adjusting packet interarrival time, handoff delays change. When the packet interarrival time decreases, wireless links become more congested due to the flooded video packets. Hence the total handoff delay (particularly the L3 and L5 ones) increases due to the longer signaling message delivery time. Among all the three considered handoff schemes, our proposed IMeX cross-layer handoff scheme can reduce up to 70% of the total handoff delay, as compared to the gateway-based one. Fig. 3.7(b) depicts the total handoff delay under different percentage of background traffic ranging from 0 to 20% between MRs, when the number of hops between the MN and the gateway is 3. In Fig. 3.7(b), as the background traffic increases, MRs become more congested and the total handoff delay increases under all the three considered handoff schemes, while our proposed IMeX can still cause the lowest total handoff delay.

3.3.2.2 Handoff Delays Using OLSR Routing Protocol

From Fig. 3.7(b), we may see that the L3 delay occupies the largest portion of the total handoff delay under all the three considered handoff schemes. Since the L3 handoff delay in WMNs largely depends on the multihop signaling message traversal delay between the MN and the gateway, an efficient multihop routing protocol,



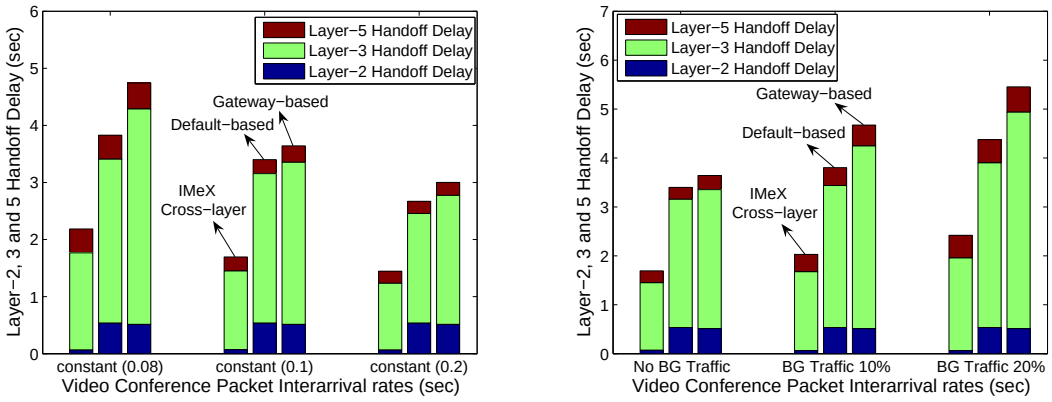
(a) L2 scanning delay (b) L3 address acquisition delay (c) L3 address update delay

(d) L3 address ack. delay (e) L5 session redirection delay (f) Total handoff delay

Figure 3.8: Various handoff delays incurred in L2, L3, and L5 handoffs (using OLSR routing protocol).

which proactively maintains every routing path so that the path discovery delay is eliminated, can also help to reduce the overall handoff delay significantly. Hence, we change the multihop routing protocol adopted by MRs from the reactive AODV protocol [59] to the proactive optimized link state routing protocol (OLSR) [61] and compare the handoff delays under the three considered handoff schemes, as shown in Fig. 3.8(a)-(e). Since the OLSR protocol builds routes and maintains them proactively independent of application traffic, it greatly reduces the delays of L3 address update, address acknowledgement, and L5 session redirection, which largely depend on the routing path discovery delay. Thus, these three delays are similar and very small under all the three schemes. However, our proposed IMeX cross-layer handoff scheme still outperforms the default- and gateway-based schemes with the minimum L3 address acquisition delay (close to zero). Though the delay gap between the proposed and the other two schemes is reduced, IMeX cross-layer handoff scheme still

results in the lowest total handoff delay, as shown in Fig. 3.8(f).



(a) (b)

Figure 3.9: Total handoff delay (using OLSR routing protocol).

Similar to the AODV case shown in Fig. 3.7, Fig. 3.9(a) and (b) show the total handoff delay under different packet interarrival time and different percentage of background traffic, when the number of hops between the MN and the gateway is 3. In Fig. 3.9, as the packet interarrival time decreases, or background traffic increases, the total handoff delay increases in all the three handoff schemes, while our proposed IMeX cross-layer handoff scheme can still cause the lowest handoff delay. Due to the proactive attribute, the overall handoff delay in the default-based and gateway-based cases is much lower than those shown in Fig. 3.7. Additionally, from both Fig. 3.7 and Fig. 3.9, we can conclude that the L3 handoff delay has a significant impact on the total handoff delay.

3.3.2.3 Tradeoffs between Delay and Overhead

Fig. 3.10 presents the tradeoff between the handoff overhead messages generated during a cross-layer handoff and the total handoff delay incurred under our proposed handoff scheme. From the figure we can see that since *XGRs* require more time to pre-set up routing paths for MNs, the total handoff delay and handoff overhead increase as the number of hops between the MN and the new gateway increases. On one hand,

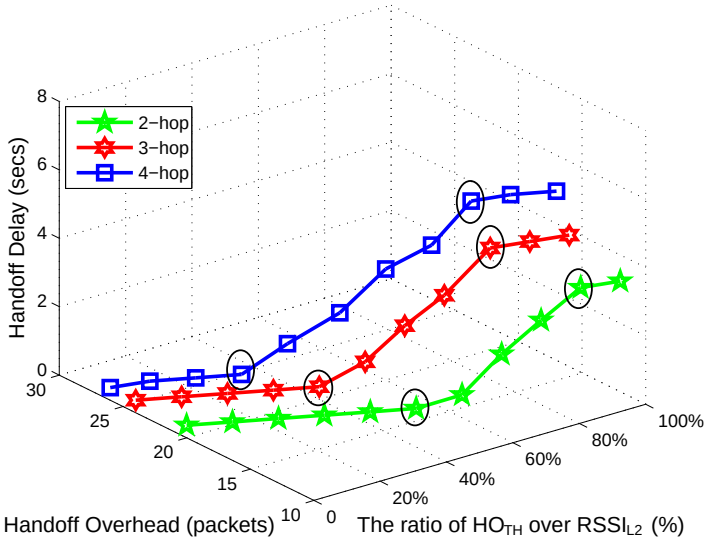


Figure 3.10: Total handoff delay and number of handoff overhead messages.

from the figure, we can see that the overhead is large if an MN is triggered to start the handoff preparation early, i.e., when the ratio of HO_{TH} over $RSSI_{L2}$ is small. On the other hand, the total delay increases if the handoff preparation is triggered late, i.e., when the ratio of HO_{TH} over $RSSI_{L2}$ is large. Therefore, it is vitally important to choose an appropriate handoff threshold in order to balance the tradeoff between the overhead generated during a handoff and the corresponding handoff delay.

3.4 Conclusion

In this chapter, a novel Planned Group Strategy based architectural design to facilitate cross-layer handoffs in WMNs is introduced. By implementing Xcast-based mesh routers(XGRs) which are strategically placed in the mesh back-bone to cover target subnets, inter-gateway handoff preparations can be proactively prepared before an MN loses its connection with the old subnet. The detailed procedure of the proposed cross-layer handoff scheme is described. Through a comprehensive simulation study using the OPNET simulator, the proposed cross-layer handoff scheme is verified to significantly reduce the total handoff delay, as compared to conventional

handoff schemes. Further reduction of the handoff delay can be achieved through efficient multihop routing and MAC protocol design.

CHAPTER 4: XCAST-BASED DATA CACHING FOR HANDOFF MANAGEMENT IN WMNS

After the cross-layer handoff scheme is applied, the remaining handoff delays on each layer may still be a vital cause for the performance degradation of delay-sensitive applications. After an MN is associated to a new AP in a different subnet, it still needs to perform the rest L3 and L5 handoff steps in order to receive packets via the new gateway, during which packet loss is inevitable. With the help of a XGR connecting the old and new subnet, data packets can be cached in cAPs in advance. Then, the MN can resume its receiving traffic right after the L2 handoff. In Figure 4.1, we further divide the handoff delay into two parts in our architecture. The first is an Inevitable Handoff period (I-Handoff) during which the MN loses its connection to the AP. The second is a Skippable Handoff period (S-Handoff) during which the handoff steps can be processed without affecting the end-to-end packet delivery, if data packets can be delivered via the old gateway. Our design goal is to further reduce the handoff delay and packet loss in order to achieve seamless handoffs for delay-sensitive applications.

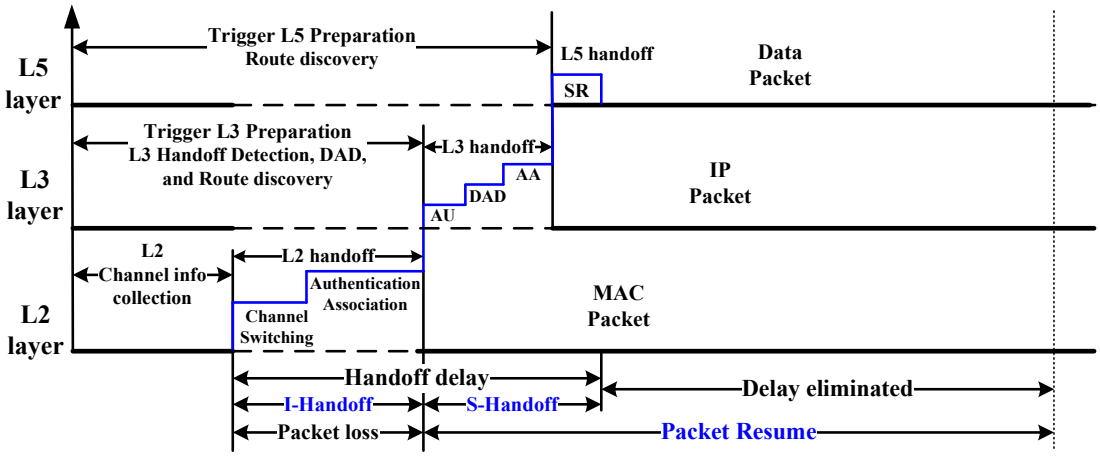
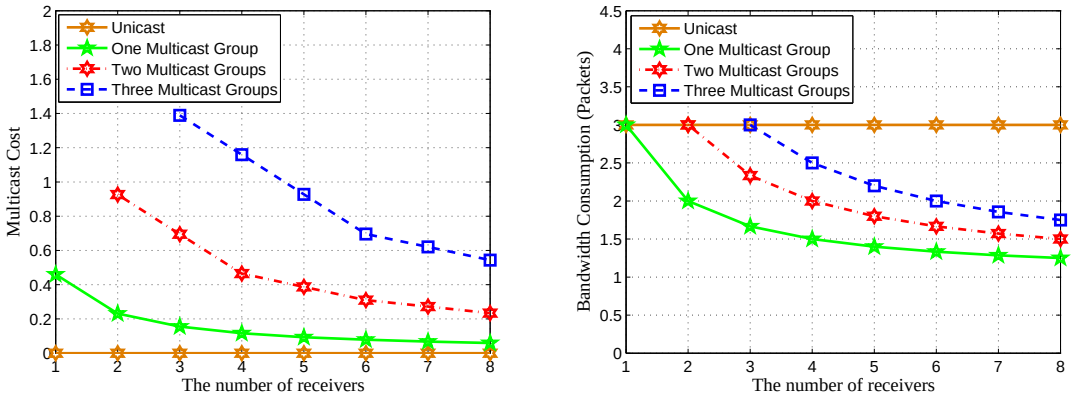


Figure 4.1: Handoff delays based on XMesh architecture.

4.1 Proposed Explicit Multicast (Xcast)-based Data Caching

Standard IP multicast is characterized by the receiver group and multicast routing protocols. Each multicast group is associated with a class-D IP address.In IP-based networks, IGMP [62] is a common protocol used between multicast receivers and their attached routers to set up and maintain the status of the receiver group. Multicast routing protocols are responsible for setting up routing paths and maintaining the membership status of each multicast group. Several third-party multicast routing protocols have been proposed to meet these requirements, such as PIM-SM[63] and DVMRP[64]. These routing protocols generate control messages to set up and maintain each multicast group, which can cause high signaling overhead.



(a) Multicast cost

(b) Bandwidth consumption

Figure 4.2: Multicast vs. unicast.

We compare standard multicast and unicast using OPNET [13] simulations. Figure 4.2(a) shows the multicast cost varying with the number of receivers. The sender is three hops away from all receivers. Here, the multicast cost is defined as the ratio of the total multicast control packets (the sum of IGMP and PIM-SM packets on each multicast rendezvous point router) over the total data packets delivered. When the number of receivers in one multicast group increases, the multicast cost drops. Note that the initial cost is higher when receivers are sparsely distributed in differ-

ent multicast groups because more rendezvous point routers are added to maintain the receiver's groups and set up the forwarding tree. For example, when there is only one receiver in each of the three multicast groups, by delivering 1 data packet, 1.4 control packets on average are generated in the network. Figure 4.2(b) depicts the bandwidth consumption. Bandwidth consumption is defined as the total number of data packets transmitted in the network per data packet delivered. The overall bandwidth consumption for the multicast case decreases as the number of receivers increases. Note that the initial bandwidth consumption is higher when receivers are sparsely distributed in different groups. From Figure 4.2, we conclude that standard multicast incurs higher control overhead when supporting a small number of receivers in each group, while unicast is also not a good option because of its higher bandwidth consumption. Therefore, a multicast-like data forwarding scheme with low control overhead is desirable.

As a complementary scheme of the standard multicast, Xcast is a source-based multicast scheme. It adopts both the simplicity and straightforward principle of unicast, while economizing the link bandwidth as the standard multicast[65]. A Xcast-based routing protocol in mobile ad hoc networks (MANETs) depending on the underlying unicast routing protocol is proposed in [66]. Two effective tree construction algorithms for Xcast based on packet encapsulation in MANETs are proposed in [67]. However, Xcast-based applications such as data caching for WMN handoff management has not been proposed.

In our proposed data caching mechanism, the data sender (gateway) is notified about the number of cAPs with their IP addresses before an MN finishes its L2 handoff. We make the following assumptions in designing the data forwarding mechanism in our handoff-support IMeX:

- The Xcast data packet construction follows a similar way in [65], where the IPv6 header's destination field is filled with a special symbol "Xcast_Group_Router".

The intermediate routers receiving such data packets need to process the routing extension header of IPv6 datagrams which stores explicit IP addresses of handoff cAPs.

- With a good MN mobility prediction, the number of handoff cAPs (i.e., Xcast receivers) can be limited. So the list of handoff cAP's addresses can be included in a data packet without incurring too much overhead.
- Handoff cAPs can be located in different subnets. XGRs connecting multiple subnets are responsible for forwarding inter-gateway Xcast data traffic for caching, which might not be via the optimal path (e.g., least number of hops). However, regular data forwarding can still follow the optimal path based on the underlying multihop routing protocol used among mesh routers.
- AODV [59] and OLSR [61] are considered in our XMesh architecture as examples of the mesh routing protocol for the reactive and proactive routing case, respectively.

In order to implement caching for handoffs, the multihop path setup to multiple receivers is critical for efficient packet delivery and low bandwidth consumption. Therefore, the design of our Xcast-based data caching mechanism abides by 1) maintaining a low control overhead by simultaneously setting up paths for multiple cAPs (the reactive protocol case); 2) keeping low bandwidth consumption for data packet delivery by ruling the selection of Branch Routers (*BRs*), which are responsible for packet duplications to multiple paths, and 3) selecting *XGRs* as *BRs* for duplicating packets to the cAPs that reside in different subnets.

According to the procedures described in the proposed cross-layer handoff scheme in Section 3.2.2, when an MN's L2 handoff preparation is triggered, it first obtains the list of handoff cAPs and then generates a message which signals the current *XGR* group (the gateway is a special *XGR* in the current subnet) to start handoff preparations. After the current gateway receives the message containing the list of

cAPs of the MN, it has to find the forwarding routes to forward data packets to each cAP for data caching.

For reactive routing protocols which are on-demand, we design a XAODV protocol which sets up an efficient and bandwidth-conserved path tree towards multiple cAPs based on the standard unicast AODV routing. The route discovery phase uses similar messages as the *Route Request* RREQ and *Route Reply* RREP in AODV. When the old gateway receives the list of cAPs from an MN, it sends out a XRREQ message including multiple cAPs' addresses. An intermediate router receiving the XRREQ message rebroadcasts the packet, until it finally reaches all the cAPs. Each cAP then sends a reply XRREP. The selection of the *BRs* for packet duplications to different paths relies on the received XRREP messages. When an intermediate router receives multiple replies from the XRREP originators, it chooses the XRREP that can aggregate the maximum number of cAP's addresses and forwards it to its precursor. After the routing path tree is set up, the gateway sends Xcast data packets. The routing extension header of each data packet is processed by each intermediate router based on their routing tables. Packets are duplicated by each *BR* and forwarded to each cAP.

Proactive routing protocols build routes and maintain them independent of application data arrivals. We modify the OLSR protocol to realize a less-sparse tree towards XGRs which are responsible for inter-gateway data caching. In order to conserve bandwidth, OLSR facilitates the selection of Multi-Point Relays (*MPRs*) which serve two purposes: a) generate topology control messages, which condense a dense mesh wireless topology by eliminating redundancies, and b) form a small set of data forwarders. While selecting the *MPRs*, *XGRs* connecting different subnets are preferred as *MPRs* than non-*XGR* routers, even though this operation may result in a larger number of *MPRs*. Selecting a large number of *MPRs* can make the size of Topology Control (TC) messages in OLSR bigger and also increase the number of

forwarding nodes. However, lower bandwidth consumption for data forwarding with shorter data end-to-end packet delivery delay for inter-gateway data caching may justify the overhead cost.

Figure 4.3 shows an example of Xcast-based data caching procedure to three cAPs (R1, R2, R3) using the proposed routing path setup for the inter-gateway handoff purpose. The dashed lines indicate route discovery between nodes. The solid lines represent the Xcast data packet forwarding path. Figure 4.3(a) shows a minimal-sparse case where *XGR2* is in the same subnet of R1, R2, R3. The gateway sends out the data packet containing the IP addresses of R1, R2, R3 to MR5. MR5 sends the packet to the next hop *XGR2*, then the data packet is duplicated at the first *BR* (*MR2*) and finally forwarded to the three cAPs. Figure 4.3(b) and 4.3(c) illustrate a similar case that two cAPs are in the subnet of one *XGR*, while the other cAP belongs to a different *XGR*. These two cases have two BRs (*XGR2, MR2*) for the packet duplication. Figure 4.3(d) shows the super-sparse case where three cAPs belong to the subnets of three different *XGRs*. The data packet is duplicated three times at the *BR* (*XGR2*): the first packet for *MR2*, the second one for (*XGR1*), and the third one for *XGR3*. Case (d) has higher bandwidth consumption compared to the other three cases.

4.2 Required Number of *XGRs* and Optimal Placement

In our IMeX architecture, *XGRs* are special MRs which have multiple IP addresses. Each IP address belongs to a different group which corresponds to one subnet. Given the number of available gateways in a WMN, i.e., the number of subnets a WMN covers, an implementation issue is that the minimum required number of *XGRs* under the proposed IMeX architecture should be obtained. In addition, how the available *XGRs* are configured to connect different groups should also be figured out before the network is deployed.

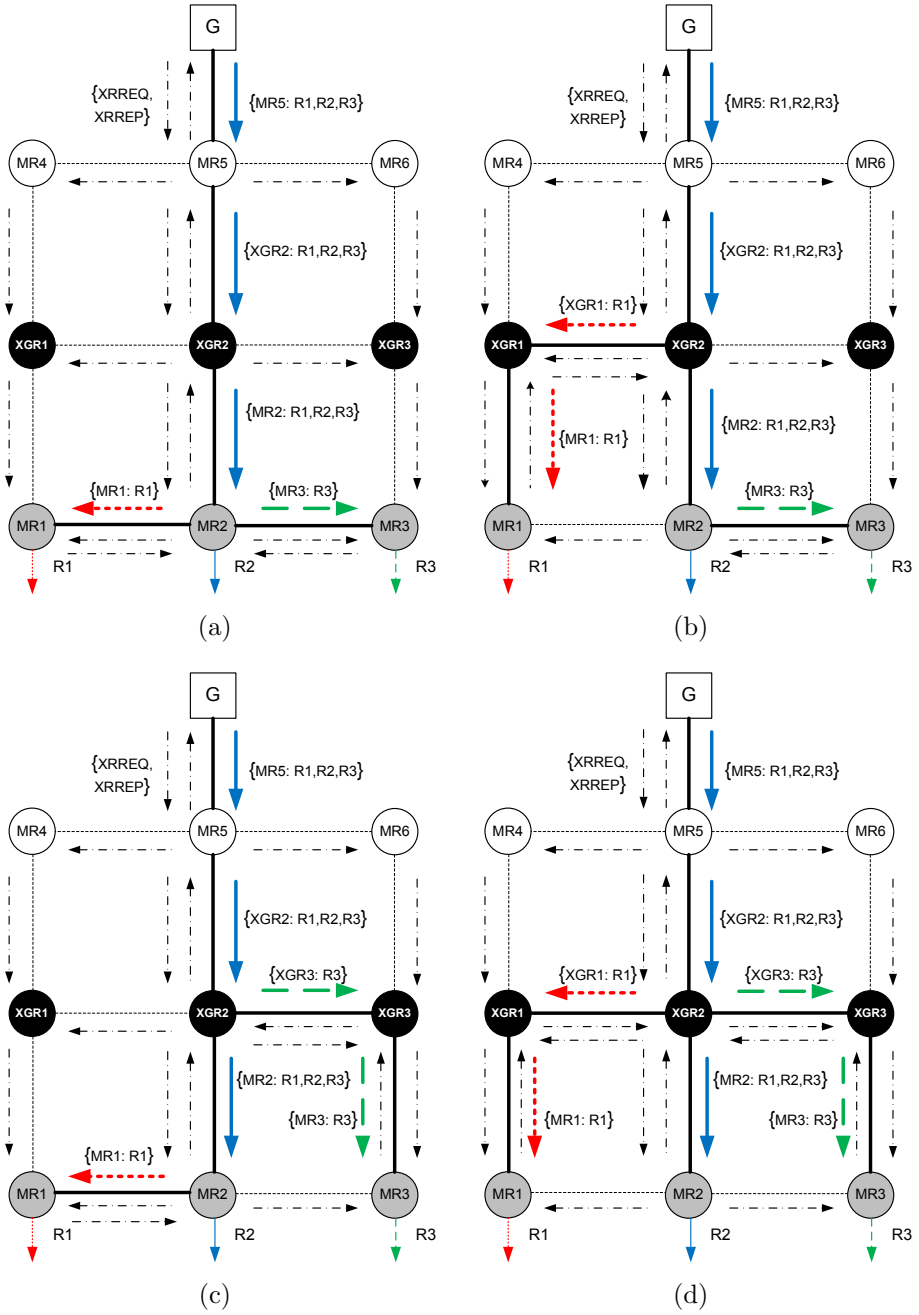


Figure 4.3: Various Xcast data caching forwarding cases (XAODV routing protocol). (a) – {XGR2:R1,R2,R3}; (b) – {XGR2:R1,R2} & {XGR3:R3}; (c) – {XGR2:R2,R3} & {XGR1:R1}; (d) – {XGR1:R1},{XGR2:R2} & {XGR3:R3}.

4.2.1 Problem Formulation

Assume that there are N gateways available and each XGR has M IP addresses. The goal of the proposed planned group strategy is to group $XGRs$ into N groups,

each rooted at one of the N gateways. Every group member has an IP address corresponding to the represented subnet of the group. The design requirement is to have at least one common *XGR* for every pair of groups, given the implementation constraints N and M.

Our objective is to find the minimum required number of *XGRs*, given a specific pair of (N, M) (M and N are natural integers, $M, N \geq 2$). This problem can be modeled as a *set covering* problem as follows:

Given a universal finite set $X = \{G_{12}, G_{13}, \ldots, G_{ij}, \ldots, G_{N-1,N}\}$, where G_{ij} is the common MR between any two groups i and j, $i \neq j$, $1 \leq i,\ j \leq N$, $|X| = C_N^2 = \frac{N(N-1)}{2}$, and a family of subsets $\mathscr{F} = \{S_1, \ldots, S_k, \ldots, S_m\}$, where S_k is a subset of X and $m = C_N^M$. Each S_k corresponds to a *XGR* which is the common MR for at most C_M^2 pairs of groups, i.e., $|S_k| = C_M^2$. The indexes i and j of the elements G_{ij} in each S_k can be at most M different integers.

Find the minimum-size set cover $\mathcal{C}^*$ such that $X = \bigcup_{S_k \in C^*} S_k$. An illustration of the set covering problem for an example of $(N = 9, M = 3)$ is shown in Fig. 4.4.

