國立交通大學

網路工程研究所

碩 士 論 文

一 個 異 質 無 線 網 路 換 手 機 制 之 研 究

A Study on Vertical Handover Scheme over Heterogeneous Wireless Network

研 究 生：王博謙

指導教授：陳耀宗　教授

中 華 民 國　一 零 零 年 八 月

一 個 異 質 無 線 網 路 換 手 機 制 之 研 究

A Study on Vertical Handover Scheme over Heterogeneous Wireless Network

研 究 生：王博謙　　Student：Po-Chien Wang

指導教授：陳耀宗　　Advisor：Yaw-Chung Chen

國 立 交 通 大 學

網 路 工 程 研 究 所

碩 士 論 文

A Thesis

Submitted to Institute of Network Engineering

College of Computer Science

National Chiao Tung University

in partial Fulfillment of the Requirements

for the Degree of

Master

in

Computer Science

August 2011

Hsinchu, Taiwan, Republic of China

中華民國一零零年八月

一個異質無線網路換手機制之研究

學生：王博謙　　指導教授：陳耀宗 博士

國立交通大學網路工程研究所

摘要

近年來各種無線網路技術迅速發展，如 WiFi、 WiMAX、LTE，讓使用者隨時隨地可以藉由行動裝置連上網際網路擷取多樣的網路服務。行動裝置的運算能力也變得越來越高並提供多模上網功能，使用者可以藉由多模上網同時連上不同網路，因此造成異質網路間的換手問題。在使用者移動到其中一個無線網路的涵蓋範圍之外時，會造成使用者在該網路的連線中斷。為了解決這樣的問題，研究人員提出許多種方法，如 Mobile IP 針對 Layer 3 的換手提出一種解決方案，但是使用 Mobile IP 需要先擴充網路裝置的能力，成本較高。另一方面 Multipath TCP 被提出，它擴充原本 Layer 4 TCP 的能力，讓使用者的應用程式可以同時用兩種介面啟始 TCP 連線，當其中一個介面連線中斷時，應用程式可以利用依然存在之另一個介面連線繼續進行通訊，讓使用者感覺不出連線中斷，間接解決了異質網路換手的問題。但是使用者同時用兩個介面發起連線的同時也造成無線資源的浪費。因此 Multipath TCP 提供了一個選項，讓使用者可以在 Multipath TCP 連線建立後，只使用其中一個介面的連線，另一個連線當作備援。本論文在 NS2 的環境下實做 Multipath TCP 的該選項並提出一個跨階層的方案幫助 Multipath TCP 在 WiFi 與 WiMAX 間進行換手，並評估其結果。結果證明在大部分的情況下，本方案可以幫助 Multipath TCP 在異質無線網路下快速並順利地換手。

A Study on Vertical Handover Scheme over Heterogeneous Wireless Network

Student: Po-Chien Wang Advisor: Dr. Yaw-Chung Chen

Institute of Network Engineering
National Chiao Tung University

ABSTRACT

In recent years, various wireless technologies such as Wi-Fi, WiMAX and LTE have advanced very rapidly. These technologies allow users to access a variety of services provided on the Internet anytime and anywhere using mobile devices. Computation power of mobile devices also becomes very powerful, and able to provide multihoming connection to heterogeneous wireless networks at the same time. Therefore, vertical handover becomes an essential issue. A mobile node moving out of the coverage area of a wireless network will cause disruption of the connection. And Mobile IP technique was proposed to provide IP layer mobility, however its deployment cost is high because Mobile IP needs support of network backend. On the other hand, Multipath TCP is proposed and it extends TCP to make applications able to initiate connections using different interfaces concurrently. If a connection from one of the interface becomes corrupt, a connection from the other interface can still be used to sustain the session of the application. This makes user not aware of the corruption of the connections, and solves vertical handover problem indirectly. But, Multipath TCP causes waste of radio resource because it keeps multiple connections from both interfaces concurrently. Multipath TCP also provides an option which allows a user to direct the traffic flow to one of the interface, while connection from the other interface is just for backup. In this thesis, we implement the priority option of Multipath TCP and propose a cross-layer scheme to support handover between Wi-Fi and WiMAX with Multipath TCP in NS-2, and evaluate the performance and save the radio resource. Results show that our scheme can support very fast handover under heterogeneous wireless network with Multipath TCP in a common scenario.

Acknowledgement

能完成這篇論文首先要感謝我的指導老師陳耀宗教授，老師指引我研究的方向並給予我許多實際的建議，老師的建議也帶給我許多研究上的靈感。老師也細心地檢視並指導我的論文，讓文句的描述更精確、內容更完整。也感謝蔡文能教授與李春良教授在口試時給予意見，指正我研究中的失誤與其修正的方法。

最後要感謝實驗室的所有成員，尤其是跟我同屆的同學，他們平時給我不少鼓勵並且在課業與生活中互相支援，減少了我許多不必要的煩惱，讓我能專心地作研究。

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

In recent years, various wireless network technologies, such as Wireless Fidelity (Wi-Fi), Worldwide Interoperability for Microwave Access (WiMAX) and Long Term Evolution (LTE), have advanced very rapidly. These allow users to access a variety of services provided on the Internet anytime and anywhere. Computation power of mobile devices, such as PDA and smart phone, is becoming more and more powerful nowadays. The mobile devices are now powerful enough to provide multihoming ability which makes attachment to heterogeneous wireless networks at the same time possible. Since the signal coverage of wireless networks using different technologies is also different, it's quite possible that a user moves out of a wireless network with smaller signal coverage area but still stays connected to the wireless network with larger signal coverage area. However, user moving out of the signal coverage area of a wireless network will cause connection using that network disrupted, and this in turn will cause the application service being interrupted. Therefore, vertical handover that deals with such problem becomes an essential issue to be studied.

To make the connections persistent when a handover happens, many approaches has been proposed. Mobile IP [3] was proposed to provide IP layer mobility, however the cost of its deployment is high because Mobile IP needs support of network backend and the related software needs to be installed in a mobile node. Session Initiation Protocol (SIP) [6] is also proposed to provide session mobility at application layer, but it needs to modify the application software and also needs support of network backend like Mobile IP. A transport layer approach, Multipath TCP (MPTCP) [9] [10] is proposed and it extends TCP to make application software capable to initiate connections using different interfaces concurrently. If a connection from one of the interfaces becomes corrupt due to the weak signal, a connection from the other interface can still be available to sustain the session of the application. This makes user not even aware of the connection used by the application is corrupt, thus solves vertical handover problem indirectly. Moreover, MPTCP is designed to be compatible with TCP so it needs neither modification of application software nor any support of network backend. But, MPTCP may cause waste of wireless network radio resource [11] because it creates multiple connections from multiple interfaces concurrently. Therefore Multipath TCP also provides an option which allows a user to direct traffic flow to one of the interface and in the mean time the connection from the other interface is just for backup.

In this thesis, we implement the priority option of Multipath TCP and propose a cross-layer scheme to support fast handover of Multipath TCP between Wi-Fi and WiMAX and save the radio resource. We evaluate the scheme by doing simulations in Network Simulator 2 (NS-2) [13]. Results show that our scheme can provide a very fast vertical handover between heterogeneous wireless networks in a common scenario.

Chapter 2

Background

In this chapter, we first explain the concept of multihoming. Then, we introduce some important concepts of Transmission Control Protocol (TCP), which explains why the connections created by TCP get broken when a handover happens. We also introduce extensions of TCP like Selective Acknowledgment TCP (TCP SACK), and MPTCP.

2.1 Multihoming

Traditionally, every computer has only one interface attached to itself, for example Ethernet card is a typically interface. However, recent computers have become more and more complicated in the network interfaces. Usually they have more than one interfaces attached to them, and these interfaces include wireless network interfaces to exploit mobility. Thus, mobile devices can have more than one IP addresses at the same time. And they usually connect to different kinds of network; say Wi-Fi and WiMAX networks or Wi-Fi and LTE networks, which usually operated by different ISPs. Such devices are called multihomed.

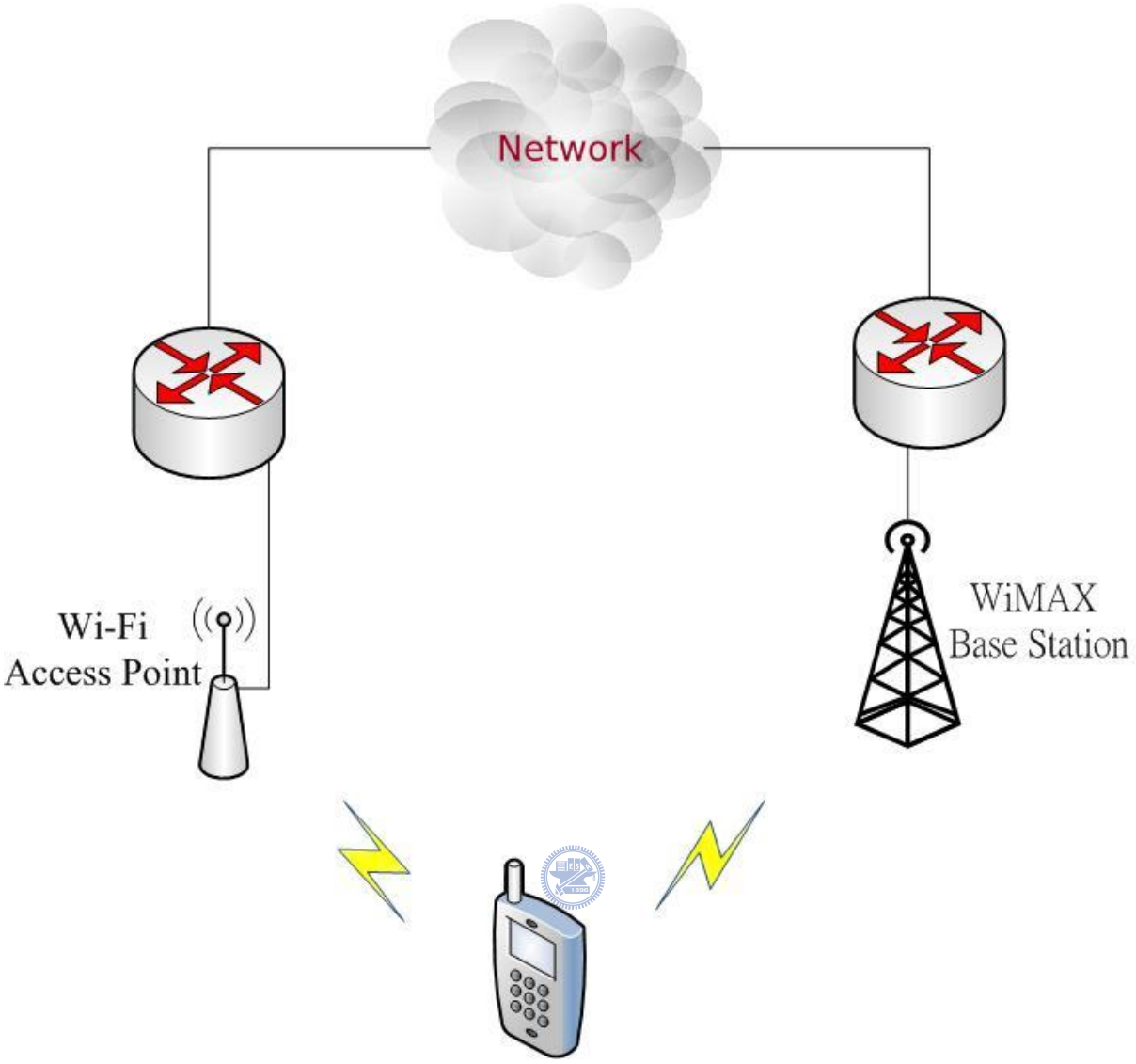

Figure 2.1 Multihomed mobile device

2.2 Review of TCP

2.2.1 TCP 3-way handshake

TCP [1] is a connection-oriented protocol, that is, before two hosts begin to transfer data between each other, they must setup a logical connection established between them, therefore a three-way handshake must be carried out first.

After the connection is established, it is identified by a 5-tuple, that is (protocol,

local-address, local-process, foreign-address, foreign-process). And every connection of a host is identified by the 5-tuple uniquely. If two connections with at least one of the 5-tuple are different, then they are identified as different connections. If a mobile node changes its point of attachment then it must be assigned a different IP address, which would make the connection identified by the original address become no longer usable. This explains why the connection of application software will get corrupted when a handover happens.

