WMM: A Mobility Management Mechanism using Location Cache for Wireless Mesh Networks

Di-Wei Huang

(*draft*)

Contents

List of Figures

List of Tables

Acknowledgement

I would like to express my sincere thanks to my advisor, Prof. Phone Lin and Dr. Chai-Hien Gan. Without their supervision and perspicacious advices, I can not complete this dissertation. Special thanks to my committee members, Prof. Yi-Bing Lin, Dr. Kwang-Yao Chang, and Dr. Jeu-Yih Jeng for their valuable comments and suggestions. Thanks also to the colleagues in the Mobile Communication Networking Lab., especially Shin-Ming Cheng, Yen-Cheng Lai, Hong-Yueh Chen, and Lu-Tsung Chang. Thanks to Chi-Chang Hsieh and Yen-Ming Chen for their help to implement the prototype system. Finally, I am grateful to my family for their encouragement and continuous support during these two years.

Abstract

Wireless Mesh Network (WMN) is one of the major technologies for the 4G high-speed mobile network. In the WMN, a mesh backhaul connects the WMN with Internet, and *mesh access points* (MAPs) provide wireless network access service to *mobile stations* (MSs). The *mesh nodes* (MNs) (including mesh backhaul and MAPs) are stationary and connected through the wireless mesh links. Due to MS mobility in WMN, Mobility Management (MM) is required to efficiently and correctly route the packets to MSs in WMN. We propose an MM mechanism named *Wireless mesh Mobility Management* (WMM). The WMM adopts the location cache approach where MNs cache the MS's location information while routing the data for the MS. The MM is exercised when MNs routes the packets. We implement the WMM and conduct an analytical model and simulation experiments to investigate the performance of WMM. We compare the signaling and routing cost between WMM and other existing MM protocols. Our study shows that WMM has light overhead and low implementation cost.

Keywords: Location Cache; Mobility Management; Wireless Mesh Network

Chapter 1

Introduction

Wireless Mesh Network (WMN) [5][7][8] is one of the major technologies for the 4G high-speed mobile networks. The WMN provides an ubiquitous solution for wireless Internet access with low deployment cost. Figure 1.1 illustrates a general WMN architecture that comprises two kinds of fixed *mesh nodes* (MNs). The mesh backhaul is a gateway between the WMN and Internet, through which all packets are delivered between the WMN and the Internet. The *mesh access point* (MAP) provides network access service to the *mobile stations* (MSs) through the wireless access links. A wireless mesh link exists between two MNs that are located within the each other's radio coverage area. The MN location is stationary.

When an MS enters the coverage area of an MAP, the MS performs the association procedure to establish a wireless access link to the MAP [7]. This MAP is known as the *serving MAP* (SMAP) of the MS. Before delivering the user data to an MS, WMN identifies

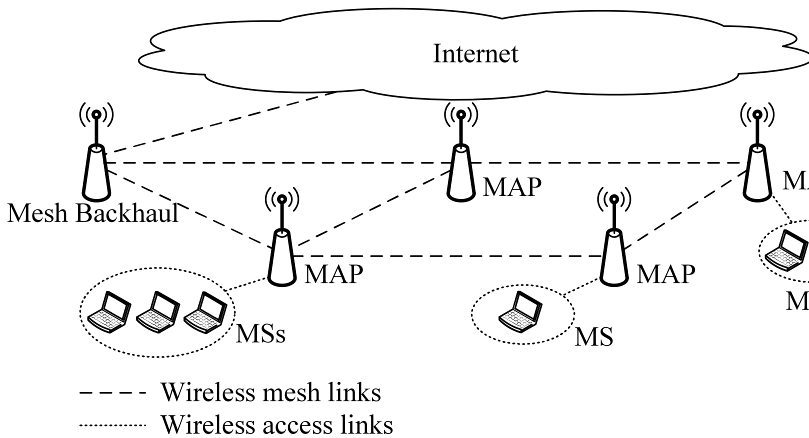


Figure 1.1: The mesh network architecture

the SMAP of this MS. Then the user data is sent to this SMAP through one or more MNs via the wireless mesh links. These MNs are known as the *relaying MAPs* (RMAPs). Since an MS may change the SMAP from time to time, *mobility management* (MM) is required for packet delivery to the moving MSs. Existing standards (such as IEEE 802.11 [7] and IEEE 802.16 [8]) for the WMN do not address the MM issue. The MM consists of *location management* and *handoff management*. With location management, WMN maintains the location information of the current SMAP for an MS. When an MS changes its SMAP, the WMN updates the SMAP information for the MS. During data transmission, if the MS changes from the old SMAP to the new SMAP, handoff management enables the old SMAP to forward user data to the new SMAP.

Existing MM protocols for mobile networks are divided into three categories, including the ad-hoc routing protocol [19], the centralized-database MM protocol [16], and the mobile IP protocol [18]. The ad-hoc routing protocol is adopted in the *mobile ad-hoc network* (MANET) where the user data is relayed hop-by-hop by MSs, and a routing path from the source to the destination is established for routing user data. Unlike MANET, the infrastructure of WMN is fixed (i.e., MNs are stationary), and the heavy-duty ad-hoc routing protocol is not needed in the WMN. The centralized-database MM protocol (where a centralized database is maintained to store MS location information) is usually adopted in the cellular network. The service area of a cellular network is partitioned into several *location areas* (LAs). Whenever an MS moves from an LA to another, the database is accessed to update MS location information. When the size of an LA is small, high signaling cost is expected. In WMN, the service area of an MAP is small. Therefore, the centralized-database MM protocol in WMN may not be a good solution for WMN. The mobile IP protocol partitions the service area of an IP network into the home network and foreign networks. Two network entities, *home agent* (HA) in the home network and *foreign agent* (FA) in the foreign network, are responsible to tunnel user data to MSs. Similar to the centralized-database protocol, the mobile IP protocol introduces signaling overhead to inform the HA of the MS's movement. Furthermore, tunneling between the home network and the foreign network lengthens the routing path, which is known as the triangle routing problem [18]. Hence, mobile IP protocol may not be an efficient solution for WMN.

In this paper, we propose a novel MM mechanism named *Wireless mesh Mobility Man-*

agement (WMM) for WMN. WMM adopts the location cache approach where the MNs cache the IP address of MS's SMAP (known as MS's location information) while routing the data for the MS. The MS's location information is distributed in the MNs that have routed the packets for the MS. The MM is exercised when MNs route the packets.

The rest of the paper is organized as follows. Chapter 2 details the WMM mechanism. Chapter 3 describes our implementation of the WMM mechanism. Chapter 4 proposes an analytical model and simulation experiments to study the performance of WMM. Chapter 5 compares the signaling and routing cost of WMM and other existing MM protocols. Chapter 6 concludes this paper.

Chapter 2

The WMM Mechanism

In the WMM mechanism, an MN maintains two cache tables, the *routing table* and the *proxy table*. The routing table is used to maintain the routing paths between the MN and other MNs. The proxy table maintains the MS location information. In WMM, when an MS enters the WMN or moves from one SMAP to another MAP, the MS registers to the new SMAP. The MS location information is carried in the packet headers. When MNs routes packets for an MS, the location information of the MS in proxy tables in the MNs are updated. Then the MN can correctly route the packets for MSs by referencing the proxy table and routing table. If the mesh backhaul does not cache MS location information when processing packet routing, a query procedure is executed to obtain the MS location information (to be elaborated in Chapter 2.3).

Several routing table maintenance protocols have been proposed in Internet or ad-hoc networks [19]. These protocols can be applied in WMM directly. In this paper, we focus on

Im Field	Is Field	Ts Field
MS's IP address	IP address of MS's SMAP	The time when the MS is associated with its SMAP

Figure 2.1: An entry in the proxy table

the proxy table maintenance for the MNs. As shown in Figure 2.1, every MN maintains an entry in the proxy table for MS, which consists of three fields: the Im field (to store MS's IP address), the Is field (to store the IP address of MS's SMAP), and the Ts field (to store the time when the MS is associated with its SMAP; also known as the "*serving timestamp*"). The serving timestamp can be obtained from the MS to ensure the non-decreasing property of the serving timestamp for the MS. We assume that all IP addresses assigned to MSs in the same WMN have the same prefix, and we can identify the WMN where the MS resides by checking the prefix of the MS's IP address.