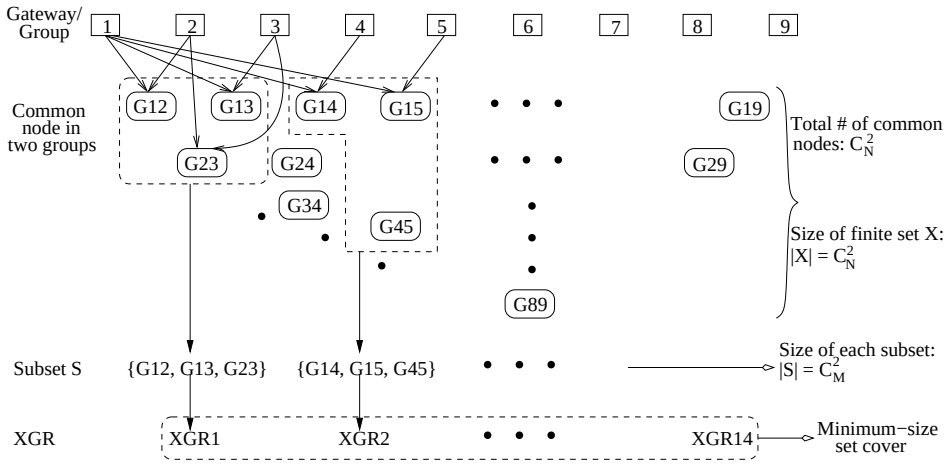


Figure 4.4: Illustration of the set covering problem.

4.2.2 Greedy Algorithm and Optimal Placement

Since a set covering problem is NP-hard, we design a greedy heuristic algorithm to find the set cover $\mathcal{C}$. The greedy algorithm picks the subset S_k that covers the

greatest number of the remaining elements that are not covered at each stage.

- *Step 1*: the greedy algorithm starts from picking the first subset S_1, i.e., the first *XGR*, R_1. Elements G_{ij} in S_1 are from $i = 1$, and $j = 2, \ldots, M$, i.e., $S_1 = \{G_{12}, G_{13}, \ldots, G_{1M}\}$. Thus, the M IP addresses of R_1 can be assigned to group $1, 2, \ldots, M$ and R_1 is the bridging node of group 1 and 2, group 1 and 3, ..., and group 1 and M.
- *Step 2*: continue Step 1 from $i = 1$ and $j = M + 1$ for elements in subset S_2, so $S_2 = \left\{G_{1(M+1)}, G_{1(M+2)}, \ldots, G_{1(2M-1)}\right\}$ and *XGR* R_2 belongs to group $1, M + 1, M + 2, \ldots, 2M - 1$. This process continues from $i = 1$ and $j = 2M$ for the next subset until $j = N$.
- *Step 3*: note that the *XGR* corresponds to the subset containing G_{1N} may have remaining available IP addresses that can be assigned to other groups. In this case, the greedy algorithm adds additional elements to this subset in a reverse direction, that is, keep $j = N$ but increase the value of i from $i = 2$ to $i = N-1$ to determine which group i should be assigned a remaining IP address. When determining the value of i, the greedy decision-making is that to choose the group that can add as many un-selected elements to the subset as possible (with ties broken arbitrarily). For example, if elements $G_{1(N-1)}$ and G_{1N} are already in a subset, when determining the next element to be selected, the greedy algorithm checks groups from $i = 2$ to $i = N - 1$ to see whether element G_{1i}, $G_{i(N-1)}$, and G_{iN} have already been added in any of the already selected subsets. Choose the group i that can add the most un-selected elements to the subset.
- *Step 4*: repeat Step 1 to Step 3 for the rest of the subsets from $i = 2$. For a fixed i, increase j from $j = i + 1$ to $j = N$. When determining whether a group j should be assigned an IP address, the greedy algorithm checks that whether this group j can result in the most un-selected elements to be added to the subset

at the same time. If yes, select this group j. The algorithm then increases the value of j and continues to determine the next group to be assigned. When j reaches N and there are remaining available IP addresses for this router, then a reverse direction search is conducted by keeping $j = N$ and increasing the value of i in order to add additional elements to this subset.

- *Step 5*: the algorithm stops when all elements in X are selected. Then we find the set cover $\mathcal{C}$ such that $X = \bigcup_{S_k \in C} S_k$. The required number of *XGRs* is $|\mathcal{C}|$.

Fig. 4.5 shows the outcome of the greedy algorithm for the case $(N = 9, M = 3)$.

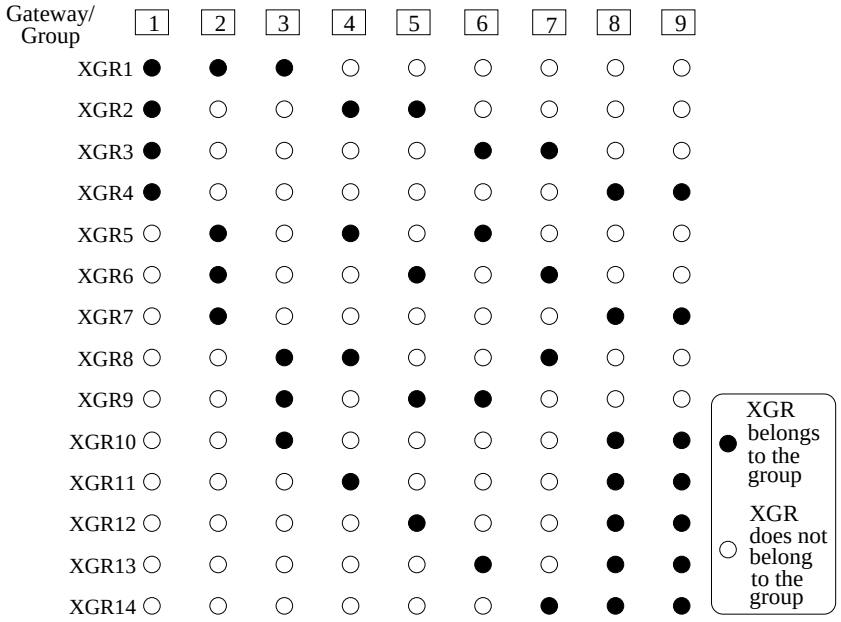


Figure 4.5: Outcome of the greedy algorithm for the case $N = 9$ and $M = 3$.

Theorem 1. *The size of the set cover found by the greedy algorithm, $|\mathcal{C}|$, is bounded by a function of the size of the minimum-size set cover, $|\mathcal{C}^*|$, and the number of the elements in X, $|X|$.*

Proof: The size of each subset, $|S_k|$, is C_M^2. After the first subset is picked by the greedy algorithm, the number of remaining un-covered elements in X is: $n_1 = |X| - C_M^2$. Among these n_1 remaining elements that need to be covered, at least

one of the remaining subsets S_i must contain at least $n_1/\left(|\mathcal{C}^*|-1\right)$ such elements because otherwise the optimal solution would have to contain more than $|\mathcal{C}^*|$ subsets. Therefore, after the greedy algorithm picks the second subset that contains the largest number of un-covered elements, the number of remaining un-covered elements is:

$$n_2 \leq n_1 - \frac{n_1}{(|\mathcal{C}^*|-1)} = n_1\left(1-\frac{1}{(|\mathcal{C}^*|-1)}\right) \leq n_1\left(1-\frac{1}{|\mathcal{C}^*|}\right). \tag{4.1}$$

Similarly, the number of remaining un-covered elements after the third subset is picked by the algorithm is:

$$n_3 \leq n_2 - \frac{n_2}{(|\mathcal{C}^*|-2)} \leq n_2\left(1-\frac{1}{|\mathcal{C}^*|}\right) \leq n_1\left(1-\frac{1}{|\mathcal{C}^*|}\right)^2. \tag{4.2}$$

In general, we have

$$n_i \leq n_1\left(1-\frac{1}{|\mathcal{C}^*|}\right)^{i-1}, \tag{4.3}$$

where n_i is the number of remaining un-covered elements in X after the ith subset is picked by the greedy algorithm. Assume $k = |\mathcal{C}|$, that is, the set cover found by the greedy algorithm has k subsets. Based on the above analysis, from (4.3), the number of remaining un-covered elements in X after k subsets are picked by the greedy algorithm is: $n_k \leq n_1\left(1-1/|\mathcal{C}^*|\right)^{k-1}$. In the worst case of the greedy algorithm, $n_k = n_1\left(1-1/|\mathcal{C}^*|\right)^{k-1}$. Since k subsets have already been picked, n_k

should be less than one. So in the worst case scenario,

$$\begin{aligned} n_1\left(1-\frac{1}{|\mathcal{C}^*|}\right)^{k-1} &< 1, \\ \left(1-\frac{1}{|\mathcal{C}^*|}\right)^{|\mathcal{C}^*|\frac{k-1}{|\mathcal{C}^*|}} &< \frac{1}{n_1} = \frac{1}{|X|-C_M^2}, \\ e^{-\frac{k-1}{|\mathcal{C}^*|}} &< \frac{1}{|X|-C_M^2}, \quad \left(\because (1-x)^{\frac{1}{x}} \approx 1/e\right) \\ \frac{k-1}{|\mathcal{C}^*|} &< \ln(|X|-C_M^2), \\ k &< |\mathcal{C}^*|\ln(|X|-C_M^2)+1. \end{aligned} \tag{4.4}$$

Therefore, from the above analysis, we can see that the size of the set cover found by the greedy algorithm, $|\mathcal{C}|$, is bounded by $|\mathcal{C}^*|\ln(|X|-C_M^2)+1$. ■

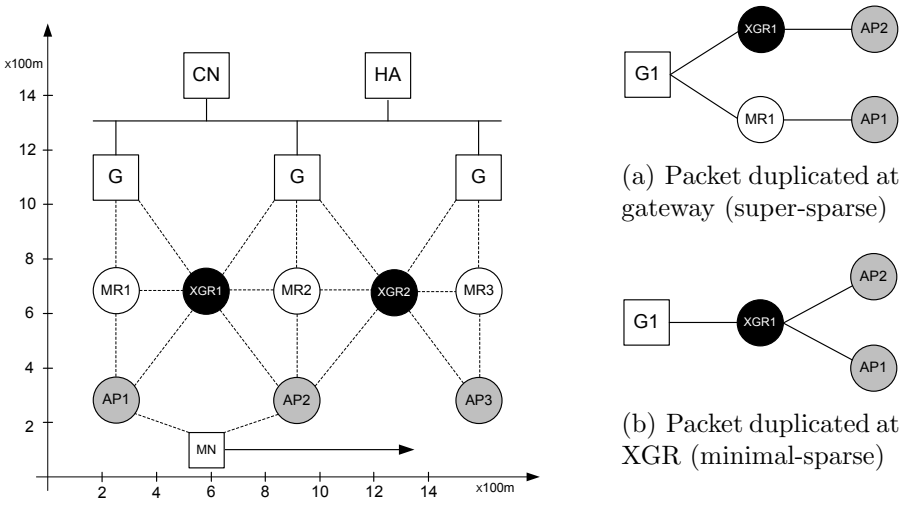
The size of the obtained set cover is the required number of *XGRs* to form a IMeX backbone. However, this number does not consider the size of the geographic area that needs to be covered by the WMN. If more MRs are needed to cover a large area, more regular MRs can be added to join the closeby group, but all the *XGRs* have to configure their IP addresses based on the outcome from the greedy algorithm to form N connected groups.

4.3 Performance Evaluation

To demonstrate the advantages of the proposed IMeX architecture, we conduct OPNET [13] simulations to evaluate the performance of the proposed Xcast-based data caching mechanism. Fig. 4.6 shows a 3-hop (MN to gateway) simulation scenario with three subnets. Based on the implementation of cross-layer design in the [13] as explained in last Chapter, we study the proposed Xcast-based data caching mechanism performance in an independent scenario as shown in Fig. 4.6(a)(b).

4.3.1 Simulation Setup

For the simplicity of simulation, the gateway has equal number of hops to all cAPs. Several performance metrics are defined for comparisons: control overhead is



(a) Packet duplicated at gateway (super-sparse)

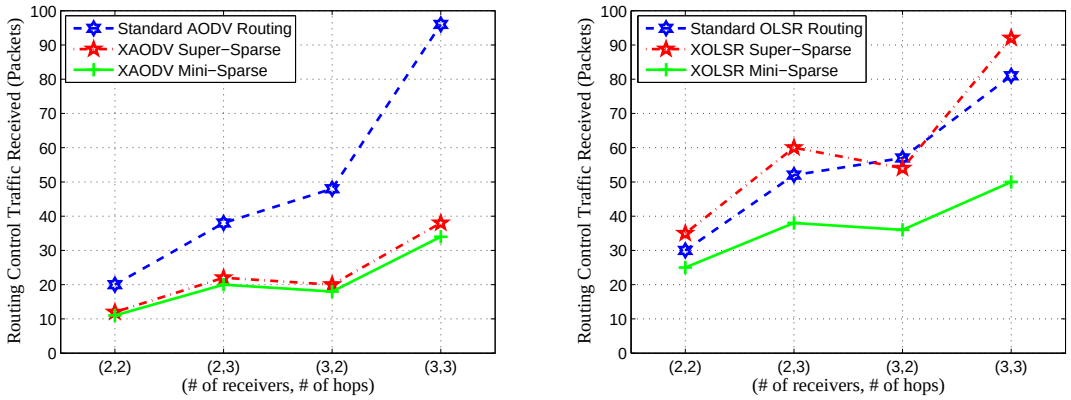
(b) Packet duplicated at XGR (minimal-sparse)

Figure 4.6: An Inter-gateway 3-hop handoff and data caching simulation scenario in OPNET.

defined as the number of routing control packets generated in the network per data packet transmitted; bandwidth consumption is defined as the number of data packets transmitted per data packet delivered; and the ETE delay is the average ETE delay of the first data packet arrival on cAPs.

4.3.2 Simulation Results

4.3.2.1 Performance of Xcast Data Caching



(a) (b)

Figure 4.7: Routing control overhead. (a) - XAODV; (b) - XOLSR

Fig. 4.7 shows the comparison of routing control overhead between the unicast

routing and Xcast-based routing (super-sparse and minimal-sparse cases). The x axis shows two sets of data corresponding to the number of handoff cAPs and the number of hops from the gateway (sender) to cAPs (receivers). For example, (2, 3) means the path length of three hops from the gateway to two cAPs. In Fig. 4.7(a) the XAODV case, the two Xcast schemes have lower control overhead than that of the unicast scheme, while the Xcast minimal-sparse case induces the lowest control overhead. In Fig. 4.7(b) the XOLSR case, the *XGR*-preferred MPR selection algorithm would possibly cause a larger number of forwarding nodes and generate more control messages. However, the small difference between the super-sparse and the unicast case is a result of the randomness in the discrete event simulation. On the other hand, the minimal-sparse case can keep the control overhead at a lower level.

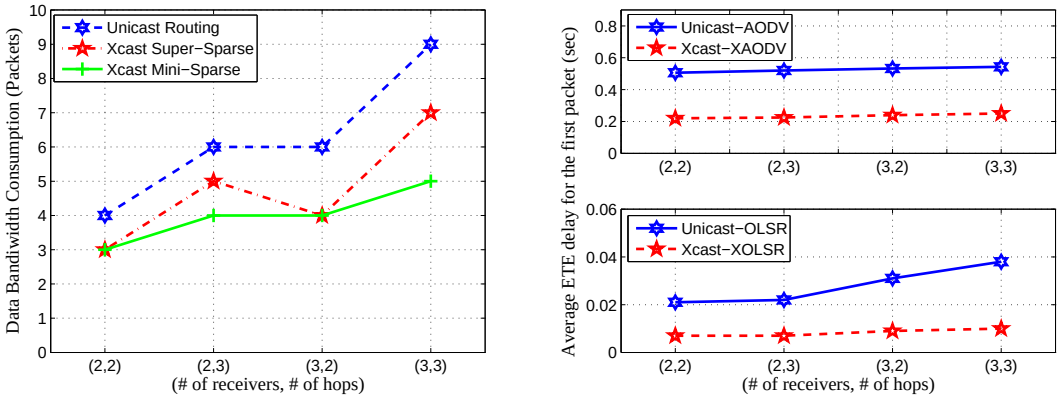
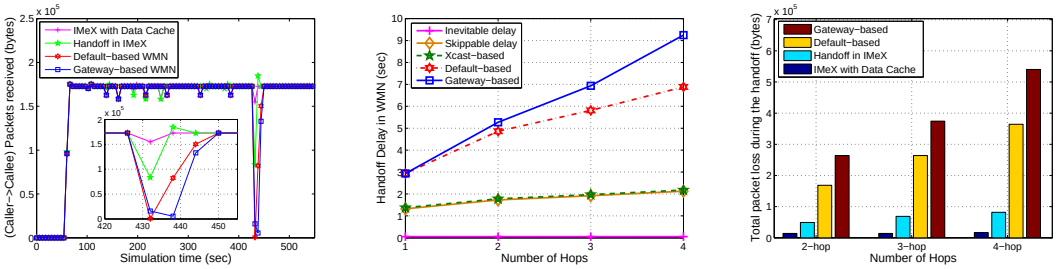


Figure 4.8: Bandwidth consumption and average ETE delay.

Fig. 4.8(a) shows the bandwidth consumption of data packet delivery using the proposed XAODV and XOLSR routing path tree setup. As expected, the two Xcast schemes have much lower bandwidth consumptions than the unicast one, while the minimal-sparse case retains the lowest bandwidth consumption. Fig. 4.8(b) shows that our Xcast-based data caching mechanism spends shorter time caching the first data packet to all cAPs. The reactive cases have much longer ETE delay than the

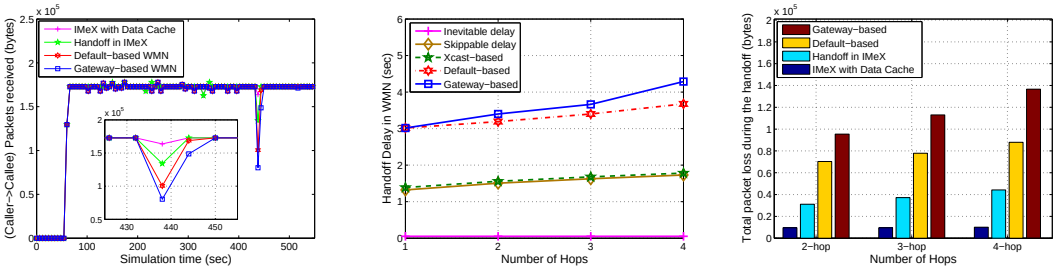
proactive ones due to the packet-initiated route discovery attribute. Hence, our Xcast-based schemes can support caching to the handoff cAPs faster so that the performance degradation during a handoff is minimized.

4.3.2.2 Packet Loss During Handoffs



(a) Video called party traffic received (b) Handoff delay (c) Packet loss during handoff

Figure 4.9: Handoff delay and packet loss (using AODV routing protocol).



(a) Video called party traffic received (b) Handoff delay (c) Packet loss during handoff

Figure 4.10: Handoff delay and packet loss (using OLSR routing protocol).

Fig. 4.9 and 4.10 illustrate the handoff features under the four considered handoff scenarios when an ETE video conferencing packet flow starts at 60 second. Fig. 4.9(a) illustrates the instantaneous ETE traffic flow under a three-hop handoff scenario with a zoomed handoff period, when AODV routing protocol is used. Usually, nearly all the packets destined to the MN are dropped during a handoff. However, the video conferencing application has much better performance in terms of ETE packet delivery in our IMeX architecture, as compared to the other two schemes. Lower packet loss

during a handoff is shown in our IMeX architecture with the Xcast-based data caching mechanism. Fig. 4.9(b) illustrates the total handoff delay (L2, L3, L5) versus the number of hops (gateway to MN) in different architectures. We also plot the inevitable delay and skippable delay during an inter-gateway handoff. By using the data caching scheme, handoff delay of a multilayer inter-gateway handoff can be reduced to that of an L2 handoff. Fig. 4.9(c) shows the packet loss during the handoff period. The figure shows that our IMeX architecture with the data caching mechanism experiences the least packet loss during a handoff. For comparison, Fig. 4.10 shows the simulation results using the OLSR routing protocol. The handoff delay and packet loss of the two default cases are reduced as compared to the AODV case due to the proactive attribute. Our IMeX architecture with the data caching mechanism still has the best performance.

4.3.2.3 End-to-end Delay and Delay Jitter

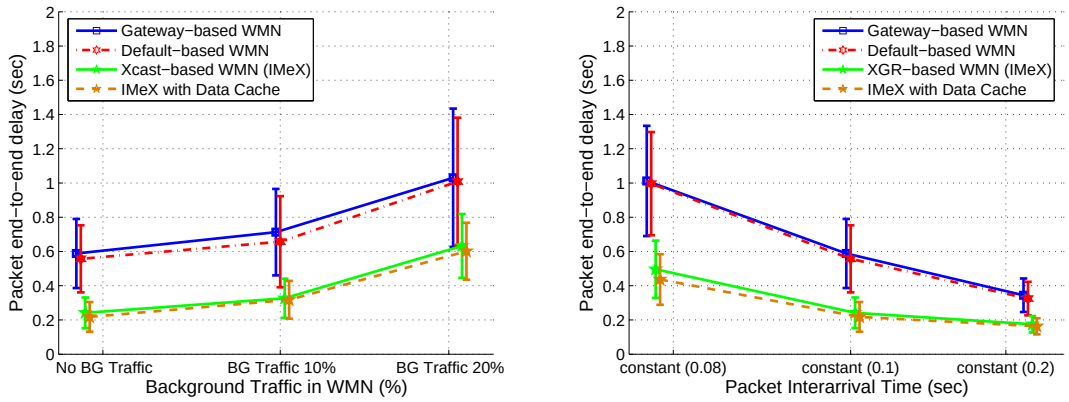


Figure 4.11: ETE packet delivery delay with deviation (using AODV).

In Fig. 4.11 and 4.12, we compare the application response time under different handoff schemes. Fig. 4.11(a) and 4.12(a) present the end-to-end packet delivery delay with deviation for video conferencing applications under the four considered handoff scenarios, with different percentage of background traffic ranging from 0 to

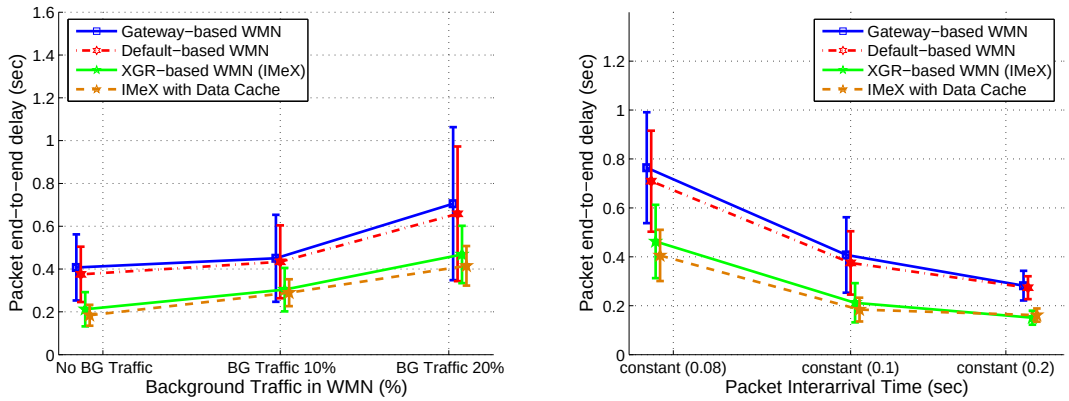


Figure 4.12: ETE packet delivery delay with deviation (using OLSR).

20% between MRs. The figures show that both the end-to-end delay and delay jitter increase as the percentage of the background traffic increases. However, our proposed IMeX handoff schemes outperform the other two schemes in terms of reducing the end-to-end delay and delay jitter, because it can reduce the handoff delay and resume the video session quickly after a handoff.

From Fig. 4.11(b) and 4.12(b), we can see that when the video conferencing packet interarrival time shortens from 0.2 second to 0.08 second, the average end-to-end packet delay and delay jitter increase. Under both default-based and gateway-based handoff schemes, the end-to-end delay and delay jitter are higher than those under the IMeX architecture.

4.3.2.4 Performance based on Various Queuing Schemes

Fig. 3.7(b), 3.9(b), 4.11, and 4.12 show that all the handoff schemes (default-based, gateway-based, and IMeX cross-layer) suffer when the background traffic increases and MRs become more congested. We investigate whether employing a priority queuing scheme at MRs can accelerate signaling and improve handoff performance. We consider two queuing schemes: (1) a non-priority first-in-first-out (FIFO) queuing discipline and (2) a priority queuing discipline with a higher priority given to

handoff-related control packets.

We change the video conferencing application to a very light video application (frame size: 172 bytes, interarrival time: 0.5 second) as a source of constant UDP traffic to shorten the simulation time with the same effect. Fig. 4.13 presents the total handoff delay based on different queuing schemes when the background traffic increases, under a 2-hop handoff scenario. The results are the average of 20 simulation trials with varying seeds. The figure shows that when handoff signaling is prioritized, the total handoff delay is reduced under each handoff scheme. However, the IMeX cross-layer handoff scheme can reduce the total handoff delay more than a priority queuing scheme.