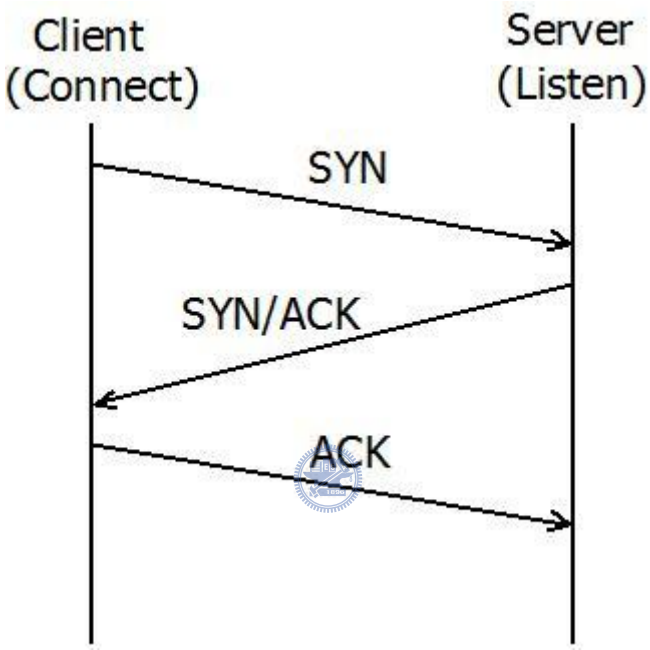

Figure 2.2 TCP 3-way handshake

2.2.2 TCP SACK

In wireless environment, packet loss usually happens more frequently than wired environment. The purpose for TCP SACK [2] is designed to improve the TCP Multiple Loss problem. TCP uses a cumulative acknowledgment scheme in which received segments that are not the first segment it should receive are not acknowledged. To find out each lost packet , the sender have to wait a round trip time or unnecessarily retransmit segments that have been correctly received by the receiver. This would severely reduce the overall throughput of TCP.

Selective Acknowledgment (SACK) is a mechanism which corrects this behavior in the face of Multiple Loss problem. With selective acknowledgments, the data receiver informs the sender about all segments that have been received successfully, so the sender needs only to retransmit the segments that have been actually lost. However, if there are multiple consecutive acknowledge packets get lost in the network, it's impossible to inform the sender which packets are correctly received by the receiver. The sender only needs to wait for the packets to be timeout then retransmits them. For example, if the receiver has received segments 1,3,4,6, and 7 then it can send a segment with acknowledgement set to 2 and the following SACK option back to the sender.

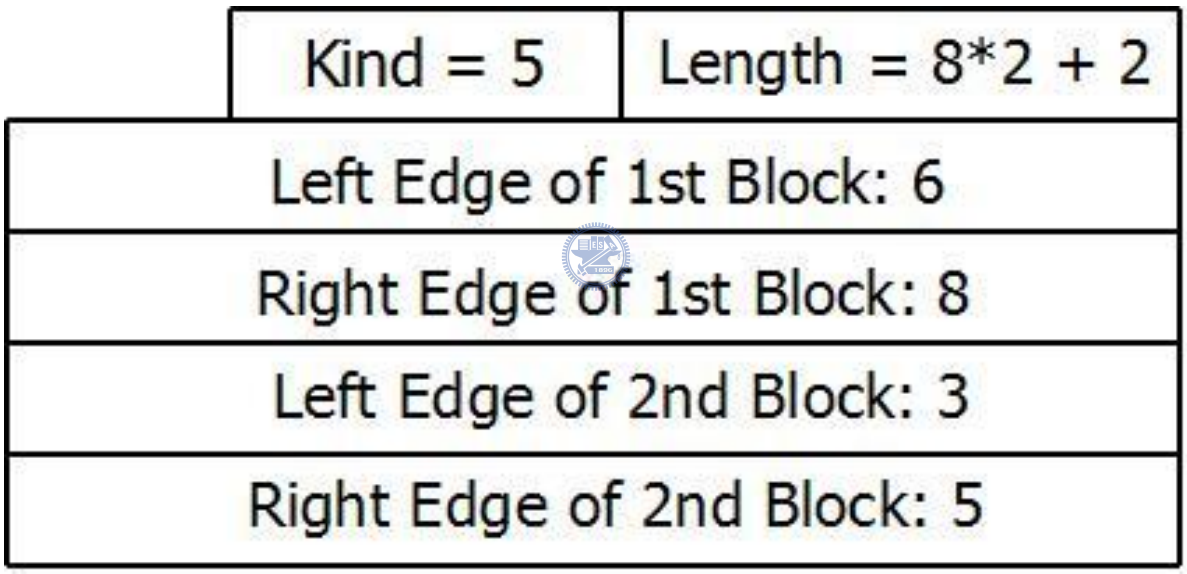

Figure 2.3 TCP SACK option example

2.3 MPTCP

2.3.1 Objectives of MPTCP

MPTCP is developed by the IETF's MPTCP working group. It develops mechanisms that add the capability of simultaneously using multiple paths to a regular TCP session.

Each TCP connection has only one path; however there often exist multiple paths between sender and receiver because of the existence of multihomed host. To take advantage of multiple paths, MPTCP can be used. MPTCP has some design objectives:

- **Improve throughput**

  MPTCP uses multiple paths to gain more throughputs, which can't be worse than just using single path TCP.

- **Improve resilience**

  If a segment can be transmitted over one path of MPTCP, then this segment can also be transmitted over the other available paths of MPTCP. That is, an MPTCP session must be no less resilient than single-path TCP.

- **Application compatibility**

  If an application is designed using single-path TCP API, then this application need not to be modified for using MPTCP.

- **Network compatibility**

  It is desirable for MPTCP to be compatible with NAT [4], firewall and other middle boxes existed on the Internet.

- **Compatibility with other network users**

  MPTCP flows and single-path TCP flows must coexist gracefully. Multiple MPTCP flows at a shared bottleneck must share bandwidth with each other fairly.

2.3.2 MPTCP 6-way handshake

Therefore if we use MPTCP, we can gain the above benefits. Fig. 2.4 illustrates how a MPTCP session is established.

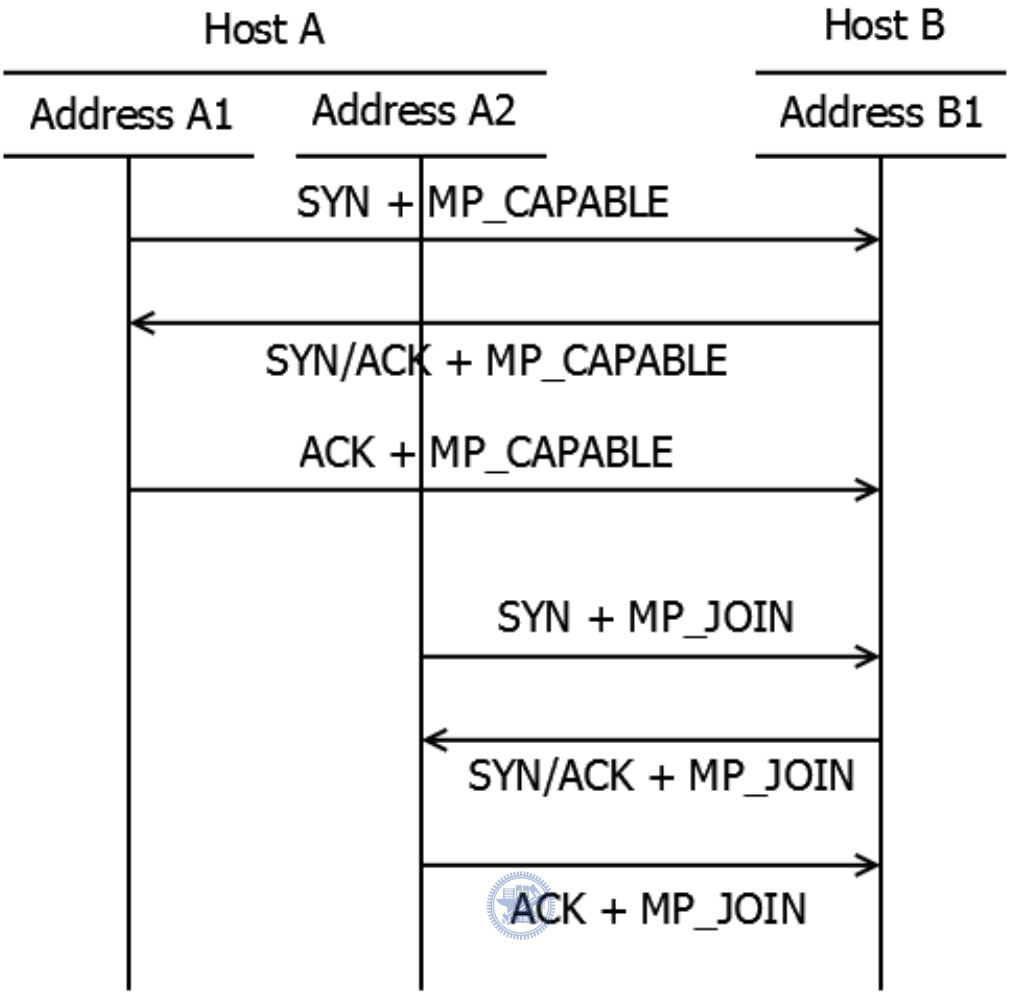

Figure 2.4 MPTCP 6-way handshake

1. Host A uses Address A1 to connect to Address B1 of Host B with SYN flag set like regular TCP. This packet carries the MP_CAPABLE option to tell Host B that Host A is capable of running MPTCP.
2. Host B responds Host A with packet in which SYN and ACK flags are set like regular TCP. This packet also carries the MP_CAPABLE option to tell Host A that Host B is capable of running MPTCP, too.
3. Host A responds Host B a packet with ACK flag set like regular TCP. Because Host A receives the MP_CAPABLE option, it responds Host B a packet carrying MP_CAPABLE option, too. At this moment, the first subflow (A1 ←→ B1) of MPTCP

connection is established.

4. Like regular TCP, Host A uses Address A2 to connect to Address B1 of Host B. But this time, the packet carries the MP_JOIN option to tell Host B that a new subflow of MPTCP connection is to be created.
5. Host B responds Host A with a packet carrying MP_JOIN, too.
6. Host A responds Host B with a packet carrying MP_JOIN. At this moment, a new subflow of MPTCP connection is established. Now, the MPTCP connection has two TCP subflows that can be used by the same session.

A MPTCP connection can be viewed as a set of TCP subflows. If one of the subflows becomes unusable, then the MPTCP still has another usable subflow. Therefore, the session that uses MPTCP connection will not feel the connection down if just one of the subflows is corrupted except that all subflows get corrupted at the same time.

2.3.3 MPTCP Data Sequence Signal Option

From application's perspective, it has a data stream to send, but finally MPTCP will distribute the data stream to send to independent subflows. MPTCP has to find a way to tell the receiver how the segments received by the receiver are to be reassembled to the original data. Here we introduce the data sequence signal (DSS) option of MPTCP.

The DSS option comprises two parts of important information needed by the operation of MPTCP. The first part of important information is Data Sequence Mapping, the other is Data Ack. The DSS option format is illustrated as follows:

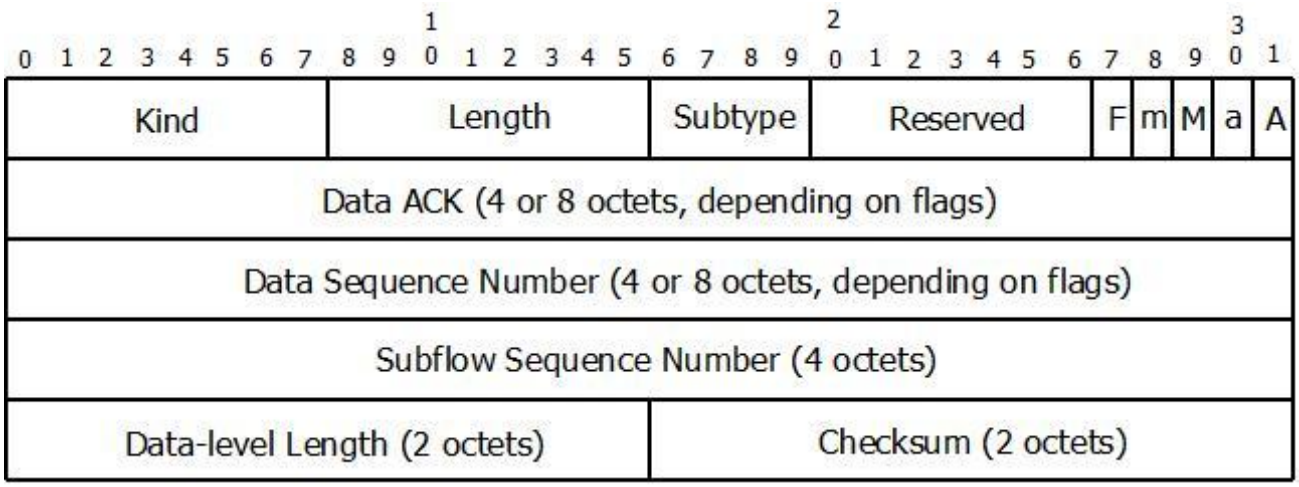

Figure 2.5 Data sequence signal (DSS) option format

We first explain the flags on the upper right.

