Master's Thesis (Academic Year 2009)

iPATH: ACHIEVING HIGH-PERFORMANCE END-TO-END PATHS FOR MULTI-HOMED MOBILE HOSTS

Keio University

Graduate School of Media and Governance

Vu Thi Huong Giang

Abstract of Master's Thesis Academic Year 2009

iPATH: ACHIEVING HIGH-PERFORMANCE END-TO-END PATHS FOR MULTI-HOMED MOBILE HOSTS

Summary

This work aims at building iPath - an intelligent path-selection system for vertical handoff in multi-homed mobile hosts. Unlike existing systems, iPath selects the best access network interface for incoming and out going packets based on end-to-end path properties (e.g., available bandwidth, delay, packet loss rate, and jitter). iPath also adopts a new metric for handoff - "switching cost", which is the cost to switch the network interfaces, including congestion control behavior of transport protocols affected by change of the source or destination address. Handoff execution based on the switching cost prevents inappropriate handoff in terms of transmission performance and smoothness. Our example also shows that path selection based on end-to-end paths' properties can reduce up to 30% of transmission time.

Keywords:

1 Vertical Handoff   2 End-to-end path properties   3 Switching cost

4 Transport protocol behaviour   5 Mobility

Keio University

Graduate School of Media and Governance

Vu Thi Huong Giang

2009(21)

iPath iPath

1 2 3
4 5

VU THI HUONG GIANG

Contents

List of Figures

List of Tables

Chapter 1

Introduction

This chapter explains the motivation and objective of my research. It also goes over background of this research, including descriptions and examples of important issues such as heterogeneous wireless networks and handoff.

1.1 Motivation and Objective

Mobile users desire to always connect to the network that provides the highest transmission performance. None of wireless access networks today; including GPRS, 3G, WiMAX and WLAN, can satisfy all the requirements of users, because their characteristics are trade-off in terms of available bandwidth, wireless coverage and power consumption. For example, WLAN provides high bandwidth in a small coverage while 3G networks support wide range mobility with low transmission speed. However, those heterogeneous wireless networks coexist and their characteristics perfectly complement each other [33]. With the advancement of multiple-interface mobile devices (e.g., smart phones, netbooks), users now have the chance to utilize their preferred networks with vertical handoff between their multiple, distinct network interfaces.

In traditional handoff mechanisms, handoff only occurs when current serving network becomes unreachable or its signal quality falls below threshold to satisfy the communication quality [1]. Received-Signal-Strength (RSS) and other parameters obtained from the wireless signal are therefore decisive indicators for the threshold to execute handoff. However, these parameters are not sufficient for selecting an access network which provides the highest performance. End-to-end path conditions, including available bandwidth, delay, packet loss rate and jitter affect the communication performance in addition to the wireless signal condition. Thus, interface selection or handoff decision based on these end-to-end parameters could significantly improve communication performance.

In addition, cost for handoff or "switching cost" is important for handoff decision. Handoff reduces transport protocol performance due to resetting congestion control parameters, such as congestion window [11, 31], which takes time to reach the previous sending rate again from the initial window size. This behavior also increases transmission jitter, which is problematic for applications that prefer smooth transmission. For these reasons, inefficient handoff should be avoided. In other words, decision to execute handoff in end-system should be done with the considering if it pays the switching cost.

In this paper, we design and implement an intelligent vertical handoff system named iPath, which maintains a high-performance end-to-end path for multi-homed mobile hosts. iPath is an end-to-end system, hence it does not require any modification to the network infrastructure. iPath has two advantages to existing work:

1. The highest-performance end-to-end path
   iPath can ensure the highest performance end-to-end path, because it selects the path based on end-to-end measurement.

2. Minimized handoff overhead
   iPath can minimize performance degradation caused by handoff execution, because it makes handoff decision with the consideration to switching cost.

iPath provides two functionalities to achieve these advantages. First, iPath collects end-to-end path properties including available bandwidth, delay, packet loss rate and transmission jitter by end-to-end path measurement from each local network interface. Second, iPath selects the highest-performance path based on the switching cost in addition to end-to-end path properties. Path selection of iPath is implemented as extensions of some widely used handoff decision algorithms.

1.2 Background

1.2.1 Heterogeneous wireless networks

The development and proliferation of wireless and mobile technologies have revolutionized the world of communications. Such technologies are evolving towards ubiquitous access for various devices and services. Recent broadband wireless access systems include wireless local area networks (WLAN), metropolitan networks and wireless personal area networks (WPAN), as well as the widely used mobile access technologies, such as General Packet Radio Service (GPRS), Wide Code Division Multiple Access (WCDMA), Enhanced Data Rate for Global Evolution (EDGE), 3G and Beyond 3G (B3G) communication systems, Worldwide Interoperability for Microwave Access (WIMAX), and Bluetooth (Figure 1.1) [6].