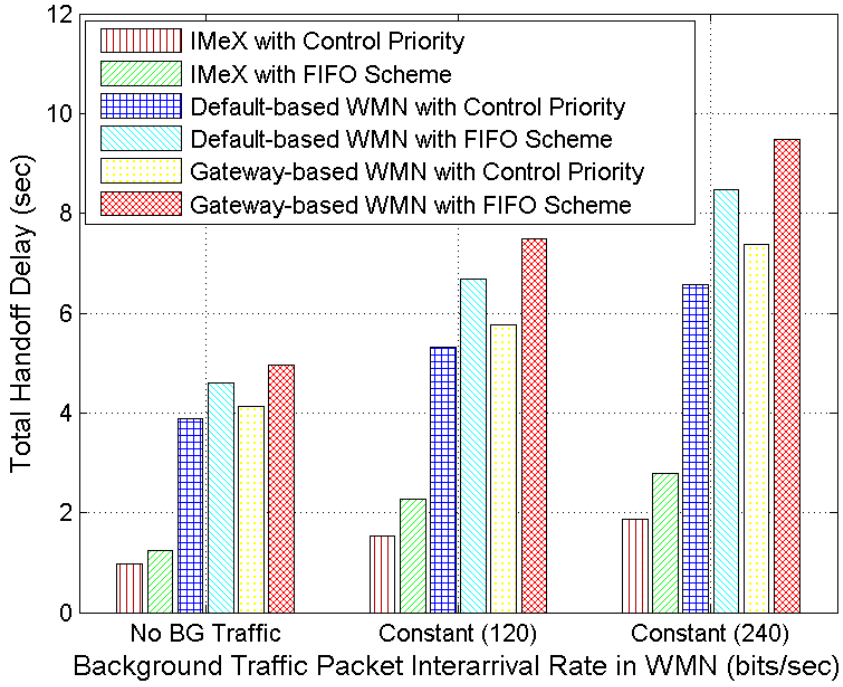
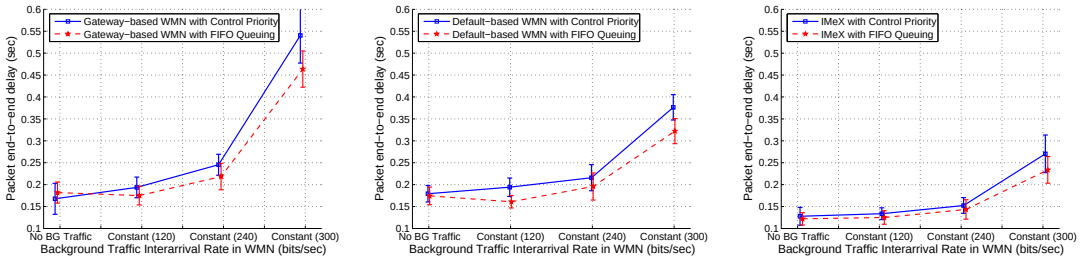


Figure 4.13: Total handoff delay based on various queuing schemes.

Fig. 4.14 illustrates the end-to-end delay with deviation for video packets based on 20 simulation trials and a confidence level 90%. Fig. 4.13 and 4.14 demonstrate a tradeoff when priority queuing is used: priority queuing can reduce the total handoff delay, but at the cost of higher end-to-end data packet delivery delay, because data

packets need to wait longer time when handoff control packets are given a higher priority. It can be seen from both figures that our proposed IMeX cross-layer handoff scheme has the lowest handoff delay and end-to-end packet delay, as compared to the other two conventional schemes.



(a) End-to-end delay in gateway-WMN (b) End-to-end delay in default-WMN (c) End-to-end delay in IMeX

Figure 4.14: Video conferencing packet end-to-end delay based on different queuing schemes.

4.3.3 Summary

From the performance results, we may see that under the default- and gateway-based handoff schemes, the total handoff delay is not acceptable for real-time multimedia applications, especially when the number of wireless hops connecting an MN to the gateway increases or the offered load at MRs is high. Under the proposed IMeX cross-layer handoff scheme with data caching, the L3 address acquisition delay is eliminated and the handoff delays caused from different layers are shortened. The proposed handoff scheme can reduce the total handoff delay to around 1 second, when a proactive routing protocol is used. However, this delay may still not be good enough for real-time multimedia applications. Further reduction of the handoff delay can be achieved through efficient multihop routing and MAC protocol design to reduce the wireless multihop signaling traffic delivery delay.

4.4 Conclusion

In chapter 3 and 4, we introduced a novel explicit multicast-based architectural design with planned group strategy to address the special L3 handoff detection chal-

lenge and facilitate cross-layer handoffs in Internet-based WMNs. By implementing Xcast MRs (XGRs) which are strategically configured in the mesh backbone to cover target subnets, inter-gateway handoff preparations can be proactively prepared before an MN loses its connection with the old subnet. In addition, data packets can be cached in cAPs across subnets for the MN to ensure minimum packet loss. The detailed procedure of the proposed cross-layer handoff scheme and the Xcast-based data caching mechanism are described. The problem of finding the required minimum number of XGRs is modeled as a set covering problem and a greedy algorithm is proposed to obtain the required number and the optimal placement of XGRs. Through a comprehensive simulation study, the proposed IMeX architecture is demonstrated to offer an interworking paradigm across subnets to assure the session continuity with significantly reduced handoff delay and packet loss during handoffs.

CHAPTER 5: INTER-GATEWAY QOS HANDOFFS IN INFRASTRUCTURE WMNS

The goal of a seamless QoS handoff is to maintain the QoS of the handoff service before and after the handoff, and at the same time, preserve the global QoS stability by maximally utilizing the total resources in the network. As shown in Fig. 5.1, the bottleneck of realizing a seamless inter-gateway QoS handoff in WMNs lies mainly in two aspects: 1) the number of gateways connected to the Internet. Given that the number of gateways in a WMN is fixed in the deployment phase, inter-domain handoff traffic is routed to only pass through the gateways that reside in the new domain after a handoff. Such architecture limitations will cause QoS degradations to either delay-sensitive handoff traffic or existing services if the gateways in the new domain after a handoff are saturated. In this chapter, the QoS gateway selection issue is addressed, which is to answer how an MN decides to (or not to) associate to a gateway for the new Internet access when an inter-domain handoff happens; 2) the QoS conditions of the intermediate forwarding mesh routers. In an inter-gateway handoff, mesh routers perform multihop routing by selecting one or more optimized routes and each mesh router selects a neighbor router to forward handoff traffic in each hop. This strategy does not always adapt well in dynamic handoff scenarios, let alone the extra handoff signaling traffic added to the existing traffic. The QoS traffic forwarding issue during a handoff is addressed, which is to answer how an intermediate mesh router decides to (or not to) forward data to a neighbor mesh router during a handoff when the global QoS conditions change.

To achieve an inter-gateway QoS handoff in WMNs while preserving the global QoS stability requires the co-design of network architecture and multi-hop traffic forwarding to address the two bottlenecks shown in Fig. 5.1 together. The key

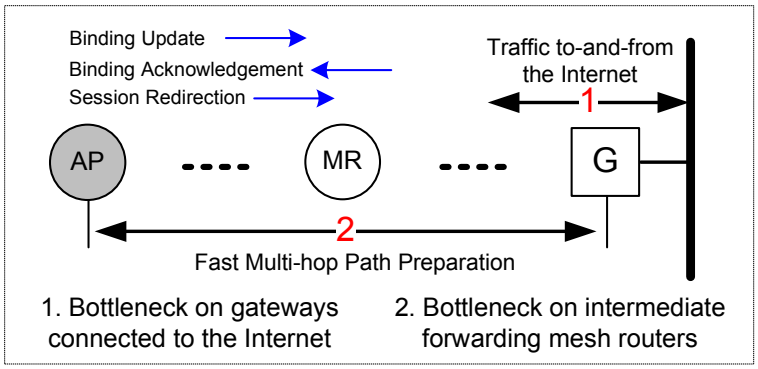


Figure 5.1: The key design aspects of multi-hop inter-gateway handoffs in WMNs.

contributions of this Chapter can be summarized as follows:

- A discussion of the interdependent design between network engineering (NE) and traffic forwarding (TF) for the inter-gateway QoS handoff scenario and the necessity of the co-design methodology for maximally utilizing global QoS resources.
- A new architectural design that offers dynamic gateway selection for inter-gateway handoffs in WMNs. The proposed design can utilize gateway resources across different domains for Internet access. A discussion and design of different L3 handoffs when choosing different gateways during handoffs are provided.
- A resilient forwarding scheme that seamlessly interacts with the IPv6 protocol to provide *neighbor QoS detection* and make resilient next-hop decisions for traffic forwarding during an inter-gateway handoff.

To the best of our knowledge, this is the first work that applies the co-design methodology of NE and TF for QoS handoffs in the inter-gateway environment that maximizes the global resource utilization.

5.1 Explore QoS Handoffs in WMNs

In this section, two different perspectives of the handoff design are examined and the co-design methodology for realizing inter-gateway QoS handoffs in WMNs is explored.

5.1.1 Network Engineering (NE) Versus Traffic Forwarding (TF)

Firstly, the inter-dependency between Network Engineering (NE) and Traffic Forwarding (TF) that are involved in the QoS handoff design is briefly discussed:

- NE is responsible for allocating bandwidth to support traffic. In an inter-gateway handoff scenario, the mesh topology design has a great impact on the accessibility of Internet connections. Hence, it determines the number of available gateways that an MN can access.
- TF is responsible for placing traffic where there is bandwidth. Integrated Services (IntServ)[68] and Differentiated Services (DiffServ)[69] are two proposals that provide fundamental QoS traffic management in the Internet.

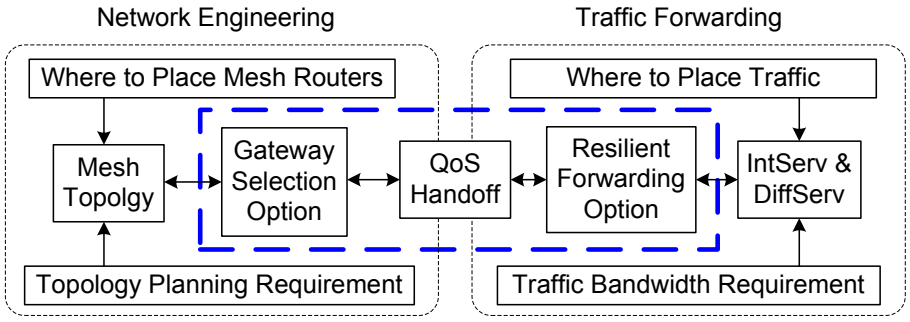


Figure 5.2: Interaction between NE & TF for inter-gateway QoS handoffs.

In Internet-based WMNs, mesh routers are usually static and the IP address of each mesh router is pre-configured during the deployment phase. If that different gateways belong to different subnets is assumed, the gateway through which an MN can access is confined only to its own subnet. Therefore, in our design, a gateway selection option is added into NE for supporting the utilization of gateways across subnets during an inter-gateway QoS handoff. In addition to the two conventional QoS traffic management methods, a resilient forwarding option is also added into TF for supporting resilient global QoS maintenance in a multi-hop routing environment. The inter-dependency between NE and TE in our design is shown in Fig. 5.2. By enabling resilient gateway selection and resilient traffic forwarding, inter-gateway QoS

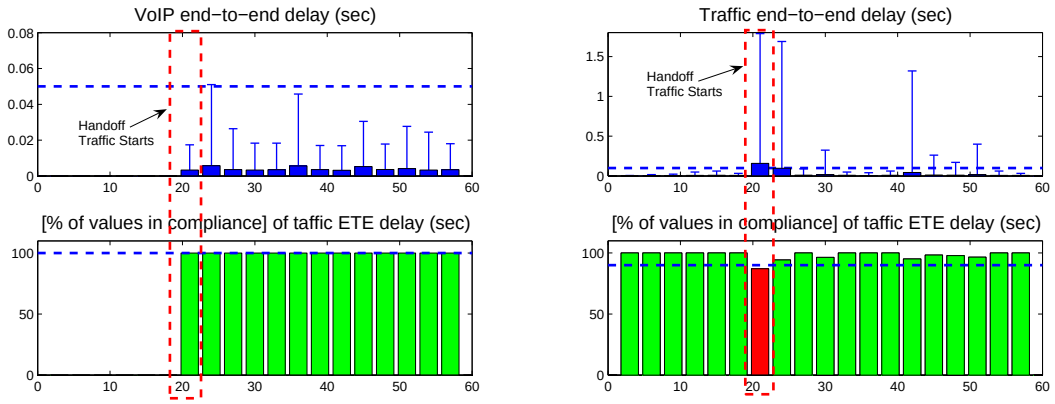
handoffs can be realized in WMNs. Therefore, the integration of architectural and resilient routing designs can facilitate inter-gateway QoS handoffs and maintain the global QoS stability by maximally utilizing resources across domains.

5.1.2 Tradeoffs and Limitations under a Separated Design

In order to depict the limitations of a separated design (e.g., NE only with a static mesh topology, TF only with conventional QoS traffic management) in inter-gateway QoS handoff scenarios, OPNET [13] simulations are conducted to see if 1) a subnet can sustain the new handoff traffic (e.g., VoIP applications) under a fixed mesh topology, and 2) the QoS conditions of both handoff and existing traffic can be maintained the same before and after a handoff.

Table 5.1: Averaged Traffic Sent & Received at Different Time

	Baseline traffic (bits)	**VoIP** (bits)
Sent **Before** Handoff (20s)	66538.46	N / A
Received **Before** Handoff (20s)	63571.42 (-4.4%)	N / A
Sent **After** Handoff (60s)	110300	8259.2
Received **After** Handoff (60s)	95900 (-13%)	8259.2



(a) VoIP Traffic

(b) Baseline Traffic

Figure 5.3: QoS handoff tradeoffs.

Fig. 5.3(a) shows that with a good handoff scheme, the QoS of the handoff

traffic can be guaranteed in terms of the service level agreement (SLA) as defined. However, the addition of the handoff traffic to the subnet can seriously affect the performance of existing baseline traffic since the performance of some traffic flows does not 100% conform to the SLA requirement, as shown in Fig. 5.3(b). From Table 5.1, the total throughput of baseline traffic can be seen drop 13% after adding the handoff traffic to the subnet. Furthermore, when the gateway in this subnet is saturated, the performance degradation can be even more. In conclusion, given a fixed mesh topology, even a good QoS routing protocol can only exhaust current available resources without considering other resources from a different subnet. In addition, sometimes a QoS handoff is realized by disrupting other existing services. Therefore, in an inter-gateway QoS handoff, a separated design on the routing protocol with a static mesh topology has the limitations of finding the best path across domains.

5.1.3 Summary

The following issues existing the current inter-gateway QoS handoff design in WMNs can be concluded as follows:

- Separated design of either NE or TE brings tradeoffs to one another in an inter-gateway handoff environment (e.g., mesh topology constraints bring routing limitations when inter-gateway handoffs occur).
- In Internet-based WMNs, during an inter-gateway handoff, a static mesh topology only allows an MN to access the Internet via the gateway in the new subnet. An awkward situation occurs when the MN needs to access a light-loaded gateway that is nearby but belongs to a different subnet.
- In a multihop WMN environment, routing may not well adapt to the rapidly changing QoS conditions on intermediate forwarding routers. Hence, QoS degradations may happen, especially in an inter-gateway handoff. In addition, the fact that QoS routing puts certain higher priority traffic on one path can introduce load oscillations and high instability in the performance seen by lower-

priority traffic (e.g., best-effort traffic). For example, when one path becomes overloaded, the traffic will be diverted onto another path. Thus, the lower-priority traffic flow on those two paths will observe high instability in the performance which is undesirable.

5.2 Proposed Inter-gateway QoS Handoffs in WMNs

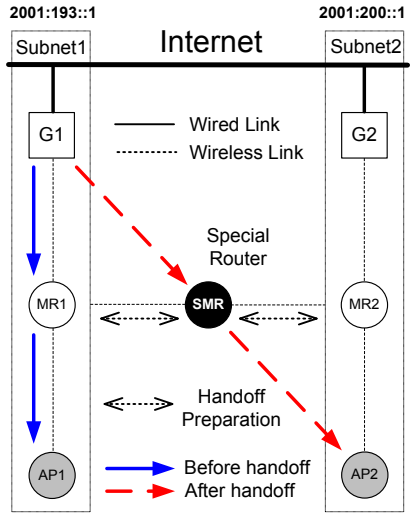
In this section, an integrated design for inter-gateway QoS handoffs in WMNs with a resilient gateway selection and traffic forwarding scheme is proposed.

5.2.1 Assumptions

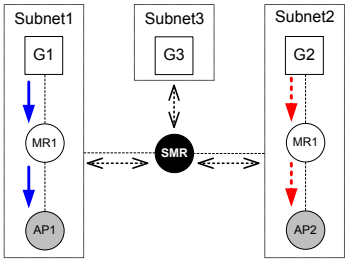
- The IPv6 stateless address autoconfiguration (construction of link-local addresses, duplicate address detection, construction of unique global addresses) and the corresponding Neighbor Discovery Protocol (NDP)[7] are adopted for the address configuration in WMNs.
- Priority queuing is applied to all mesh routers in WMNs. All mesh routers have two queues: the first queue is a priority queue solely used for real-time traffic, while the second queue based on weighed fair queuing (WFQ) allocates bandwidth to other non-real-time packets (data and control packets).
- AODV [59] and OLSR [61] are considered in our WMN architecture as the mesh routing protocol for the reactive and proactive routing scenario, respectively.

5.2.2 A Resilient Architecture for Inter-gateway QoS Handoffs

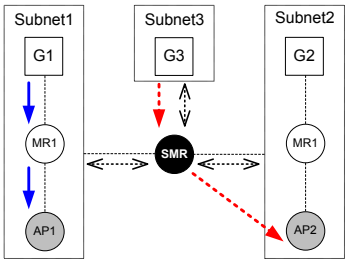
A novel mesh architectural design to facilitate cross-layer handoff preparations is proposed in [70]. In the design, handoffs to a new domain can be proactively prepared with the help of a set of special mesh routers (SMR). Based on the design in [70], that different gateways belong to different subnets is assumed. A SMR with multiple IP addresses belonging to different subnets has the capability of routing traffic to gateways which reside in different subnets. As shown in Fig. 5.4, solid arrows stand for the old routing path before a handoff and dashed arrows stand for the new routing path after a handoff. Double-headed arrows indicate the handoff preparations with



(a) Inter-gateway handoffs via the old gateway



(b) Access the gateway in the handoff subnet



(c) Access the gateway in another subnet

Figure 5.4: Gateway selection for inter-gateway QoS handoffs.

the help of the selected SMR. In order to pick up an appropriate gateway among the available ones that a selected SMR can access, a gateway selection algorithm for maximally utilizing gateway resources across domains during inter-gateway handoffs is proposed. Common notations used throughout the algorithms are listed in Table 5.2.

Table 5.2: NOTATIONS USED FOR ALGORITHMS

Notations	Definition
g_i	A gateway with index i
q_i	Maximum achievable QoS value of gateway g_i
δ	The highest value of maximally achievable QoS among gateways
G	Set representing a list of available gateways a SMR can access
ζ	Set containing measured QoS values of the gateways
Φ	Set containing measured RTT values indicating the queue length of neighbor routers

The gateway selection algorithm, as shown in Algorithm 2, provides a procedure

to find a set of gateways that an MN can access with the help of a SMR for the inter-gateway handoff preparations. In this algorithm, the SMR with multiple IP addresses belonging to different subnets sends an ICMPv6 [71] ping message to all gateways contained in a basic set G. The shorter the round trip time (RTT) observed from a gateway g_i, the smaller queue the gateway has for processing real-time traffic. The QoS value is defined as inversely proportional to the detected queue length of a gateway. The SMR compares the q_i obtained from each gateway g_i with a variable δ representing the highest value among what have been obtained for maximally achievable QoS value. From the comparisons, the SMR can help the MN to decide whether it needs to change a gateway and which gateway is the best option for Internet access when inter-gateway handoffs occur.

Algorithm 2: Gateway Selection Algorithm

```
1  Let i ← 0, initialize δ: min. QoS value for a handoff;
2  for g_i ∈ G do
3  |  Perform QoS inquiry to find q_i of gateway g_i;
4  |  if δ < q_i then
5  |  |  let ζ ← ζ ∪ {q_i}, δ ← q_i;
6  if ζ = ∅ then
7  |  return false and inter-gateway handoff rejected;
8  else if g_{M ∈ argmax_i {∀q_i ∈ ζ}} belongs to the old subnet then
9  |  return g_M        /* choose the gateway from the old subnet */;
10 else                          /* choose one from a new subnet instead */
11 |  M is any index from the set argmax_i {∀q_i ∈ ζ};
12 |  return g_M;
```

5.2.3 A Resilient Traffic Forwarding Scheme

"*There's a difference between knowing the path and walking the path.*" This quotation from the film "The Matrix" well explains the two stages that compose an intermediate router's routing-and-forwarding behavior: the routing protocol and the forwarding scheme. The routing part answers what each intermediate router along a chosen path is supposed to do, while the forwarding decides what each interme-

diate router really behaves. In other words, forwarding sometimes will not depend on routing in the sense that it needs to cope with real-time conditions. If the traffic forwarding direction needs to change to avoid a congested link, sometimes there is no need to invoke routing at intermediate routers to re-calculate the path and update the routing table. Using a resilient traffic forwarding scheme, a local signaling can replace the global signaling required by routing to reduce considerable overhead. In a word, a better traffic forwarding scheme will not replace but better support the routing protocol to form a resilient routing path based on QoS requirements.

Algorithm 3: Intermediate Mesh Router Selection Algorithm

```
1  Let RTT ← 0, i ← 0;
2  Send Neighbor Solicitation message to neighbor routers ;
3                /* Perform the NQD for the neighbor routers */;
4  for received Router Advertisement from neighbor routers do
5  |  Calculate RTT_i;
6  |  Φ ← Φ ∪ {RTT_i};
7             /* Obtains the best next-hop intermediate router */;
8  if Φ = ∅ then
9  |  return false and perform the regular forwarding;
10 else
11 |  Let M ← argmin_i { ∀ RTT_i∈Φ };
12 |  return RTT_M;
```

Moreover, most popular routing protocols achieve the up-to-date QoS conditions at the cost of extra signaling overhead. Little consideration has been given to utilizing the existing control messages. Our proposed resilient forwarding scheme provides a seamless interaction with the existing IPv6 NDP [7]. The NDP contains a combination of router discovery, router advertisement, and neighbor unreachability detection messages, etc. The number of packets in queues in a neighbor router can be estimated by measuring the difference between the observed RTT and the base RTT defined as the round trip time when there is no queuing. Based on the NDP, using a dynamic timer on the interface, a mesh router multicasts a neighbor solicitation message (a packet which is enqueued in the data queue) to all neighbor routers and

records the RTT. As soon as finishing processing the priority queue, neighbor routers reply the neighbor advertisement message. Upon receiving the messages, the sender can calculate the RTT to estimate the queue length of each neighbor. Additionally, heavy loaded neighbors can multicast an NDP unsolicit message to inform neighbors about their full load of queues. The multicast neighbor receivers will then update the status of that neighbor router. Packets can be redirected to an idle mesh router for fast queue processing. By doing so, mesh routers are aware of the neighbors that are available for fast packet forwarding and can direct traffic to an alternative path when the primary path fails to conform with the QoS requirement. For the inter-gateway QoS handoff purpose, our proposed *neighbor QoS detection* (NQD) mechanism is combined with the duplication address detection (DAD) procedure after an MN obtains a Care-of-address (CoA). The procedures to obtain the NQD from a set of intermediate nodes is shown in Algorithm 3. By seamlessly interacting with the NDP, our proposed traffic forwarding scheme can work with existing routing protocols, especially the reactive ones, to provide up-to-date QoS routing paths while generating no new control messages.

5.2.4 Handoff Scenarios Involved in Inter-gateway Roaming

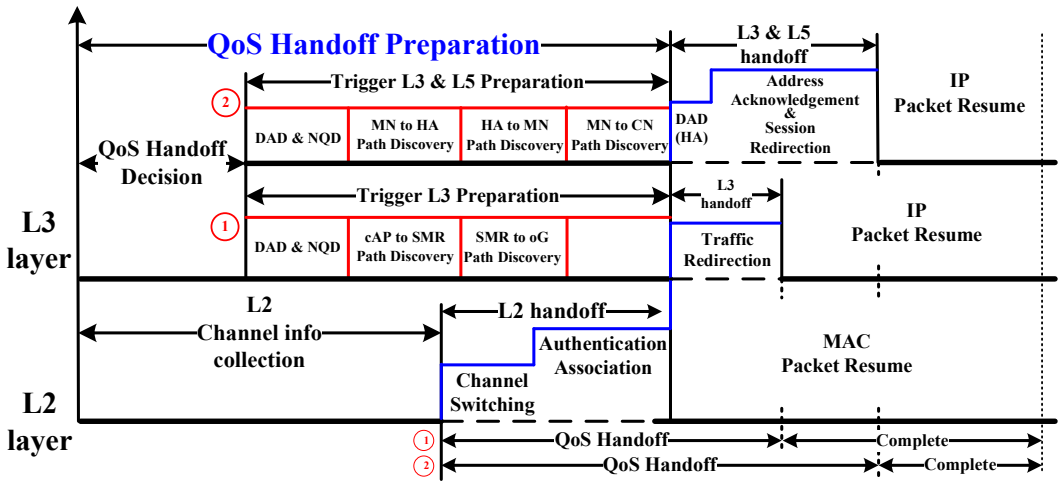


Figure 5.5: Cross-layer QoS-handoff procedures.

Based on our proposed gateway selection algorithm, inter-gateway QoS handoffs

can happen in two scenarios:

5.2.4.1 A light L3 handoff

In this case, the MN does not change the gateway when it is handed off to a new subnet, as shown in Fig. 6.5(a). Similar to [70], a selected SMR in the architecture helps the inter-gateway handoff preparations for the MN. After forming a new CoA for the MN, the SMR continues the DAD and NQD steps to retrieve the QoS information of the intermediate routers. Based on the NQD acquired, the SMR prepares a routing path between the candidate AP (cAP) and the old gateway for the MN. Note that there is no need to send a *Binding Update* to the home agent (HA) and correspondent node (CN) because the MN still accesses the old subnet via the old gateway. The old gateway can redirect traffic to the cAP in the new subnet via the selected SMR. Fig. 6.5(a) and Fig. 5.5 show the light L3 handoff architecture and detailed handoff steps on each layer, respectively.