We utilize the *options* field in the IP header to store the MS location information, including the IP address of MS's SMAP and MS's serving timestamp. The options field is filled or modified by MNs when they route the packets for an MS. The options field (consisting of 16 bytes) is divided into four subfields: the ISS field (to store the IP address of the sender's SMAP), the SST field (to store the sender's serving timestamp), the IRS field (to store the IP address of the receiver's SMAP), and the RST field (to store the receiver's serving timestamp). There are three WMM procedures: the registration procedure, the routing procedure, and the query procedure. Details of these procedures are described in the following sections.

2.1 The Registration Procedure

The registration procedure is executed to register an MS to its SMAP. Suppose that MS_1 moves from its SMAP MAP_1 to another MAP MAP_2. Figure 2.2 illustrates the message flow for this procedure with the following steps.

Step RE1. MS_1 sends a registration request message, REREQ(MS_1's IP Address, Previous SMAP's IP Address, Selected SMAP's IP Address), to MAP_2. The previous SMAP's IP address is set to `null` if the previous SMAP is unavailable. In this example, the previous SMAP is MAP_1.

Step RE2. Upon receipt of REREQ at t_1, MAP_2 first checks whether an entry for MS_1 exists in its proxy table. If the entry exists, MAP_2 updates the entry. Otherwise, MAP_2 creates an new entry for MS_1. MS_1's entry in MAP_2's proxy table is updated as: Im is set to MS_1's IP Address; Is is set to MAP_2's IP Address; Ts is set to t_1. Then MAP_2 checks the previous SMAP's IP address carried in REREQ. If it is `null` (i.e., there is no previous SMAP for MS_1), the procedure proceeds to the next step. Otherwise, MAP_2 sends a update request message, UREQ(MS_1's IP Address, Selected SMAP's IP Address, t_1), to MAP_1. Upon receipt of UREQ, MAP_1 updates the entry

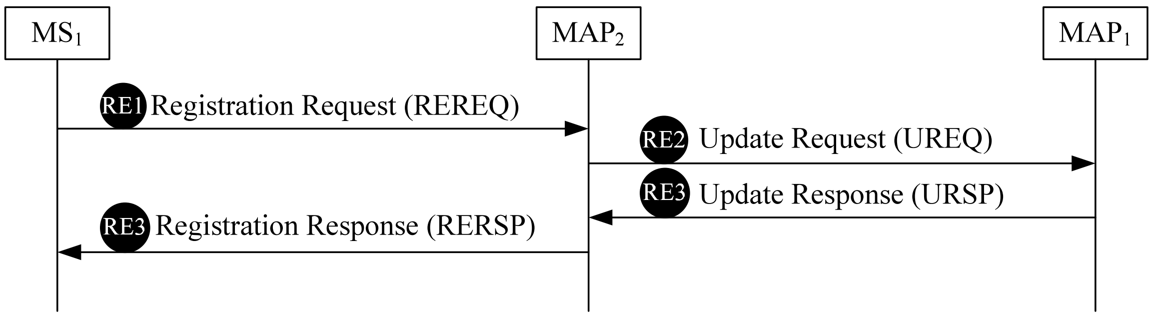


Figure 2.2: The message flow for the registration procedure

for MS_1 in its proxy table as: Im is set to MS_1's IP Address; Is is set to MAP_2's IP Address; Ts is set to t_1.

Step RE3. MAP_1 responds MAP_2 an update response message, URSP. Then, MAP_2 sends a registration response message, RERSP, to MS_1, which indicates that the registration request has been completed.

Note that MS_1 may have ongoing sessions during the movement, where handoff management is required to ensure session continuity. Existing handoff management mechanisms such as enhanced IAPP [6] may be adopted in this procedure, where packets are buffered in MAP_1 and then forwarded to MAP_2. This paper converges the study on the location management. The details of handoff management are not included in this paper.

After Step RE3, MS_1's location information is kept in the proxy tables of both MAP_1 and MAP_2, and the location management for MS_1 is done at MAP_1 and MAP_2. For other MNs with obsolete MS_1's location information (i.e., Is field for MS_1 stores MAP_1's IP address), the packets are first routed to MAP_1, and then MAP_1 retrieves its proxy table to forward the packets to MAP_2.

2.2 The Routing Procedure

The routing procedure is executed by MNs when the MNs route the packets for an MS, which consists of two parts: Location Information Synchronization and Packet Routing.

Part 1: Location Information Synchronization. In this part, the MS location information in the proxy table of the MN and that carried in the IP header of the packets are updated as the latest MS location information. Suppose that MS_1 (sender) is sending IP packets to MS_2 (receiver), where MAP_3 is one of the MNs along the routing path. Figure 2.3 illustrates the flow chart for this part. Steps L1 and L2 update the location information for the sender (i.e., MS_1).

Step L1. Upon receipt of an IP packet, MAP_3 first checks the prefix of MS_1's IP address to determine whether MS_1 is in the WMN. If the packet is sent from Internet into WMN (i.e., MS_1 is out of the WMN), the procedure jumps to Step L3. Otherwise, the procedure proceeds to the next step.

Step L2. MAP_3 checks the options field in the IP header. Two cases are considered.

Case L2.I. The options field is `null`, i.e., MAP_3 is MS_1's SMAP, whose proxy table contains MS_1's current location information. MAP_3 updates the options field in the IP header: ISS is set to the Is value of MS_1's entry in MAP_3's proxy table; SST is set to the Ts value of MS_1's entry in MAP_3's proxy table; IRS is set to `null`; RST is set to the `null`.

Case L2.II. The options field is not `null`. If MS_1's entry exists in MAP_3's proxy table, MAP_3 updates MS_1's location information. Otherwise, an entry is created for MS_1 in MAP_3's proxy table. MS_1's entry in MAP_3 is set as: Im is set to MS_1's IP Address; Is is set to the ISS value in the IP header; Ts is set to the IST value in the IP header.

The following two steps (Steps L3 and L4) update the location information for the receiver (i.e., MS_2).

Step L3. MAP_3 checks the prefix of MS_2's IP address to determine whether MS_2 is in WMN. If MS_2 is out of WMN, the procedure exits. Otherwise, the procedure proceeds to the next step.

Step L4. This step synchronizes MS_2's location information carried in the IP header and that stored in the proxy table. Let t_t be the Ts value in MS_2's entry, and t_p be the RST value in the IP header. Without loss of generality, if MS_2's entry does not exist, $t_t = 0$, and if the RST value is `null`, $t_p = 0$. We consider three cases:

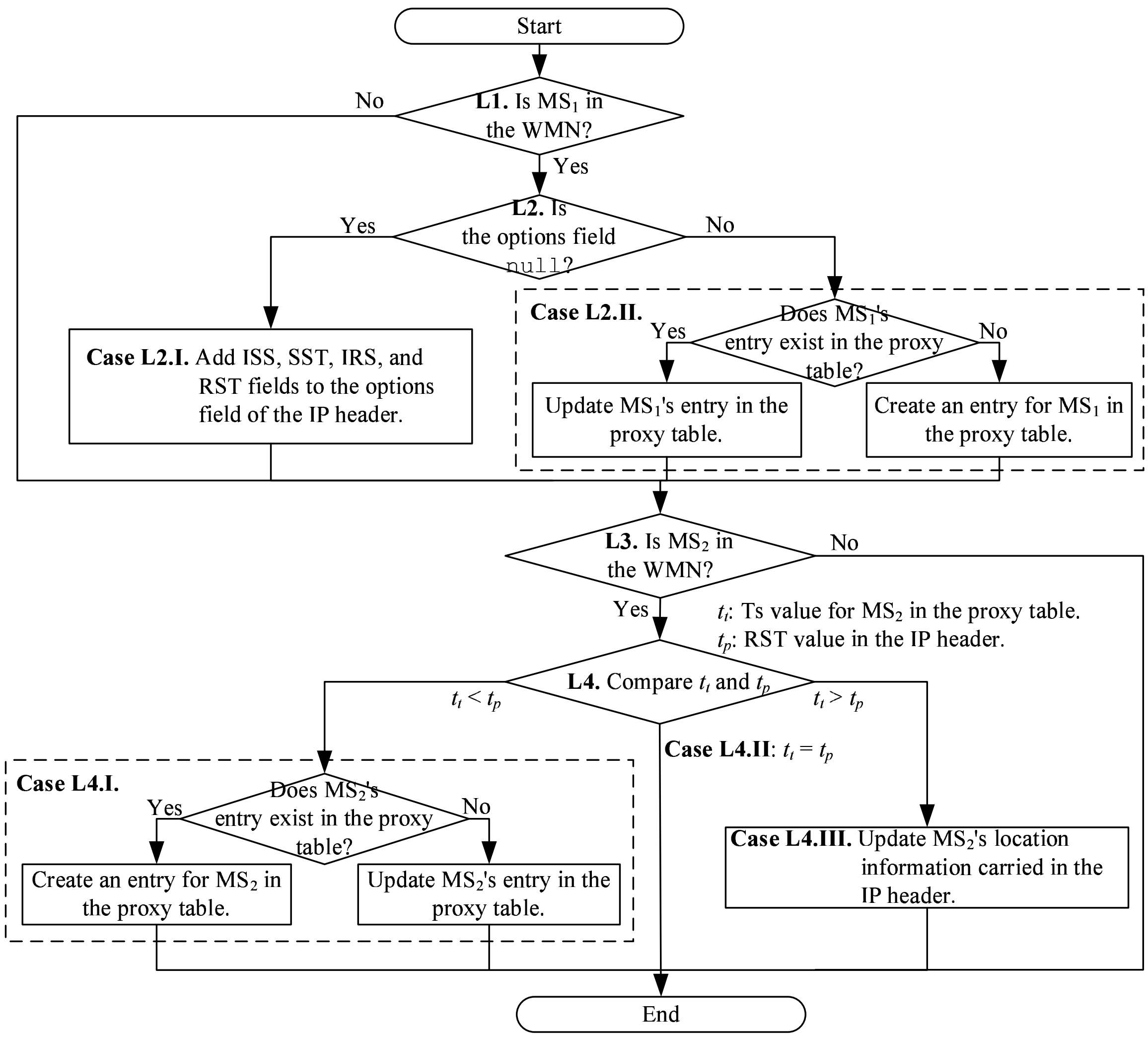


Figure 2.3: The flow chart for Part 1 of the routing procedure

Case L4.I. $t_t < t_p$, i.e., MS_2's location information carried in the IP header is fresher than that stored in the proxy table. MS_2's location information in MAP_3's proxy table is updated as MS_2's location information carried in the IP header. MS_2's entry in MAP_3 is updated as: Im is set to MS_2's IP Address; Is is set to the RSS value in the IP header; Ts is set to the RST value in the IP header.

Case L4.II. $t_t = t_p$. MS_2's location information carried in the IP header is the same as that in MAP_3's proxy table. The procedure does nothing.

Case L4.III. $t_t > t_p$, i.e., MS_2's location information in the proxy table of MAP_3 is fresher than that carried in the IP header. MS_2's location information carried in the IP header is filled with MS_2's location information in MAP_3's proxy table. The options field in the IP header is filled as: ISS is not changed; SST is not changed; IRS is set to the Is value of MS_2's entry in MAP_3's proxy table; RST is set to the Ts value of MS_2's entry in MAP_3's proxy table.

After Part 1 finishes, the sender's (i.e., MS_1's) current location information is stored in MAP_3's proxy table and in the IP header. MS_2's location information stored in MAP_3's proxy table and that carried in the IP header are synchronized.

Part 2: Packet Routing. In this part, MAP_3 routes the packets for MS_2 by referencing the IRS value in the IP header. MAP_3 determines whether MS_2 is in WMN by checking the prefix of MS_2's IP address, and then routes the packets by considering four cases.

Case R1. MS_2 is not in WMN. If MAP_3 is the mesh backhaul, it simply routes the packet to Internet. Otherwise (i.e., MAP_3 is not the mesh backhaul), it routes the packet to the next hop (that is close to the mesh backhaul) by referencing the routing table.

Case R2. MS_2 is in WMN, and the IRS field in the IP header is `null` (i.e., MS_2's SMAP is unknown). If MAP_3 is the mesh backhaul, it exercises the query procedure (to be elaborated in the next section) to obtain the IP address of MS_2's SMAP, and then routes the packet to the next hop that is close to MS_2's SMAP. Otherwise (i.e., MAP_3 is not the mesh backhaul), MAP_3 routes the packet to the next hop that is close to the mesh backhaul by referencing the routing table.

Case R3. MS_2 is in WMN, and the IRS field in the IP header specifies MAP_3's IP address (i.e., MAP_3 is MS_2's SMAP). The packet is directly delivered to MS_2.

Case R4. MS_2 is in WMN, and the IRS field in the IP header contains the value other than MAP_3's IP address. MAP_3 references the routing table to route the packet to the next hop that is close to MS_2's SMAP.

2.3 The Query Procedure

The query procedure is exercised by the mesh backhaul to obtain the IP address of receiver's SMAP when the mesh backhaul routes a packet for the receiver, and the receiver's SMAP is unknown (see Case R2 in the routing procedure). Suppose that MS_2 is the receiver of the packet. The query procedure consists of the following three steps.

Step Q1. The mesh backhaul broadcasts a route request message, RREQ(MS_2's IP Address), to all MAPs. The mesh backhaul starts a timer T_q and then expects to receive a route response message, RRES, before the timer expires.

Step Q2. Upon receipt of the RREQ message, MS_2's SMAP replies a route response message, RRES(IP Address of MS_2's SMAP, MS_2's Serving Timestamp), to the mesh backhaul.

Step Q3. If the RRES message is received before T_q expires, the mesh backhaul updates MS_2's location information carried in the IP header and that in the proxy table. After query procedure, MAP_3 can route the packet. Otherwise (i.e., T_q expires), the mesh backhaul discards the packet.

Note that the query procedure requires flooding signaling messages to all MNs in the WMN, which is a high cost operation.

Chapter 3

Implementation of the WMM Mechanism

We have implemented the WMM mechanism. In our implementation, we set up a WMN with four MAPs and a mesh backhaul. Each MAP is emulated by a laptop computer (running the Linux operating system) with two 802.11b WLAN cards. One serves as an AP for MSs, which is running in the infrastructure mode. The other is responsible for the communication with other MAPs, which is running in the ad-hoc mode. The mesh backhaul is emulated by a laptop computer (running the Linux OS) with an 802.11b WLAN card and an Ethernet card. The WLAN card is responsible for communication between the mesh backhaul and other MAPs, which is running in the ad-hoc mode. The Ethernet card connects the WMN with Internet through a wired link. We use the device with 802.11b WLAN capability to emulate MSs.

The software environment is listed in Table 3.1. Our implementation is based on the

Table 3.1: The software environment

OS	Linux-2.6. patched with ipdivert
Language	C++
Tools	hostap-driver-0.3.9, iptables-1.3.1, Wireless Tools for Linux
API	Unix Network API (raw socket)

Linux-2.6 OS [3] patched with the ipdivert package [1] that provides divert socket functionality for programmers to manipulate IP packets. We adopt C++ programming language. The software tools include hostap-driver-0.3.9 [2] (a driver to drive a WLAN card in the infrastructure mode), iptables-1.3.1 [21] (to direct the IP packets to the divert socket), and wireless tools for linux [22] (for the WLAN card manipulation). The Unix Network API [20] is adopted for the raw socket programming.

A simplified version of the software architecture for our implementation is shown in Figure 3.1. Due to the page limitation, we will separate the details of the implementation as another technical report. For the details and the software, readers may refer `http://pcs.csie.ntu.edu.tw/wmm/`. The iptales (Figure 3.1 (2)) is responsible to pass packets from the WLAN card driver (Figure 3.1 (1)) to the divert socket (Figure 3.1 (3)). We maintain a database (Figure 3.1 (5)) for the proxy table and the route table. Before the main procedure (Figure 3.1 (4)) routes packets from the divert socket, it looks up tables to obtain the routing information of the packets. Then by referencing the routing information, the main procedure sends the packets through raw sockets (Figure 3.1 (6)) to the WLAN

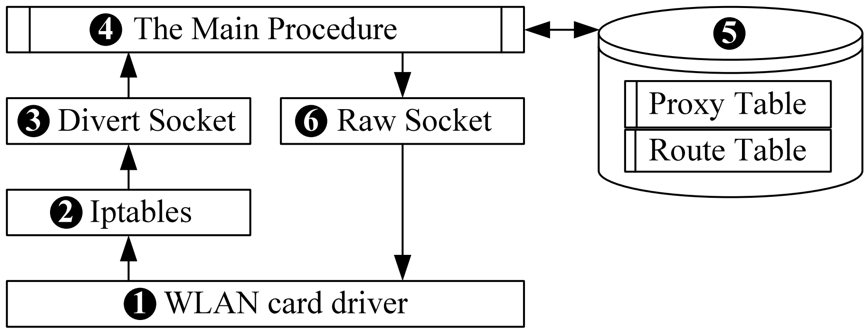


Figure 3.1: The software architecture of the implementation

card driver.