These wireless access technologies have characteristics that perfectly complement each other as illustrated in table 1.1 [33]. Wireless Wide Area Networks (WWANs) including 3G, GPRS, GSM provide broad coverage areas, full mobility and roaming, but offer relatively low bandwidth connectivity and high cost. Meanwhile, WLANs provide considerably high bandwidth at low cost, but only within a limited area. WIMAX can supply broader coverage and lower bandwidth than WLAN does, but its coverage is smaller and bandwidth is higher than WWANs are. More specifically, WLANs are expected to provide access to IP-based services (including telephony and multimedia conferencing) at high data rates and reduced coverage in public and private areas. Current WLANs offer a bit rate of up to 144Mbps with

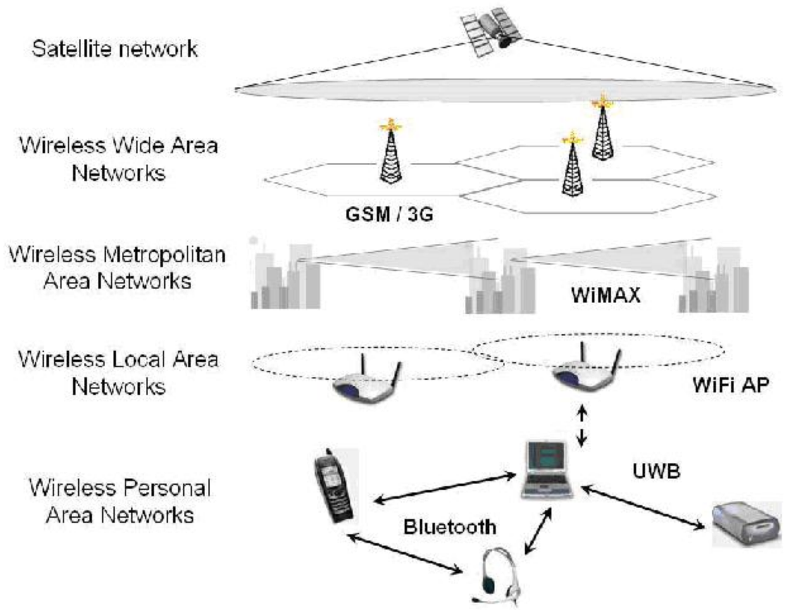

Figure 1.1: Layers of Heterogeneous Wireless Networks

IEEE 802.11n in the 2.4Ghz frequency band.

Table 1.1: Diversity in existing and emerging wireless technologies.

Network	Coverage	Data rates	Mobility	Cost
Satellite	World	Max. 144kbps	High	High
GSM/GPRS	Approx. 35km	9.6-144 kbps	High	High
IEEE 802.16a	Approx. 30km	Max 70Mbps	High	Medium
UMTS	20km	Up to 2Mbps	High	High
IEEE 802.11a	30m	54Mbps	Medium/Low	Low
IEEE 802.11n	182m	40-144Mbps	Medium/Low	Low
Bluetooth	10m	Max 700kbps	Very low	Low

1.2.2 Handoff

Handoff or handover (HO) is the process by which an active mobile node (MN) switch its point of attachment to the network, or when such a change is attempted without terminating services. Handoff can be classified into two types: horizontal handoff and vertical handoff. Horizontal handoff (HHO) involves MNs moving between access points that use the same technology. For example, in cellular telecommunication, horizontal handoff is the process of transferring an ongoing call or data session from one channel connected to the core network to another; in satellite communications it is the process of transferring satellite control responsibility from one earth station to another without loss or interruption of service. Vertical handoff (VHO) refers to a network node changing the type of connectivity it uses to access a supporting infrastructure, usually to support node mobility. For example, a suitably equipped laptop might be able to use both a high speed wireless LAN and a cellular technology for Internet access. The laptop user might want to use a wireless LAN connection whenever one is available, and to 'fail over' to a cellular connection when the wireless LAN is unavailable. Vertical handoffs refer to the automatic failover from one technology to another in order to maintain communication [42]

The difference between a horizontal and vertical handover is vague. For example, a handover from an AP with 802.11b WLAN link to an AP with 802.11g WLAN link maybe considered as either a vertical or a horizontal handover [24]. Figure 1.2 illustrates the different between horizontal handoff and vertical handoff. Seamless handoff is defined as a handoff scheme that maintains the connectivity of all applications on the mobile device when handoff happens.