- A: When set, the Data ACK is presented; otherwise the Data ACK is not presented
- a: When set, the Data ACK is 8 octets; otherwise the Data ACK is 4 octets.
- M: When set, Data Sequence Number, Subflow Sequence Number, Data-level Length, and Checksum is presented; otherwise those are not presented.
- m:When set, Data Sequence Number is 8 octets; otherwise Data Sequence Number is 4 octets.
- F: When set, this option is used to indicate the end of the MPTCP session. There is no more data to send by the sender.

The 3-tuple (Data Sequence Number, Sub-flow Sequence Number, Data-level Length) is called a Data Sequence Mapping. It's used to reassemble the data received from a subflow to the data the application can consume.

- Data Sequence Number: This number specifies the sequence number in the Data-level, i.e. the final position on which the segment should be placed in the application.
- Subflow Sequence Number: This number is used by a TCP subflow to identify the segment sent. This number locates the temporary location at which the segment should

be placed in a subflow. This number is generated by the same way as a regular TCP connection. To reassemble to the data the application can consume, this number has to be mapped to Data Sequence Number so that the MPTCP session can reassemble it to the data the application can consume.

- Data-level Length: This number specifies the length of data starting from the Subflow Sequence Number or the Data Sequence Number.

Data ACK is Data Acknowledgements, this number is generated according to Data Sequence Number and is a cumulative acknowledgement in data level. This is used by receiver's MPTCP session to tell the sender's MPTCP session which segment in data level to send next.

2.4 Comparison between IEEE 802.11 and IEEE 802.16

There are some different characteristics between WiFi and WiMAX wireless technologies. Table 2.1 summarizes important differences between them. In short, from throughput's perspective, WiFi is higher than WiMAX, and from coverage area's perspective, WiMAX is larger than WiFi. Because the coverage area of WiMAX is very large, we expect a mobile node is always connected to some WiMAX BS when it moves at a low speed.

Table 2.1 Comparison between 802.11b/g and 802.16d

	802.11b/g	802.16d
Frequency band	2.4GHz	3.5/5.8GHz
Data rate	11Mbps/54Mbps	Point to point: 70Mbps
		Point to multipoint: 54Mbps
Coverage area	40-100m in radius	30km in radius

Number of users	Dozens	Thousands
Throughput	Higher	Lower

2.5 Related works

Mobile IP provides mobility support for mobile hosts in network layer; however it needs the support of network backend because it needs the deployment of mobility agents such as home agent and foreign agent. It also has the triangular routing problem, which means that a packet routed by Mobile IP must pass through the home agent and therefore the path is not optimized path between the two end hosts. The operation of Mobile IP also needs to do some extra encapsulation/decapsulation of packets, which in turn causes a lot of overhead. And when Mobile IP is operated with NAT, the IP-in-UDP tunnel [5] created between the HA and the NAT further increases the overhead in terms of keep alive feature and encapsulation/decapsulation. The drawbacks of providing mobility in IP layer encourage us to transfer the mobility support to transport layer [17].

Stream Control Transmission Protocol (SCTP) [7] is a reliable transport protocol. It locates at the same layer as UDP and TCP. Differing from TCP and UDP, the SCTP is used for ‘multi-streaming’ and ’multi-homing’. The multihoming ability of SCTP enables SCTP to be used for Internet mobility support, without support of network routers or special agents. Mobile SCTP (mSCTP) [8] is SCTP extended by Dynamic Address Reconfiguration (DAR). And it use address configuration (ASCONF) message to dynamically add or remove IP address of an SCTP association. However, modern NAT-capable routers don’t even recognize the SCTP protocol number [12]. To add SCTP support to a NAT, the NAT needs to understand the operation of SCTP. The operation is very complicated and not as easy as TCP/UDP for NAT. Therefore, an SCTP association must use only public IP addresses. However, we are facing IP addresses shortage in recent days, so it’s better to avoid public IP address if possible.

MPTCP supports multihoming and also supports dynamically configuration of IP address like mSCTP, and MPTCP is NAT-compatible as mentioned before. So MPTCP can use private IP address when initiating MPTCP connection. However, the default operation of MPTCP wastes radio resource because it uses multiple subflows concurrently. It’s not very kind to Internet Service Provider (ISP)

Chapter 3

Proposed Scheme

In this chapter, we first describe the objectives of our proposed scheme. Then, we describe the concepts of the scheme. Finally, we describe how we implement the proposed scheme in Network Simulator 2 (NS2).

3.1 Objectives of the proposed scheme

- We want the proposed scheme not to need any support of network backend.
- Because TCP is still the most widely used transport layer protocol, we want to provide the solution based on TCP.
- We want our proposed scheme to have some way to know the quality of the connection.
- We want our proposed scheme to be able to avoid the waste of radio resource, so that the traffic can always be directed to the more efficient wireless network only.
- We want the vertical handover to be performed bi-directionally, that is; we can switch from Wi-Fi to WiMAX and from WiMAX to Wi-Fi.

3.2 Concepts of proposed scheme

- We use the protocol that supports multihoming to exploit the advantage of multiple addresses, instead of using only one address all the time.
- We modify the behavior of MPTCP to make it supporting the vertical handover of MN and modify nothing else.
- We use cross-layer [18] messages to make MPTCP know the signal strength, which is a metric of the connection of a mobile device. And we use the metric to determine when to perform a handover.
- To reduce the waste of radio resource reduce, we make the MPTCP to select the subflow with higher throughput to generate traffic, and minimize the data traffic appeared at other subflows.
- Each subflow of MPTCP has binding to one of the host's interfaces, so if we can change the policy the MPTCP selects the subflow, then we can always switch from Wi-Fi to WiMAX or from WiMAX to Wi-Fi.

3.3 Implementation and detailed description of proposed scheme

We download a patch that can make NS-2.34 to support MPTCP [15]. MPTCP supports multihoming originally, so it can be used to perform vertical handovers. And the MPTCP implementation is based on TCP SACK. We analyze the source code and find that the MPTCP sender can be modified to use only one subflow to send application data. This finding can be used to reduce the waste of radio resource. And the draft of MPTCP also states that the

MPTCP has a priority option which enables MPTCP to direct the traffic to a specific subflow and the other subflows are used just for backup.

In NS2, every object of class Packet has a variable of class PacketStamp named txinfo_. The txinfo_ variable records the signal strength when receiving the packet. This information can be directly read by MPTCP although in fact this is a piece of cross-layered information.

To make NS-2.34 support simulation of wireless heterogeneous networks, we download the National Institute of Standards and Technology (NIST) [14] WiMAX extension for NS-2.31, and manually install it on NS-2.34. Now, our NS2 can support simulation of wireless heterogeneous networks, that is, Wi-Fi and WiMAX.

It’s difficult to simulate a node with multiple interfaces in NS2. However, because we are modifying the transport layer, the packet routing between interface nodes doesn’t really bother us at all.

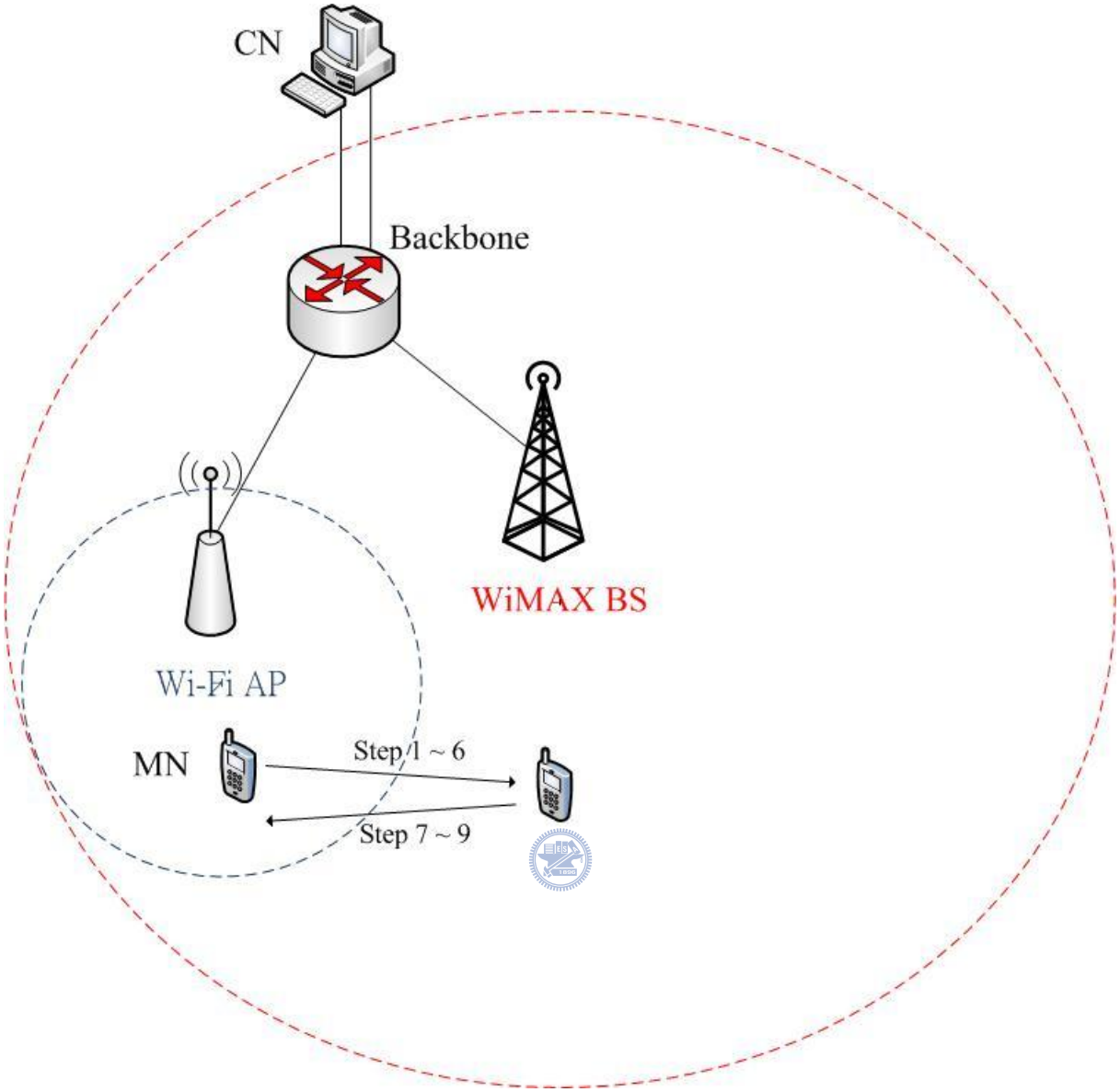

Figure 3.1 Scenario of the proposed scheme

Our proposed scheme is described as follows:

1. We assume that the MN (receiver) is located at the overlapped area of Wi-Fi and WiMAX signal coverage at the beginning. Because of that, the MN is able to create two subflows using MPTCP.
2. The CN (sender) executes some application over MPTCP (For example, CN sends a file using FTP). The CN starts the MPTCP session and 6-way handshake with MN.
3. For every received packet (including the handshake packet) of MN, the MPTCP of MN can

detect the signal strength of its interfaces using txinfo_ variable of Packet. Then the MPTCP of MN decides to give Wi-Fi a higher priority and sends Acks to CN with priority option to tell CN that MN wants to use the subflow with Wi-Fi interface to receive data. For convenience, we call the subflow binding to MN's Wi-Fi interface subflow1 and the subflow binding to MN's WiMAX interface subflow2.