5.2.4.2 A full L3 and L5 handoff

The MN changes to a new gateway after it is handed off to a new subnet, as shown in Fig. 6.5(b)(c). Similar to the first case, the SMR performs the DAD and NQD steps to form a set of intermediate routers for assisting the following routing path preparations. A L2 channel switch triggers the inter-gateway handoff which is completed after the CN redirects the traffic to the new gateway in a new subnet. The full inter-gateway handoff procedure is shown in Fig. 5.5 and in Algorithm 4.

5.3 Performance Evaluation

In this section, the performance of our proposed inter-gateway QoS handoff design using OPNET[13] is evaluated.

5.3.1 Simulation Scenarios and Setup

Based on the WMN architecture proposed in [70], the gateway selection algorithm and traffic forwarding mechanism are further implemented in order to support the inter-gateway QoS handoffs. The simulation topologies of WMNs are shown in Fig.

Algorithm 4: L3 & L5 Handoffs for Different QoS Decisions

```
1  if the selected g_M belongs to the old subnet then
2      /* Perform the light L3 handoff preparations for the MN */;
3      Perform the DAD & NQD ;
4      Routing path preparation between the cAP and old gateway via the SMR;
5      Old gateway redirects the traffic to the cAP via the SMR;
6  else
7      if the selected g_M belongs to a new subnet then
8          Perform the DAD & NQD;
9          Routing path preparation from the cAP to new gateway, HA, and CN via
           the SMR;
10         New gateway redirects the traffic to the cAP;
```

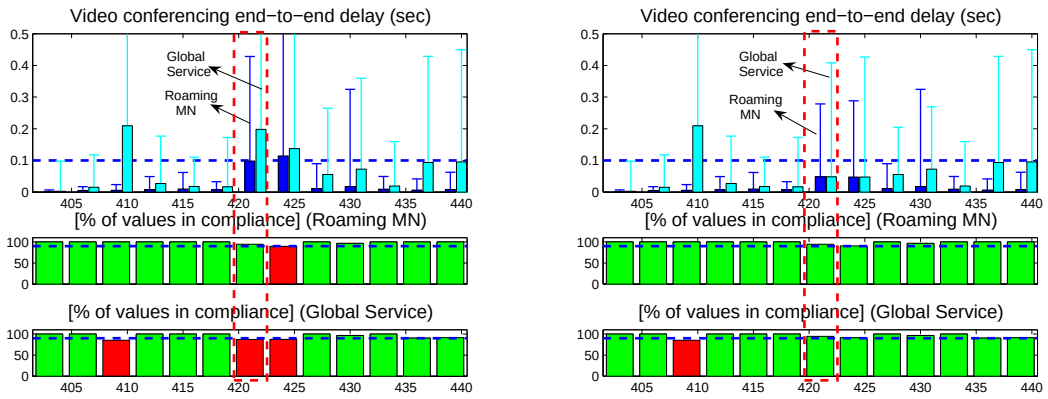
5.4. Only SMRs with multiple IP addresses can route traffic across subnets. Two flows of video conferencing are added to each subnet for simulating global QoS-aware services. One roaming MN moves at constant speed across subnets. The service level agreement (SLA) is used as the criterion for evaluating the performance conformance for different types of traffic. Under our definition of the SLA, the performance is in compliance with the SLA if the end-to-end (ETE) delay of video conferencing is below 0.1 second 95 percent of each 3 seconds (in simulation time). This means that an SLA violation will be shown if the ETE delay is above 0.1 second more than five percent of each 3 seconds. A simulation list of parameters for network and traffic attributes are shown in Table 5.3.

5.3.2 Results Analysis

Fig. 5.6 shows the ETE delay with SLA conformance for both the inter-gateway handoff traffic and global QoS-aware services. A green bar indicates conformance to the defined SLA and a red bar shows a violation. Fig. 5.6(a) shows the default inter-gateway handoff scenario in which the roaming MN is handed off to a new gateway which happens to be heavily loaded. Observe that during the handoff period around 420 seconds, the default inter-gateway handoff causes performance degradation to other existing QoS-aware services as the gateway in the new subnet is excessively utilized. Consequently, some flows of the global traffic do not conform to the SLA

Table 5.3: Simulation Parameters

Parameters for Network Attributes		
Mesh Router Settings	AP transmit power (W)	0.05
	Buffer size (bits)	256000
	Packet reception power (dbm)	-95
	AP beacon interval (sec)	0.02
Control Messages	AODV *HELLO* message (sec)	uniform (1, 1.1)
	AODV active route timeout (sec)	3
	OLSR *HELLO* message interval (sec)	2
	OLSR *TC* message interval (sec)	5
	NDPv6 messages interval (sec)	uniform (0.5, 1)
Parameters for Traffic Attributes		
Video Conferencing	Start time (sec)	60
	Frame size (bytes)	172
	Frame interarrival time (sec)	constant (0.5)
SLA of ETE Delay	Value below 0.1 sec	95%
	Bucket duration (sec)	3.0
Delay Variation	Bucket duration (sec)	15.0
	Values per statistic	500



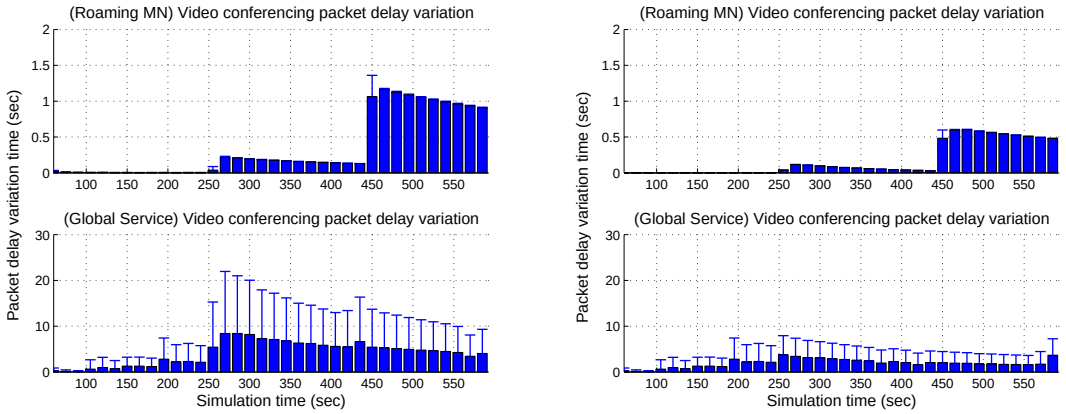
(a) Default inter-gateway handoff

(b) The inter-gateway QoS handoff

Figure 5.6: Performance comparisons of video conferencing packet ETE delay.

requirement, as seen in Fig. 5.6(a). In contrast, using the proposed dynamic gateway selection algorithm, the roaming MN still chooses the old gateway for Internet access after it moves to a new subnet since the selected SMR detects poor QoS conditions of the gateway in the new subnet. Hence, a full L3 and L5 handoff can be reduced

to a light L3 handoff. The inter-gateway handoff is completed without affecting other existing services since the old gateway is used for delivering the handoff service. The corresponding global QoS stability of all services is preserved and the handoff performance of the roaming MN is satisfactory, as seen in Fig. 5.6(b).



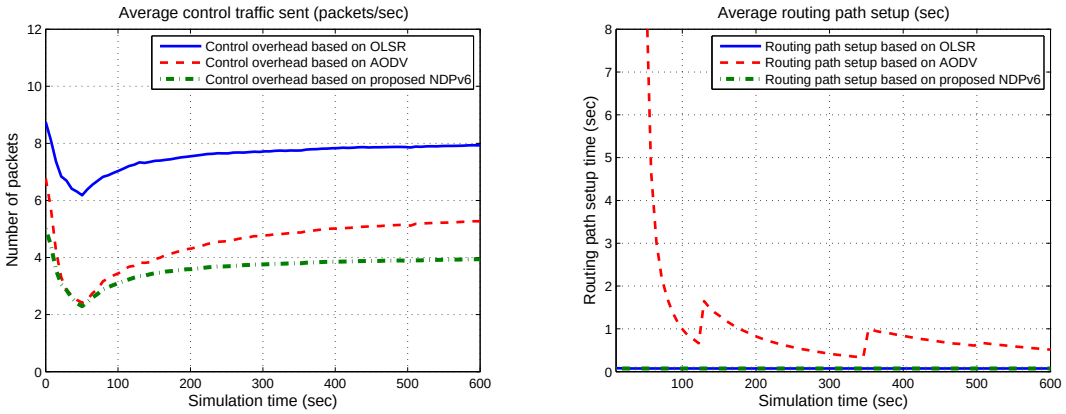
(a) Default inter-gateway handoff

(b) The inter-gateway QoS handoff

Figure 5.7: Comparisons of packet delay variation.

The corresponding packet delay variation (PDV) of the handoff and global QoS-aware traffic under two scenarios are shown in Fig. 5.7. The PDV in our proposed inter-gateway QoS handoff is much lower and the stability of global services is preserved before and after the inter-gateway handoff, as compared to those in the default inter-gateway handoff.

Fig. 5.8 shows comparisons of the control message overhead and the average path setup time between two regular routing protocols and our proposed scheme. In the proactive OLSR routing design, low path setup time is achieved at the price of constant control messages to maintain the multihop paths. On the other hand, the reactive AODV is inappropriate for inter-gateway handoffs as it incurs long-path discovery delay, which is detrimental for delay-sensitive applications. Since the proposed traffic forwarding scheme keeps exchanging the latest QoS information for helping routing path setup, it has the same low discovery delay as the OLSR.



(a) Control overhead (b) Average path setup time

Figure 5.8: Comparisons of control overhead & the average path setup time.

Therefore, by utilizing network resources gracefully, our proposed co-design methodology can guarantee the performance of inter-gateway handoffs while preserving the stability to global services. In addition, our proposed traffic forwarding scheme utilizing the NDPv6 can provide up-to-date QoS routing path as proactive protocols without generating new control messages.

5.4 Conclusion

In this chapter, a WMN QoS-handoff framework is introduced which includes 1) a resilient mesh architecture that offers dynamic gateway selection for inter-gateway handoffs and 2) a resilient forwarding scheme that allows intermediate mesh routers to make resilient next-hop decisions for traffic forwarding. With an integrated design and inter-dependency linkage of network architecture and traffic forwarding, inter-gateway QoS handoffs can be realized in Internet-based WMNs.

CHAPTER 6: A DYNAMIC LOCATION MANAGEMENT SOLUTION IN INTERNET-BASED WMNS

As one of the key designs in mobility management, location management 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 (an LU action), so that during the PD procedure, the network can locate the MN for the delivery of incoming packets.

Location management protocols proposed for mobile ad hoc networks (MANETs) [54, 72, 73] are generally not appropriate for Internet-based infrastructure WMNs (Ii-WMNs). These protocols are designed in consideration of the characteristics unique to MANETs, e.g., infrastructurelessness, energy constraints, node mobility, and dynamic topology. However, location management design in IiWMNs is different from previous proposals in MANETs because MRs in WMNs are usually static and unlike MANETs where traffic is inside a network, IiWMNs are primarily used for Internet-based applications [74].

In this chapter, a framework, *DoMaIN* is proposed, to provide location management for a large number of *silently* roaming MNs (sMNs) residing under an IiWMN. In the proposed *DoMaIN* framework, sufficient location information is provided by the network to each sMN before a proper LU action is triggered. In an IiWMN with multiple gateways, the proposed *DoMaIN* can help each sMN decide whether an intra- or inter-gateway LU action is needed and provide the *best* path for PD in an Internet session scenario. In addition, by minimizing LU overhead in the mesh backbone, the proposed *DoMaIN* provides a *scalable* location management design

targeting to support a large number of sMNs. Moreover, the proposed *DoMaIN* facilitates the implementation of hop-based LU which can further reduce the frequency of LU actions needed, thus preserves the power consumption on the sMN side. To be specific, the salient features of the *DoMaIN* framework are as follows:

1. The *DoMaIN* framework ensures the best location management performance in terms of data PD delay for each sMN under random mesh topologies with arbitrary MN movements.
2. With PD performance guarantees, the *DoMaIN* framework minimizes location management protocol overhead in terms of the LU overhead in the mesh backbone caused by each intra-gateway LU action and it is scalable to support location management for a large number of sMNs.
3. The *DoMaIN* framework facilitates the implementation of *dynamic hop-based* LU in the wireless mesh backbone, which is different from previous time-, movement-, and distance-based LU.
4. The *DoMaIN* framework considers the practicability and applicability issues by exploring the characteristics of an IiWMN and leveraging designs on the network side, thus minimizes changes on end users (sMNs). Hence, the proposed *DoMaIN* framework for IiWMNs is deployable.

To the best of the knowledge, *DoMaIN* is the first attempt to study dynamic WMN location management with the consideration of the special design challenges of WMNs and scales to support a large number of silently roaming MNs. The proposed location management is evaluated via comprehensive simulations and case studies. The performance of location management in terms of PD delay and LU overhead in the mesh backbone is substantially improved with the design.

6.1 Background and Motivations

Fig. 7.1(a) shows a typical IiWMN architecture for location management which includes the following entities: the home agent (HA) and correspondent node (CN)

located in the Internet; common MRs with access point (AP) functions; and MRs with gateway functions (G_1, G_2) connected to the Internet. In addition, MNs residing in the mesh backbone can be categorized into two groups: active MNs (aMNs) which currently have active end-to-end data sessions and explicit location information (i.e., the IP address of its associated MR); silently roaming MNs (sMNs) which currently do not have an active data session and only have implicit location information (i.e., the IP address of the last updated MR/gateway). In order to save battery consumption, an aMN with no active session for a while enters a power saving mode and becomes an sMN. On the contrary, an sMN becomes an aMN when initiating an active data session or when there are packets destined to this MN and it is paged by the network. If an sMN silently roams without performing any LU and relies only on the network to locate it when there are packets destined to it, the sMN battery consumption can be preserved but a large amount of paging traffic is generated since the sMN could reside under any MR. On the other hand, if an sMN performs LUs every time it visits a different MR, the network always knows the exact location of the sMN, but this is not a power-saving solution. Hence, there are two main criteria to evaluate the efficiency of a location management design. The first is packet forwarding delay induced by the paging procedure until the requested sMN is found. The second is the amount of power consumed on the sMN side by performing LUs.

As shown in Fig. 7.1(b), all existing dynamic location management schemes focus on addressing *how often* an LU action on the sMN side needs to be triggered so as to balance the tradeoff between the power consumption on the sMN side and the corresponding paging delay. *Where* an LU message should be sent to and *how* this decision affects the performance of location management have not been properly addressed in the literature.

The MR through which an an sMN performs the latest full LU to the HA to update its location (i.e., the IP address of its associated MR) is named as uMR. When the

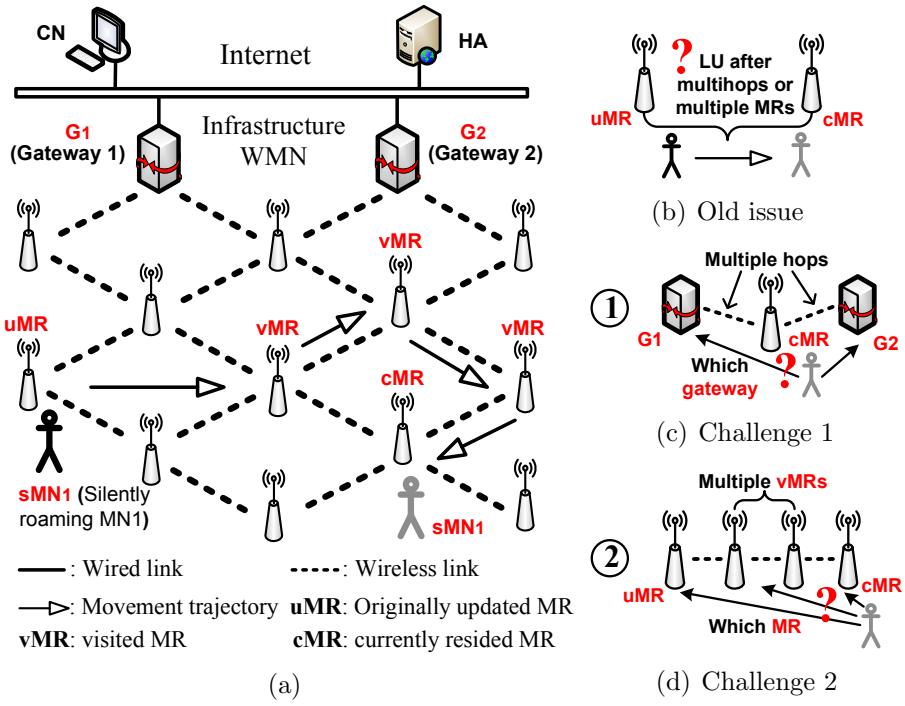


Figure 6.1: (a) The architecture of an Internet-based infrastructure WMN with silently roaming MNs residing under. (b)-(d) One old issue and two new challenging issues for location management.

sMN is residing under the current MR (cMR), the MRs the MN has passed during arbitrary movements are called visited MRs (vMRs) as shown in Fig. 7.1(a). During this movement trajectory, two new design challenges arise for location management:

- Challenge 1: assume that an sMN initially residing under its uMR chooses gateway G_1 for potential Internet data sessions. As this sMN silently roams, how can it be aware that it approaches a different gateway (e.g., G_2) that provides better location management performance in terms of a lower PD delay than G_1 and how to trigger the sMN to perform an inter-gateway LU are new challenging issues, since the sMN may be multiple wireless hops away from the gateway, as shown in Fig. 7.1(c).
- Challenge 2: assume that an sMN has visited several MRs (uMR, vMRs, or cMR) before an intra-gateway LU action is triggered as shown in Fig. 7.1(d).

With PD performance guarantees addressed in Challenge 1, how to decide which location entity (uMR, vMRs, or cMR) the sMN should report its location to that minimizes location management protocol overhead in terms of lower LU overhead in the mesh backbone is another new challenging issue.

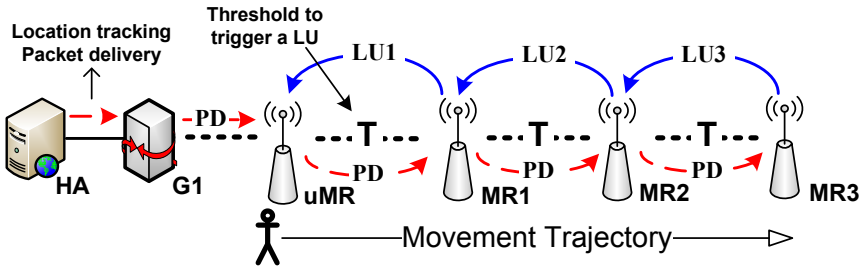
To the best of the knowledge, the above two design challenges have not been addressed in any existing work on location management in WMNs and they are the focus of the design. The proposed *DoMaIN* framework can work with any dynamic location update triggering mechanism that addresses the issue shown in Fig. 7.1(b) but provide novel yet practical designs to specifically address the two new challenges in IiWMNs shown in Fig. 7.1(c) and (d).

6.2 Exploring Location Management Designs in WMNs

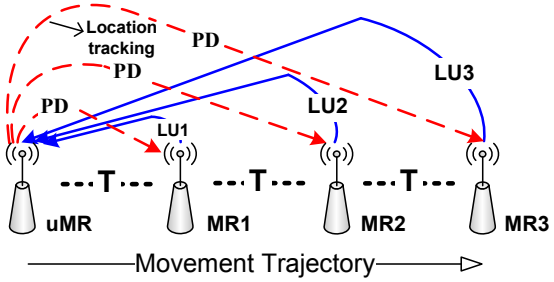
Before introducing the proposed *DoMaIN* for location management in IiWMNs, three straightforward location management designs are first described in this section, depending on how an LU is performed. These three designs will be used as the basis for performance comparison with the proposed design.

6.2.1 Location Tracking Chain based on Movement (LTC-M)

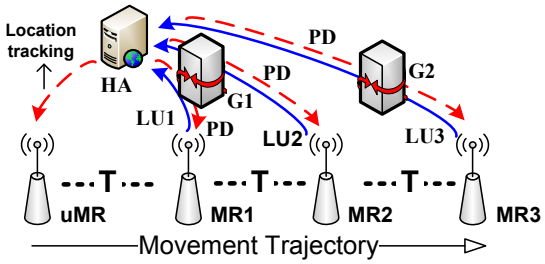
The first location management design, a resemblance to the location tracking chain scheme proposed for Mobile IP networks [26], is shown in Fig. 6.2(a). In this design, an LU action is required from an sMN only after it has visited T different MRs, where T can be different for different sMNs and dynamically changed. As shown in the figure, uMR, MR_1, MR_2, MR_3 are T hops away from each other. An sMN without any active session initially residing under its uMR follows a movement trajectory passing MR_1, MR_2, and finally reaches MR_3. During the movement, three LU actions are triggered and the sMN performs each LU to update the MN's cMR address to its previously updated MR (e.g., MR_1 to uMR, MR_2 to MR_1, etc.). Hence, a location tracking chain based on movement (LTC-M) within the wireless mesh backbone is formed.



(a) LTC-M scheme



(b) LTC-R scheme



(c) LTC-H scheme

Figure 6.2: Location update and packet delivery procedures under three straightforward location management designs.

However, the LTC-M scheme cannot address Challenge 1 as explained in Section II, e.g., MR_3 might have a better gateway other than G_1 to provide the Internet session PD. Even if G_1 is the only gateway MR_3 can reach, the best routing path for PD from G_1 to MR_3 may be different from the one formed by the LTC-M scheme ($G_1 \rightarrow uMR \rightarrow MR_1 \rightarrow MR_2 \rightarrow MR_3$). In addition, LTC-M also cannot address Challenge 2 since each LU is made statically to the previously updated MR without considering minimizing the LU overhead.

6.2.2 Location Tracking Chain based on Routing (LTC-R)

Another location management design is shown in Fig. 6.2(b). Similarly, MR_1, MR_2, and MR_3 are T hops apart from each other. In this design, the MN performs an LU after visiting T different MRs directly to the uMR using its cMR address. Hence, a new location tracking chain, LTC-R, 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.

Similar to the LTC-M scheme, LTC-R also cannot address Challenge 1. On the other hand, the best routing path between the uMR and cMR for PD can be ensured under the LTC-R scheme, since the path between the uMR and cMR is determined based on the routing protocol adopted. However, performing LUs always to the uMR can cause high LU overhead in the mesh backbone, thus can cause severe *scalability* problem when the number of LU actions increases.

6.2.3 Location Tracking Chain based on LU to HA (LTC-H)

In this scheme, an sMN always performs LUs directly to the HA, as shown in Fig. 6.2(c). The LTC-H scheme can always choose the best gateway for Internet data sessions to address Challenge 1. Even in the Intra-gateway LU scenario, the PD path between the gateway and cMR is optimized since it is determined by the routing protocol adopted. However, LTC-H can cause high LU overhead in the mesh backbone among all the three schemes since each LU is made all the way to the HA which is located in the Internet.

6.2.4 A Hybrid Location Tracking Chain (Hatch)

Based on LTC-M and LTC-H, a *hybrid* LTC scheme (*Hatch*) [75] is proposed, where an sMN initially performs LUs using the LTC-R procedure but can trigger an LTC-H procedure when the hop distance between the latest update MR and uMR reaches a certain threshold. Hence, the *Hatch* scheme has better location management performance than both LTC-R and LTC-H. Under the *Hatch* 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 5.

However, *Hatch* also inherits the issues of LTC-R and LTC-H. It can not always address Challenge 1 and 2 under a random mesh topology and arbitrary sMN movements.

Algorithm 5: LU Procedure of the Proposed Hybrid Scheme

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

6.2.5 Summary

Each of the above designs may provide satisfactory PD performance under certain scenarios, depending on the network topology and sMN's movement trajectory. LTC-M provides better PD performance when each movement of an sMN is topological farther to previously updated location entities. In contrast, LTC-R provides better PD performance when each movement of an sMN is topological closer to previously updated location entities. On the other hand, LTC-H provides better PD performance when the movement of an sMN leads to the same topology distance to different gateways.

However, none of the above location management schemes can always provide the best PD performance under random mesh topology with arbitrary sMN movements, while minimizing the LU overhead incurred in the mesh backbone.

6.3 The Proposed *DoMaIN* Framework for Location Management in WMNs

In this chapter, a *DoMaIN* framework is proposed which is fundamentally different from previous approaches to support location management for sMNs in WMNs. Four

main goals for the *DoMaIN* design are identified:

1. *Satisfactory location management performance in terms of PD delay*: The performance of data session delivery for sMNs should be satisfied with efficient PD procedures.
2. *Minimized location management protocol overhead in terms of LU overhead*: With PD performance guarantees, minimized LU overhead in the mesh backbone helps the IiWMN scale well to support a large number of sMNs.
3. *Adaptivity to support any network topology and arbitrary sMN roaming scenarios*: *DoMaIN* should be able to adapt to the changes of network topology and MN movements.
4. *Minimum changes on the MN side*: The design of *DoMaIN* should induce the least changes on the sMN side by exploring the characteristics of WMNs.