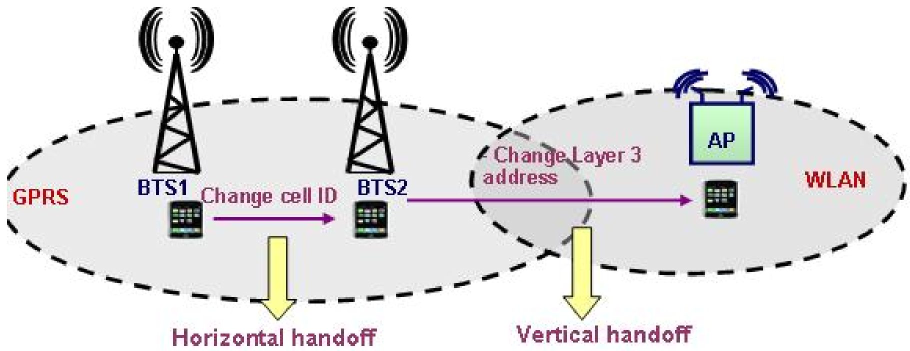

Figure 1.2: Horizontal Handoff and Vertical Handoff

Received signal strength (RSS) and related metrics (such as Signal-to-Noise ratio SNR, Signal-to-Interference ratio) are currently decisive parameters for handoff. To support the moving of mobile hosts between networks in heterogeneous wireless networks, vertical handoff should be performed.

1.3 Organization

The rest of the thesis is organized as follows. Chapter 2 describes challenges for a handoff system as well as problems of proposed handoff systems in heterogeneous wireless networks. Chapter 3 describes operation scope of iPath system and its architecture design, including all main modules in details. In chapter 4, implementation of iPath system is explained. Chapter 5 presents the testing environment and evaluations to investigate the performance of the proposed system. Chapter 6 positions our work with respect to the related work in this area. Conclusion and future work are put on chapter 7.

Chapter 2

Problem Definition

This chapter details problems of existing handoff systems and vertical handoff algorithms.

2.1 Handoff Metric

The coexistence of many types of wireless technologies (heterogeneous wireless networks) is believed to be an inevitable trend. Each single network has its benefits and drawbacks as mentioned in section 1.2.1. For example, WLAN provides high bandwidth in a small coverage while 3G networks support wide range mobility with low transmission speed. Which handoff metrics that are most effective in choosing best wireless network in term of performance is an issue in research on vertical handoff system in heterogeneous wireless networks.

Figure 2.1 illustrate the situation in which traditional handoff metrics such as RSS and the other parameters obtained from the wireless signal are not sufficient for selecting an access network which provides the highest performance. In area (1), mobile host receives signal of three networks: GPRS, WLAN 1 and WLAN 2 and signal strength from all three networks are good enough for communication. Suppose that GPRS provides the highest signal strength, WLAN 1 provides higher signal strength than WLAN2, WLAN2 provides the highest available bandwidth, the lowest packet loss rate, the

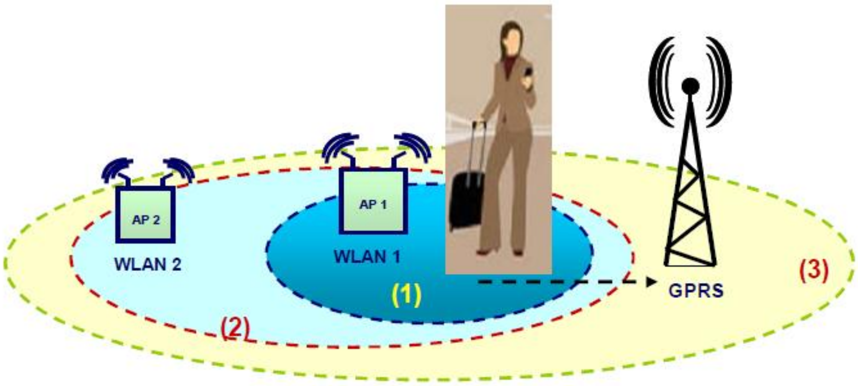

Figure 2.1: Mobile host in multiple network area.

smallest delay and jitter. If RSS is the handoff metric, GPRS network is chosen but in fact the network has highest performance is WLAN2.

In this case, interface selection or handoff decision based on end-to-end parameters (available bandwidth, delay, packet loss rate and jitter) will provide mobile hosts high-performance end-to-end paths for communication.

2.2 Switching Cost for Vertical Handoff Algorithm

Vertical handoff algorithm is another issue of vertical handoff system. Vertical handoff algorithms choose the best network for the mobile host from many networks and based on many handoff criteria. Some algorithms have been proposed and they are proved to be efficient.

The problem of current algorithms is that they do not consider "switching cost" for handoff. In unstable environment like wireless networks, parameters of the networks can change quickly; hence, redundant inefficient handoffs might occur, reducing transport layer performance. Handoff reduces transport protocol performance due to resetting congestion control parameters, such as congestion window [11, 31], which takes time to reach the previous sending rate again from the initial window size.