4. The CN receives the Ack and sends data to MN via subflow1 but nothing via subflow2.
5. The MN begins to move and, as time goes further, the MN is going to moves out of the signal coverage of Wi-Fi. However, before MN moves out of the signal coverage of Wi-Fi, its MPTCP session would detect the signal strength of Wi-Fi weakening via txinfo_ variable. And the MPTCP of MN sends Acks to CN via subflow1 with priority option that subflow2 has a higher priority than subflow1 to ask the MPTCP of CN to redirect the data traffic to subflow2.In other words, when MN performs a handover from Wi-Fi to WiMAX, it sends a request to ask CN to redirect the data traffic from Wi-Fi to WiMAX via subflow1.
6. (a) If the MPTCP of CN receives the Ack with the priority option the MPTCP of MN sent via subflow1, then it redirects the data traffic via subflow2 at once. (b) If the MPTCP of CN doesn't receive the Acks carrying the priority option, then after a timeout within which the MPTCP of CN doesn't receive any Ack, the MPTCP of CN redirects the data traffic to subflow2 automatically.
7. As time goes by, the MN, for some reason, returns to the signal coverage of WiFi and the MPTCP of MN detects that the IP address of WiFi interface is configured and ready to send packets, then the MPTCP sends Acks via subflow2 with priority option that subflow1 has a higher priority than subflow2. Here, we assume MN is still in the coverage of WiMAX. In other words, when MN performs a handover from WiMAX to WiFi, it sends a request to

ask the CN to redirect the data traffic from WiMAX to WiFi via subflow2.

8. The MPTCP of CN receives the Ack with the priority option and redirects the data traffic to subflow1. The Ack will be received by the MPTCP of CN with nothing wrong, because subflow2 is bound to WiMAX and doesn't get corrupted.
9. The MN stays at the signal coverage of WiFi and the MPTCP of MN continues to receive the data traffic generated by CN.
10. If MN decides to move out of the signal coverage of WiFi again, then go to step 3 and repeat.

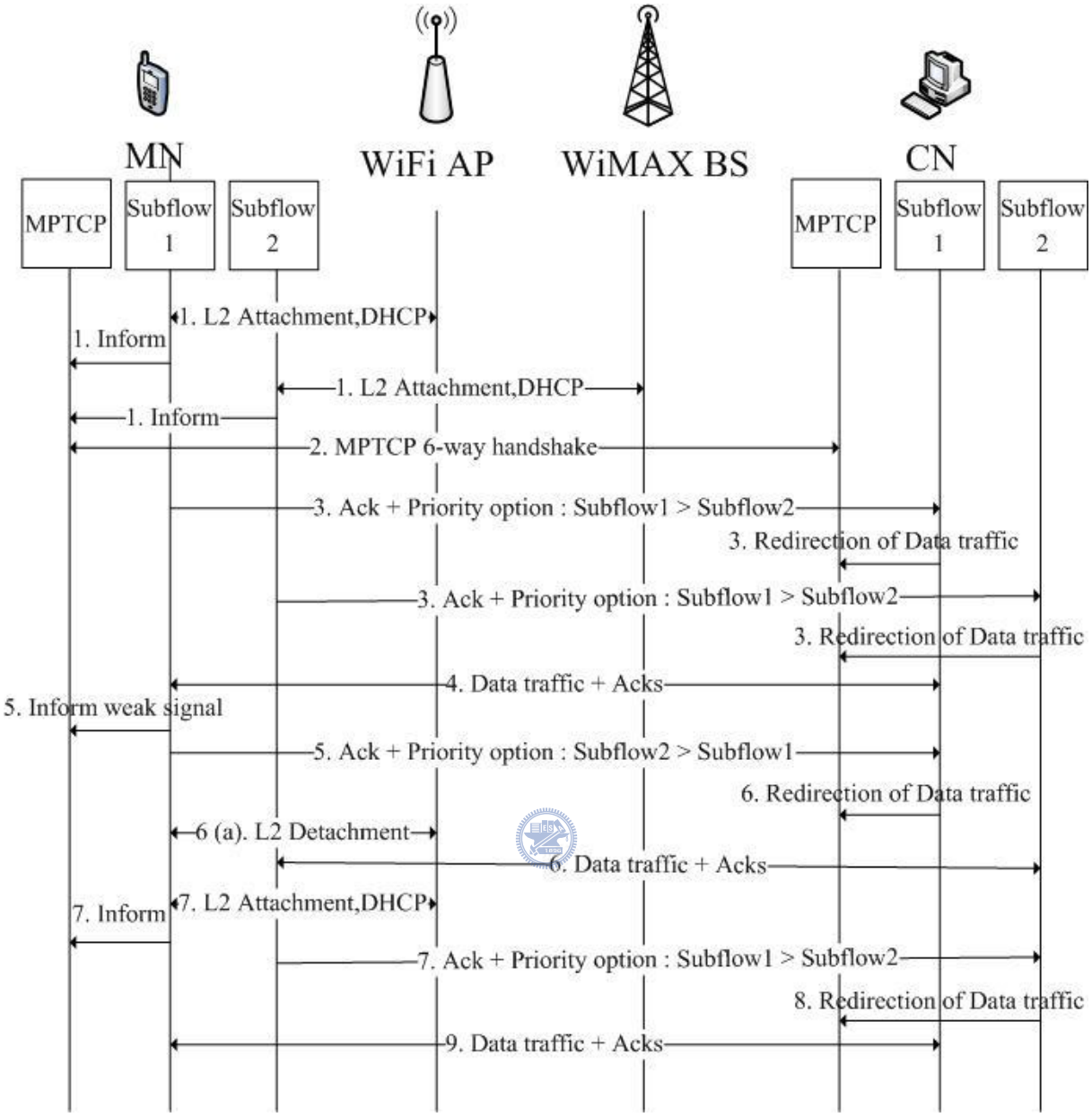

Figure 3.2 Message flow of the proposed scheme in case 6(a)

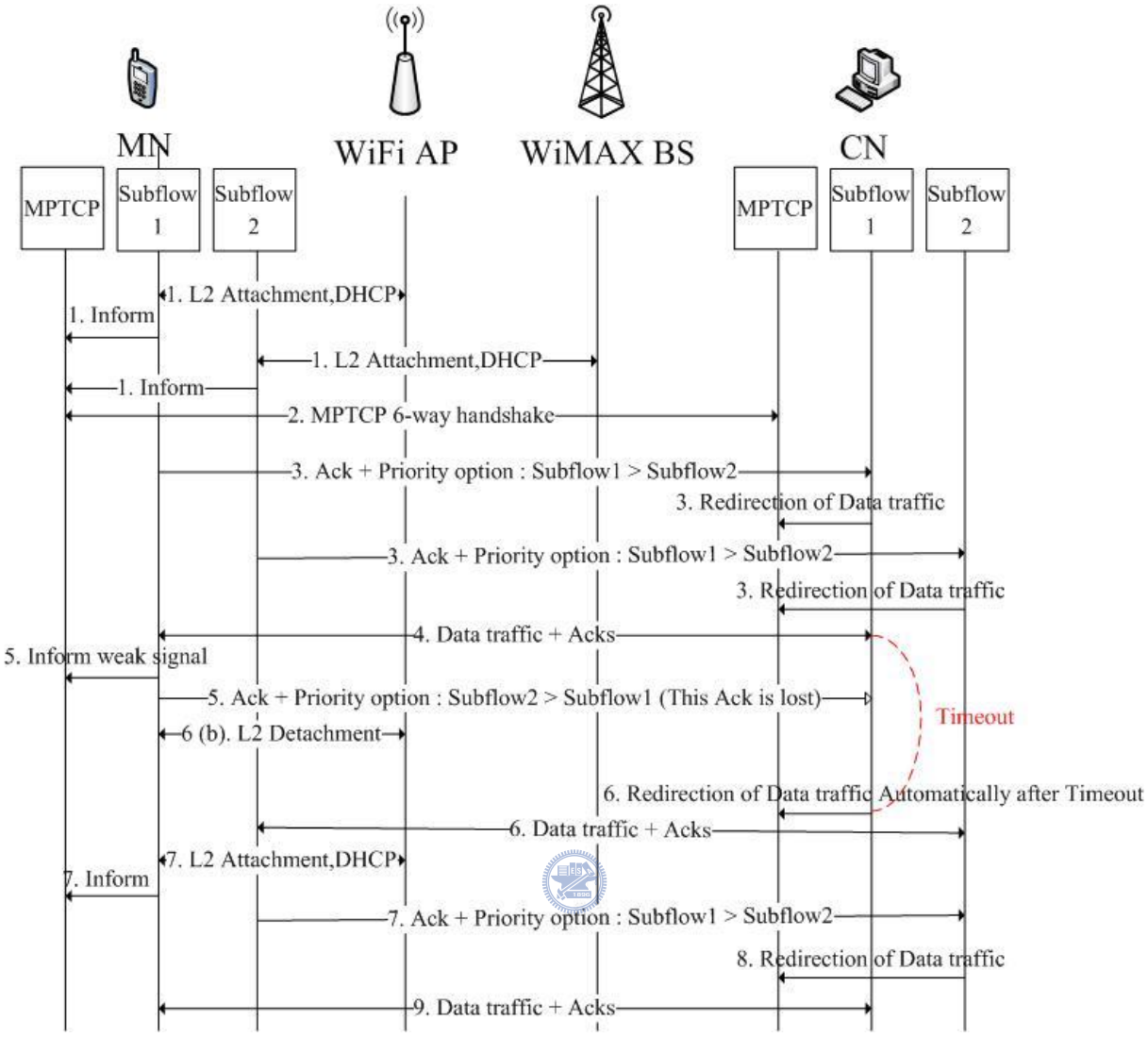

Figure 3.3 Message flow of the proposed scheme in case 6(b)

We now explain the reason why the step 7 is legal. Here, we assume that when MN returns to the signal coverage area of WiFi, it will be assigned the same IP address as before it moves out of the signal coverage of WiFi. It is possible to configure AP and assign IP address to it based on the MAC address of MN's WiFi interface. And we assume the subflow1 information in MN's MPTCP session is not removed because MN moves out of the signal coverage of WiFi.

If MN is not assigned the same IP address as before, it's still OK. It's because MPTCP supports dynamically add new subflow if handover of MN happens. If MN returns to WiFi

and is assigned a new IP address, it can create a new subflow with the new IP address and do MP_JOIN with CN via the new subflow. The MP_JOIN operation can be referred to the step 4~6 of the 6-way handshake of MPTCP. This operation enables both CN and MN to know that a new subflow is to be added in the MPTCP session and the new subflow is used to transfer the data traffic of the same application as before. If the subflow with the old IP address won't be used again, then it can be removed safely.

Chapter 4

Simulation Result

In this chapter, we describe the simulation scenario, which includes the parameters, and the figures. Then we analyze the result based on our proposed scheme.

4.1 Scenario I

4.1.1 Network topology

The CN is a multihomed node with two wired interfaces. However, CN needs not to be a multihomed interface in the proposed scheme. We let CN be multihomed just for simulation convenience. The MN is a multihomed node with two wireless interfaces, WiFi and WiMAX. The WiFi AP has a coverage of 40m radius, and the WiMAX BS has signal coverage of 500m radius.

The bandwidth of the first interface of CN is 100Mbps, and that of the second interface of CN is 100 Mbps, too. The bandwidth of the link connecting the backbone and WiFi AP is

100 Mbps, and the bandwidth of the link connecting the backbone and WiMAX BS is 100Mbps, too. The bandwidth of the WiFi interface of MN is 54Mbps, and the bandwidth of the WiMAX interface of MN is variable data rate. The MPTCP is used to generate TCP data packets.

The total simulation time is 160 seconds, and we use MPTCP/FTP traffic between CN and MN. The MPTCP/FTP traffic begins at 1 second and ends at 160 seconds.