In order to achieve the above goals, the design includes two parts. In the first part, changes on the network side are necessary so as to periodically provide location information to sMNs and to minimize changes on the MN side. In the second part, sMNs perform location estimation when an LU is triggered based on network-provided location information.

6.3.1 Network Design

In IPv6-based wireless networks, *router advertisement* (RAs) messages broadcasted by each MR can be utilized by MNs to find out whether its movement is an intra- or inter-subnet movement, because RAs can indicate whether a change of subnet occurs. However, such information is insufficient for sMNs to make proper LU decisions to address the aforementioned Challenge 1 and 2. Hence, new location information needs to be provided from the network side to sMNs.

In the design, new location information includes the gateway information for addressing Challenge 1 and the neighbor MR information shared among neighboring MRs for addressing Challenge 2. In addition, a new data structure, location report is

introduced, for facilitating sMNs' LU decisions during the location estimation stage.

6.3.1.1 Gateway Information

To address Challenge 1, information on the availability of gateways is vital since IiWMNs are primarily used for Internet-based applications. An Internet-based MANET architecture was proposed in [76, 77] where a gateway periodically disseminates the *gateway advertisement* (GA) message containing its gateway ID to all non-gateway MANET nodes. In this way, each MANET node can be aware of the availability of all gateways it can reach. Motivated by this, in the design, modified *gateway advertisement* (mGA) messages are added with new fields (G_{ID}, G_{HOP}) (G_{ID} and G_{HOP} represent the gateway ID and the corresponding number of hop distance to reach this gateway, respectively). Likewise, each gateway needs to disseminate its gateway information by propagating the mGA message to all MRs. When an mGA message arrives at an intermediate MR, either a new mGA entry is recorded or an existing one gets updated with a newer G_{HOP} (a lower hop number to the gateway G_{ID}). Then, the value of G_{HOP} in the mGA is incremented by one and the mGA is rebroadcasted again. In this way, each MR in the mesh backbone can obtain the number of gateways it can reach the Internet and the corresponding shortest hop distance (G_{HOP}) associated to each gateway (G_{ID}).

Different from Internet-based MANETs, MRs in IiWMNs are mostly static. Hence, the topology of the mesh backbone seldom changes and mGA messages are only generated once and propagated during the network deployment phase. By using the above gateway information and the number of hops chosen as a criterion, each MR knows the gateway ID G_{ID} with the shortest hop distance among all available ones (potential inter-gateway LU candidate) and the shortest hop distance G_{HOP} to each gateway (best PD for intra-gateway LU). The gateway information can be utilized by sMNs for location estimation to address Challenge 1.

6.3.1.2 Neighbor MR Information

A scalable location management protocol needs to address Challenge 2, i.e., to minimize the LU overhead in the mesh backbone by helping sMNs perform an LU to a proper MR when an LU action is triggered. Hence, the network needs to provide more location information for sMNs, namely, neighbor MR information, with which an sMN can be aware of the network topology during its movements.

To obtain neighbor MR information, the HELLO message is utilized, a message used in many existing routing protocols to check the availability of neighbors [78]. Neighbor MR information needs to include the information of the availability of neighboring MRs and the corresponding hop distance to them. Hence, the modified HELLO (mHELLO) message has a nested tuple data structure $\{M_{ID}, (N_{ID}, N_{HOP}), (G_{ID}, G_{HOP})\}$. Here, N_{ID} and N_{HOP} represent the ID of a neighbor MR and the corresponding number of hops to reach that MR, respectively. M_{ID} is the ID of an MR, which can be used to indicate different senders of an mHELLO message.[1]

After obtaining the gateway information, each MR starts to broadcast the mHELLO messages. Upon receiving an mHELLO message sent by MR_i, MR_j processes and/or relays the mHELLO message as needed. Different from the mGA, which can be relayed to all MRs in the mesh backbone, how far or how many hops neighbor MR information needs to propagate can be dynamically determined.

Apparently, the farther away the mHELLO message propagates, the more neighbor MR information an MR can obtain and the more network topology information can be obtained by an MR. However, the high extra overhead can easily cause the scalability issue. On the other hand, if only the minimal information of the network topology is provided, sMNs need to retrieve location information from a cMR every time they make a movement (e.g., visit a different MR). Therefore, the minimal num-

[1] In this Chapter, for the purpose of simple demonstration, an integer is used to represent the ID while IP addresses can be used to represent the ID in real implementations. In addition, X_i is used to represent a variable/data structure and $X[i].Y$ stands for the element Y of X_i.

ber of hops an mHELLO message needs to propagate should be determined to suffice for sMNs' location estimation.

In the design, the action an MN moving from an MR to another MR is called "*one movement*". The two MRs during one movement of an sMN can be maximally h-hop (a *worst* movement scenario that causes the longest hop distance between the two MRs) is assumed to be apart from each other in the multihop mesh backbone. Then, in order to obtain the information of an MR that is h hops away, an mHELLO message should propagate to other MRs within $\lceil \frac{h}{2} \rceil$ hops. In this way, an MR can directly obtain the information of those MRs within $\left[1, \lceil \frac{h}{2} \rceil\right]$ hops and indirectly obtain information of those MRs within $\left[\lceil \frac{h}{2} \rceil, h\right]$ hops with the help of a neighbor MR that is within $\lceil \frac{h}{2} \rceil$ hop distance away. Hence, an MR can receive two types of mHELLO messages, namely, direct mHELLO entries and indirect mHELLO entries. In direct mHELLO entries, the M_{ID} is "Null" on the sender side but replaced by the ID of the receiving node. As shown in Fig. 6.3 Part I, each MR ($R1$, $R2$, $R3$) initially forms one direct entry after receiving the gateway information sent by $G1$. In Fig. 6.3 Part II, MR $R2$ receives two mHELLO messages 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. 6.3 Part III, $R3$ can receive its 2-hop neighbor $R1$'s mHello message forwarded by $R2$, in which the M_{ID} is R2. This indirect mHELLO entry indicates the relations of neighboring MRs ($R2$ and $R1$). Moreover, a direct entry can also be created from one direct and one indirect ones (e.g., $R3$ can create a direct entry $(R3, R1, 2, G1, 3)$ from $E1$ and $E2$ as shown in Fig. 6.3 Part III. In a word, direct entries can be used by sMNs to perform location estimation for the movement within $\left[1, \lceil \frac{h}{2} \rceil\right]$ hops. Indirect entries can be used by sMNs to perform

location estimation for the movement within $\left[\left\lceil \frac{h}{2} \right\rceil, h\right]$ hop distance.

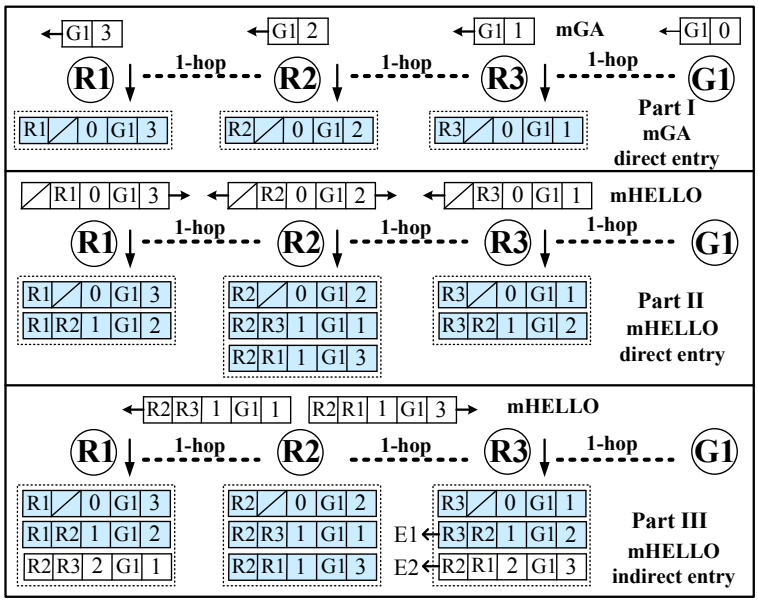


Figure 6.3: An example of neighbor MR information formation. Direct mHELLO entries are in shaded color to be differentiated from indirect ones.

The pseudocode of gateway and neighbor MR information mGA and mHELLO formation is shown in Algorithm 6. Similar to the gateway information, neighbor MR information can be obtained during the IiWMN deployment phase which does not cause "extra burden" on the network if the topology does not change.

6.3.1.3 Location Report

With the information obtained from mHELLO messages, a new data structure, location report is formed, for facilitating sMN location estimation during its movements. This is inspired by the neighbor report defined in IEEE 802.11k supporting information exchange between APs by aggregating multiple data entries/instances into one report.

Generally, the desired mHELLO entries are the G_{ID} with the lowest G_{HOP} (possibly leading to the lowest data PD delay) and the N_{ID} with the lowest N_{HOP} (possibly leading to the lowest LU overhead in the mesh backbone). However, since all mHELLO entries in a location report are received randomly, if h increases, searching

Algorithm 6: Pseudocode of gateway and neighbor MR information (mGA and mHELLO) Formation

```
1  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 = ⌈h/2⌉;
2        /* During the IiWMN deployment phase, each gateway triggers the mGA procedure */;
3  Use its IP address as G_ID & initialize G_HOP = 0 ;
4  Encapsulate mGA in an IPv6 packet with TTL = 255 ;
5  Broadcast the mGA message;
6                              /* MRs process the received mGA message */;
7  G_HOP = G_HOP + 1 and get G_ID;
8  if G_ID already exists in the G_info_table then
9      if new G_HOP ¡ old G_HOP then
10         Update mGA entry in the G_info_table;
11     else
12         Discard this mGA message;
13 else                                                          /* this is a new mGA entry */
14     Add mGA entry to the G_info_table;
15 if TTL != 1 then
16     Encapsulate mGA in an IPv6 packet with TTL = TTL - 1 ;
17     Rebroadcast mGA ;
18 else
19     Discard this mGA message;
20       /* After the establishment of G_info_table, each MR triggers the mHELLO procedure */;
21 Use its IP address as N_ID, initialize N_HOP = 0, M_ID = NULL;
22 for i = 0; i < num_G; i++ do
23     Get the mGA entry in the G_info_table;
24 Insert each mGA entry to an mHELLO entry;
25 Encapsulate mHELLO in an IPv6 packet with TTL = 255;
26 Broadcast the mHELLO message;
27                           /* MRs process the received mHELLO message */;
28 N_HOP = N_HOP + 1;
29 if N_HOP == TH_h or N_HOP == 2 * TH_h then
30     Discard this mHELLO message;
31 else
32     Get M_ID, N_ID, and G_ID from mHELLO;
33 if N_ID already exists in the N_info_table then
34     if new N_HOP ¡ old N_HOP then
35         if M_ID == NULL and new N_HOP < TH_h then
36             Update M_ID using its IP address;
37         Update mHELLO entry in the N_info_table;
38     else                                  /* check whether this mHELLO has a new G_ID */
39         if G_ID already exists in the G_info_table then
40             Discard this mHELLO message;
41         else
42             if M_ID == NULL and new N_HOP < TH_h then
43                 Update M_ID using its IP address;
44             Update mHELLO entry in the N_info_table;
45 else                                                      /* this is a new mHELLO entry */
46     Add mHELLO entry to the N_info_table;
47 Encapsulate mHELLO in an IPv6 packet with TTL = TTL - 1 ;
48 Rebroadcast the mHELLO message;
```

all the mHELLO entries in a location report to obtain a desired one can cause a long searching time, thus affect a timely LU decision.

Algorithm 7: 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);
```

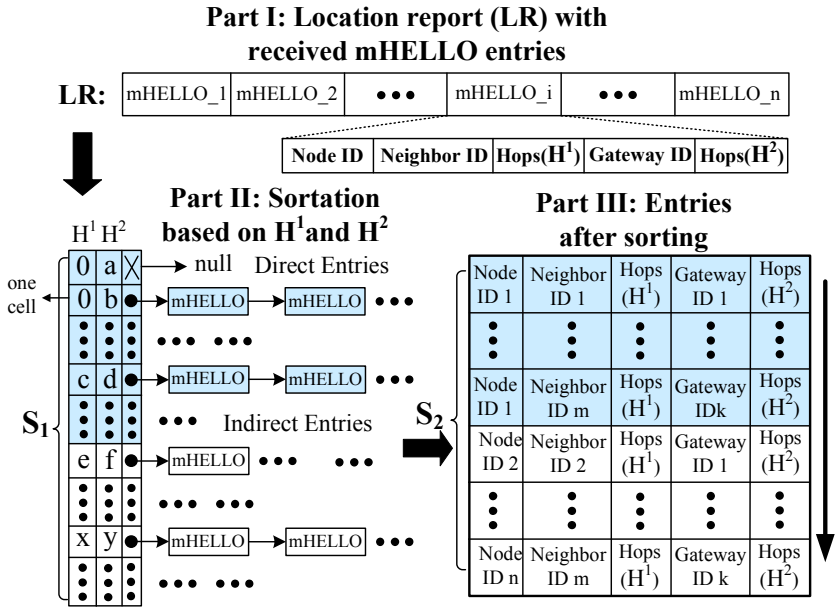


Figure 6.4: Location report formation and its data structure on each MR based on gateway information and mHELLO message in the proposed *DoMaIN* framework.

Hence, the mHELLO entries in the location report need to be sorted. Specifically, the direct mHELLO entries is desired to be accessed earlier than the indirect ones. In each category of mHELLO entries, the hop distance N_{HOP} is used as `KEY 1` (H^1) to form an ascending order, while the corresponding gateway hop distance G_{HOP} is used as `KEY 2` (H^2) to further sort the entries. The detailed formation of a sorted location report under the *DoMaIN* scheme is shown in Algorithm 7. As shown in Fig. 6.4 Part I: there are n randomly received mHELLO entries $mhello_1, \ldots, mhello_n$ in the location report. Each mHELLO entry is an ordered tuple with two keys: H^1 and H^2. H^1 is more significant over H^2. A location report is considered sorted in an ascending order with respect to the keys if and only if for every pair of mHELLO entries $mhello_i$, $mhello_j$ $(i < j)$, $mhello[i].H^1 \leq mhello[j].H^1$. When $mhello[i].H^1 = mhello[j].H^1$, the entries $mhello_i \leq mhello_j$ if and only if $mhello[i].H^2 \leq mhello[j].H^2$. As noticed, there could be multiple mHELLO entries when the value of two keys are equal. Hence, the mHELLO entries with the same key value are linked as a linked list. After the sortation is done, *Remove_empty_cells* removes the empty cells in the location report

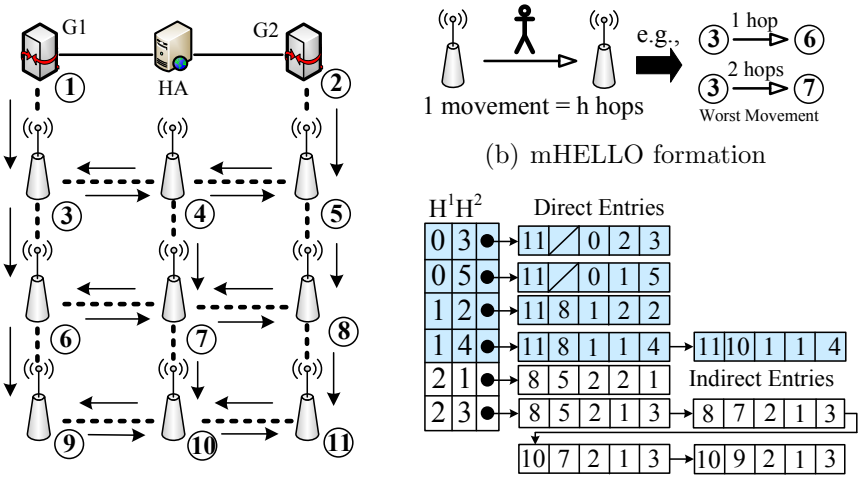
whose entries have been moved to the linked lists and returns the total number of cells S_1. The final data structure of the linked lists with two keys is shown in Fig. 6.4 Part II.

The sorted location report has the following three properties: 1) the gateway with the lowest G_{HOP} is always placed in the top (entries in the first linked list) for facilitating sMNs to make a timely inter-gateway LU decision; 2) each MR always has the lowest G_{HOP} to each gateway by keeping only one entry to each gateway with the lowest G_{HOP}; and 3) the sorted mHELLO entries for each neighbor MR in the location report also indicate the corresponding LU overhead in an ascending order. The visiting sequence of mHELLO entries in the formed data structure is shown in Fig. 6.4 Part III. If $List\_length(i)$ is the total number of mHELLO entries in the ith linked list, then the number of mHELLO entries in one location report $S_2 = \sum_{i=0}^{s_1-1} List\_length(i)$.

The formed location report on each MR does not change most of the time. Then, each sMN can now build its own location database during its movements, learning the information of the network topology by receiving a location report sent by each MR that the sMN has visited.

6.3.1.4 Example

Let us consider a grid mesh topology with two gateways, as shown in the Fig. 6.5(a). For simplicity, an sMN is required to perform an LU whenever making a movement (e.g., visiting a different MR) is assumed. Thus, the *worst* movement of an sMN can cause a maximum 2-hop distance (e.g., moves from MR_3 to MR_7). Thus, the mHELLO messages only need to be exchanged between 1-hop neighboring MRs. After the propagation of mHELLO messages, the corresponding location report can be formed. For instance, 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



(a) A grid mesh topology

(b) mHELLO formation

(c) Location report on MR ⑪

Figure 6.5: An example of location report formation in a grid mesh topology with two gateways.

underneath the direct ones. All entries under each category are sorted by H^1 and H^2.

6.3.2 Location Estimation based on Location Report

Since sMNs are in the *silently* roaming mode, the goal of the *DoMaIN* framework is to preserve the sMN power consumption but utilize its "*listen*" capability to make proper LU decisions. Therefore, before an LU action is triggered, an sMN performs location estimation using the received location reports from the MRs it has visited. In the following, sMN's location database and how to use this new database to make LU decisions in order to address Challenge 1 and 2 are described.

6.3.2.1 Preliminary

An sMN starts to form its location database when it first registers to an MR (uMR). Specifically, the MN's location database includes three tables, as shown in Fig. 6.6:

1. **Movement History Table** (MHT): this table records each MR that an sMN has visited. Let $MHT = \{R_1, R_i, \ldots, R_m\}$ after m movements. R_i is the ID of MR_i. Each entry in the MHT is obtained from the received modified *RA* messages. Each RA message is modified by adding the ID of the MR. In IPv6-

based wireless networks, RA messages are periodically broadcasted by each MR. The MHT has the following features: each entry is added sequentially into the MHT to form the sMN's movement history. The fist entry is uMR's ID; intermediate ones are vMRs' IDs; and the latest one corresponds to cMR's ID. Neighboring entries in the MHT are different to represent one movement. However, the same entry can appear in the MHT more than once indicating that the sMN moves back and forth.

2. **LU History Table** (LHT): this table is a subset of the MHT at movement m. Let $LHT = \{R_1, R_j, \ldots, R_n\}$, $(n < m)$ and $LHT \subseteq MHT$. During an sMN's arbitrary movements, the LHT keeps the records of each MR that the sMN has performed an LU to in the LTC-M scheme, while in the proposed *DoMaIN* scheme, the LHT only keeps some of them. The LHT also shares similar features as the MHT.
3. **Location Information Table** (LIT): this table keeps the location reports the sMN has received during the m movements. Let $LIT = \{LR_1, \ldots, LR_m\}$.

During an sMN's intra-gateway movement, the number of entries in the MHT and LIT increases. The number of entries in the LHT varies depending on both the sMN's movement and LUs. However, the sMN resets all three tables after it performs an inter-gateway LU (i.e., the uMR of the sMN changes).

6.3.2.2 Location Estimation Procedure

In the following, how an sMN estimates its location and makes a proper LU decision based on the location database with the above three tables is illustrated. As shown in Fig. 6.6, when an sMN reaches an MR at the movement m, it receives R_m from the modified RA message sent by the MR, inserts its entry into the MHT, and places the received mth location report LR_m into the LIT. The detailed location estimation procedure under the *DoMaIN* scheme is shown in Algorithm 8. $LIT[\alpha, \beta]$ stands for the βth mHEELO entry of the αth location report in the LIT. Benefited from the sorted entries in the location report, G_{ID} entries in the LR_m can be accessed

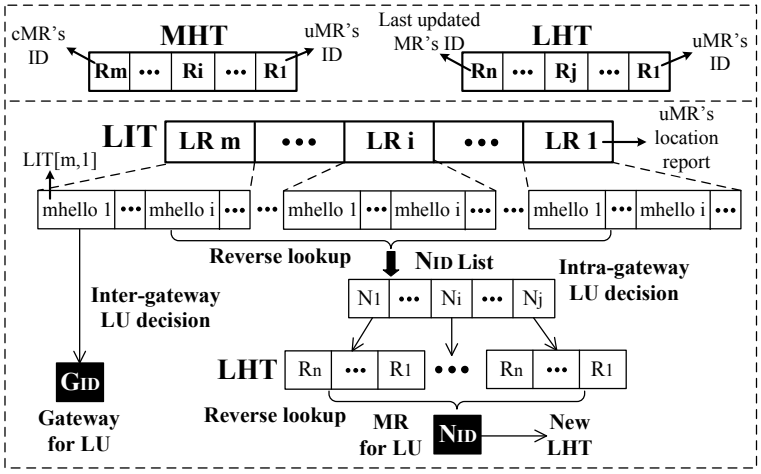
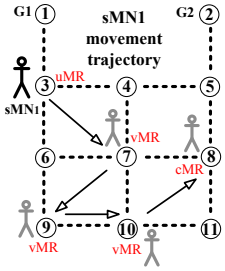
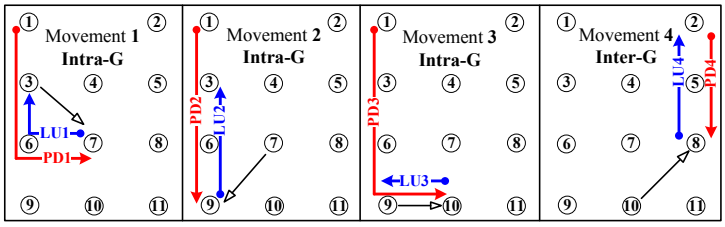


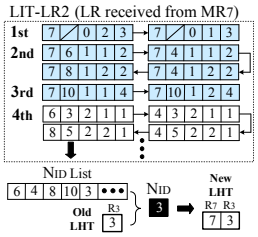
Figure 6.6: sMN's location database for location estimation in the *DoMaIN* framework including three table formation and lookup.



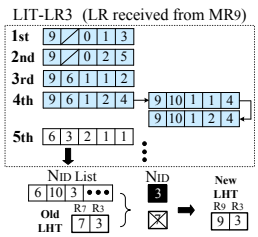
(a) A Grid Mesh Topology



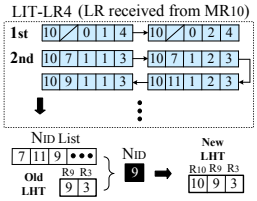
(b) The four LU and PD procedures in the proposed DoMaIN



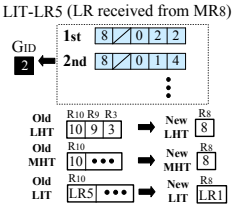
(c) Movement 1



(d) Movement 2



(e) Movement 3



(f) Movement 4

Figure 6.7: An example of location report formation in grid mesh topology with two gateways.

in an ascending order. Firstly, an sMN accesses the 1st mHELLO entry $LIT[m,1]$ to check whether the G_{ID} of $LIT[m,1]$ is different from its current G_{ID} for Internet access ($LIT[1,1].G_{ID}$). If so, the sMN needs to further check the next entry to see whether the G_{HOP} of $LIT[m-1,1]$ is higher than that of $LIT[m,1]$. If yes, the sMN

can now be assured that it obtains a better gateway than its current one and performs an inter-gateway LU to the HA via a new gateway ($LIT[m,1].G_{ID}$). If not, the sMN continues to check $LIT[m-2,1].G_{HOP}$, and so on. One special case is that multiple gateways have the same lowest G_{HOP} to MR_m. Hence, the search time before making an inter-gateway LU decision is between 2 to num_G entries (num_G is the number of gateways).

Under the case when the G_{HOP} of the sMN's current gateway ($LIT[1,1].G_{ID}$) is the lowest or it shares the same lowest value with one or more gateways, an intra-gateway LU decision is needed. Now, the sMN needs to determine the N_{ID} for the LU, the closest MR in the LHT such that the number of hops of the best path between the $LIT[1,1].G_{ID}$ and cMR formed by this LU is $LIT[m,1].G_{HOP}$. In Algorithm 8, $Get\_jth\_list(\alpha,\beta)$ returns the βth mHELLO entry of the αth linked list. Starting from the 1st mHELLO entry of the linked list where $N_{HOP}=1$ in the LIT, the sMN compares the N_{ID} of each entry with the entries in the LHT in a reverse order from R_n to R_1. The procedure stops with a matched N_{ID} from both tables. The search time for the desired N_{ID} in the LIT is between 1 and $m*S_2$ entries (S_2 is the number of entries in one location report). Then, to indicate the newly formed LU entry chain in the LHT, the entries between the latest and desired one need to be removed from the LHT. The current entry R_m from MR_m is added to the LHT. Finally, the sMN performs an intra-gateway LU to the desired N_{ID}.

6.3.2.3 Example

Fig. 6.7(a) shows an example of an 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. Assume that the sMN initially resides under the uMR (MR_3) and G_1 is chosen as the default gateway for Internet access and 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. 6.7(c)-(f) shows the sMN's LU cases corresponding to the four movements. In

Algorithm 8: 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 */;
```

movement 1 ($MR_3 \rightarrow MR_7$), location estimation finds G_1 and G_2 in the first two mHELLO entries in the 1st linked list in the LIT sharing 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 1st 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. 6.7(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. 6.7(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. 6.7(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 $G_{HOP} = 2$ which is 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. 6.7(f). The corresponding paths for the LU and PD procedures in each step of the location estimation are shown in Fig. 6.7(b).