This behavior also increases transmission jitter, which is problematic for applications that prefer smooth transmission such as video streaming and voice. Therefore, decision if the end-system executes handoff should be done with considering if it pays the switching cost. This switching cost gives the current serving network the higher priority compared to other networks in selection list.

2.3 Approach

To deal with mentioned problems, in this thesis, I propose a vertical handoff system named iPath, which maintains a high-performance end-to-end path for multi-homed mobile hosts. iPath has two advantages to existing work:

Network interface selection based on end-to-end path properties: iPath can ensure the highest performance end-to-end path, because it selects the path based on end-to-end measurement (available bandwidth, packet loss rate, delay and jitter). The calculated performance of each path is inferred by handoff algorithms from retrieval parameters and the chosen path is the path that provides best performance.

Switching cost for handoff: iPath can minimize performance degradation caused by handoff execution, because it makes handoff decision with considering switching cost. Switching cost value will be applied to some proposed vertical handoff algorithms. Current algorithms calculate the vertical handoff decision vectors F for interfaces then compare between them to find out the best interface. We change these vertical handoff decision vectors to F', which is given by:

$$F'_i = \begin{cases} F_i & \text{if } i \text{ is the current serving path.} \\ F_i - cost(i) & \text{if } i \text{ is not the current serving path.} \end{cases}$$

The end-to-end path which can pay the switching cost and has higher calculated performance than current serving path will be chosen.

2.4 Summary

This chapter presented two existing problems in current vertical handoff systems: 1) Requirement for the new handoff metrics to improve network performance. 2) The lack of switching cost for handoff, which prevents redundant insufficient handoff. From these defined problems, the approach of my research is a vertical handoff system named iPath, which selects end-to-end paths for communication based on end-to-end parameters and adopts switching cost in vertical handoff algorithms. The next chapter presents the design of iPath and describes the proposed calculation of switching cost and proposed method to apply this switching cost to current handoff algorithms.

Chapter 3

iPath System Design

This chapter explains in details the design of iPath system and insights to its modules, including Path Property Retrieval Module, Handoff Decision Module and Interface Switching module.

3.1 Overview of iPath System Design

There are a wide range of mobility scenarios in terms of speed, available wireless coverage and available wireless media. An ideal handoff system is the system that has different handoff strategies based on moving speed of mobile host, available wireless coverage and available wireless media. Figure 3.1 shows scenarios with different moving speeds of the mobile host and available wireless media with assumed solution of each case.

1. Case 1: Mobile host is moving fast in the multiple-network area (WLANs and carrier network (e.g.,GPRS))
   Mobile host always connects to the carrier network (vertical handoff selection is not performed).

2. Case 2: Mobile host is moving slowly in the multiple-network area (WLANs and carrier network (e.g.,GPRS))
   Vertical handoff selection is performed.

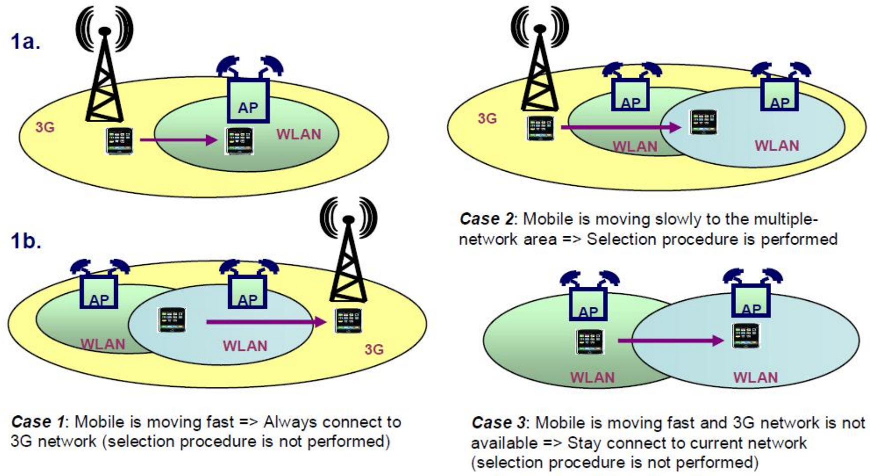

Figure 3.1: Mobility assumption and handoff strategies.

3. Case 3: Mobile host is moving fast and no carrier network is available Handoff is triggered based on Received Signal Strength and related metrics (signal-to-noise ratio SNR, bit error rate BER).

In the scope of this thesis, my work focuses on mobile hosts moving slowly between wireless networks. In this case, the interface selection procedure is performed as illustrated in Figure 3.2. iPath first retrieves end-to-end path conditions that include available bandwidth, delay, packet loss and jitter of all interfaces of which receive signal strength RSS exceeds $RSSthreshold$. After the iPath processes path parameters to find out the vertical handoff decision value of each interface, iPath switches to the interface that have highest vertical handoff decision value.