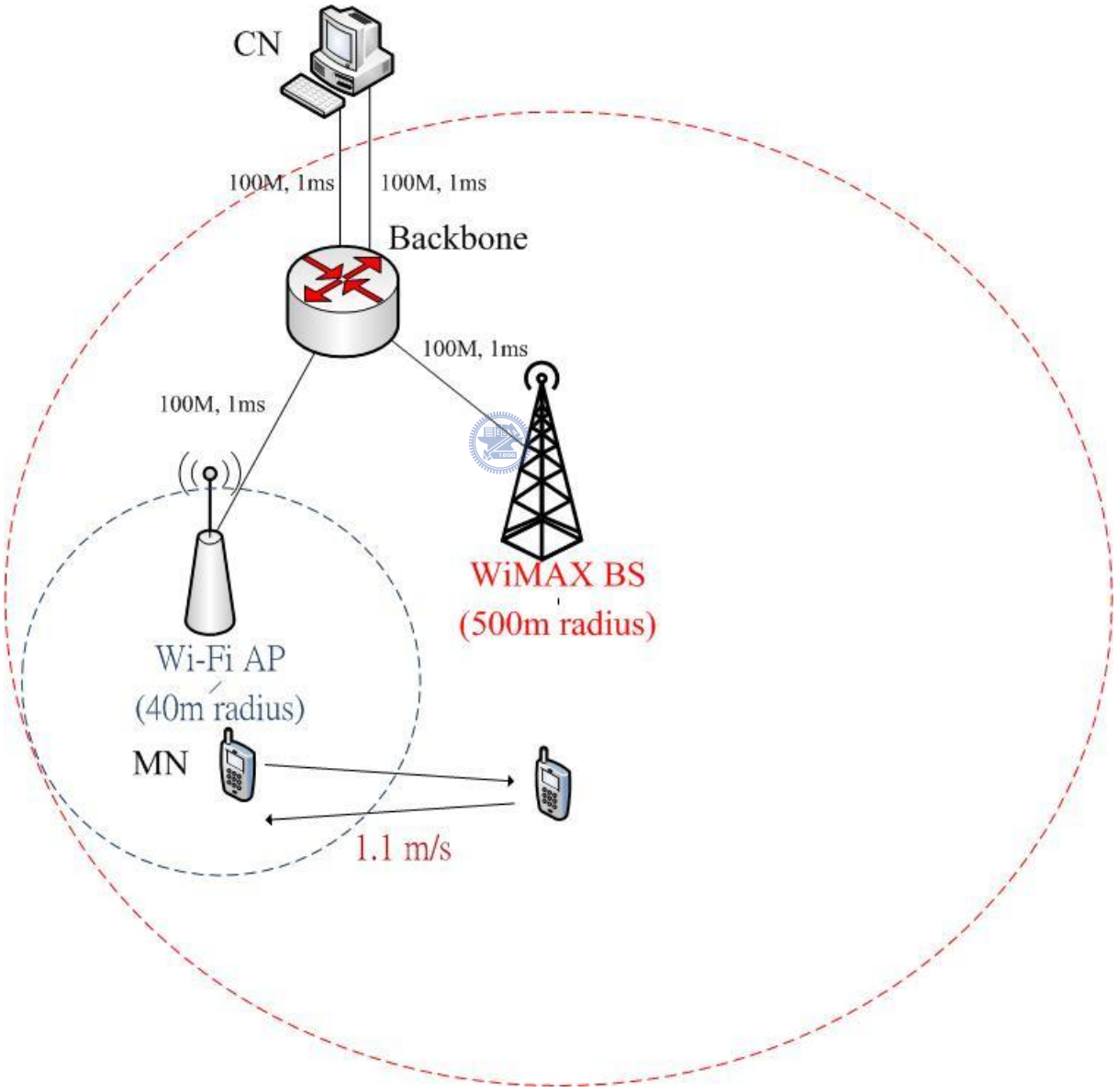

Figure 4.1 Network topology of simulation

The MN, at the beginning, is located at the signal coverage of WiFi, and then it begins to

move out of the signal coverage of WiFi to the east at a speed of 1.1 m/s. The MN begins to move at 1 second, and turns around at 80/1.1 + 1 = 73.72 seconds. The MN will return to the starting point in the end. The further detailed configuration of simulation scenario can be referred to Table 4.1.

Table 4.1 Parameters of simulation in Scenario I

Parameter	Value
Position Configuration	
Flat Grid Size	2000m * 2000m
WiFi AP Position	(999,1000)
WiMAX BS Position	(1000,1000)
MN Initial Position	(1000,990)
MN Turn-Around Position	(1080,990)
MN Speed	1.1 m/s
WiFi AP Signal Coverage	40m in radius
WiMAX BS Signal Coverage	500m in radius
Simulation Time	160 seconds
Bandwidth Configuration	
CN Interface1 ←→ Backbone	100M
CN Interface2 ←→ Backbone	100M
Backbone ←→ WiFi AP	100M
Backbone ←→ WiMAX BS	100M
Delay	1 milliseconds (ms)
WiFi Configuration	

Data rate	54M
Beacon interval	100 ms
MinChannelTime	0.005 seconds
MaxChannelTime	0.011 seconds
WiMAX Configuration	
Dcd_interval	5 seconds
Ucd_interval	5 seconds
Default modulation	OFDM_64QAM_3_4
Frame duration	0.004 seconds
Traffic Configuration	
Application	FTP
Protocol	MPTCP
Max TCP (Subflow) Transmission Window	100 segments
MPTCP Configuration	
Timeout that Triggers Redirection of Data Traffic	0.2 seconds
Signal Strength that Triggers Redirection of Data Traffic	-56.544073 dBm

4.1.2 Simulation steps

Time 0: The WiFi interface of MN attaches to WiFi AP; The WiMAX interface of MN attaches to WiMAX BS. The MPTCP session of MN enters the state of listen. The MPTCP session of CN connects to the MPTCP session of MN.

Time 1: The FTP application of CN starts. MN begins to move from position (1000, 990)

to position (1080, 990) at a speed of 1.1 m/s.

Time 73: MN reaches (1080, 990). MN begins to return to position (1000, 990).

Time 146: MN reaches (1000, 990).

Time 160: The MPTCP/FTP traffic terminates.

A value needs to be determined. That's the signal strength at which the MPTCP of MN begins to trigger the redirection of data traffic from WiFi to WiMAX.

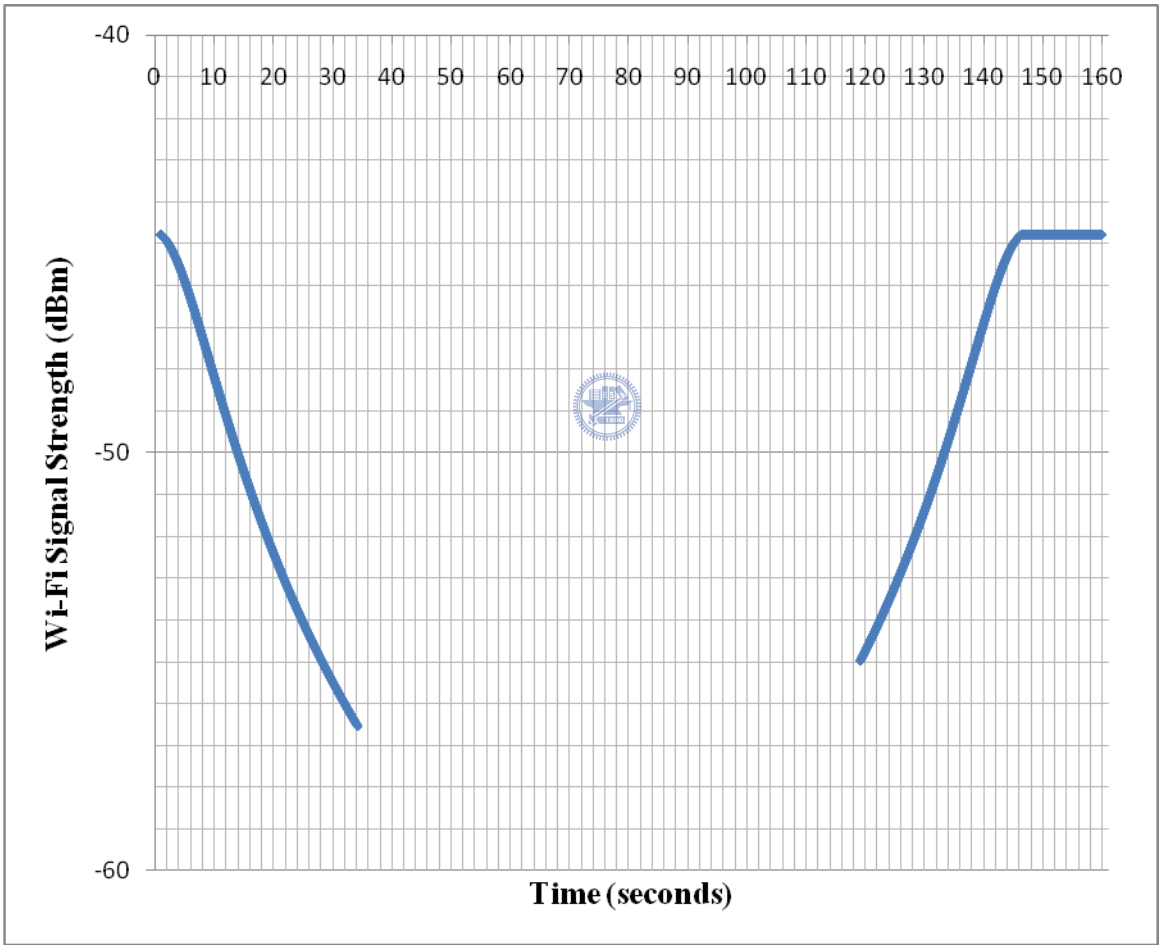

Figure 4.2 WiFi signal strength sensed by the interface of MN in simulation

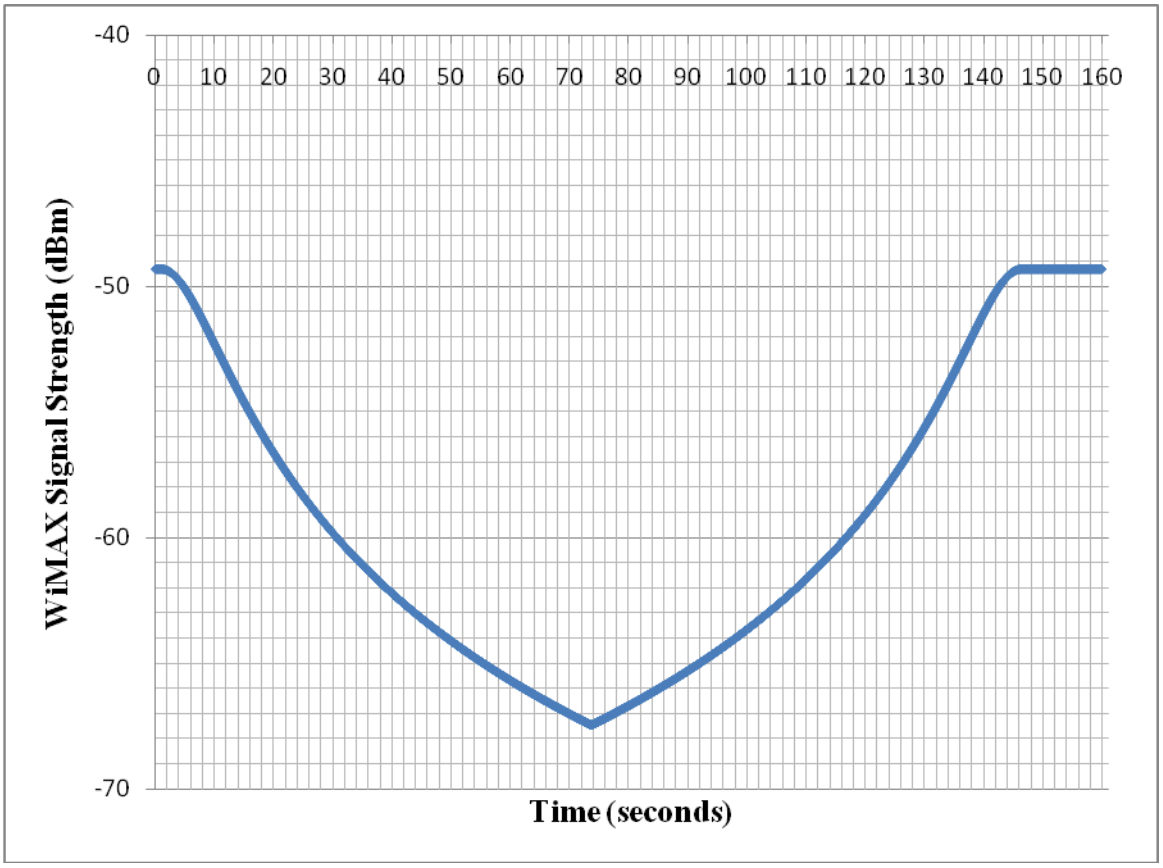

Figure 4.3 WiMAX signal strength sensed by the interface of MN in simulation

Fig. 4.2 is WiFi signal strength distribution in simulation. Fig. 4.2 is derived from the signal strength sensed by the WiFi interface of MN along the way the MN moves at a speed of 1.1m/s. Fig. 4.3 is WiMAX signal strength distribution in simulation. Fig. 4.3 is derived from the signal strength sensed by the WiMAX interface of MN along the way the MN moves at a speed of 1.1m/s. By investigaing the above two figures, we can find that the WiMAX signal strength doesn't get interrupted or disapeared. Therefore, because MN is always connected to WiMAX,we set the WiMAX to be the backup path of MPTCP,and the WiFi to be the primary path. The primary path always has a higher priority than the backup path if it is usable. If the primary path is not usable, then the backup path has a higher priority than the primary path.