Table 6.1: 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

6.3.3 Dynamic LU Trigger

As mentioned earlier, the proposed *DoMaIN* framework can address the two new challenges independent of the underneath adopted LU triggering mechanism. However, the *DoMaIN* framework can bring extra benefits for implementing different dynamic LU triggering methods. Among the three dynamic LU triggering methods, time-based LU only requires the implementation on the sMN side, thus it can be directly applied to WMNs without changes in the network. However, since both movement- and distance-based schemes need information from the network, changes must be made before applying them to the multihop mesh backbone.

The sMN's last updated location entity is chosen as a reference to discuss the movement- and hop-based LU triggering designs. For the movement-based LU, it can be implemented for an sMN based on its previously formed MHT. The sMN counts the number of MRs it has visited during its arbitrary movements. When the number reaches a user-defined threshold of the movement-based scheme, it triggers an LU action. Moreover, in the wireless mesh backbone, the performance of either LU or data PD relies on the number of wireless hops the control or data packets traverse in the wireless mesh backbone. Thus, a dynamic hop-based LU triggering mechanism is developed as an alternative. The dynamic hop-based LU can be realized in the wireless mesh backbone by utilizing the proposed LIT, since the N_{HOP} in each mHELLO entry can provide the hop distance from the last updated location entity. As shown in Fig. 6.8, when the threshold of the movement-based scheme is $m = 2$ in scenario 1 to 3, the hop distance can vary from 0 to 2. In scenario 4, one movement $m = 1$ can cause a multi-hop distance ($h = 2$) is observed.

6.3.4 Implementation Issue

In the 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

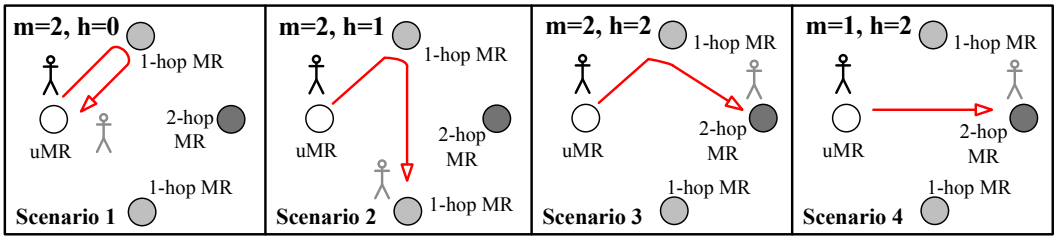


Figure 6.8: Scenarios for movement- and hop-based LU triggers. (m: the number of movements; h: the number of hops)

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, an sMN needs to keep the location reports it receives to perform location estimations. Considering the fast pace of technological advancements, an 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.

6.3.5 Summary

Given an adopted dynamic LU triggering method, the proposed *DoMaIN* framework can always help an sMN choose the best gateway to perform an LU to, namely, an intra- or inter-gateway LU, which overcomes the shortcomings of the three location management schemes described in Section 3. In addition, in an intra-gateway LU case, the proposed *DoMaIN* framework can always help an sMN choose the closest MR to perform an LU to, which minimizes location management protocol overhead incurred in the mesh backbone and simultaneously guarantees the corresponding PD performance. Finally, with the formed MHT and LIT, both dynamic movement- and hop-based LU can be realized in IiWMNs. The comparison of different location management schemes is shown in Table 6.1.

6.4 Performance Analysis

In this section, the performance of the proposed dynamic location management (*DoMaIN*) framework is evaluated with three other location management schemes (LTC-M, LTC-R, and LTC-H) using OPNET[13] simulations.

6.4.1 Simulation Scenarios and Assumptions

First, two LU triggering methods are implemented in OPNET. The movement- and hop-based LU methods can be adopted by the four considered location management designs for performance comparisons. Second, the LTC-M, LTC-R, LTC-H, and *DoMaIN* are implemented independent of the above two LU triggering methods. Then, Two simulation scenarios in OPNET for different purposes are created.

In the first scenario, performance analysis of a single sMN which follows a predefined movement trajectory in the mesh backbone under the four location management schemes is provided. Fig. 6.9(a) to (e) show the sMN's movement trajectory where the black dots with a number indicate the sequence of MRs the sMN visits. Here in this scenario, assume that the LU action is triggered by one movement, i.e., the sMN visits a different MR. Thus, in the predefined movement trajectory, the sMN starts moving from MR_1 and finally resides under MR_6. Five LU actions in total are performed by the sMN under the four location management schemes.

Moreover, this scenario can be easily extended to a second scenario, a general one where the *Random Waypoint Mobility* model [79] is adopted to characterize sMNs' mobility. The performance of the four location management schemes are evaluated and compared by changing 1) the LU triggering method and its threshold; 2) the number of sMNs residing under the mesh backbone. In these two scenarios, location management performance is defined as the delay of data packet delivery and location management protocol overhead is defined as the control overhead of LUs incurred in the mesh backbone from the sMN to the MR that the sMN performs an LU to (i.e., the sum of LU messages and the corresponding routing discovery messages). AODV

[59] is considered in the architecture as the mesh routing protocol to interact with the implemented location management protocol. The parameters used in the simulations are shown in Table 6.2.

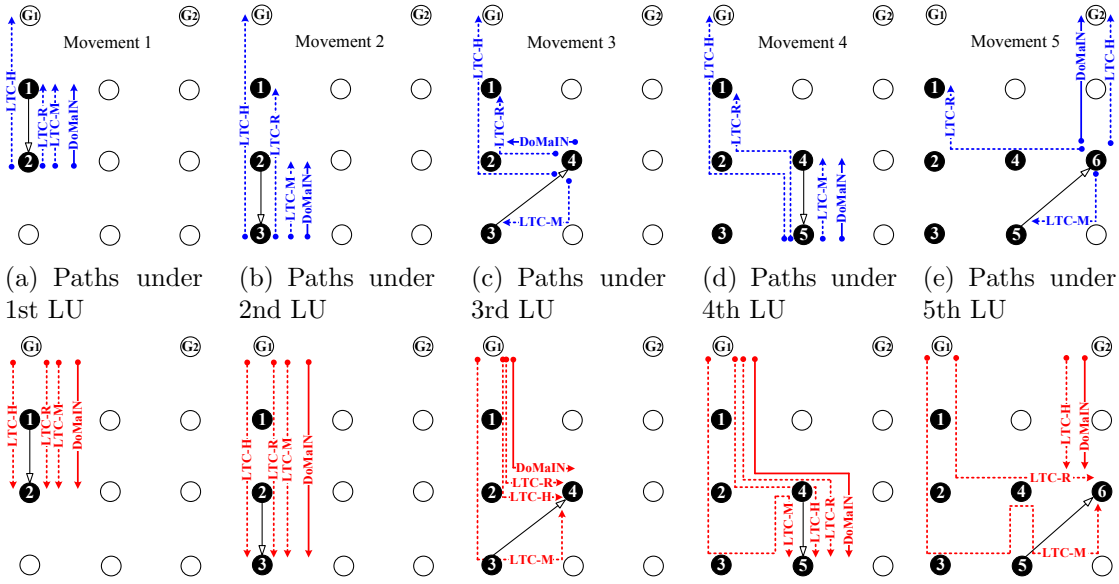
Table 6.2: Simulation Parameters

IiWMN Parameters	
MR transmit power (W)	0.05
Packet reception-power threshold (dbm)	-95
Buffer size (bits)	256000
IPv6 interface routing protocol	RIPng
Multihop routing protocol	AODV
AODV active route timeout (sec)	3.0
IPv6 *Router Advertisement* interval (sec)	constant (20)
MR queuing scheme	FIFO
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)

6.4.2 Results Analysis

6.4.2.1 Performance Analysis on A Single sMN with A Predefined Movement Trajectory

In this scenario, the sMN's movements follow a predefined trajectory (from MR_1 to MR_6) with a constant velocity. An LU action is triggered at the sMN whenever it makes a movement. The interarrival time of data packets destined to this sMN from the Internet is close to the time it takes the sMN moving from one MR to another MR. In this way, the PD delay corresponding to the LU overhead at the moment of sMN's each LU action can be shown. The detailed multihop paths in the mesh backbone chosen for the LU signaling and data PD procedures under the four location management schemes are shown in Fig. 6.9(a) to (j). The solid line shows the proposed *DoMaIN* framework, while the dashed lines show three other

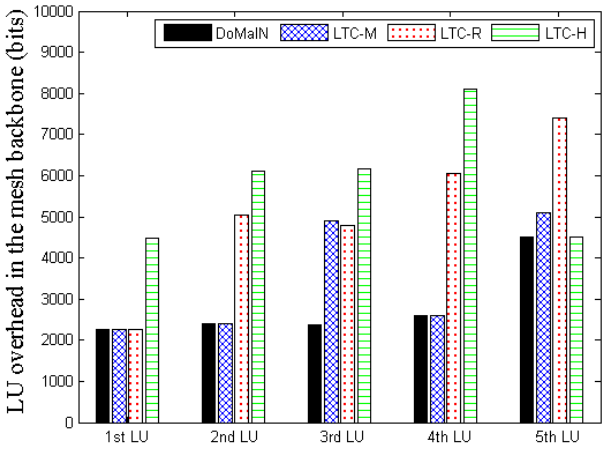


(a) Paths under 1st LU (b) Paths under 2nd LU (c) Paths under 3rd LU (d) Paths under 4th LU (e) Paths under 5th LU

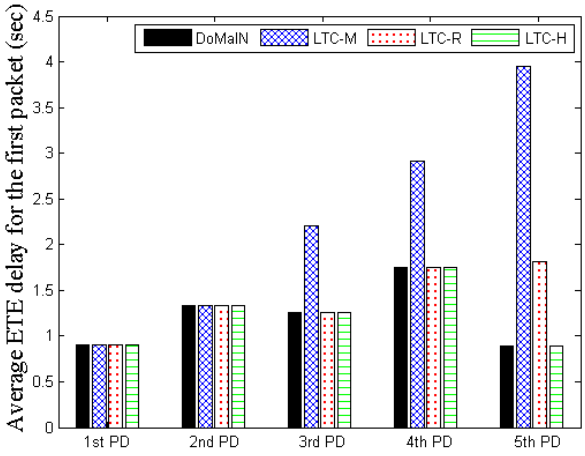
(f) Paths under 1st PD (g) Paths under 2nd PD (h) Paths under 3rd PD (i) Paths under 4th PD (j) Paths under 5th PD

Figure 6.9: Multihop paths for LU signaling and data PD in a grid mesh topology under four location management schemes.

schemes (LTC-M, LTC-R, and LTC-H). For movement 1 (MR_1 to MR_2), the path for delivering the LU signaling packets under the LTC-H scheme is longer than the ones under the LTC-M, LTC-R, and *DoMaIN* schemes, while the corresponding path formed for the data PD procedure under the four schemes are the same. At movement 2 (MR_2 to MR_3), the paths for delivering LU signaling packets under the LTC-M and *DoMaIN* schemes are shorter than the ones under the LTC-R and LTC-H schemes. At movement 3 (MR_3 to MR_4), the path for LU signaling packets under the *DoMaIN* scheme is the shortest among all the four schemes is observed. In addition, by always performing the LU to its previous vMR, the LTC-M can cause the longest path in the corresponding PD procedure as shown in Fig. 6.9(h) to (j). For instance, the LTC-M induces the redundant path for the data PD procedure at movement 4 (MR_4 to MR_5) in Fig. 6.9(h). At movement 5 (MR_5 to MR_6), the proposed *DoMaIN* and LTC-H schemes make the sMN perform an inter-gateway LU which results in the shortest path among the four schemes for the data PD.



(a) LU overhead comparison under four schemes



(b) PD delay comparison under four schemes

Figure 6.10: LU overhead and PD delay at the moment of each LU action under four location management schemes.

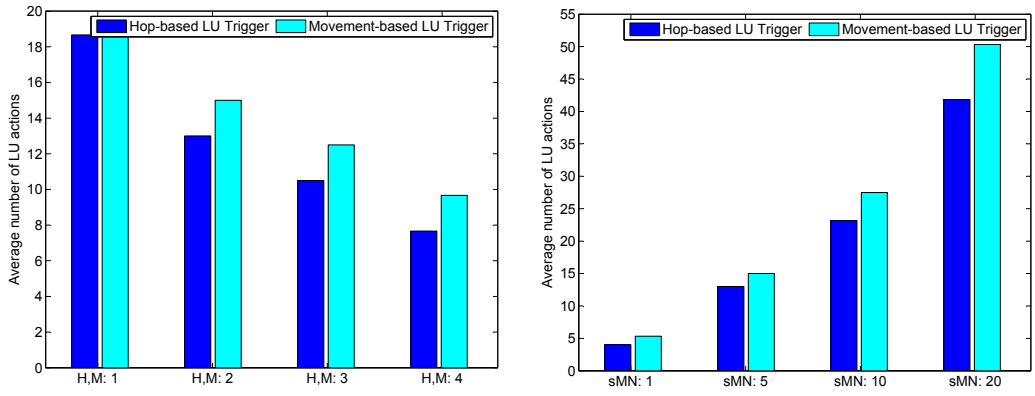
Corresponding to the detailed path setup for both LU signaling and data PD, the LU overhead and data PD delay of each LU action are shown in Fig. 6.10. As expected, the longer the path formed by the location management scheme, the higher LU overhead and longer data PD delay it induces. The LTC-M scheme is seen cause relatively lower LU overhead from the first to 4th intra-gateway LU (movement 1 to 4), as compared to the LTC-R and LTC-H schemes as shown in Fig. 6.10(a), whereas

the LU overhead using the LTC-H is the highest since it makes the sMN perform each LU always to the HA. Meanwhile, the proposed *DoMaIN* can properly decide 1) whether an intra- or inter-gateway LU action is needed, e.g., intra-gateway LUs from movement 1 to 4 and an inter-gateway LU at movement 5 and 2) which MR should be chosen to minimize LU overhead in the mesh backbone and guarantees PD performance in an intra-gateway LU action. For example, at movement 3, the proposed *DoMaIN* outperforms three other schemes due to the lowest LU overhead introduced in the mesh backbone. Correspondingly, in Fig. 6.10(b), the LTC-M can cause the highest PD delay during movement 3 to 5. On the contrary, the LTC-R, LTC-H, and *DoMaIN* have the same lowest PD delay under intra-gateway LU scenarios (movement 1 to 4). Moreover, at movement 5, the LTC-H and proposed *DoMaIN* schemes have the same lower PD delay as compared to the LTC-M and LTC-R. Hence, among the four location management schemes, the *DoMaIN* can always cause the minimal LU overhead in the mesh backbone while simultaneously maintain the lowest delay for the data PD.

6.4.2.2 Performance Analysis of sMNs with Arbitrary Movements

In the second scenario, based on the same grid mesh topology as shown in Fig. 6.9, the mobility model of sMNs is changed to the *Random Waypoint* model. The parameters of the model and data packet inter-arrival time for sMNs are defined in Table 6.2. First the two implemented LU triggering methods (movement- and hop-based LU triggers) are compared and observed how they affect the frequency of LU actions by varying the value of the thresholds and the number of sMNs residing under the mesh backbone.

As shown in Fig. 6.11(a), both LU triggering methods can effectively reduce the average number of LU actions between two consecutive inter-gateway LUs on the sMN side as the value of the thresholds increases while the number of sMNs residing under is fixed. In addition, when the value of the threshold is fixed (e.g., $m, h = 2$) for



(a) Thresholds of LU triggers (number of sMNs: 5)

(b) The number of sMNs varies ($m, h = 2$)

Figure 6.11: Performance comparison under the movement- and hop-based LU triggering methods.

each LU triggering method, the average number of LU actions triggered by the hop-based LU method is less than the movement-based one. The advantage of using the hop-based method over the movement-based one can also be seen when the number of sMNs increases from 1 to 20 as shown in Fig. 6.11(b). Hence, under the grid mesh topology, the hop-based LU triggering method is preferable to be adopted by sMNs in the mesh backbone than the movement-based one as it preserves the power consumption on the sMN side better during its arbitrary movements.

Next, performance comparisons in terms of the average PD delay and LU signaling overhead in the mesh backbone under the four location management schemes are studied in Fig. 6.12. The two LU triggering methods are adopted, the threshold of each method and the number of sMNs residing under the mesh backbone are varied. How different the location management performance under the four schemes can be affected by using different LU triggering methods and the advantage of using the proposed *DoMaIN* over the other three schemes are studied.

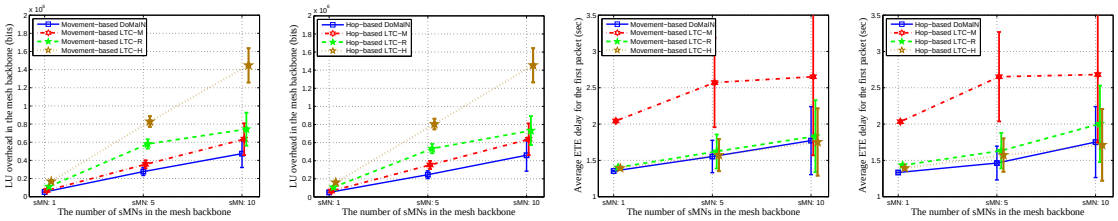
As shown in Fig. 6.12, under the same type of LU triggering method and the same threshold value (e.g., $m, h = 2$ in Fig. 6.12(e)(f)), a lower average LU overhead can be seen in the LTC-M scheme compared to the LTC-R and LTC-H schemes, whereas

a lower average PD delay can be seen in the LTC-H among the three schemes (e.g., in Fig. 6.12(g)(h) where $m, h = 2$). However, the proposed *DoMaIN* outperforms these three designs since it always incurs the lowest average PD delay with the lowest average LU overhead among the four schemes. Moreover, as compared to Fig. 6.12(e), better performance in terms of lower average LU overhead under all four schemes can be seen in Fig. 6.12(f). Since the hop-based LU triggering method in Fig. 6.12(f) generates less LU actions than the movement-based one in Fig. 6.12(e), it reduces the total LU overhead incurred in the mesh backbone. By increasing the value of the threshold in both movement- and hop-based schemes, the average LU overhead under the four schemes can be greatly reduced as expected. As the number of sMNs increases, LU overhead rises as well under the four schemes. However, the proposed *DoMaIN* still keeps the lowest LU overhead among all four schemes, thus provides more scalable location management than the other three schemes (LTC-M, LTC-R, and LTC-H).

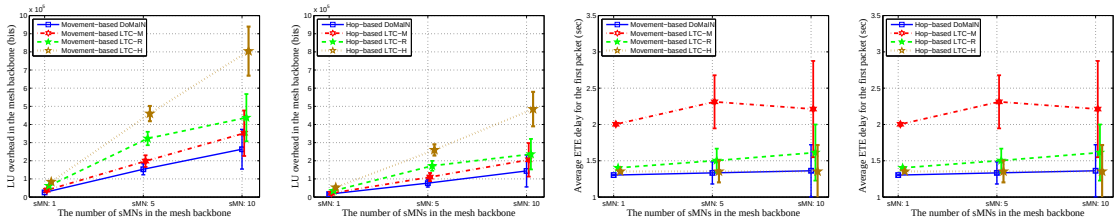
Moreover, notice that there is not much difference in the average PD delay of each location management scheme by using different LU triggering methods. However, as the threshold of m and h increases from 1 to 2 (e.g, Fig. 6.12(c)(d) and Fig. 6.12(g)(h)), the average PD delay of each scheme slightly drops due to the fact that a higher threshold may induce a shorter length of location tracking chain. However, this does not mean that increasing the threshold of the LU trigger can always reduce the average PD delay, since a higher threshold can also cause a higher paging delay from sMN's last known location entity to finding the sMN. That is why the average PD delay of each scheme in Fig. 6.12(k)(l) increases slightly as compared to those in Fig. 6.12(g)(h)).

In summary, given an LU triggering method with a certain threshold, the proposed *DoMaIN* scheme outperforms the other three location management schemes (LTC-M, LTC-R, and LTC-H) by inducing the lowest LU signaling overhead to achieve the

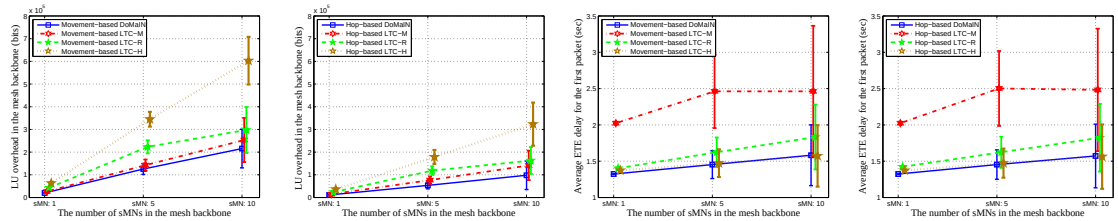
lowest data PD delay. Furthermore, in the simulated grid mesh topology, the proposed *DoMaIN* scheme with a hop-based LU triggering method outperforms the movement-based one in terms of even lower LU signaling overhead in the mesh backbone.



(a) Average LU ($m = 1$) (b) Average LU ($h = 1$) (c) Average PD ($m = 1$) (d) Average PD ($h = 1$)



(e) Average LU ($m = 2$) (f) Average LU ($h = 2$) (g) Average PD ($m = 2$) (h) Average PD ($h = 2$)



(i) Average LU ($m = 3$) (j) Average LU ($h = 3$) (k) Average PD ($m = 3$) (l) Average PD ($h = 3$)

Figure 6.12: The average LU overhead and the corresponding PD delay with standard deviation under four location management schemes.

6.5 Conclusion

In this Chapter, a novel dynamic location management framework (*DoMaIN*) is proposed for MNs silently roaming under Internet-based infrastructure WMNs. As the saying goes, *"The bow whispers to the arrow before it speeds forth - Your freedom is mine." – by Rabindranath Tagore*. In the proposed *DoMaIN* framework, sufficient location information is provided by the network to each sMN before an LU action is triggered. The proposed *DoMaIN* framework can ensure location management performance in terms of the lowest PD delay for mobile users under arbitrary move-

ments in a random mesh topology. In addition, by dynamically guiding users to perform LUs to a desirable location entity, the proposed *DoMaIN* can minimize location management protocol overhead in terms of LU overhead in the mesh backbone. The proposed *DoMAIN* helps the mesh backbone scale up to support a large number of mobile users. The performance of the proposed *DoMaIN* outperforms other location management schemes in different case studies using OPNET simulations are evaluated and verified.

CHAPTER 7: A RESILIENT LOCATION AREA DESIGN IN INTERNET-BASED WMNS

Previous centralized location caching and management schemes proposed for cellular and wireless local area networks (WLANs) are not suitable in an IiWMN environment due to the scalability issues [74]. Moreover, location management in cellular networks adopts a fixed location area (LA) design, where several cells are grouped into an LA during the network deployment phase [80]. LUs are only performed when an MN moves from one LA to another. However, fixed LA design cannot adapt to the changes of traffic load and MN mobility. Hence, the LA design in IiWMNs should be self-configured rather than deployment-based.

A distributed location caching scheme for WMNs is proposed in [38] that caches each MN's location information in MRs while routing data for the MN. However, this scheme only considers LUs when an MN initiates an active data session, but does not consider the case if an MN only receives data packets but not send, or the MN silently roams with no active data sessions. Moreover, when the location query for an MN fails, the gateway is invoked to start the paging procedure by flooding paging messages to all MRs in the mesh backbone, which can cause serious scalability problems in WMNs.

In this chapter, a resilient location area design, *ReLoAD* is proposed, for location management in IiWMNs. Under *ReLoAD*, the size of LAs can adapt to the changes of both paging and service load in the network. Based on the premise that preserving the QoS performance of existing traffic of active MNs is the design goal, the proposed *ReLoAD* can achieve a reasonable tradeoff between signaling overhead caused by the paging procedure and MN power consumption caused by the LU procedure for both intra- and internet sessions. More specifically, the contributions of the *ReLoAD* are:

- Utilizing the *user information* (explained in Section 7.2.1), important location entities which are potential high paging sources construct resilient location areas (RLAs) which can adapt to the changes of paging and traffic load in the network.
- The formation rule of RLAs that separates heavily loaded location entities from paging traffic preserves the QoS performance of existing traffic on them. Furthermore, the adaptivity of RLAs also balances paging overhead and LUs from the silently roaming MNs.
- Practical implementation of *ReLoAD* is possible by integrating with the network-layer multihop routing protocol, IPv6 protocols, and IPv6 address management without adding new functional entities. Hence, the proposed *ReLoAD* for Ii-WMNs is practical.