To justify our implementation, we developed a monitor program to watch the flow of packets. When an MAP routes the packet, it sends the sender's IP address and receiver's IP address of the packet to the monitor program. When the proxy table is modified, the modification of the proxy table (including the Im, Is, and Ts values) is informed to the monitor program. Figure 3.2 shows the appearance of the monitor program.

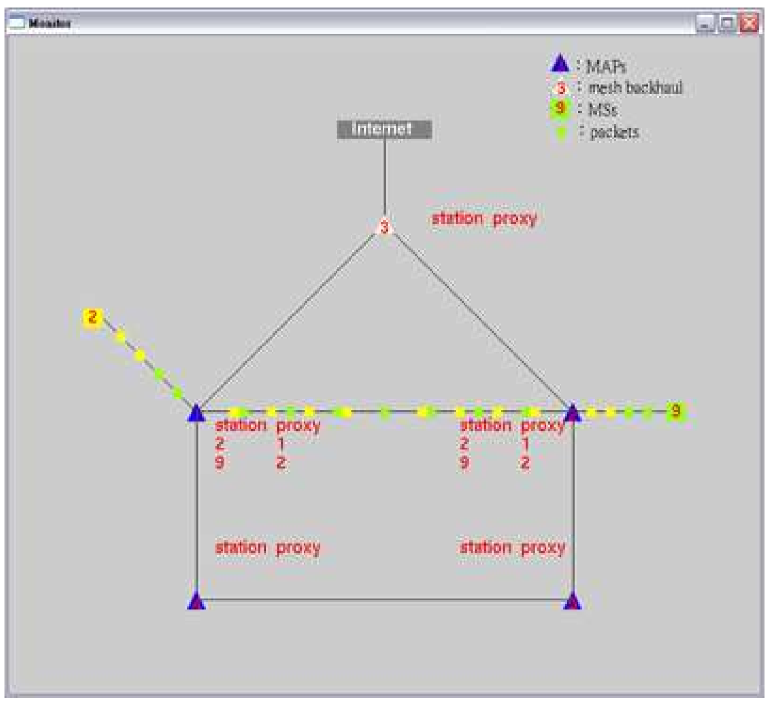


Figure 3.2: The appearance of the monitor

Chapter 4

An Analytical Model for Query Overhead

As described in Chapter 2.2, when the mesh backhaul routes the packet whose receiver's location information (i.e., the IP address of the receiver's SMAP) can not be determined, it exercises the query procedure to obtain the information (see Case R2). The query procedure requires flooding signaling messages to the WMN, which results in signaling overhead. This chapter proposes an analytical model and simulation experiments to study this performance issue.

We classify the traffic in WMN into two categories: Internet and intranet sessions. The Internet session involves an MS and a server (or a host) out of the WMN. The packets for Internet sessions must be routed through the mesh backhaul, and the sender's location information in the mesh backhaul's proxy table is updated. The intranet session involves two MSs in the same WMN.

Consider the timing diagram in Figure 4.1. Suppose that MS_0 enters WMN at t_0 and then stays in WMN for a time period w where w is assumed to have a general distribution with the density function $f_w(\cdot)$, and the corresponding Laplace transform $f_w^*(s)$. Let x be the time period between t_0 and the time when the MS originates the 1st Internet session, and y be the time period between t_0 and the time when the MS originates the 1st intranet session. Suppose that the Internet session arrivals originated by an MS form a Poisson process with rate λ_I. Then, with the memoryless property of the exponential distribution, we have the density function $f_x(x)$ for x as

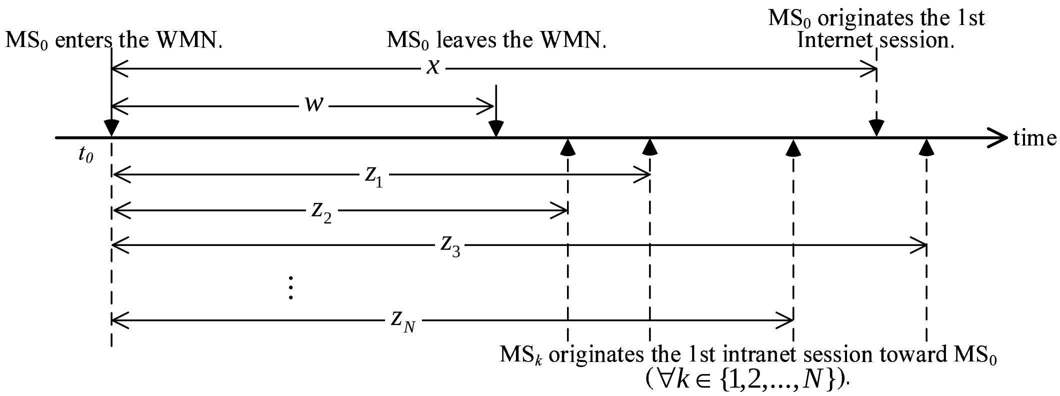
$$f_x(x) = \lambda_I e^{-\lambda_I x}.$$

Assume that the intranet session arrivals originated by an MS form a Poisson process with rate λ_a. Then, the density function $f_y(y)$ for y is

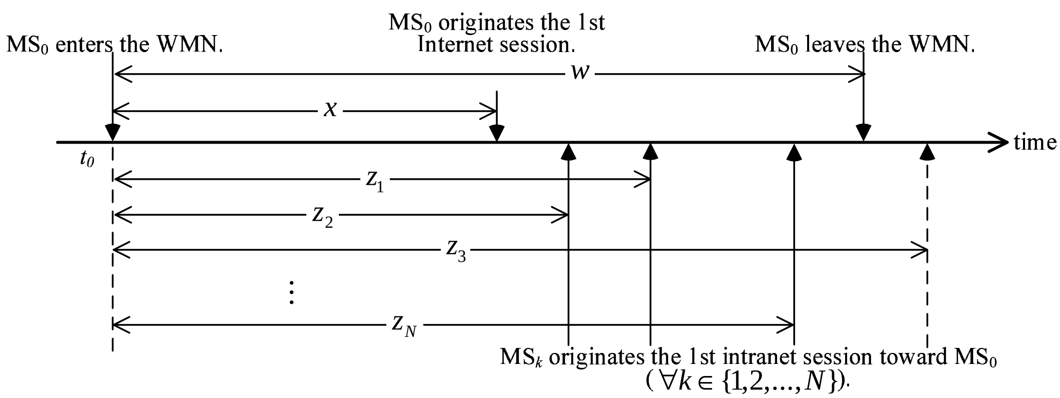
$$f_y(y) = \lambda_a e^{-\lambda_a y}.$$

Suppose that when MS_0 enters WMN, there are another N MSs (denoted as MS_1, MS_2, ..., MS_N) in the WMN, and all MSs in the WMN are identical. Let z_k be the time period between t_0 and the time when MS_k (where $1 \le k \le N$) originates the 1st intranet session toward MS_0. Then the intranet session arrivals from a specific MS_k to MS_0 also forms a Poisson process with rate $\dfrac{\lambda_a}{N}$, and the density function $f_z(z)$ for z_k as