Because the WiMAX interface of MN is always usable, we can just care about Fig. 4.2 to find the value of the signal strength that triggers the redirection of data traffic of MPTCP to

the backup path.

By very closely investigating Fig. 4.2, we find that the value of the WiFi signal strength that triggers the redirection of data traffic of MPTCP from the WiFi interface to the WiMAX interface is -56.544073 dBm (at a speed of 1.1m/s).We choose this value because it's a value very close to the weakest value the WiFi interface can sense, and the value doesn't trigger the redirection of data traffic of MPTCP because of the step 6(b) mentioned in section 3.3.

Therefore, when MN is moving, if MN finds that the WiFi signal strength sensed by the interface is less than -56.544073 dBm ,it will trigger the redirection of data traffic of MPTCP from WiFi to WiMAX. But when MN finds that the WiFi signal strength sensed by the interface is higher than -56.544073 dBm, it is not necessarily to trigger the redirection of data traffic of MPTCP from WiMAX to WiFi. It's because the condition that the signal strength higher than -56.544073 dBm doesn't necessarily mean the IP address configured on the WiFi interface and is ready to send packets. After the IP address is configured, the MPTCP of MN begins to trigger the redirection of data traffic.

We have mentioned that it's possible the priority option sent within the Ack by the MPTCP of MN get lost and not received by the MPTCP of CN. This condition may happen because when the MPTCP of MN sends the Ack, the MN is located out of the signal coverage of WiFi or the Ack simply get lost in the network. If the MPTCP of CN doesn't receive the Ack, then it will keep sending data traffic via the same subflow.

Therefore, we have defined a timeout for automatically redirection of data traffic of MPTCP. For example, if the MPTCP of CN doesn't receive any Ack via subflow1, then after the timeout, the MPTCP of CN will send data traffic via subflow2 automatically because subflow2 is the backup path. That is, the MPTCP of CN redirects the data traffic to subflow2

because of timeout. In our experiment, we have defined the timeout to be 0.2 seconds. If we define the timeout to be less than 0.2 seconds, then the redirection of data traffic of MPTCP will get triggered too frequently because the congestion of the network or the bandwidth of WiFi/WiMAX make the RTT of data traffic higher than 0.2 seconds. This will make the performance of our proposed scheme degrade a lot.

4.1.3 Simulation results

Fig. 4.4 shows the cumulative number of packet loss as MN moves at a speed of 1.1m/s. The handover of MN happens at time 34.35 seconds. Before the time 34.35 seconds, the MPTCP of MN use subflow1 to receive the data traffic generate by the MPTCP of CN.

The number of packet loss increases very slowly when MN is in the signal coverage of WiFi because WiFi provides higher performance and lower rate of packet loss. In fact, there is still a little data traffic appeared at subflow2 before the time 34.35 seconds, because MPTCP needs to perform 6-way handshake. However, after the connection of subflow2 of MPTCP is established, the subflow2 stays at the connection established state and doesn't send or receive any data traffic because the MPTCP of CN doesn't generate data traffic at subflow2. There is no data traffic appeared at subflow2 after the subflow2 of MPTCP enters the state of connection established and before the redirection of data traffic of MPTCP is triggered.

At time 34.35 seconds, the MPTCP of CN receives the Ack with priority option that subflow2 has a higher priority than subflow1 and it redirects the data traffic to subflow2 via subflow1. After the redirection of data traffic, the number of packet increases much quicker than that before the handover of MN, because WiMAX provides lower performance and higher rate of packet loss.

At time 121.1 seconds, the MN returns to the signal coverage of WiFi. After the WiFi interface configures the IP address and is ready to send packets. The MPTCP of MN sends Acks with priority option that subflow1 has a higher priority than subflow2 via subflow2. The Ack triggers the redirection of data traffic of MPTCP from subflow2 to subflow1. The number of packet loss increases slower and the rate of packet loss is the same as that before the time 34.35 seconds.

At about time 146 seconds, MN returns to the starting point and the MPTCP of MN continues to receive data traffic via subflow1 till the end of simulation.

We now explain why the waste of radio resource is reduced. If before time 34.35 seconds, the MPTCP of CN generates data traffic via both subflow1 and subflow2 concurrently, then the rate of packet loss before time 34.35 seconds will be much higher because subflow2 is used to generate the data traffic according to the number of packet loss between time 34.35 seconds and 121.1 seconds. The jitter of number of packet loss will be lower if we used only subflow1 than that if both the subflow1 and subflow2 are used together. In other words, the number of packet loss before time 34.35 seconds will vary much more if MPTCP uses both subflows than that shown in the Fig. 4.4. Therefore, Fig. 4.4 demonstrates that our proposed scheme wastes only a little radio resource of WiMAX when the WiMAX interface is usable but not necessary to be used.

Fig. 4.5 shows the instant throughput of MPTCP when MN is moving at a speed of 1.1m/s. Before time 34.35 seconds, MN is in the signal coverage of WiFi and uses the subflow1 instead of subflow2 of MPTCP to receive the data traffic generated by the MPTCP of CN. And the instant throughput of MPTCP is at about 6.5 Mbps.

At time 34.35 seconds, Fig. 4.5 shows a transition of throughput from about 6.5Mbps to a very low value. At this moment, MN performs a handover from WiFi to WiMAX and sends

an Ack with priority option that subflow2 has a higher priority than subflow1 via subflow1. We can find that the instant throughput of MPTCP never reaches 0 Kbps between time 34.35 seconds and 40 seconds. It's because the redirection of data traffic of MPTCP is triggered by the Ack sent by the MPTCP of MN and is not triggered by the timeout at MPTCP of CN for redirection of data traffic. If the redirection is triggered by the timeout at MPTCP of CN, the instant throughput will degrade to 0Kbps temporarily for about 0.2 seconds. However, because we have chosen a very proper value of signal strength of WiFi that triggers redirection of data traffic, the redirection will not be triggered by timeout in our configuration.

Between time 34.35 seconds and 121.1 seconds, the MPTCP of MN can only use subflow2 to receive data traffic. At time 121.1 seconds, the MPTCP of MN finds that the WiFi interface becomes usable and is ready to send packets. So, the MPTCP of MN triggers the redirection of data traffic. The throughput shown in Fig. 4.5 at time 121.1 seconds has a smooth transition from 1.5Mbps to 6.5Mbps.

After time 121.1 seconds, the MPTCP of MN continues to receive data traffic at an instant throughput of about 6.5Mbps.

At about time 146 seconds, MN reaches the starting point. The instant throughput keeps at about 6.5Mbps till the end of simulation.

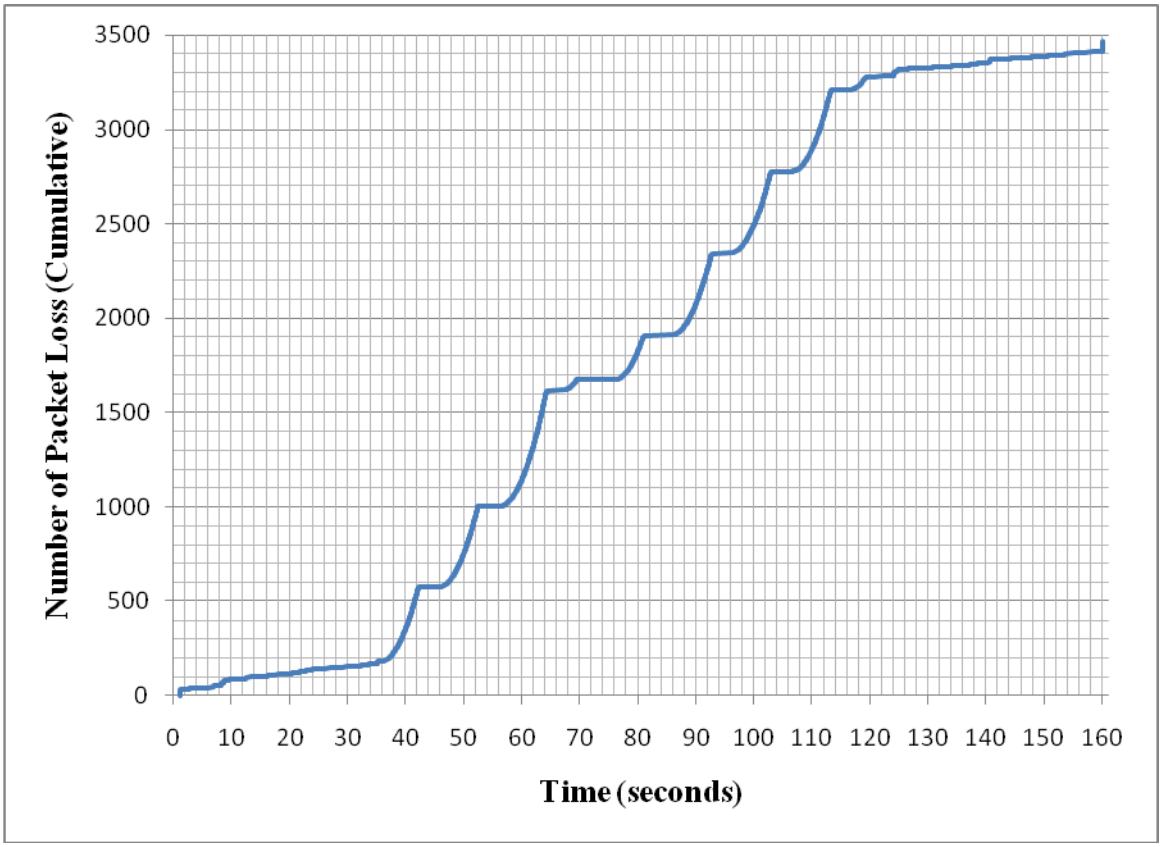

Figure 4.4 Cumulative number of packet loss in simulation

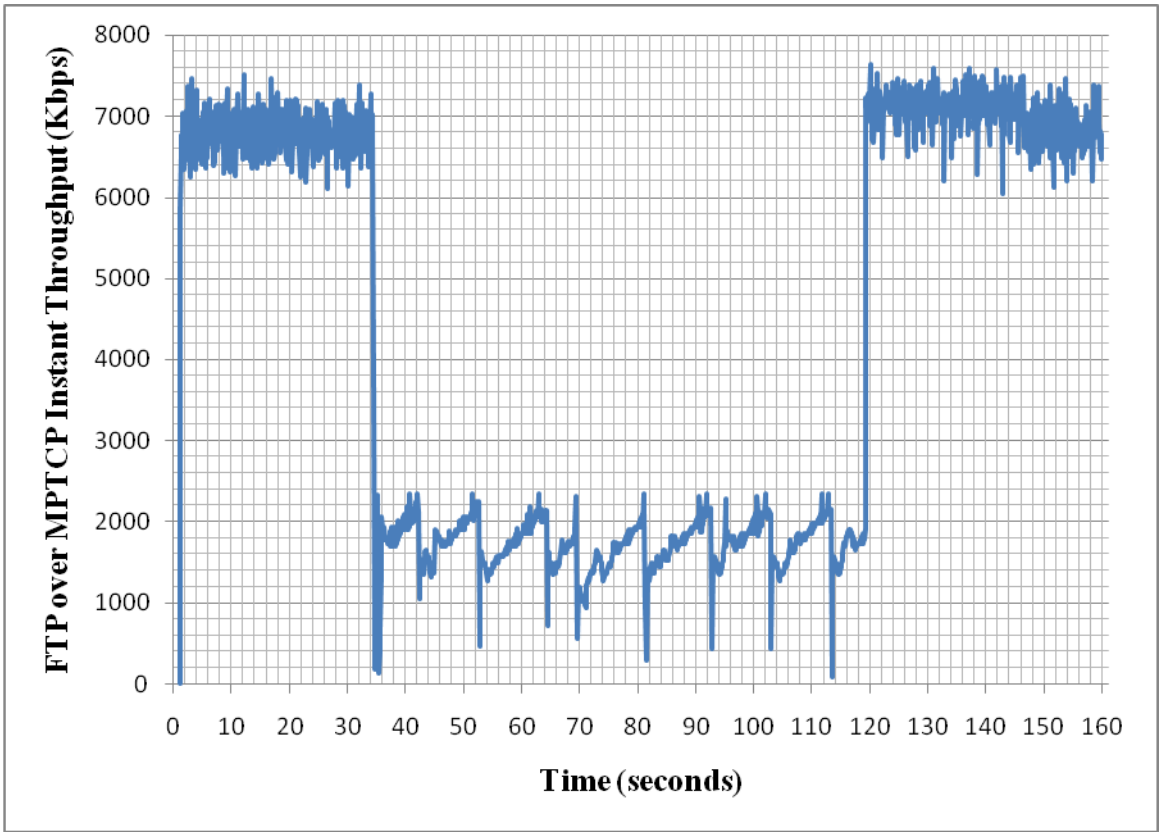

Figure 4.5 Instant throughput of FTP over MPTCP in simulation

4.2 Scenario II

This scenario is constructed to compare the proposed scheme with other vertical handover schemes in the literature [16].