7.1 Background and Motivation

7.1.1 Architecture Characteristics of IiWMNs

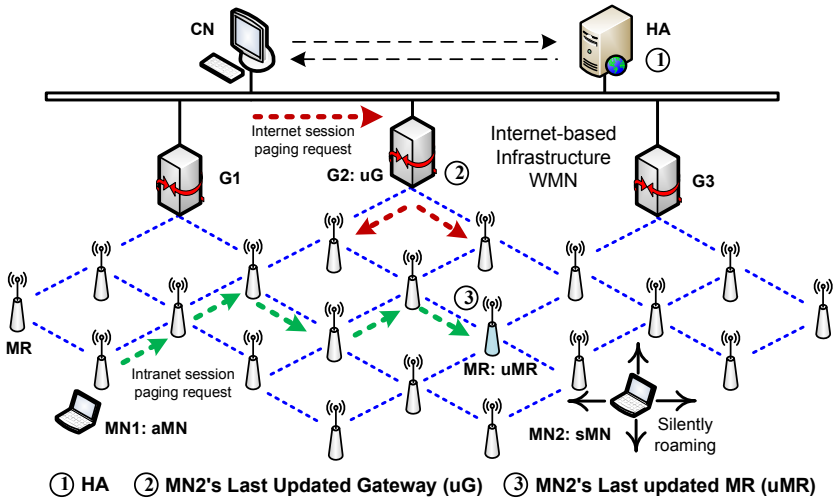


Figure 7.1: Location entities for location management in IiWMNs.

7.1.1.1 Application Categories and Characteristics of MNs

The traffic of various applications in an IiWMN can be classified into two categories: 1) intranet session, i.e., an active session between two MNs inside the WMN, e.g., in Fig. 7.1, MN1 communicates with MN2 that is multiple hops away in the mesh

backbone; 2) internet session, i.e., an active session between an MN in the WMN and a node in the Internet, e.g., a CN in the Internet communicates with MN2 located in the mesh backbone.

In addition, MNs residing in the mesh backbone can be categorized into two groups: active MNs (aMNs) which currently have active end-to-end data sessions and explicit location information (i.e., the IP address of its associated MR); and silently roaming MNs (sMNs) which currently do not have an active data session and only have implicit location information (i.e., the IP address of the last updated MR/gateway). In order to save battery consumption, an aMN with no active session for a while enters a power saving mode and becomes an sMN. On the contrary, an sMN becomes an aMN when initiating an active data session or when there are packets destined to this MN and it is paged by the network.

7.1.1.2 Location Entities and Criteria for Location Area Design

To locate an sMN is a non-trivial task since the exact location of the sMN is unknown to the network. The paging procedure to locate an sMN which silently roams in the mesh backbone is triggered at this sMN's last updated location. Paging traffic may flood all MRs in the whole LA until a response from the sMN is received. If the number of sMNs to be paged increases, significant paging traffic is generated and the load of involved MRs increases.

Besides the home agent (HA)(①) in Fig. 7.1, there are two other entities which may participate in locating sMNs: the last updated gateway (uG)(②) of the sMN for locating the sMR before setting up an internet session and the last updated MR (uMR)(③) for locating an sMR before setting up an intranet session.

If an sMN silently roams without performing any LU and relies only on the network to locate it when there are packets destined to it, the sMN battery consumption can be preserved but a large amount of paging traffic is generated since the sMN could reside under any MR. On the contrary, if an sMN performs LUs every time it visits a

different MR, the network always knows the exact location of the sMN, but this is not a power-saving solution. Hence, there are two main criteria to evaluate the efficiency of an LA design. The first is control overhead induced by the paging procedure until the requested sMN is found. The second is the amount of power consumed on the sMN by performing LUs. In addition, location management should not undermine the performance of existing active sessions in IiWMNs.

7.1.2 Motivation of New LA Design in IiWMNs

In order to illustrate the special design challenge in IiWMNs, OPNET [13] simulations are conducted to evaluate the network performance as the number of sMNs to be paged increases. First of all, the paging procedure in the mesh backbone can be designed utilizing the neighbor discovery protocol (NDP) [7] in the IPv6 protocol suite. The *neighbor solicitation* (NS) message can be modified as the paging request message containing the original address of an sMN. Since IPv6 has replaced broadcast with multicast, an NS message is destined to all MRs in a subnet with the address FF01::2 (IPv6 default all-router multicast address). Upon receiving an NS message, sMN's current MR (cMR) replies a *neighbor advertisement* (NA) message to the sender which is modified as the paging reply message.

Assume that a number of sMNs need to be paged scaling from 4 to 20 in the IiWMN. The uG (G2) of these sMNs is triggered to start the paging procedure within the mesh backbone. In addition, an end-to-end active session running between a CN and an aMN exists in the network. As shown in Fig. 7.2, the network performance becomes poor when the number of sMNs to be paged increases. The gap (dropped data packets) between the sent and received packets of the existing active session begins to increase when the number of sMNs is 12. However, this packet loss is not due to buffer overflow on MRs because the buffer overflow on the network occurs only when the number of sMNs reaches 20. The reason for this performance degradation of the existing active session, as shown in the dash rectangular in Fig. 7.2 when

the number of paged sMNs is 12 and 16, is "retry threshold exceeded" on the MAC-layer. The paging messages generated for multiple sMNs pass through those MRs with active traffic load, compete the scarce wireless resources with data packets, and thus cause the MAC-layer contentions. This issue is more severe in IiWMNs when the number of paged sMNs increases due to the existence of multihop wireless links in the mesh backbone. Therefore, the LA design in IiWMNs (which determines how large an area paging requests are broadcasted) should consider the support of scalability and exclude heavily loaded MRs with existing active sessions from participating in the paging process.

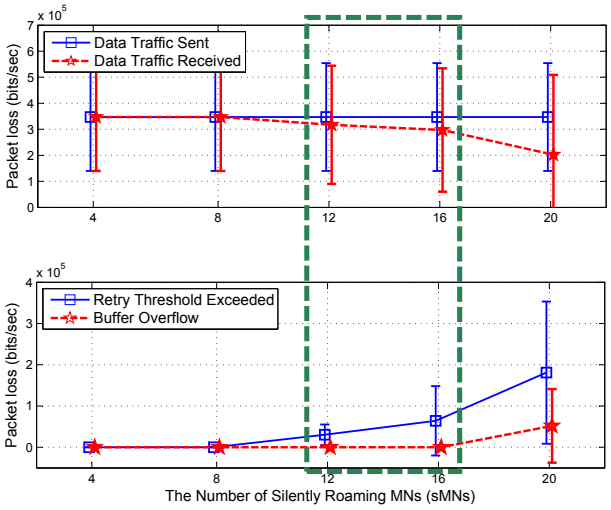


Figure 7.2: Poor scalability in an IiWMN when the number of paged MNs increases.

7.2 Proposed Resilient Location Area Design (*ReLoAD*) for IiWMNs

A resilient location area design (*ReLoAD*) is proposed which can facilitate scalable paging for both intranet and internet session paging requests in IiWMNs. The proposed resilient location area (*RLA*) can adapt to changes of both paging load (for sMNs) and active service load (for aMNs) on each MR/gateway in the mesh backbone.

7.2.1 Proposed RLA Formation

7.2.1.1 The Formation of RLA

Assume that there are N MNs residing in the network which consists of A aMNs and S sMNs, $N = A+S$. Generally, the number of aMNs is proportional to the traffic load on MRs/gateways, while the number of sMNs indicates both the potential paging and traffic load on the registered MR/gateway. Hence, the tuple (A, S) is a key piece of *user information* cached in each MR/gateway and needs to be exchanged in the mesh backbone when forming RLAs.

Firstly, in the design, each MR/gateway keeps a user database (MN_ID, MN_STATUS) associated with a countdown timer. The value of the timer is set to a predefined value MAX_TIME. The timer starts to count down when an MR/gateway receives a packet to or from an MN associated to this MR/gateway. The timer is reset to MAX_TIME if the MR/gateway receives another packet to or from the same MN before MAX_TIME decreases to zero. In this way, each MR/gateway is able to check the number of aMNs associated to this MR and the number of sMNs last updated its location through this MR following the conditions: 1) change the status of an sMN to 'aMN' if it receives a data packet to or from the sMN (packet interruption) and 2) change the status of an aMN to 'sMN' if the value of the aMN's timer decreases to zero (event interruption). The algorithm of updating MN status is depicted in Fig. 7.3 (Part I).

Secondly, assume that all MRs/gateways periodically check its (A, S). The MRs/-gateways with a large value of A is selected to be the location designated routers (*LDRs*) and these *LDRs* will form RLAs. In addition, in order to prevent performance degradation on aMNs while paging sMNs, the formed RLAs also need to vary based on conditions to control the potential high paging load caused by the increasing number of S in an RLA.

To form an RLA, an important performance factor affecting the scalability of WMNs, the number of hops between the *LDR* and its farthest members in the RLA,

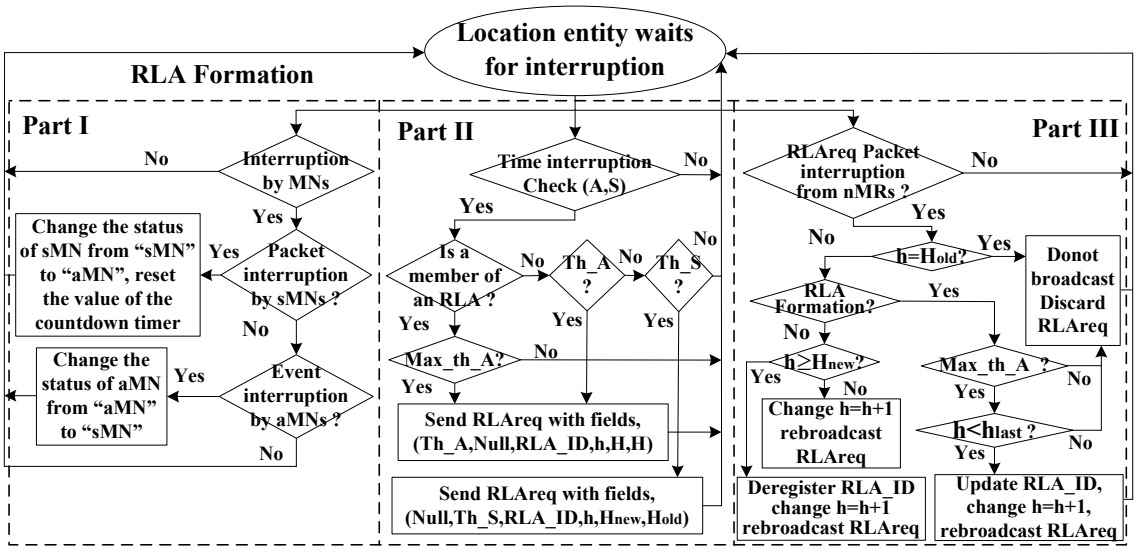


Figure 7.3: The proposed algorithms for RLA formulation.

should not exceed a hop threshold H. Assume that the default LA for each MR/-gateway is the size of the mesh backbone. During the time interruption as shown in Fig. 7.3 (Part II), if the number of A and S on an MR/gateway exceeds a threshold Th_A and Th_S, respectively, it starts to exchange *RLA request* (RLAreq) messages between neighboring MRs/gateways (nMRs/nGs). The general fields of an RLAreq include the information of (Th_A, Th_S, RLA_ID, h, H_{new}, H_{old}). Here, RLA_ID represents the ID of an RLA, while h stands for the number of hops the RLAreq has propagated. For the purpose of RLA formation, the fields of an RLAreq are (Th_A, Null, RLA_ID, h, H, H). For the purpose of RLA deregistration, the fields of an RLAreq can be (Null, Th_S, RLA_ID, h, H_{new}, H_{old}). Assume that the growth of S on the LDR reaches Th_S, it intends to reduce the potential paging overhead by adjusting the size of its current RLA (e.g., with H_{old} hops) to a smaller one (e.g., with H_{new} hops). Hence, MRs/gateways within H_{new} to H_{old} hops receiving the RLAreq message can deregister with this RLA_ID. In a special case, the RLA is deregistered when $H_{new} = 0$.

The process of processing the arriving RLAreq messages on the intermediate nM-R/nGs is shown in Fig. 7.3 (Part III). Assume that an MR_j receives an RLAreq

from MR_i. It first checks whether h reaches H_{old}. If not, it processes the message accordingly. If the incoming RLAreq message is for the RLA formation purpose, $Max\_th\_A_{ij}$ is introduced to depict the threshold Th_A which can be illustrated as $|(Th\_A_i - Th\_A_j)/Th\_A_j|$. When this value reaches $Max\_th\_A_{ij}$, nMR_j/nG_j needs to further check whether the value of h of this arriving RLAreq is lower than the one of the last accepted RLAreq. If yes, it updates its RLA_ID using MR_i's RLA_ID, increases h by 1, and rebroadcasts the RLAreq message. In this way, MR_j joins the RLA of MR_i. Otherwise, nMR_j/nG_j rejects to join the RLA of MR_i by discarding this RLAreq message. If the incoming RLAreq message is for the deregistration purpose and the value of h exceeds H_{new}, nMR_j/nG_j deregisters the RLA_ID and rebroadcasts the message.

7.2.1.2 RLA_ID: Multicast Addressing for Paging in an RLA

The paging procedure in the mesh backbone can be developed by utilizing the neighbor discovery protocol (NDP) [7] in the IPv6 protocol suite. The *neighbor solicitation* (NS) message can be modified as the paging request message containing the original address of an sMN. Since IPv6 has replaced broadcast with multicast, an NS message is destined to all MRs in a subnet with the address FF01::2 (IPv6 default all-router multicast address). Upon receiving an NS message, sMN's current MR (cMR) replies a *neighbor advertisement* (NA) message to the sender which is modified as the paging reply message. In the proposed *ReLoAD*, instead of sending paging messages to all MRs in the current subnet with a default all-router multicast address FF01::2 defined in IPv6, the flooding area of paging messages of an *LDR* is confined in its RLA. To achieve this, each MR/gateway formulates an RLA_ID during the network deployment phase, a source-based multicast address [81] for MRs within its RLA. Based on this, for instance, an MR/gateway with a unicast IPv6 prefix of 2001:0:1::/48 can also form a unicast prefix-based multicast prefix of FF3X:0030:0:2001:0:1::/96 (where 'x' is any valid scope), which is used as the RLA_ID for differentiating different

RLAs.

7.2.2 Proposed Location Update and Paging in ReLoAD

7.2.2.1 Location Area Information from the Formed RLA

In the proposal, *router advertisement* (RA) messages periodically broadcasted by MRs/gateways should include additional information, as shown in Fig. 7.4, to help sMNs to decide "*when and where an LU is needed*" so that the paging procedure does not undermine the QoS performance of existing traffic. When an sMN visits a cMR, it listens to the RA messages broadcasted on the link of the cMR. In Fig. 7.4, initially, there are two RLAs, *RLA_M1* and *RLA_M2* formed by MR_1 and MR_2, respectively. As an sMN roams to a cMR which is a member of the RLA, it can be aware of the RLA_ID based on the received RA message. Originally, MR_7 belongs to RLA_M2. Later, it becomes a heavily loaded MR and forms its own *RLA_M7* as shown in the shadow area. All RLAs are excluded from each other.

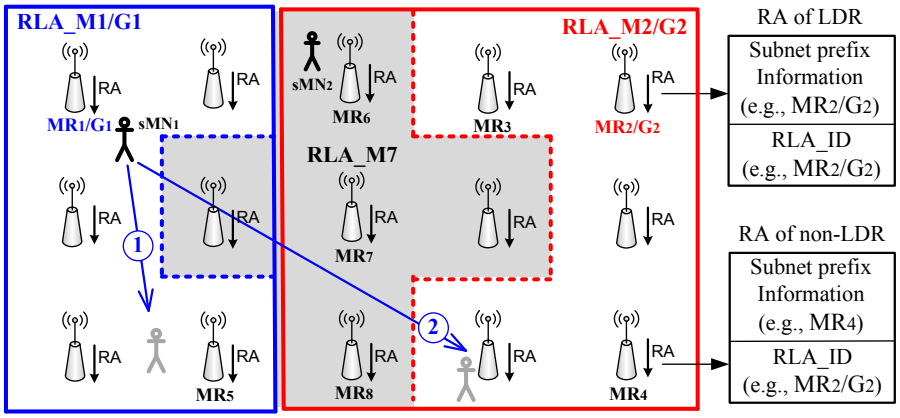


Figure 7.4: sMNs' location-aware movement in RLAs. Two types movement trajectories of an MN: ① Intra-RLA and ② Inter-RLA movement.

7.2.2.2 Mandatory LU When Changing RLAs

If the received RA messages indicate that an sMN has moved out of the current RLA, a mandatory LU to the HA is required. In Fig. 7.4, the sMN_1 originally resides under $RLA\_M1/G1$ and it needs to perform an LU to the HA when moving out of the RLA. Based on the formation of RLAs, there are two types of movement

trajectories, representing the intra-RLA (movement ①) and inter-RLA (movement ②). No LU is required during movement ① since the movement is within the same RLA. On the other hand, under the scenario that the change of RLA_ID caused by either sMN's inter-RLA movement (e.g., (movement ②)) or the change of RLAs (e.g., MR_6 the sMN_2 residing under migrates from *RLA_M2* to *RLA_M7*), the MN needs to perform an LU to the HA containing its cMR's address. The sMN also needs to send a deregistration message to the uMR.

7.2.2.3 Paging Procedure in ReLoAD

For the internet session paging request, the traffic for an sMN is intercepted by the HA, it obtains the address of the last updated uMR/uG of the sMN. Paging is triggered in the corresponding RLA which includes the uMR/uG. For the intranet session paging request, the traffic for an sMN is intercepted by the uMR/uG in which the sMN resided before changing from 'aMN' to 'sMN'. If the uMR/uG does not receive sMN's deregistration message, then it performs the paging procedure in this RLA. Otherwise, the sMN has changed to a new RLA. Its current uMR/uG receives the traffic from the sMN's previous uMR/uG and is triggered to perform the paging procedure in the current RLA.

7.2.3 Summary

In the proposed RLA design, choosing the heavily loaded MRs/gateways to form separate RLAs gives the following two benefits: 1) potential high paging signaling overhead induced by the paging procedure is confined in an RLA; and 2) since different RLAs exclude each other, the QoS performance of the ongoing active traffic sessions on those heavily loaded MRs is preserved. Moreover, the size of each RLA can be dynamically adjusted to balance paging overhead in the mesh backbone and sMNs' LU overhead. Hence, the proposed *ReLoAD* provides a scalable solution for location management under both intra- and internet sessions in IiWMNs in order to accommodate a large number of MNs, including the ones in the silently roaming

mode.

7.3 Performance Analysis

In this section, the performance of the proposed resilient location area design is evaluated using OPNET[13] simulations.

7.3.1 Simulation Scenario and Assumptions

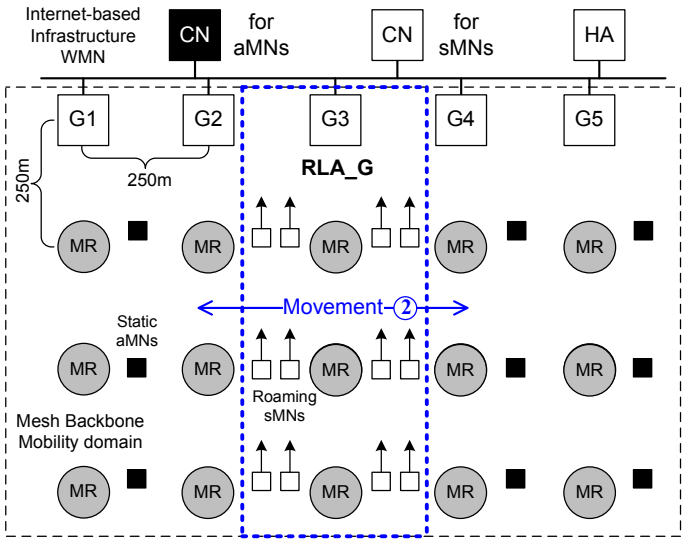


Figure 7.5: A simulation scenario in OPNET.

Fig. 7.5 shows an IiWMN scenario in OPNET with multiple gateways, MRs, MNs, CNs, and an HA. The gateways and MRs are deployed as a 4×5 grid in a $1250m \times 1000m$ area and their locations are fixed. Each MR is equipped with two network interfaces, one for mesh connectivity and the other for MN access. The two interfaces are configured to operate following IEEE 802.11b. The solid squares and the hollow squares with arrows stand for the static aMNs and roaming sMNs, respectively. To characterize sMNs' mobility, the *Random Waypoint Movement* model [79] is adopted in the simulation where an sMN picks a random destination and a random velocity. After reaching the destination, it pauses for a certain amount of time. This is repeated until the simulation ends. The movement of sMNs is restricted to the mobility area which is the same size as that of the mesh backbone. One RLA formed by a gateway (G3) is considered in this simulation scenario, the size of which

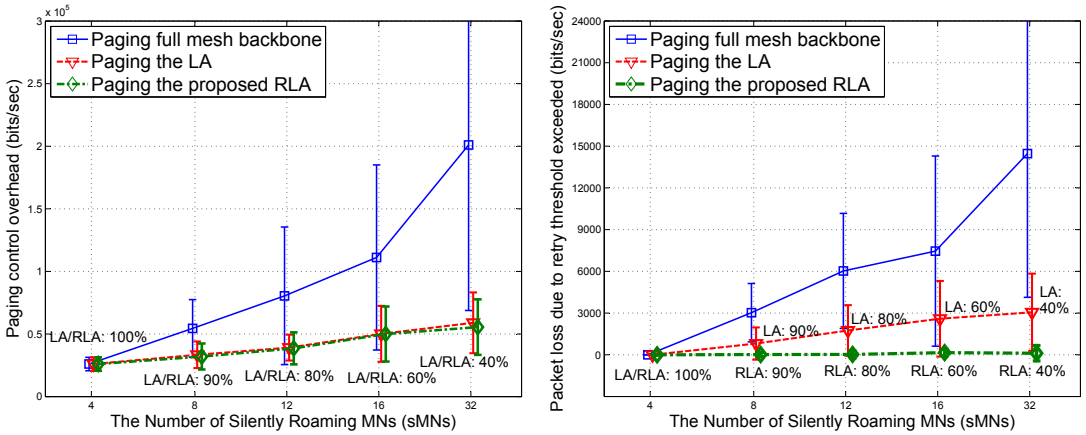
is 20% of the full mesh backbone (i.e., includes 20% of the total MRs). The mobility of sMNs causes an LU (movement ②) procedure when moving out the RLA (G3). A light video application is modeled as the existing active internet sessions between a CN (for aMNs) and static aMNs. The delay of packet traversal through the Internet is set to be constant 0.05 second. The detailed parameters used in the simulations are shown in Table 7.1.

Table 7.1: Simulation Parameters for *ReLoAD*

AP transmit power (W)	0.05
AP data rate (Mbps)	5.5
Packet reception-power threshold (dbm)	-95
AP beacon interval (sec)	0.02
Buffer size (bits)	256000
IPv6 interface routing protocol	RIPng
Multihop routing protocol	OSPFv3
IPv6 *Router Advertisement* interval (sec)	uniform (0.5, 1)
NDPv6 messages interval (sec)	uniform (1, 2)
OSPFv3 *HELLO* message interval (sec)	uniform (1, 1.1)
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
Light Video Application for aMNs	
Start time (sec)	60
Frame size (bytes)	172
Frame interarrival time (sec)	constant (0.5)
Internet Session Paging Request for sMNs	
Start time (sec)	10
Frame interarrival time (sec)	constant (10)

7.3.2 Results Analysis

Fig. 7.6 shows the performance of the paging procedure for sMNs initially residing under G3 under different LA schemes, namely, paging the full mesh backbone, paging the LA with adjustable size, and paging the RLA (heavily loaded MRs exclusion), as the number of paged sMNs increases from 4 to 32. As shown in Fig. 7.6(a), the paging procedure under the full mesh backbone causes much higher control overhead than

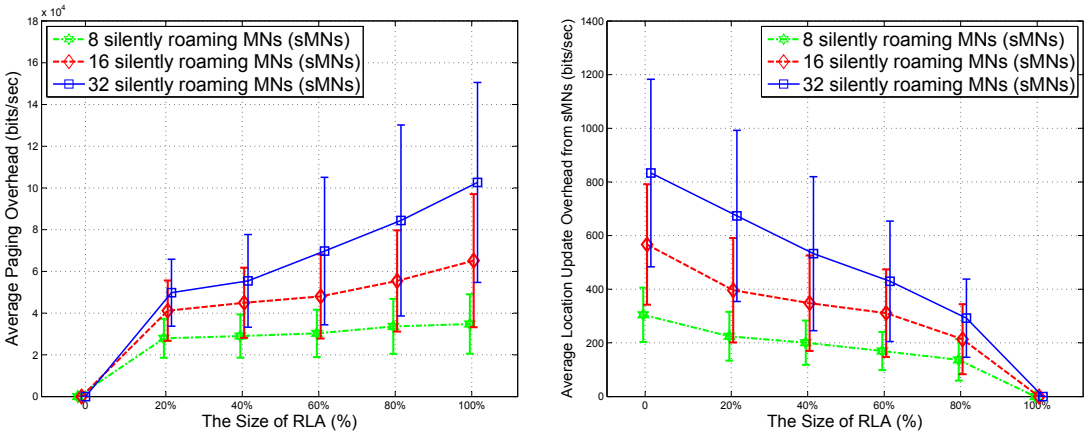


(a) Paging control overhead (b) Retry threshold exceeded

Figure 7.6: The performance of the paging procedure under different LA schemes.

the other two schemes which control the size of the LA/RLA to limit control overhead caused by the paging procedure. Paging control overhead under the LA/RLA scheme is more or less the same since paging messages are confined within the same size of LAs/RLAs. However, as shown in Fig. 7.6(b), the least data packet loss due to retry threshold exceeded of the existing active sessions can be achieved by the proposed RLA scheme since paging can exclude the heavily loaded MRs to avoid MAC contentions so that the QoS performance of the existing active video sessions on aMNs can be preserved.