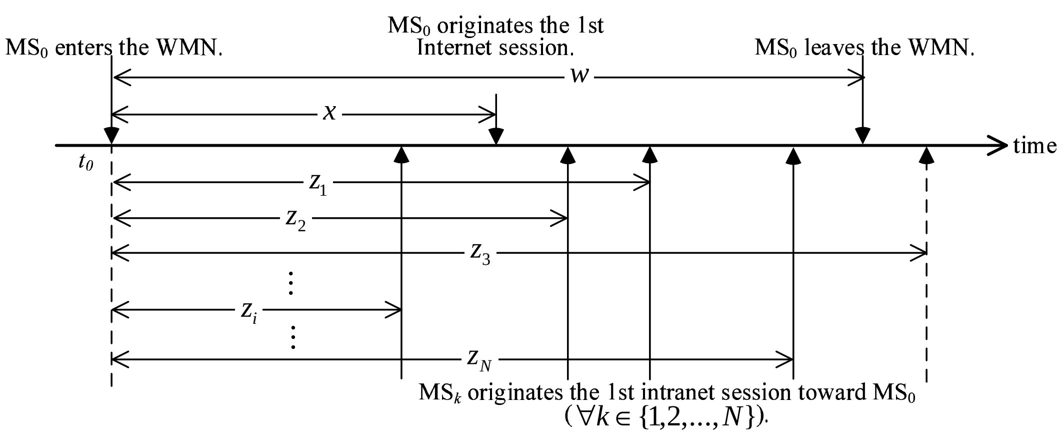
$$f_z(z) = \left(\frac{\lambda_a}{N}\right) e^{-\left(\frac{\lambda_a}{N}\right) z}.$$



(a) $w \le x$ and $\forall k \in \{1,2,\ldots,N\}$, $w \le z_k$



(b) $x \le w$ and $\forall k \in \{1,2,\ldots,N\}$, $x \le z_k$



(c) $\exists i \in \{1,2,\ldots,N\} \ni z_i < w$ and $z_i < x$

Figure 4.1: The timing diagram

Let P_q be the probability that during the time period w, the query procedure is invoked by the mesh backhaul to obtain MS_0's location information. As shown in Figure 4.1, three cases are considered to derive the P_q probability:

Case 1. $w \leq x$ and for all $k \in \{1, 2, ..., N\}$, $w \leq z_k$ (see Figure 4.1 (a)). In this case, during the w period, there are no Internet session initiated from MS_0 and no intranet session terminated at MS_0. Thus, there is no Internet session toward MS_0. The query procedure is not invoked during the w period.

Case 2. $x \leq w$ and for all $k \in \{1, 2, ..., N\}$, $x \leq z_k$ (see Figure 4.1 (b)). In this case, during the w period, MS_0 initiated the 1st Internet session. At $t_0 + x$, the mesh backhaul creates an entry to store MS_0's location information. After $t_0 + x$, if there are packets (either for Internet sessions or intranet sessions) to be routed to MS_0, these packets can be correctly routed to MS_0 without invoking the query procedure.

Case 3. There exists $i \in \{1, 2, ..., N\}$, such that $z_i \leq w$ and $z_i \leq x$ (see Figure 4.1 (c)). In this case, during the w period, at least one MS_i ($1 \leq i \leq N$) initiates the 1st intranet session toward MS_0 at $t_0 + z_i$ before MS_0 initiates the 1st Internet session at $t_0 + x$. At $t_0 + z_i$, if any of the MNs along the routing path between MS_i's SMAP and the mesh backhaul stores MS_0's location information, then the query procedure is not invoked during the packet transmission for the intranet session from MS_i to MS_0.

Let A be the probability that Case 1 occurs, B be the probability that Case 2 occurs, and C be the probability that given Case 3, the routing MNs contain MS_0's location information. Then we have the P_q probability as

$$P_q = 1 - A - B - C.$$

Probability C is highly dependent on the network topology and the relative positions of the MSs. The analysis for C is too complicated. In this study, we derive an upper bound $\tilde{P}_q$ for P_q, that is

$$\tilde{P}_q = 1 - A - B \geq P_q. \tag{4.1}$$

The A and B probabilities are derived as follows:

$$\begin{aligned}
A &= \Pr[w \leq x \text{ and } \forall k \in \{1, 2, ..., N\},\ w \leq z_k] \\
&= \int_{w=0}^{\infty} \int_{x=w}^{\infty} \left[\prod_{k=1}^{N} \int_{z_k=w}^{\infty} f_z(z_k) dz_k \right] f_x(x) f_w(w) dx dw \\
&= \int_{w=0}^{\infty} \int_{x=w}^{\infty} \left[\prod_{k=1}^{N} \int_{z_k=w}^{\infty} \left(\frac{\lambda_a}{N} \right) e^{-\left(\frac{\lambda_a}{N}\right) z_k} dz_k \right] \lambda_I e^{-\lambda_I x} f_w(w) dx dw \\
&= \int_{w=0}^{\infty} \int_{x=w}^{\infty} \left[e^{-\left(\frac{\lambda_a}{N}\right) w} \right]^N \lambda_I e^{-\lambda_I x} f_w(w) dx dw \\
&= \int_{w=0}^{\infty} e^{-(\lambda_I + \lambda_a) w} f_w(w) dw \\
&= f_w^*(\lambda_I + \lambda_a)
\end{aligned} \tag{4.2}$$

and

$$\begin{aligned}
B &= \Pr[x \le w \text{ and } \forall k \in \{1,2,...,N\},\, x \le z_k] \\
&= \int_{x=0}^{\infty}\int_{w=x}^{\infty}\left[\prod_{k=1}^{N}\int_{z_k=x}^{\infty} f_z(z_k)dz_k\right] f_x(x)f_w(w)dwdx \\
&= \int_{x=0}^{\infty}\int_{w=x}^{\infty}\left[\prod_{k=1}^{N}\int_{z_k=x}^{\infty} \left(\frac{\lambda_a}{N}\right) e^{-\left(\frac{\lambda_a}{N}\right)z_k}dz_k\right] \lambda_I e^{-\lambda_I x} f_w(w)dwdx \\
&= \lambda_I \int_{x=0}^{\infty}\int_{w=x}^{\infty}\left[e^{-\left(\frac{\lambda_a}{N}\right)x}\right]^N e^{-\lambda_I x} f_w(w)dwdx \\
&= \lambda_I \int_{x=0}^{\infty} e^{-(\lambda_I+\lambda_a)x}\left[1 - F_w(x)\right]dx \\
&= \lambda_I \left[\frac{1 - f_w^*(s)}{s}\right]\Bigg|_{s=\lambda_I+\lambda_a}
\end{aligned} \tag{4.3}$$

where $F_w(w) = \int_0^w f_w(t)dt$ is the cumulative distribution function for w. Apply (4.2) and (4.3) into (4.1), and $\tilde{P}_q$ is expressed as

$$\tilde{P}_q = \left(\frac{\lambda_a}{\lambda_I + \lambda_a}\right)\left[1 - f_w^*(\lambda_I + \lambda_a)\right]. \tag{4.4}$$

In this study, we take the Gamma distribution as an example for the distribution of w because the Gamma distribution can approximate many other distributions very well [9][13][17]. The Laplace transform $f_w^*(s)$ for $f_w(\cdot)$ with shape parameter α, mean $\frac{1}{\mu_w}$, and variance $v_w = \frac{1}{\alpha\mu_w^2}$ is

$$f_w^*(s) = \left(\frac{\alpha\mu_w}{\alpha\mu_w + s}\right)^{\alpha}. \tag{4.5}$$

Applying (4.5) into (4.4), we have

$$\tilde{P}_q = \left(\frac{\lambda_a}{\lambda_I + \lambda_a}\right)\left[1 - \left(\frac{\alpha\mu_w}{\alpha\mu_w + \lambda_I + \lambda_a}\right)^{\alpha}\right]. \tag{4.6}$$

This study also conducts simulation experiments to investigate the P_q performance. We adopt discrete event-driven approach in our simulation, which is widely used to simulate the mobile communication networks in several studies [10][11][12][14][15]. The WMN is modeled as a regular hexagonal topology [8]. Each hexagon represents the coverage area of an MAP. The mesh backhaul is located at the center of the WMN. All MSs are identical. The movement of an MS follows a 2-D random walk model [4], where an MS resides in an MAP's coverage area for a period and then moves to one of its neighboring MAPs with the same probability 1/6. Due to the page limitation, the details of the simulation are omitted in this paper.