4.2.1 Network topology

The parameters are configured to be the same as that of the scenario in the handover scheme in the literature.

Table 4.2 Parameters of simulation in Scenario II

Parameter	Value
Position Configuration	
MN Speed	1.0 m/s
WiFi Configuration	
Data rate	11M
Beacon interval	100 ms
MinChannelTime	0.02 seconds
MaxChannelTime	0.06 seconds
WiMAX Configuration	
Dcd_interval	5 seconds
Ucd_interval	5 seconds
Default modulation	OFDM_16QAM_3_4
Frame duration	5 ms
Traffic Configuration	
Application	FTP
Protocol	MPTCP
Max TCP (Subflow) Transmission Window	100 segments
MPTCP Configuration	
Timeout that Triggers Redirection of Data Traffic	0.2 seconds
Signal Strength that Triggers Redirection of Data Traffic	-56.544073 dBm

4.2.2 Simulation steps

Time 0: The WiFi interface of MN attaches to WiFi AP; The WiMAX interface of MN attaches to WiMAX BS. The MPTCP session of MN enters the state of listen. The MPTCP session of CN connects to the MPTCP session of MN.

Time 1: The FTP application of CN starts. MN begins to move from position (1000, 990) to position (1080, 990) at a speed of 1.0 m/s.

Time 73: MN reaches (1080, 990). MN begins to return to position (1000, 990).

Time 146: MN reaches (1000, 990).

Time 160: The MPTCP/FTP traffic terminates.

4.2.3 Simulation results

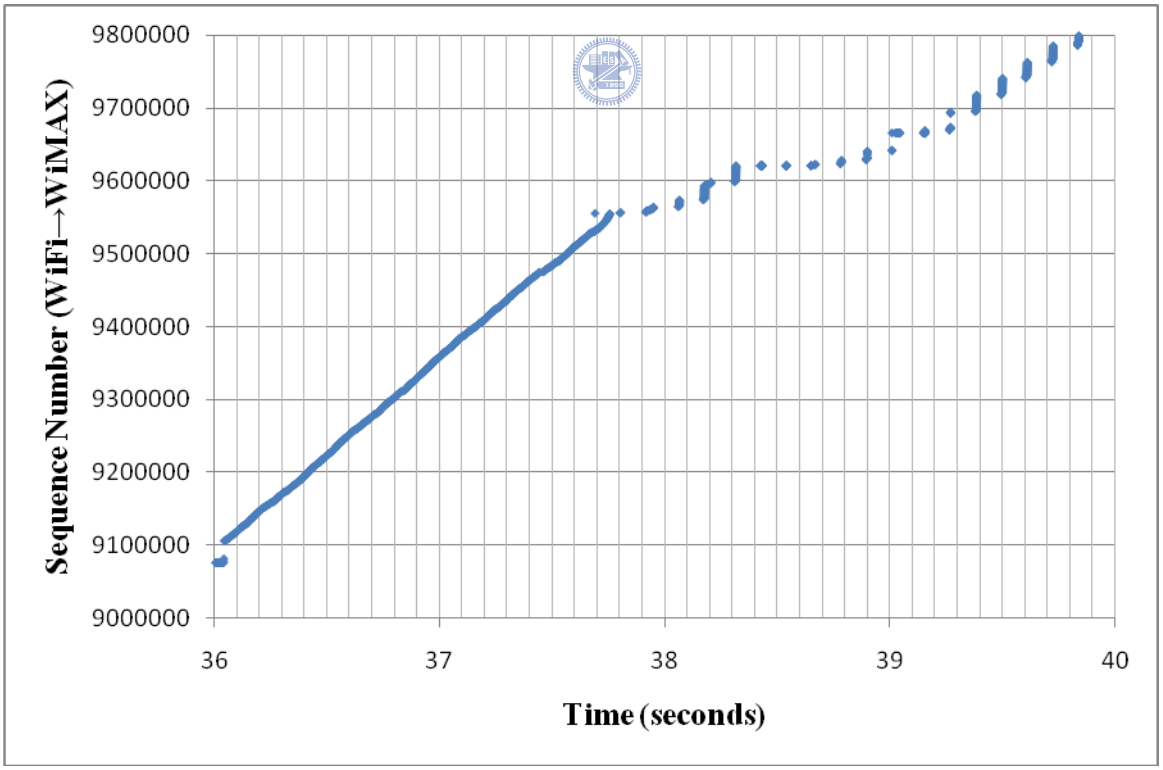

Figure 4.6 Service disruption time of handover from WiFi to WiMAX

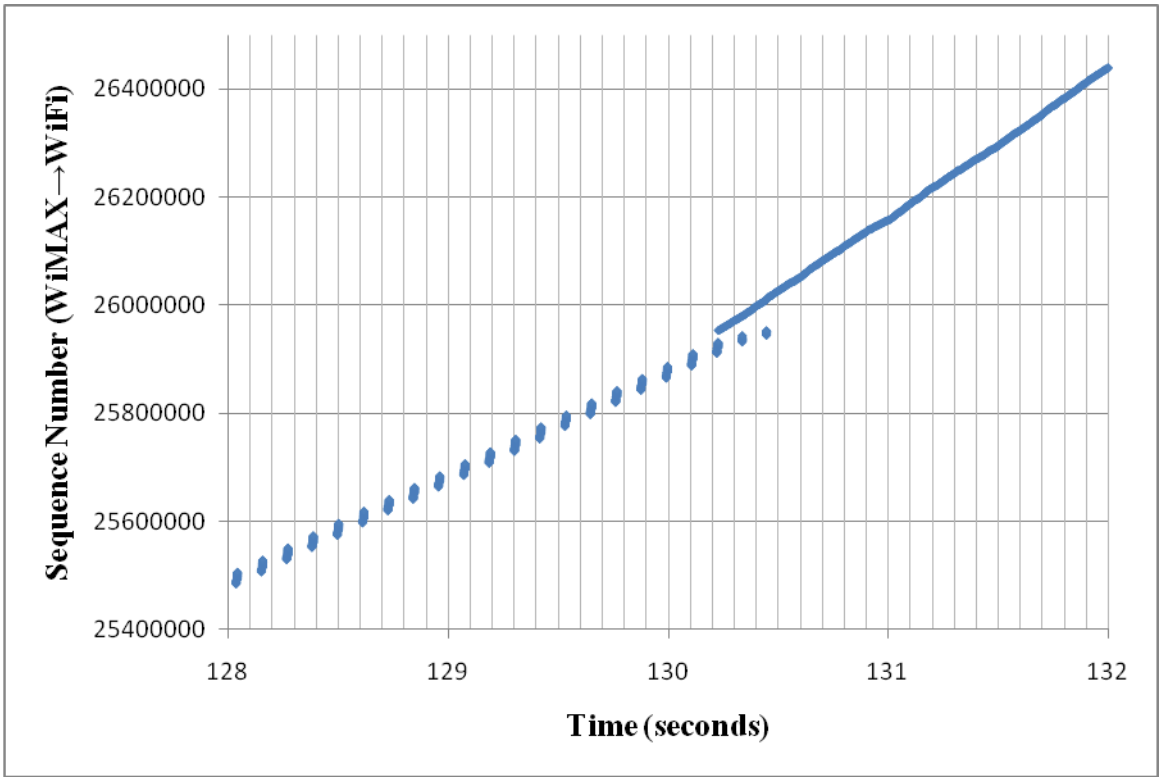

Figure 4.7 Service disruption time of handover from WiMAX to WiFi

Fig. 4.6 and Fig. 4.7 show the service disruption time when MN performs a handover in simulation. In Fig. 4.6, the MN performs a handover form WiFi to WiMAX at time around 37.67 seconds. When handover happens, the subflow2 is already usable so there is only a little or no service disruption time in our proposed scheme. In Fig. 4.7, the MN performs a handover form WiMAX to WiFi at time around 130.22 seconds. The MPTCP of MN doesn't send Acks with priority option that subflow1 is higher than subflow2 until subflow1 is usable, so the service disruption time when handover happens is short and less than 100 ms.

Table 4.3 Comparison of service disruption time with other vertical handover schemes

Handover latency	WiFi→WiMAX	WiMAX→WiFi
Standard	1426.213 ms	1364.244 ms
Literature	393.344 ms	338.604 ms

Proposed scheme	45.853 ms	5.531 ms

The scheme in the literature is based on standard and it presents a cross-layer design, which employs MIH (Media Independent Handover) and FMIPv6 (Fast Mobile IP version 6). It uses three main mechanisms including Pre-DAD (Duplicate Address Detection) procedure, parallel handover, and buffer mechanism to assist handover procedure.

Table 4.3 shows the service disruption time of the proposed scheme. The result is computed by take the average service disruption time of the simulations for 10 times. The result shows that the proposed scheme outperforms the scheme in the literature more than 8 times in both directions.