Figure 3.3 illustrates the overview architecture of iPath system. Three main modules of iPath are: Path Property Retrieval Module, Handoff Decision Module and Interface Switching Execution Module. To retrieve end-to-end path parameters, iPath should be built on top of the transport layer or in the application layer. We suppose that iPath can run with all transport protocols, including TCP/UDP and SCTP. In other words, iPath should work independently from transport protocols. Mobile IP [28, 16] or other

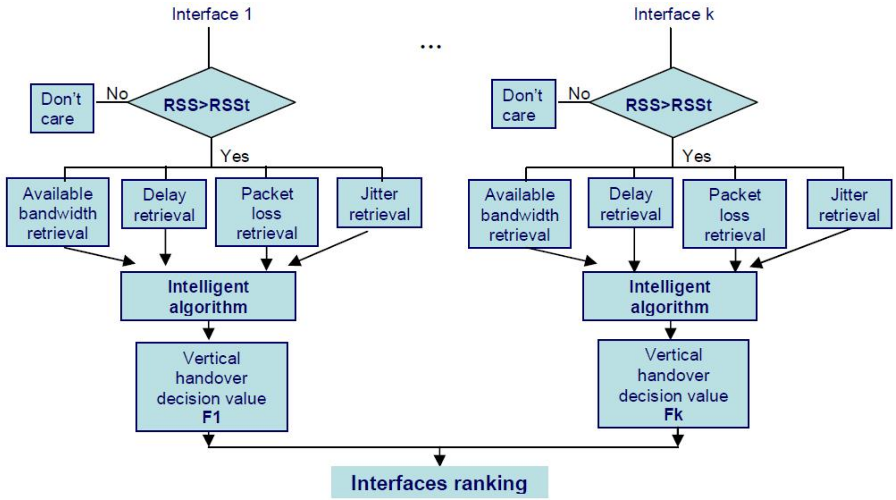

Figure 3.2: How the system works (case 2).

session maintenance protocols [26, 25] would be used to support TCP.

Details of iPath's modules will be described in the following sections.

3.2 Path Property Retrieval Module

Path Property Retrieval Module collects end-to-end path properties; including available bandwidth, delay, packet loss rate and jitter, as handoff metrics (in this system, delay is represented by round-trip time (RTT)). End-to-end path properties are measured on every source and destination pair between end systems. To get the characteristics of a connection before it is really established is always difficult. Requirements for this module are:

- Reliable estimation of parameters
- Low intrusiveness
- Small measurement time

Among required parameters, end-to-end available bandwidth measurement might be the most difficult task. Our idea is modifying one of current

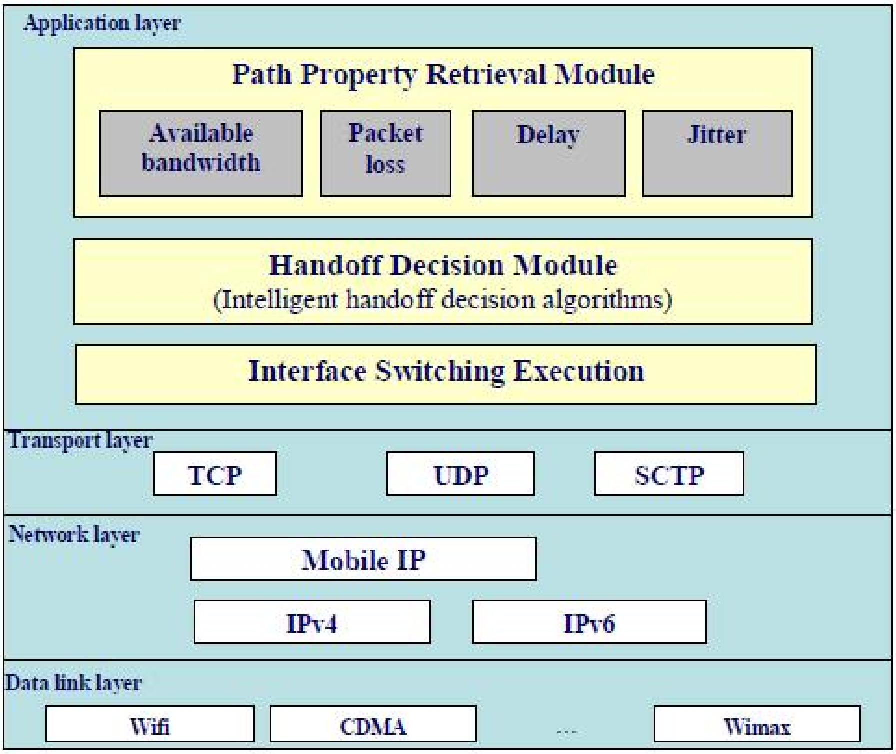

Figure 3.3: Design of iPath.

available bandwidth estimation tools (ABETs) to retrieve not only available bandwidth but all the necessary parameters.