Figures 4.2-4.4 compare the results (see the solid curves) of the analytical model against that (see the dashed curves) of the the simulation model. The input parameters λ_I and λ_a are normalized by μ_w. For example, if the expected time that an MS stays in the WMN is $\frac{1}{\mu_w} = 30$ minutes, then $\lambda_I = 10\mu_w$ indicates that the expected Internet session inter-arrival time for an MS is 3 minutes. The figures indicate that P_q of the simulation model is properly bounded by $\tilde{P}_q$. The trends of P_q and $\tilde{P}_q$ curves are the same. The impacts of the input parameters are discussed below.

Effects of Intranet Traffic Ratio $\frac{\lambda_a}{\lambda_I+\lambda_a}$. Figure 4.2 plots $\tilde{P}_q$ and P_q as increasing functions of $\frac{\lambda_a}{\lambda_I+\lambda_a}$, where $\lambda_I+\lambda_a$ is fixed to $10\mu_w$ and $N = 50$ with various v_w setups. As $\frac{\lambda_a}{\lambda_I+\lambda_a}$ increases, it is more likely that an intranet session terminated at an MS occurs

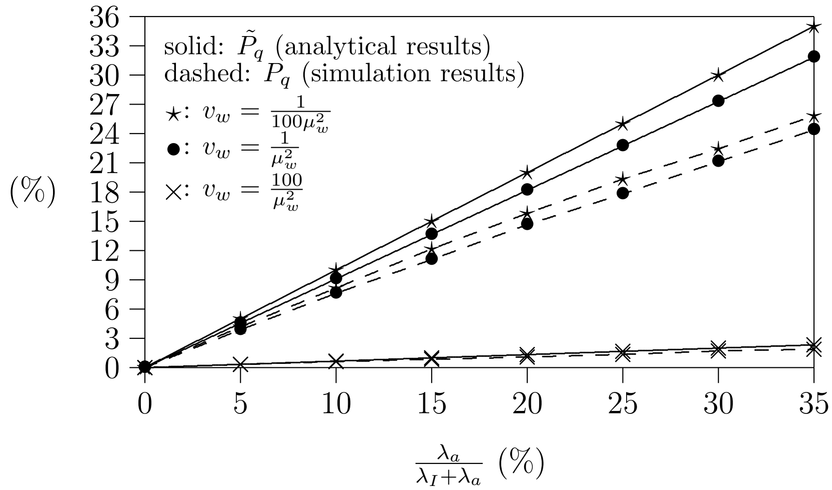


Figure 4.2: Effects of proportion of λ_a ($\lambda_I + \lambda_a = 10\mu_w$; $N = 50$)

before the MS initiates the 1st Internet session. The mesh backhaul has worse chance to contain the MS location information. Therefore, $\tilde{P}_q$ increases as $\dfrac{\lambda_a}{\lambda_I + \lambda_a}$ increases. In the real situation, most traffic of an MS is from Internet sessions (e.g., $\dfrac{\lambda_a}{\lambda_I + \lambda_a}$ is less than 10%). When $\dfrac{\lambda_a}{\lambda_I + \lambda_a} < 10\%$, $\tilde{P}_q$ is bounded by 10%. Consequently, WMM mechanism does not incur significant query (or signaling) overhead.

Effects of Number of MSs N. In Figure 4.3, we set $\lambda_I + \lambda_a = 10\mu_w$ and $\dfrac{\lambda_a}{\lambda_I + \lambda_a} = 10\%$, and change the variance v_w. This figure shows that the $\tilde{P}_q$ and P_q performances are not affected by N. For various v_w setups, we observe the same trend. This phenomenon can be explained by interpreting Equation (4.6) based on our analysis, where the query probability is independent of N.

Effects of Total Traffic $\lambda_I + \lambda_a$. Figure 4.4 studies the effects of total traffic on the $\tilde{P}_q$

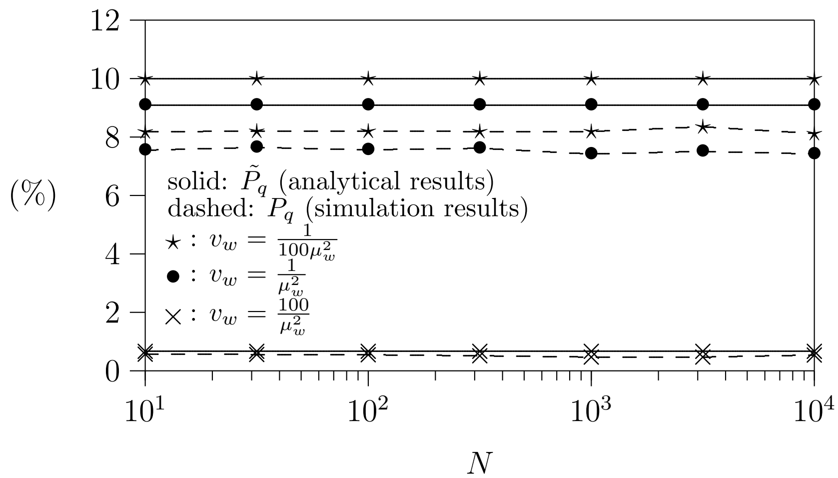


Figure 4.3: Effects of N ($\lambda_I + \lambda_a = 10\mu_w$; $\frac{\lambda_a}{\lambda_I+\lambda_a} = 10\%$)

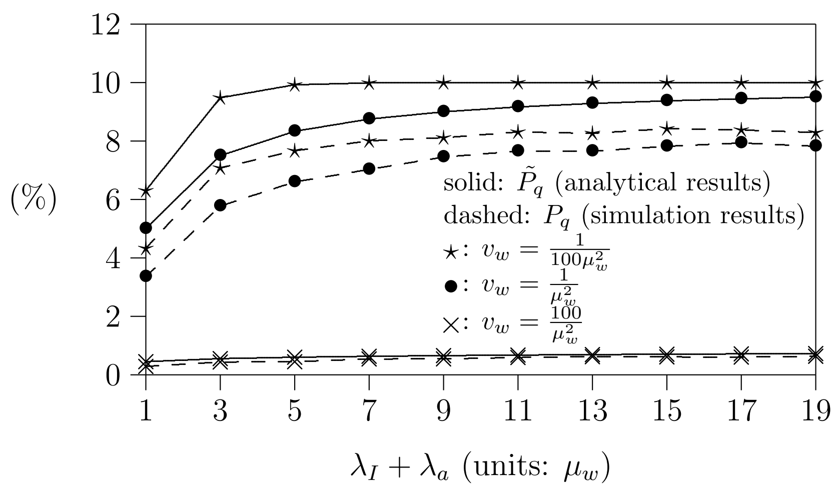


Figure 4.4: Effects of $\lambda_I + \lambda_a$ ($\frac{\lambda_a}{\lambda_I+\lambda_a} = 10\%$; $N = 50$)

and P_q performance, where we set $\frac{\lambda_a}{\lambda_I + \lambda_a} = 10\%$ and $N = 50$. For different v_w setups, we observe the following two phenomena:

Phenomenon I. For $v_w = \frac{1}{100\mu_w^2}$ (i.e., $\alpha = 100$) or $v_w = \frac{1}{\mu_w^2}$ (i.e., $\alpha = 1$), when $\lambda_I + \lambda_a$ is low (i.e., $\lambda_I + \lambda_a < 5\mu_w$), $\tilde{P}_q$ and P_q increase significantly as $\lambda_I + \lambda_a$ increases. On the other hand, when total traffic is high (i.e., $\lambda_I + \lambda_a > 5\mu_w$), the effects of total traffic become minor.

Phenomenon II. For $v_w = \frac{100}{\mu_w^2}$ (i.e., $\alpha = 0.01$), $\tilde{P}_q$ and P_q increase slightly as $\lambda_I + \lambda_a$ increases, and bounded by 1%.

We explain the above two phenomena by considering (4.6). For $\alpha = 100$ or $\alpha = 1$, as $\lambda_I + \lambda_a$ increases, $1 - \left(\frac{\alpha\mu_w}{\alpha\mu_w + \lambda_I + \lambda_a}\right)^\alpha$ increases and approaches to 1. In this figure, we fix $\frac{\lambda_a}{\lambda_I + \lambda_a} = 10\%$. Hence, we observe that $\tilde{P}_q$ increases and approaches to $\frac{\lambda_a}{\lambda_I + \lambda_a} = 10\%$. For $\alpha = 0.01$, $1 - \left(\frac{\alpha\mu_w}{\alpha\mu_w + \lambda_I + \lambda_a}\right)^\alpha$ approaches to 0. Therefore, we observe small $\tilde{P}_q$ and P_q values for various $\lambda_I + \lambda_a$ setups.