Fig. 7.7(a) shows that the control overhead of paging in the formed RLA of G3 increases when its size increases from 20% to 100% of the full mesh backbone and the number of sMNs to be paged increases from 8 to 32. On the contrary, the corresponding LU overhead on the sMN side decreases, as shown in Fig. 7.7(b). Based on these two figures, the tradeoff between control overhead caused by the paging procedure and LU overhead can be seen on the sMN side. Correspondingly, Fig. 7.8 shows the performance of existing active internet sessions running on aMNs. The service level agreement (SLA) is employed as the criterion for evaluating the performance conformance for existing active internet sessions. Under the definition



(a) Paging control overhead (b) LU control overhead

Figure 7.7: Control overhead caused by the paging procedure in an RLA and LU overhead on the sMN side.

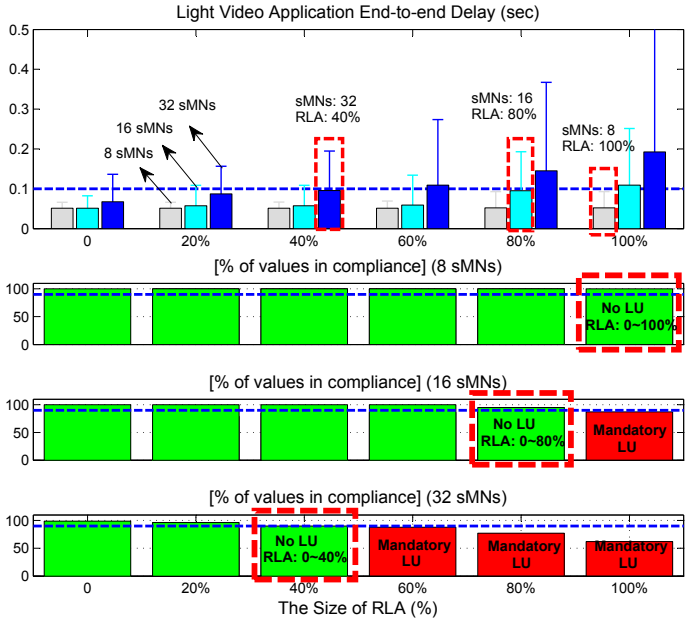


Figure 7.8: The QoS performance of existing active sessions of aMNs.

of the SLA, the performance is in compliance with the SLA if the end-to-end (ETE) delay of an active session is below 0.1 second 90 percent of one simulation time. This means that an SLA violation will be shown if the ETE delay is above 0.1 second more than 10 percent of each simulation time. A green bar indicates conformance to the defined SLA and a red bar shows a violation. The dash rectangle stands for the

largest possible size of an RLA under which the paging traffic is still tolerable by the existing active internet sessions. Consider the case when the number of sMNs to be paged is 8. The size of an RLA can cover the full mesh backbone without undermining the SLA of the existing active internet sessions, thus power consumption caused by performing LUs from sMNs can be minimized. However, under the cases when the number of sMNs to be paged in an RLA is 16 and 32, control overhead of the paging procedure in the RLA causes the performance degradation on the existing active internet sessions of aMNs when the size of an RLA reaches 100% and 60%∼100% of the full mesh backbone, respectively. Moreover, under these two cases, no LUs are needed for sMNs when the size of an RLA is less than or equal to 80% and 40%, respectively.

7.4 Conclusion

In this chapter, a resilient location area design *(ReLoAD)* is proposed to facilitate scalable location management in IiWMNs. Under the proposed design, LAs can adapt to the changes of both paging and service load in the wireless mesh backbone. Hence, the formed *RLA* can achieve a reasonable tradeoff between signaling overhead caused by the paging procedure and MN power consumption caused by the LU procedure for both intra- and internet sessions, while simultaneously preserve the QoS performance of existing traffic of active MNs. Through OPNET simulation study and analysis, the proposed *ReLoAD* offers important design guidelines is demonstrated for location management in IiWMNs in order to accommodate a large number of MNs, including the ones in the silently roaming mode.

CHAPTER 8: CONCLUSION

8.1 Completed Work

The following research work has been completed:

- A novel architectural design was proposed to facilitate the L3 handoff detection and cross-layer handoffs [70]. The proposed architecture relieves the restrictions that all handoff steps have to be executed one by one sequentially. Instead, some handoff steps can be executed parallelly. Secondly, a novel caching scheme was proposed [82] that allows data packets to be cached in a small group of candidate APs (cAPs) in advance to guarantee minimum packet loss during an inter-gateway handoff. By doing so, once an MN is associated to a new AP, it can continue the L3 and L5 handoff process via the new gateway and meanwhile, keep the packet reception from the old gateway via special mesh routers. In addition, the required number and optimal placement of special mesh routers that form the proposed IMeX architecture are modeled as a set covering problem which is solved based on a greedy algorithm. Finally, a QoS-handoff mechanism was developed [83] based the proposed IMeX. OPNET simulations were conducted to evaluate the performance of the proposed designs of handoff management.
- A dynamic location management (DoMaIN) framework in Internet-based infrastructure WMNs was proposed, which addresses the new location management challenges in Internet-based WMNs. In addition, the proposed DoMaIN framework facilitates a new dynamic location update triggering method which is suitable for the multihop wireless mesh backbone. On the other hand, a resilient location area design (ReLoAD) was proposed [84] which can achieve a

balanced tradeoff between signaling overhead caused by the paging procedure and MN power consumption caused by the LU procedure for both intra- and internet sessions, while simultaneously preserve the QoS performance of existing traffic of active MNs. Comprehensive OPNET simulations were conducted to study the performance of the proposed location management designs.

8.2 Future Work

Based on the contributions of this thesis, some suggestions for future work are given below:

- Based on the proposed IMeX handoff architecture, future work such as efficient multihop routing and MAC protocol design can further reduce handoff delays in order to support end-to-end real time applications in the Internet-based wireless mesh backbone.
- Based on the proposed DoMaIN framework, a hybrid movement- and hop-based LU triggering method can be implemented in Internet-based infrastructure WMNs to provide a desirable tradeoff between signaling overhead caused by the paging procedure and MN power consumption caused by the LU procedure.

8.3 Published and Submitted Work

Weiyi Zhao and Jiang Xie. Inter-Gateway Cross-layer Handoffs in Wireless Mesh Networks. Proceedings of IEEE GLOBECOM 2009, December 2009.

Weiyi Zhao and Jiang Xie. A Novel Xcast-based Caching Architecture for Inter-gateway Handoffs in Infrastructure Wireless Mesh Networks. Proceedings of IEEE INFOCOM 2010, March 2010.

Weiyi Zhao and Jiang Xie. Network engineering and traffic forwarding (NETF): An Integrated Design for Inter-gateway QoS Handoffs in Infrastructure Wireless Mesh Networks. Proceedings of IEEE GLOBECOM 2010, December 2010.

Weiyi Zhao and Jiang Xie. Hatch: The Design of a Hybrid Location Tracking Chain in Internet-based Wireless Mesh Networks. to appear in Proceedings of IEEE GLOBECOM 2011, December 2011.

Weiyi Zhao and Jiang Xie. ReLoAD: A Resilient Location Area Design for Internet-based Infrastructure Wireless Mesh Networks. to appear in Proceedings of IEEE GLOBECOM 2011, December 2011.

Weiyi Zhao and Jiang Xie. OPNET-based Modeling and Simulation Study on Handoffs in Internetbased Infrastructure Wireless Mesh Networks. Computer Networks (Elsevier), vol.55, no.12, pp.2675-2688, August 2011.

Weiyi Zhao and Jiang Xie. IMeX: Inter-gateway Cross-layer Handoffs in Internet-based Infrastructure Wireless Mesh Networks. Accepted for Publication in IEEE Transactions on Mobile Computing, August 2011.

Weiyi Zhao and Jiang Xie. DoMaIN: A Novel Dynamic Location Management Solution for Internet-based Infrastructure Wireless Mesh Networks, August 2011. Ready for Submission.

REFERENCES

[1] I. F. Akyildiz, X. Wang, and W. Wang, "Wireless mesh networks: a survey," *Computer Networks (Elsevier)*, vol. 47, no. 4, pp. 445–487, March 2005.

[2] R. Bruno, M. Conti, and E. Gregori, "Mesh networks: commodity multihop ad hoc networks," *IEEE Communications Magazine*, vol. 43, no. 9, pp. 23–30, Sept 2005.

[3] C. E. Perkins, "IP mobility support for IPv4," RFC 3220, IETF, Januray 2001.

[4] D. Johnson and C. Perkins, "Mobility support in IPv6," RFC 3775, IETF, Januray 2004.

[5] R. Koodli, "Fast handovers for Mobile IPv6," RFC 4068, IETF, July 2005.

[6] H. Soliman, C. Castelluccia, K. E. Malki, and L. Bellier, "Hierarchical Mobile IPv6 mobility management (HMIPv6)," RFC 4140, IETF, Auguest 2005.

[7] T. Narten, E. Nordmark, W. Simpson, and H. Soliman, "Neighbor discovery for IP version 6 (IPv6)," RFC 4861, IETF, Januray 2007.

[8] K. Wong, A. Dutta, H. Schulzrinne, and K. Young, "Simultaneous mobility: analytical framework, theorems and solutions," *Wireless Communications & Mobile Computing*, vol. 7, no. 5, pp. 623–642, June 2007.

[9] H. Jung, E. Kim, J. Yi, and H. Lee, "A scheme for supporting fast handover in hierarchical mobile IPv6 networks," *Electronics and Telecommunications Research Institute (ETRI) Journal*, vol. 27, no. 6, pp. 798–801, 2005.

[10] M. Gerla, K. Tang, and R. Bagrodia, "TCP performance in wireless multi-hop networks," in *Proc. IEEE Workshop on Mobile Computer Systems and Applications WMCSA 1999*.

[11] G. Holland and N. Vaidya, "Analysis of TCP performance over mobile ad hoc networks," in *Proc. MOBICOM 1999*.

[12] J. Camp, V. Mancuso, O. Gurewitz, and E. Knightly, "A measurement study of multiplicative overhead effects in wireless networks," in *Proc. IEEE INFOCOM*, 2008, pp. 511–519.

[13] OPNET Technologies, Inc., http://www.opnet.com/.

[14] W. Kim, M. Kim, K. Lee, C. Yu, , and B. Lee, "Link layer assisted mobility support using SIP for real-time multimedia communications," in *Proc. MOBIWAC 2004*, 2004.

[15] S. Pack, J. Choi, T. Kwon, and Y. Choi, "Fast-handoff support in IEEE 802.11 wireless networks," *IEEE Communications Survey & Tutorials*, vol. 9, no. 1, pp. 2–12, 1st Quarter 2007.

[16] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Perterson, R. Sparks, M. Handley, and E.Schooler, "SIP: Session initiation protocol," RFC 3261, IETF, June 2002.

[17] "Feasibility study on 3GPP system to WLAN interworking," 3GPP TR 22.934 v1.2.0, May 2002.

[18] A. Mishra, M. Shin, and W. Arbaugh, "An empirical analysis of the IEEE 802.11 MAC layer handoff process," *ACM SIGCOMM Computer Communication Review*, vol. 33, no. 2, pp. 93–102, April 2004.

[19] M. Shin, A. Mishra, , and W. Arbaugh, "Improving the latency of 802.11 handoffs using neighbor graphs," in *Proc. ACM International Conference on MobiSys*, June 2004, pp. 70–83.

[20] S. Shin, A. G. Forte, A. S. Rawat, and H. Schulzrinne, "Reducing MAC layer handoff latency in IEEE 802.11 wireless LANs," in *Proc. ACM International Workshop on Mobility Management and Wireless Access Protocols (MobiWac)*, September 2004, pp. 19–26.

[21] I. Ramani and S. Savage, "Syncscan: Practical fast handoff for 802.11 infrastructure networks," in *Proc. IEEE INFOCOM*, 2005.

[22] H. Wu, K. Tan, Y. Zhang, and Q. Zhang, "Proactive scan: fast handoff with smart triggers for 802.11 wireless LAN," in *Proc. IEEE INFOCOM*, 2007.

[23] V. Brik, V. Mishra, and S. Banerjee, "Eliminating handoff latencies in 802.11 WLANs using multiple radios: applications, experience, and evaluation," in *Proc. ACM Internet Measurement Conference*, 2005, pp. 299–304.

[24] N. Montavont and T. Noel, "Handover management for mobile nodes in IPv6 networks," *IEEE Communications Magazine*, vol. 40, no. 8, pp. 38–43, August 2002.

[25] A. Misra, S. Das, A. Dutta, A. McAuley, and S. Das, "IDMP-based fast handoffs and paging in ip-based 4g mobile networks," *IEEE Communications Magazine*, vol. 40, no. 3, pp. 138–145, 2002.

[26] 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.

[27] A. T. Campbell, J. Gomez, S. Kim, A. G. Valko, C.-Y. Wang, and Z. R. Turanyi, "Design, implementation, and evaluation of cellular IP," *IEEE Personal Communications Magazine*, pp. 42–49, August 2000.

[28] R. Ramjee, K. Varadhan, L. Salgarelli, S. R. Thuel, S.-Y. Wang, and T. LaPorta, "HAWAII: A domain-based approach for supporting mobility in wide-area wireless networks," *IEEE/ACM Transactions on Networking (TON)*, vol. 10, no. 3, pp. 396–410, June 2002.

[29] P. Eronen, "IKEv2 mobility and multihoming protocol (MOBIKE)," RFC 4555, IETF, June 2006.

[30] M. Buddhikot, A. Hari, K. Singh, and S. Miller, "MobileNAT: A new technique for mobility across heterogeneous address spaces," *ACM Mobile Networks and Applications*, vol. 10, no. 3, pp. 289–302, June 2005.

[31] K. N. Ramachandran, M. M. Buddhikot, G. Chandranmenon, S. Miller, E. M. Belding-Royer, and K. C. Almeroth, "On the design and implementation of infrastructure mesh networks," in *Proc. First IEEE Workshop on Wireless Mesh Networks*, August 2005.

[32] S. Speicher and C. H. Cap, "Fast layer 3 handoffs in AODV-based IEEE 802.11 wireless mesh networks," in *Proc. Third International Symposium on Wireless Communication Systems (ISWCS)*, 2006, pp. 233–237.

[33] S. Speicher, "OLSR-FastSync: fast post-handoff route discovery in wireless mesh networks," in *Proc. IEEE Vehicular Technology Conference (VTC06-Fall)*, 2006, pp. 1–5.

[34] R. Huang, C. Zhang, and Y. Fang, "A mobility management scheme for wireless mesh networks," in *Proc. IEEE Global Telecommunications Conference (GLOBECOM)*, November 2007, pp. 5092–5096.

[35] V. Navda, A. Kashyap, and S. R. Das, "Design and evaluation of iMesh: an infrastructure-mode wireless mesh network," in *Proc. Sixth IEEE WoWMoM*, June 2005, pp. 164–170.

[36] Y. Amir, C. Danilov, M. Hilsdale, R. Musaloui-Elefteri, and N. Rivera, "Fast handoff for seamless wireless mesh networks," in *Proc. ACM International Conference on MobiSys*, 2006, pp. 83–95.

[37] A. Mishra, M. Shin, and W. Arbaugh, "Context caching using neighbor graphs for fast handoffs in a wireless network," in *Proc. IEEE INFOCOM*, vol. 1, March 2004, pp. 351–361.

[38] D. Huang, P. Lin, and C. Gan, "Design and performance study for a mobility management mechanism (WMM) using location cache for wireless mesh networks," *IEEE Trans. Mobile Computing*, vol. 7, no. 5, pp. 546–556, May 2008.

[39] S. Lakshmanan, K. Sundaresan, and R. Sivakumar, "On multi-gateway association in wireless mesh networks," in *Proc. 2nd IEEE Workshop on Wireless Mesh Networks (WiMesh)*, 2006, pp. 64–73.

[40] D. Nandiraju, L. Santhanam, N. Nandiraju, and D. P. Agrawal, "Achieving load balancing in wireless mesh networks through multiple gateways," in *Proc. IEEE MASS*, October 2006.

[41] F. Li, Y. Wang, X. Y. Li, and A. Nusairat, "Gateway placement for throughput optimization in wireless mesh networks," *ACM MONET Special Issue on Advances in Wireless Mesh Networks*, pp. 198–211, March 2008.

[42] Q. Xue and A. Ganz, "QoS routing for mesh-based wireless LANs," *Kluwer International Journal of Wireless Information Networks*, vol. 9, no. 3, pp. 179–190, 2002.

[43] V. Kone, S. Das, B. Zhao, and H. Zheng, "QUORUM - Quality of Service routing in wireless mesh networks," *Mobile Networks and Applications*, vol. 12, no. 5-6, pp. 358–369, December 2007.

[44] J. Tang, X. Guo, and W. Zhang, "Interference-aware topology control and qos routing in multi-channel wireless mesh networks," in *Proc.ACM MobiHoc 2005.*

[45] W. Ma and Y. Fang, "A pointer forwarding based local anchoring (POFLA) scheme for wireless networks," *IEEE Trans. Vehicular Technology*, vol. 54, no. 3, pp. 1135–1146, May 2005.

[46] J. Xie and I. F. Akyildiz, "A distributed dynamic regional location management scheme for Mobile IP," in *Proc. IEEE INFOCOM*, vol. 2, 2002, pp. 1069–1078.

[47] 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.

[48] I. F. Akyildiz, J. S. M. Ho, and Y. B. Lin, "Movement-based location update and selective paging for PCS networks," *IEEE/ACM Transactions on Networking (TON)*, vol. 4, no. 4, pp. 629–638, August 1996.

[49] Y. Fang, "Movement-based mobility management and trade off analysis for wireless mobile networks," *IEEE Transactions on Computers*, vol. 52, no. 6, pp. 791–803, June 2003.

[50] C. K. Ng and H. W. Chan, "Enhanced distance-based location management of mobile communication systems using a cell coordinates approach," *IEEE Transactions on Mobile Computing*, vol. 4, no. 1, pp. 41–55, 2005.

[51] G. Y. Lee, Y. Lee, and Z. J. Haas, "Hybrid location-update scheme for mobile networks," *IEEE Transactions on Vehicular Technology*, vol. 58, no. 1, pp. 338–348, 2009.

[52] A. Bar-Noy, I. Kessler, and M. Sidi, "Mobile users: To update or not to update ?" *ACM/Baltzer J. Wireless Networks,*, vol. 1, no. 2, pp. 175–195, July 1995.

[53] S. Pack, B. Lee, T. Kwon, and Y. Choi, "A pointer forwarding scheme with mobility-aware binding update in Mobile IPv6 networks," *Computer Communications*, vol. 31, no. 5, pp. 873–884, 2008.

[54] S. M. Das, H. Pucha, and Y. C. Hu, "Performance comparison of scalable location services for geographic ad hoc routing," in *Proc. IEEE INFOCOM*, vol. 2, March 2005, pp. 1228–1239.

[55] The Network Simulator version 2., http://www.isi.edu/nsnam/ns/.

[56] M. Abolhasan, T. Wysocki, and E. Dutkiewicz, "A review of routing protocols for mobile ad hoc networks," *Ad Hoc Networks (Elsevier)*, vol. 2, no. 1, pp. 1–22, January 2004.

[57] R. Draves, J. Padhye, and B. Zill, "Routing in multi-radio, multi-hop wireless mesh networks," in *Proc. ACM MobiCom*, September 2004, pp. 114–128.

[58] Y. Yang, J. Wang, and R. Kravets, "Designing routing metrics for mesh networks," in *Proc. First IEEE Workshop on Wireless Mesh Networks (WIMESH)*, September 2005.

[59] C. Perkins, E. Belding-Royer, and S. Das, "Ad hoc on-demand distance vector (AODV) routing," RFC 3561, IETF, July 2003.

[60] G. Athanasiou, T. Korakis, and L. Tassiulas, "A 802.11k compliant framework for cooperative handoff in wireless networks," *EURASIP Journal on Wireless Communications and Networking Volume*, 2009.

[61] T. Clausen and P. Jacquet, "Optimized link state routing protocol (OLSR)," RFC 3626, IETF, October 2003.

[62] W. Fenner, "Internet group management protocol (Version 2)," Request for Comments (RFC) 2362, Internet Engineering Task Force (IETF), November 1997.

[63] D. Estrin, D. Farinacci, and A. Helmy, "Protocol independent multicast-sparse mode (PIM-SM)," Request for Comments (RFC) 2362, Internet Engineering Task Force (IETF), Januray 1998.

[64] D. Waitzman, C. Partridge, and S. Deering, "Distance vector multicast routing protocol," Request for Comments (RFC) 1075, Internet Engineering Task Force (IETF), Januray 1988.

[65] R. Boivie and N. Feldman, "Explicit multicast (Xcast) concepts and options," Request for Comments (RFC) 5058, Internet Engineering Task Force (IETF), November 2007.

[66] L. Ji and M. Corson, "Differential destination multicast - a MANET multicast routing protocol for small groups," in *Proc. IEEE INFOCOM*, vol. 2, 2001, pp. 1192–1201.

[67] K. Chen and K. Nahrstedt, "Effective location-guided tree construction algorithms for small group multicast in MANET," in *Proc. IEEE INFOCOM*, vol. 3, 2002, pp. 1180–1189.

[68] R. Braden, D. Clark, and S. Shenker, "Integrated services in the Internet architecture: an Overview," Request for Comments (RFC) 1633, Internet Engineering Task Force (IETF), June 1994.

[69] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss, "An architecture for differentiated services," Request for Comments (RFC) 2475, Internet Engineering Task Force (IETF), Necember 1998.

[70] W. Zhao and J. Xie, "Inter-gateway cross-layer handoffs in wireless mesh networks," in *Proc. IEEE Global Telecommunications Conference (GLOBECOM)*, November 2009.

[71] A. Conta, S. Deering, and M. Gupta, "Internet control message protocol ICMPv6 for the internet protocol version 6 (IPv6) specification," RFC 4443, IETF, March 2006.

[72] T. Park and K. G. Shin, "Optimal tradeoffs for location-based routing in large-scale Ad Hoc networks," *IEEE/ACM Transactions on Networking (TON)*, vol. 13, no. 2, pp. 398–410, April 2005.

[73] Z. Ye and A. A. Abouzeid, "Optimal stochastic location updates in mobile Ad Hoc networks," *IEEE Transactions on Mobile Computing*, vol. 10, no. 5, pp. 638–652, May 2011.

[74] J. Xie and X. Wang, "A survey of mobility management in hybrid wireless mesh networks," *IEEE Network*, vol. 22, no. 6, pp. 34–40, 2008.

[75] W. Zhao and J. Xie, "Hatch: The design of a hybrid location trackin chain in Internet-based wireless mesh networks," in *Proc. IEEE Global Telecommunications Conference (GLOBECOM)*, December 2011.

[76] R. Wakikawa, J. Malinen, C. Perkins, A. Nilsson, and A. Tuominen, "Global connectivity for IPv6 mobile ad hoc networks," draft-wakikawa-manet-globalv6-05.txt, Internet Engineering Task Force (IETF), March 2006.

[77] S. Singh, J. Kim, Y. Choi, K. Kang, and Y. Roh, "Mobile multi-gateway support for IPv6 mobile ad hoc networks," draft-singh-manet-mmg-00.txt, Internet Engineering Task Force (IETF), June 2004.

[78] L. Chen and W. Heinzelman, "Qos-aware routing based on bandwidth estimation for mobile ad hoc networks," *IEEE Journal on Selected Areas in Communications (JSAC)*, vol. 23, no. 3, pp. 561–572, 2005.

[79] E. Hyytia, P. Lassila, and J. Virtamo, "Spatial node distribution of the random waypoint mobility model with applications," *IEEE Trans. Mobile Computing*, vol. 5, no. 6, pp. 680–694, June 2006.

[80] I. F. Akyildiz, J. McNair, J. S. M. Ho, H. Uzunalioglu, and W. Wang, "Mobility management for next generation wireless systems," *Proceedings of the IEEE*, vol. 87, no. 8, pp. 1347–1384, August 1999.

[81] B. Haberman and D. Thaler, "Unicast-prefix-based IPv6 multicast addresses," Request for Comments (RFC) 3306, Internet Engineering Task Force (IETF), August 2002.

[82] 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.

[83] W. Zhao and J. Xie, "Network engineering and traffic forwarding (NETF): An integrated design for inter-gateway QoS handoffs in infrastructure wireless mesh networks," in *Proc. IEEE GLOBECOM*, December 2010.

[84] W. Zhao and J. Xie, "ReLoAD: Resilient location area design for Internet-based infrastructure wireless mesh networks," in *Proc. IEEE Global Telecommunications Conference (GLOBECOM)*, December 2011.