Techniques for estimating available bandwidth are classified into passive measurement and active probing. Passive measurement is limited to network paths that have recently carried user traffic. In this system we want to measure paths of all the interfaces; therefore, active measurement methods is used.

Several available bandwidth methodologies have been proposed, such as packet pair [12] and SLoPS [13] or self-induced congestion [30]. As in [30] [32], packet pair is not robust in multi-hop networks which are common in practice, while SLoPS are equally suited to both single and multiple hop paths. Pathload [14] and PathChirp [30] are based on SLoPS. [32] shows that Pathload is more stable in dynamic traffic conditions compared to PathChirp (PathChirp further degrades and it is the most inaccurate in multiple-bottleneck-link paths). To ensure the accuracy in different network situations, Pathload is adopted for our Path Retrieval Module.

Operation of Pathload will be described briefly. Suppose that sender sends a periodic stream of K packets to receiver at a rate R, starting at an arbitrary time instant. The packet size is L bytes, and so packets are sent with a period of T=L/R time units. A is available bandwidth. If $R > A$, K packets of the periodic stream will arrive at receiver with increasing one-way delays (OWDs), while if $R < A$ the stream packets will encounter equal OWDs. With each rate R, sender sends a fleet of N streams to have exact estimation of OWDs trend. When the calculated available bandwidth belong to a range of (Rmin, Rmax) and Rmax-Rmin ¡ defined resolution (resol), the calculation stops.

Pathload injects multiple packet streams to the measured paths. A drawback of Pathload is long measurement period. For faster measurement, we reduce the number of packets of a periodic stream (K) and the number of streams (N) of same transmission speed. In default mode, current parameters of Pathload are:

- Packet size L=800B
- T = packet spacing = 100us
- Number of packets per stream K = 100 packets
- Number of streams per fleet N = 12 streams

Pathload is modified for iPath so it can also measure RTT and packet loss rate. Jitter is inferred from RTT. iPath maintains them as path conditions in addition to the available bandwidth. The modified Pathload for Path Property Retrieval can specify the interface which it wants to measure.

3.3 Handoff Decision Module

Existing handoff algorithms do not consider cost for handoff or switching cost. We define the switching cost as the value that is adopted in handoff algorithms to gives higher priority for current serving network. It prevents inefficient handoff. I aim to optimize current handoff algorithms by finding cost functions, which has the form:

$$cost(i) = \begin{cases} 0 & \text{if i is the current serving path.} \\ C_i & \text{if i is different from current serving path.} \end{cases}$$

i is the index of a network interface.

When the interface switching is completed, the mobile host starts using the new path. If the transport protocol is TCP or SCTP, the mobile host does not immediately inject a lot of segments to the network, because it follows slow-start from the initial window size, which will take time t for the new path reach the steady state (Figure 3.4). This makes the actual performance of the new path is smaller than the performance estimated by Path Retrieval Module. We consider this transport protocol behavior as the switching cost for our system.

To ensure the reliability and stability of path parameters, Path Property Retrieval Module performs two continuous measurements in one measurement period. If it chooses the same interface which is not the current serving interface in both measurements, the mobile host switches to the new network interface; if not, it keeps using the current interface.

3.3.1 Calculation of Switching Cost

Suppose that a measurement period of system is T. t_0 is the time that system starts running and measured values $B_d(x)$, $P_d(x)$, $RTT_d(x)$, $J_d(x)$ are the predicted values for period from t_0 to (t_0 + T) of interface d. $B_C(x)$ is the available bandwidth of current path. Suppose d is the chosen interface.

In slow-start phase, the congestion window of a path is doubled every RTT. In congestion avoidance phase, congestion window is increased by one

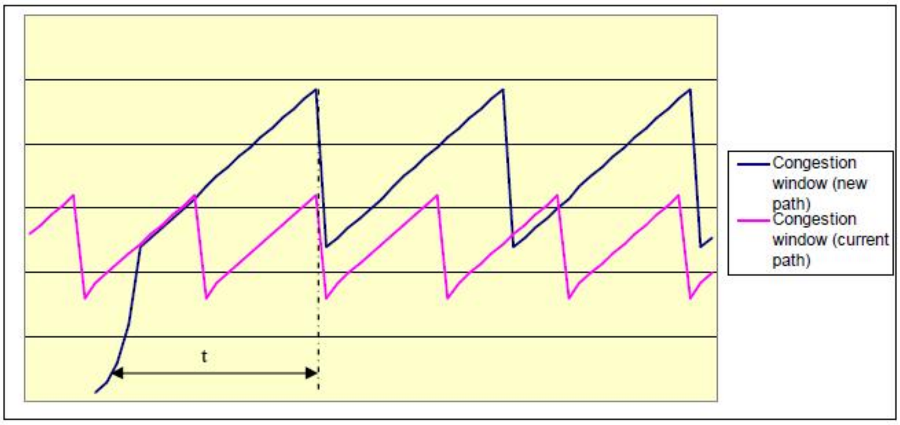

Figure 3.4: Congestion windows (switching and without switching interfaces)

segment every RTT. It is possible to calculate the time a path needs to reach the maximum congestion window size.