Effects of Variance v_w. Figures 4.2-4.4 also study the impacts of v_w. As these figures show, $\tilde{P}_q$ or P_q for small v_w are larger than that for large v_w. From (4.6), it is obvious that when v_w is larger (i.e., α is smaller), $\left(\frac{\alpha\mu_w}{\alpha\mu_w + \lambda_I + \lambda_a}\right)^\alpha$ becomes large, and $\tilde{P}_q$ decreases. Therefore, we observe this phenomenon. To summarize, WMM works better when the variance is large.

Chapter 5

Comparison between WMM and Existing MM Protocols

This chapter compares the WMM mechanism with other existing MM protocols (including ad-hoc routing protocol, centralized-database MM protocol, and mobile IP protocol) in terms of signaling cost for location management and routing cost. Specifically, we study the cost C_u (for WMN to update MS location information when an MS changes its SMAP; also known as "*location update*"), the cost C_t (for WMN to query the location information for destination MS to route packets for a session; also known as "*location tracking*"), and the cost C_r (for WMN to route packets to a destination MS) for different MM protocols. In our study, the C_u and C_t are defined as the average numbers of MNs and MSs that exchange signaling messages for a location update operation (executed when an MS changes its SMAP) and a location tracking operation (executed when a session is initiated toward an MS), respectively. The C_r is defined as the average number of MNs or MSs that route a packet to a destination MS. In

WMM and the three existing MM protocols, the number of signaling messages (exchanged between two nodes) for location management is fixed to one or two. Therefore, by analyzing C_u and C_t, we can understand the signaling overhead introduced by the MM protocols, i.e., the larger C_u or C_t means the higher signaling overhead. By analyzing C_r, we can understand the routing overhead introduced by the MM protocols. We consider a WMN consisting of M MNs and N MSs. The C_u, C_t, and C_r costs for WMM and other three existing MM protocols are compared in the following sections. The notations used in this comparison is listed below.

- M (N): the number of MNs (MSs) in the WMN
- $\bar{L}$: the average number of MNs in the routing path from MSs to a centralized node (e.g., a centralized-database or an HA)
- $\bar{R}$: the average number of MNs in the routing path between two MSs
- P_q: the probability that the query procedure in WMM mechanism is invoked for an MS
- r: the average number of sessions initiated toward an MS in the WMN
- δ: a small number to reflect the routing overhead of WMM

5.1 Signaling and Routing Cost for WMM

In WMM, when an MS changes its SMAP, location update is done through the registration procedure and the routing procedure. The registration procedure is executed between an MS, the MS's current SMAP, and the MS's pervious SMAP. The number of nodes involved in this procedure is 3. The routing procedure is done by an MN while it routes a packet for the MS. No signaling messages are required for the routing procedure. Thus, we have $C_u = 3$ for WMM.

When a session is initiated toward an MS, location tracking is processed through the routing procedure and the query procedure. As described above, the routing procedure does not incur signaling cost. On the other hand, the query procedure requires flooding signaling messages to all MNs. The number of nodes involved in the query procedure equals to the total number of MNs (i.e., M). However, the query procedure may not be invoked for the MS during the time when the MS stays in the WMN. Actually, the query procedure is executed with probability P_q for an MS (see Chapter 4), and it is invoked at most once for the MS. Let r ($r > 0$) be the number of sessions initiated toward an MS during the time when the MS stays in the WMN. Consequently, C_t for WMM can be estimated as $\dfrac{M \cdot P_q}{r}$.

Suppose that MS_1 is sending packets to MS_2, where MAP_1 and MAP_2 are SMAPs of MS_1 and MS_2, respectively. Let MAP_2' be MS_2's previous SMAP. Three cases are considered to count C_r for WMM:

Case 1. MAP_1's proxy table contains MS_2's current location information. The packets can be routed directly to MS_2, and C_r is $\bar{R}$.

Case 2. MAP_1's proxy table contains obsolete MS_2's location information. The packets are first routed to MAP_2' (with routing cost $\bar{R}$), and then MAP_2' routes the packets to MAP_2. The C_r is larger than $\bar{R}$.

Case 3. MS_2's entry does not exist in MAP_1's proxy table. The packets will be routed to the mesh backhaul. If an MN along the routing path between MAP_1 and the mesh backhaul contains MS_2's location information, then the packet can be routed to MS_2. The C_r cost is larger than $\bar{R}$.

In most of the cases in the real system, communications between two MSs are bidirectional (i.e., both MS_1 and MS_2 exchange packets with each other). It is likely that MS_1's SMAP caches MS_2's current location information and vice versa. Cases 2 and 3 are not likely to occur. Thus, the C_r cost for WMM can be estimated as $\bar{R} + \delta$, where δ is a small number (to reflect that the C_r is slightly affected by Cases 2 and 3).

5.2 Signaling and Routing Cost of Ad-hoc Routing Protocol

Two basic approaches, proactive (also known as table-driven) and reactive (also known as demand-driven) are proposed for the ad-hoc routing protocol [19]. In the proactive approach, an MS maintains a routing table to store all routing paths between the MS and other MSs. Location update is done by notifying all MNs and MSs of the MS's movement, and we have $C_u = M + N$ for the proactive ad-hoc routing protocol. In the proactive approach, when an MS routes a packet to the destination MS, it references its own routing table, and no signaling messages are required for location tracking. Thus, $C_t = 0$ for the proactive ad-hoc routing protocol. Furthermore, in the proactive ad-hoc routing protocol, since the routing tables always contain current location information for MSs, packets can be routed directly to the destination. We have $C_r = \bar{R}$ for the proactive ad-hoc routing protocol.

In the reactive approach, the MS discovers routes when it has packets to be sent. No location update operation is executed in this approach, and we have $C_u = 0$ for the reactive ad-hoc routing protocol. When an MS has packets to be sent to the destination MS, it obtains a route through flooding signaling messages to the WMN, and the location tracking is done at the same time. Thus, we have $C_t = M + N$ for the reactive ad-hoc routing protocol. Since the discovered routes are updated just before the packets are sent, we have $C_r = \bar{R}$ for the reactive approach.

To summarize, the proactive approach introduces extra signaling overhead for routing table maintenance especially when MS mobility is high. The reactive approach not only introduces extra signaling overhead but also spends time to establish the route before user data is delivered, which significantly delays user data transmission.

5.3 Signaling and Routing Cost of Centralized-database MM Protocol

In centralized-database MM protocol, a centralized MM database is maintained to store the location information for all MSs. Whenever an MS moves from an LA to another, a registration procedure is triggered for location update. The registration procedure is executed between the MS and the database to update the LA ID stored in the database. Thus, we have $C_u = \bar{L}$ for the centralized-database MM protocol.

When an MS has packets to be sent to the destination MS, it queries the database for the destination MS's location information, where the signaling messages for location tracking are exchanged between the MS and the database. Thus, we have $C_t = \bar{L}$ for the centralized-database MM protocol.

In the centralized-database MM protocol, the destination MS's current location information is stored in the centralized database. The packets can be routed correctly to the destination MS. We have $C_r = \bar{R}$ for the centralized-database MM protocol.

5.4 Signaling and Routing Cost of Mobile IP Protocol

In mobile IP protocol, the HA in the home network and the FA in the foreign network are responsible to tunnel packets for MSs. When an MS moves from the home network to the foreign network, a registration procedure is triggered for location update, which is executed between the MS, the FA, and the HA to inform the FA and HA of the MS's movement. Typically, the MS is close to the FA, and we can omit the signaling and routing cost between the MS and the FA. Thus, we have $C_u = \bar{L}$ for the mobile IP protocol.

When MNs route packets to an MS, the packets are first routed to the HA, and then the HA routes the packets to the destination MS directly (if the destination MS is in its home network) or delivers the packets to the destination MS by tunneling them from the HA to the FA (if the destination MS is in the foreign network). There is no signaling messages required for location tracking, and we have $C_t = 0$ for the mobile IP protocol.