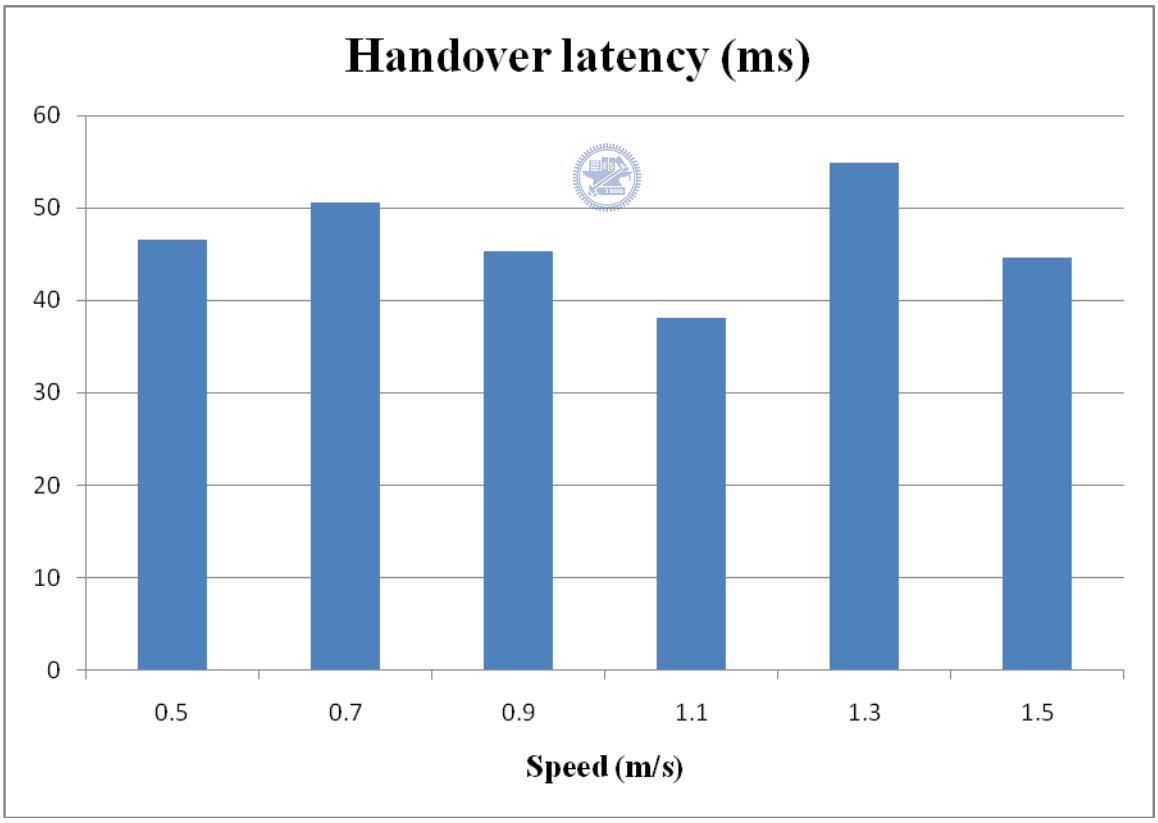

Figure 4.8 The handover latency from WiFi to WiMAX with various speed

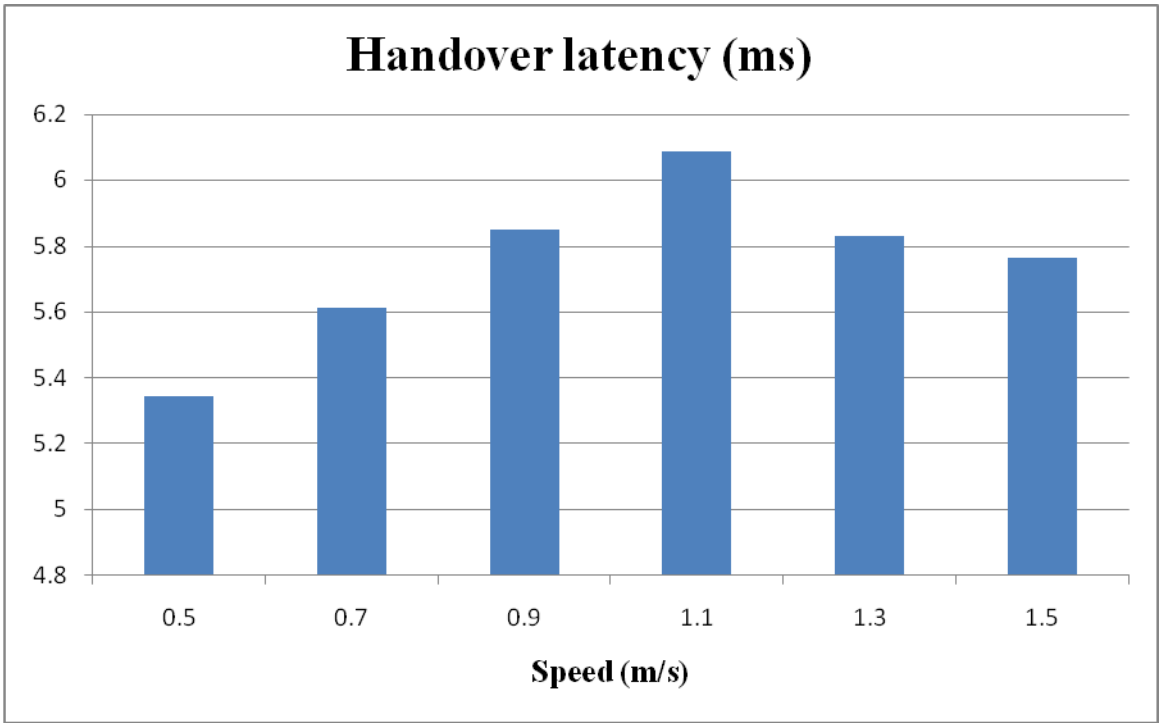

Figure 4.9 The handover latency from WiMAX to WiFi with various speed

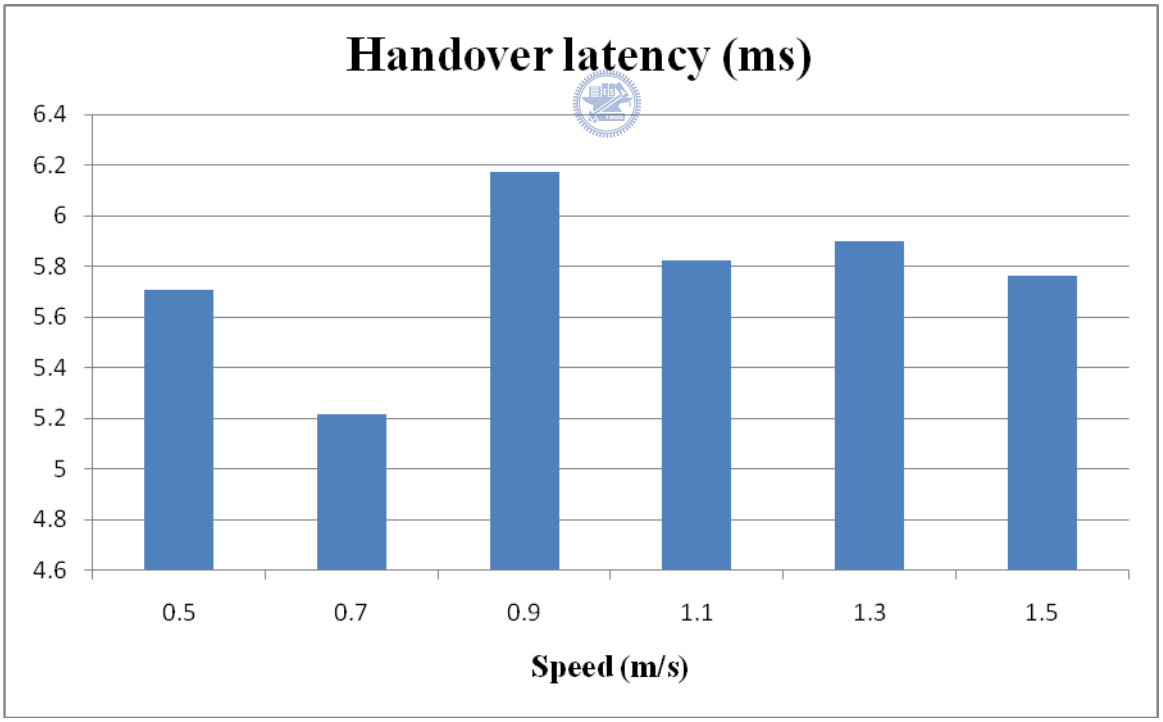

Figure 4.10 The handover latency from WiMAX to WiFi with various speed (repeated)

Fig. 4.8 and Fig 4.9 shows the handover latency when MN moves with various speed from 0.5 m/s to 1.5 m/s. The latency is computed by take the average latency of the

simulations for 10 times. The handover latency can be kept short because the signal strength sensed by the Wi-Fi interface that triggers the redirection of data traffic is very properly chosen. However, if the speed is too high (more than 1.5 m/s), the case 6(b) mentioned in section 3.3 may happen very frequently. Fig 4.10 is derived using the same setting as that of Fig 4.9. The handover latency in Fig 4.9 seems to be related with the speed of MN so we repeated the simulation again to observe the relation between them. However, Fig 4.10 shows that the handover latency has no significant relation with the speed of MN and the difference between handover latencies with different speed is only 1 ~ 2 milliseconds. As a result, the latency in both directions is so low that the proposed scheme can be applied to real-time applications to gain a very high quality of experience.

In our proposed scheme, because we assume the MN has at least a usable subflow, the service disruption time is less than 100 ms. To keep the service disruption time short, MN must have at least one usable all the time. MN must have a usable IP address, which is the cost of our proposed scheme. However, recall that MPTCP is compatible with NAT mentioned in Chapter 2, so the IP address of MN needs not to be a public IP address. Therefore, MN only has to maintain his private address while keeps the service disruption time very short. So, the cost of proposed scheme is very low for MN and can be compared to nothing.

On the other hand, the proposed scheme avoids the existence of network bottleneck. For example, in the network topology of simulation, the Backbone router that connects with the WiFi AP and WiMAX BS can be a network bottleneck if the default MPTCP is used. It's because the default MPTCP generates data traffic at all the usable subflows concurrently, which in turn may cause a big burden on the Backbone router, WiFi AP and WiMAX BS. The proposed scheme avoids the existence of network bottleneck by limit the data traffic to appear

at only the most suitable subflow.

Table 4.4 Comparison with other mobility protocol

	Proposed scheme	Mobile IP	mSCTP
NAT-compatibility	Yes	Yes	No
Multihoming	Yes	No	Yes
Mobility agent	No	Yes	No
Operation layer	Cross layer	Network layer	Transport layer
Reaction to the signal strength of the interface	Yes	No	No
Route optimization	Yes	Need an extension	Yes
Service disruption time	Lowest	High	Low

Chapter 5

Conclusion & Future Work

In this thesis, we implement the priority option of MPTCP, and evaluate the service disruption time of the proposed scheme in NS-2 simulation. The proposed scheme using a cross-layer design monitors the signal streanth sensed by interfaces and determine the the most suitible subflow to use for MPTCP when MN performs a handover between wireless heterogeneous networks. The proposed scheme reduces the waste of radio resource compared with the default operation of MPTCP and avoids the existance of network bottleneck. The simulation result shows that the proposed scheme provides a very short service disruption time which is less than 100 ms when handover happens in a common scenario. The proposed scheme is NAT-compatible and therefore the public IP addresses can be saved while providing mobility support for MN.

Future works may include divising new algorithms to determine the best subflow to generate traffic for MPTCP,evaluating the service disruption time if Add/Remove feature of MPTCP is considered when handover happens or applying multipath concept mentioned in the thesis to transport layer protocols other than TCP.

Reference

[1] J. Postel, Transmisson Control Protocol, IETF RFC 793, Sep. 1981.

[2] M. Mathis, et al., TCP Selective Acknowledgement Options", IETF RFC 2018, 1996

[3] C. Perkins, “IP Mobility Support for IPv4,” Internet Engineering Task Force, RFC 3344, Aug. 2002.

[4] P. Srisuresh and K. Egevang, “Traditional IP Network Address Translator (Traditional NAT),” Internet Engineering Task Force, RFC 3022, Jan. 2001.

[5] O. Levkowetz, J. Forslow, and H. Sjostrand, “NAT Traversal for Mobile IP using UDP Tunneling,” Internet Engineering Task Force, Internet-Draft, Jul. 2001. Available: http://tools.ietf.org/id/draft-levkowetz-mobileip-nat-tunnel-00.txt

[6] M. Handley, H. Schulzrinne, E. Schooler, and J. Rosenberg, “SIP: session initiation protocol,” RFC 2543, Internet Engineering Task Force, Mar. 1999.

[7] R. Stewart et al., “Stream Control Transmission Protocol,” IETF RFC 2960, Oct. 2000.

[8] M. Riegel and M. Tuxen, “Mobile SCTP”, March 2006, http://www.ietf.org/internet-drafts/draft-riegel-tuexen-mobile-sctp-06.txt

[9] A. Ford, C. Raiciu and M. Handley, “TCP Extensions for Multipath Operation with Multiple Addresses”, March 2010

[10] A. Ford et al., “Architectural Guidelines for Multipath TCP Development,” IETF RFC 6182, March 2011

[11] G. Hampel and T. Klein, “Enhancements to Improve the Applicability of Multipath

TCP to Wireless Access Networks," IETF ,Internet-Draft, June 2011

[12] Q. Xie et al., "SCTP NAT Traversal Considerations," IETF Internet-Draft, November 2007

[13] http://isi.edu/nsnam/ns/

[14] http://www.nist.gov/itl/antd/emntg/ssm_seamlessandsecure.cfm

[15] http://code.google.com/p/multipath-tcp/

[16] Yue-Huei Huang and Yaw-Chung Chen, "A Cross-Layer Media-Independent Handover Mechanism in Heterogenous WiMAX/WiFi Networks," Proceedings of Eighth International Network Conference (INC 2010), 6-8 July 2010, Heidelberg,Germany, 2010, pp. 121-130.

[17] Seok Joo Koh, Moon Jeong Chang, and Meejeong Lee, "mSCTP for Soft Handover in Transport Layer", IEEE Communication Letters, VOL. 8, NO. 3, March 2004

[18] G. Carneiro et al., "Cross-Layer Design in 4G Wireless Terminals,". IEEE Wireless Communications, Apr. 2004