Maximum congestion window size of a path can be calculated using the equation 3 of [9] (with b=1/2):

$$T_b = W \cdot \frac{2 - \frac{1}{2}}{2 \cdot RTT} \tag{3.1}$$

in which W is the maximum congestion window size of path d and T_b is the available sending rate in packets per second. The congestion window is initiated by value $IW = min(4.SMSS, max(SMSS, 4380))$($SMSS$ is the size of the largest segment that the sender can transmit) [29].

Time t to reach the maximum congestion window size includes: t_1 that is the time for the congestion window to increase from the initial value IW to $W/2$; and t_2 that is the time for the congestion window to increase from $W/2$ to W.

$$t = t_1 + t_2 \tag{3.2}$$

$$= RTT \cdot \log_2 \frac{\frac{W}{2}}{IW} + RTT \cdot \frac{W}{2} \tag{3.3}$$

The amount of data that the end system can transmit from IW to $W/2$

(A_1) is:

$$
\begin{aligned}
A_1 &= IW + 2 \cdot IW + \ldots + IW \cdot 2^{\log_2 \frac{W}{2 \cdot IW}} && (3.4) \\
&= \frac{IW \cdot (2^{log_2(\frac{W}{2 \cdot IW}+1)} - 1)}{\frac{1}{2}} && (3.5) \\
&= 2 \cdot IW \cdot (\frac{W}{IW} - 1) && (3.6)
\end{aligned}
$$

The amount of data that the end system can transmit from IW to $W/2$ (A_2) is:

$$
\begin{aligned}
A_2 &= \frac{W}{2} + (\frac{W}{2} + 1) + (\frac{W}{2} + 2) + ... + W && (3.7) \\
&= \frac{(\frac{W}{2} + 1) \cdot (\frac{W}{2} + W)}{2} && (3.8) \\
&\approx \frac{3 \cdot W^2}{8} && (3.9)
\end{aligned}
$$

If the mobile host does not change interface and keep using the current path, the amount of data mobile host transmits in period T is:

$$A_c = B_c \cdot T \tag{3.10}$$

If mobile host changes interface and uses new path d, the amount of data mobile transmit in period T is:

$$A_d = A_1 + A_2 + B_d \cdot (T - t) \tag{3.11}$$

So, the switching cost for path d is:

$$C_d = A_c - A_d \tag{3.12}$$

3.3.2 Applying switching cost to vertical handoff algorithms

We will apply cost value to some proposed vertical handoff algorithms. Current algorithms calculate the vertical handoff decision vectors F for interfaces then compare between them to find out the best interface. We change these vertical handoff decision vectors to F', which is given by:

$$F'_i = \begin{cases} F_i & \text{if } i \text{ is the current serving path.} \\ F_i - cost(i) & \text{if } i \text{ is not the current serving path.} \end{cases}$$

It might be considered that, the interface switching makes the available bandwidth of the next path reduces an amount data per second U compared to measured available bandwidth:

$$U = C_d/T \tag{3.13}$$

SAW and TOPSIS [40] and Hongwei Liao's algorithm [21] are applied to Handoff Decision Module. Different from systems proposed in [40] [21], in iPath, these algorithms are the functions of end-to-end parameters.

SAW and Hongwei Liao's algorithm: SAW and Hongwei Liao's algorithm in iPath calculate the vertical handoff decision values by the function:

$$F_i(x) = w_B \cdot \mu_{B,i} + w_D \cdot \mu_{D,i} + w_P \cdot \mu_{P,i} + w_J \cdot \mu_{J,i} \tag{3.14}$$

in which $\mu_{B,i}$, $\mu_{P,i}$, $mu_{D,i}$, $\mu_{J,i}$ are normalized degrees of available bandwidth, packet loss, delay and jitter, respectively (which are calculated from measured parameters by membership function or normalizing functions proposed in [40, 21]); w_B, w_P, w_D, w_J: are weight vectors for available bandwidth, packet loss, delay and jitter normalized degrees, respectively.

As explained above, U has the same unit with available bandwidth; therefore, U is normalized and has the weight vector of available bandwidth to become $\mu_{U,i}$. The new vertical handoff decision value when switching cost is considered equal to $F_i'(x)$:

$$F_i'(x) = w_B.\mu_{B,i} + w_D.\mu_{D,i} + w_P.\mu_{P,i} + w_J.\mu_{J,i} - w_B.\mu_{U,i} \tag{3.15}$$

TOPSIS algorithm: In TOPSIS, the candidate path to use next is the one which is closest to the ideal solution (and farthest from the worst solution). The ideal solution is obtained by using the best value for each metric. In this algorithm, to adopt switching cost, We make the measured available bandwidth of paths, which is not the current serving path, become($B_d - U$); then all the steps described in [40] can be done normally.