In mobile IP protocol, the packets are always routed to the HA and then to the destination (i.e., the triangle routing problem), the C_r cost for the mobile IP protocol can be estimated as $2\bar{L}$.

Table 5.1: Comparison between WMM and other MM mechanisms

	The C_u Cost	The C_t Cost	The C_r Cost
Proactive Ad-hoc Routing Protocol	$M+N$	0	$\bar{R}$
Reactive Ad-hoc Routing Protocol	0	$M+N$	$\bar{R}$
Centralized-database MM Protocol	$\bar{L}$	$\bar{L}$	$\bar{R}$
Mobile IP Protocol	$\bar{L}$	0	$2\bar{L}$
WMM Mechanism	3	$\frac{M \cdot P_q}{r}$	$\bar{R}+\delta$

5.5 Comparison

Table 5.1 lists the C_u cost and the C_t cost for the four existing MM protocols and the proposed WMM mechanism. As shown in Table 5.1, the C_u cost for WMM is constant and is lower than that for the proactive ad-hoc routing protocol, the centralized-database MM protocol, and the mobile IP protocol. The C_u cost for WMM is slightly higher than that for the reactive ad-hoc routing protocol. However, the C_t for WMM is much lower than that for the reactive ad-hoc routing protocol.

The C_t cost for WMM is estimated as $\dfrac{M \cdot P_q}{r}$. Typically, the service area of a WMN is not huge, and the number (i.e., M) is not large number. As shown in our study (see Chapter 4) P_q is low, which is bounded by 10%). Furthermore, when the traffic load to a MS increases (i.e., more sessions are initiated to an MS; i.e., r increases), we gain better C_t performance for WMM. From this analysis, we have that the WMM mechanism does not incur heavy C_t cost. Obviously, compared with the reactive ad-hoc routing protocol (whose C_t is estimated as $M+N$) and the centralized-database MM protocol (whose C_t is estimated as $\bar{L}$), WMM has the lower C_t cost. The C_t for WMM is slightly higher than that for the proactive ad-hoc routing protocol and the mobile IP protocol. However, the WMM significantly outperforms the proactive ad-hoc routing protocol and the mobile IP protocol in terms of the C_u cost.

The C_r cost for WMM is obviously lower than that for the mobile IP protocol, whose C_r cost is estimated as $2\bar{L}$. The C_r cost for WMM is slightly higher than that for the proactive ad-hoc routing protocol, the reactive ad-hoc protocol, and the centralized-database MM protocol. However, as mentioned above, the ad-hoc routing protocols and the centralized-database protocol incurs much heavier signaling overhead (i.e., C_u and C_t cost) than WMM does. To summarize, the proposed WMM mechanism is capable of correctly and efficiently routing packets for MSs with lighter overhead than the existing MM protocols do.

Chapter 6

Concluding Remark

This paper designed a novel MM protocol, the WMM mechanism, for WMN by capturing the characteristics of WMN. In the WMM mechanism, location caches to cache MSs' location information are added in the MAPs so that the network can more efficiently (i.e., fast and low signaling cost) routes packets to mobile users. The fields in IP header are utilized to carry MSs' location information. Location update can be done at the same time when the MN routes packets for MSs, and the signaling cost for location update is reduced. The MSs' location information is distributed within WMN. We have implemented a prototype of the WMM mechanism in the real system.

If the MS's location information can not be determined, the query procedure is executed to find the MS's location, where signaling cost may incur to the network. We conducted an analytical model and simulation experiments to study this performance issue. Our study shows that

- The probability P_q (that during the time period when an MS stays in WMN, the query procedure exercises) is reasonably low and is not affected by the number of MSs in WMN.

- With higher variance of the time when an MS stays in WMN, the WMN gains the better P_q performance (i.e, P_q is the lower).

In the end of the paper, we made a comparison between WMM and other existing MM protocols (including ad-hoc routing protocol, centralized-database MM protocol, and mobile IP protocol). Our study concluded that the WMM mechanism can provide correct and efficient packet routing for MSs with lighter signaling and routing cost than the existing MM protocols do.

Bibliography

[1] Divert Sockets for Linux. [Online]. Available: http://sourceforge.net/projects/ipdivert/.

[2] Host AP Driver for Intersil Prism2/2.5/3, Hostapd, and WPA Supplicant. [Online]. Available: http://hostap.epitest.fi/.

[3] The Linux Kernel Archives. [Online]. Available: http://www.kernel.org/.

[4] Akyildiz, I. F., Lin, Y.-B., Lai, W.-R., and Chen, R.-J. A New Random Walk Model for PCS Networks. *IEEE Journal on Selected Areas in Communications*, 18(7):1254–1260, July 2000.

[5] Akyildiz, I. F., Wang, X., and Wang W. Wireless Mesh Networks: A Survey. *Computer Networks Journal, Elsevier*, 47(4):445–487, March 2005.

[6] Chou, C.-T. and Shin, K. G. An Enhanced Inter-Access Point Protocol for Uniform Intra and Intersubnet Handoffs. *IEEE Journal on Mobile Computing*, 14(4):321–334, July-August 2005.

[7] IEEE. IEEE Std 802.11-1997 Information Technology- telecommunications And Information exchange Between Systems-Local And Metropolitan Area Networks-specific Requirements-part 11: Wireless LAN Medium Access Control (MAC) And Physical Layer (PHY) Specifications. *IEEE Std 802.11-1997*, November 1997.

[8] IEEE. IEEE Standard for Local and Metropolitan Area Networks Part 16: Air Interface for Fixed Broadband Wireless Access Systems. *IEEE Std 802.16-2004 (Revision of IEEE Std 802.16-2001)*, May 2004.

[9] Lin, P. and Lin, Y.-B. Implementation and Performance Evaluation for Mobility Management of a Wireless PBX Network. *IEEE Journal on Selected Areas in Communications*, 19(6):1138–1146, June 2001.

[10] Lin, P. and Tu, G.-H. An Improved GGSN Failure Restoration Mechanism for UMTS. *ACM Wireless Networks*, 12(1):91–103, February 2006.

[11] Lin, P., Chang, H.-M., Fang, Y., and Cheng, S.-M. HISNs: Distributed Gateways for Application-level Integration of Heterogenous Wireless Networks. *ACM Wireless Networks*, Accepted for publication.

[12] Lin, P., Lai, W.-R., and Gan, C.-H. Modeling Opportunity Driven Multiple Access in UMTS. *IEEE Transactions on Wireless Communications*, 3(5):1669–1677, September 2004.

[13] Lin, P., Lin, Y.-B., and Chlamtac, I. Modeling Frame Synchronization for UMTS High-Speed Downlink Packet Access. *IEEE Transactions on Vehicular Technology*, 50(1):132–141, January 2003.

[14] Lin, P., Lin, Y.-B., and Chlamtac, I. Module Count-Based Overflow Control Scheme for UMTS High-Speed Downlink Packet Access. *IEEE Transactions on Vehicular Technology*, 53(2):425–432, March 2004.

[15] Lin, P., Lin, Y.-B., and Chlamtac, I. Overflow Control for UMTS High-Speed Downlink Packet Access. *IEEE Transactions on Wireless Communications*, 3(2):524–532, March 2004.

[16] Lin, Y.-B. and Chlamtac, I. *Wireless and Mobile Network Architectures*. John Wiley & Sons, 2001.

[17] Lin, Y.-B. and Lin, P. Performance Modeling of Location Tracking Systems. *ACM Mobile Computing and Communications Review*, 2(3):24–27, July-August 1998.

[18] Perkins, C. E. Mobile Networking through Mobile IP. *IEEE Internet Computing*, 2(1):58–69, January 1998.

[19] Royer, E. M. and Toh, C.-K. A Review of Current Routing Protocols for Ad Hoc Mobile Wireless Networks. *IEEE Personal Communications*, 6(2):46–55, April 1999.

[20] Stevens W., Fenner B., and Rudoff A. *Unix Network Programming, Vol. 1: The Sockets Networking API, Third Edition.* Addison-Wesley Professional, October 2003.

[21] The Netfilter Core Team. The netfilter.org "iptables" project. [Online]. Available: http://www.netfilter.org/projects/iptables/.

[22] Tourrilhes J. Wireless Tools for Linux. [Online]. Available: http://www.hpl.hp.com/personal/Jean_Tourrilhes/Linux/Tools.html.