3.4 Interface Switching Module

When the best interface is chosen by the Handoff Decision Module, Interface Switching Execution Module changes the outgoing and incoming interface. It should be noted that iPath maintains the transport layer connections by using other mechanisms, such as Mobile IP [28, 16] or SCTP [35], because they already support session maintenance against change of IP addresses. Mobile IP enables mobile host to change its point of attachment from one network to another without changing its IP address; thus, there is no effect to transport layer connection. Meanwhile, SCTP provides the means for each SCTP endpoint to provide the other endpoint (during association startup) with a list of transport addresses and mobile host can transmit data from/to any address.

In the end-system that obeys TCP on top of Mobile IP, Mobile IP changes the next-hop router with Binding Update for switching both outgoing and incoming interface. iPath hence triggers such Mobile IP behavior up based on the decision of the interface.

In the end-system that obeys SCTP, iPath changes the next-hop router to switch the outgoing interface. This is because, SCTP supports multiple local and remote addresses in an association, but SCTP itself does not change the outgoing interface. For the change of incoming interface, iPath also makes SCTP implementation transmit an ASCONF Chunk [36] with Set Primary parameter, which specifies the IP address of new incoming interface. When the corresponding node receives the ASCONF Chunk, it will immediately sends data to the specified IP address and the changing of incoming interface is completed.

3.5 Summary

This chapter proposed the design of iPath. iPath consists of three main modules: Path Property Retrieval Module, Handoff Decision Module and Interface Switching Execution Module. The description of modules' design was explained in details.

We proposed a method to calculate switching cost for handoff based on the behavior of transport protocols. We also described the method to apply this switching cost to some current vertical handoff algorithms (SAW, TOPSIS and Hongwei Liao's algorithms). The next chapter will present the implementation of a prototype system which follows this design.

Chapter 4

Prototype System Implementation

This section describes the implementation of a prototype system in details. It also provides information on implementing platform and setup parameters of each module.

4.1 Overview

An iPath prototype system was implemented in which SCTP is supposed to be transport layer protocol (as illustrated in Figure 4.1, yellow parts) in FreeBSD 7.0 Release. Programming language is C.

4.2 Path Property Retrieval Module

Pathload is modified for Path Property Retrieval module as described in Sec. 3.2. Parameters of Pathload in iPath are:

- Number of packets per stream $K = 9$ packets.
- Number of streams per fleet $N = 1$ streams.

Beside that, modified Pathload can retrieve not only end-to-end available bandwidth but also RTT, packet loss rate and jitter of a specified interface.

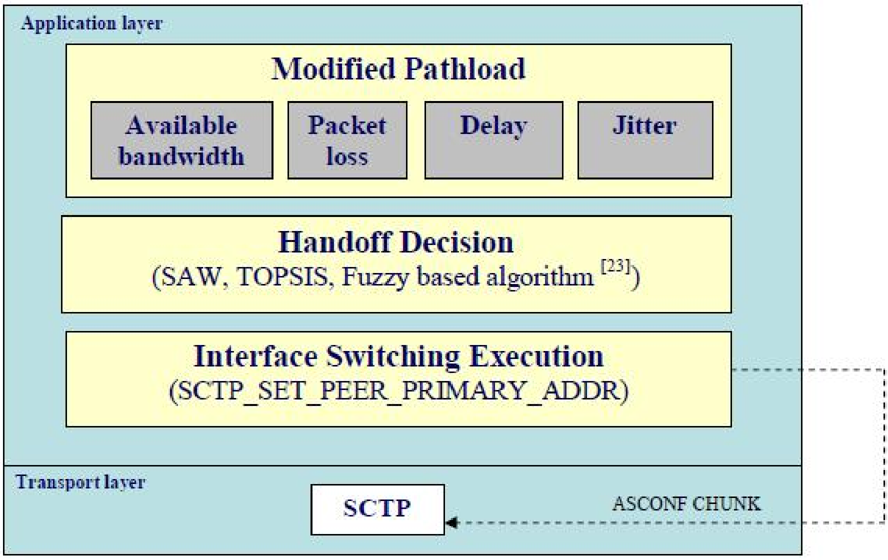

Figure 4.1: iPath prototype system.

4.3 Handoff Decision Module

In Handoff decision module, we implement three algorithms: Hongwei Liao's, SAW and TOPSIS algorithms with cost function as proposed in section 3.3.1. For automatic calculation of weight vector, the method proposed in [21] is used to find the weight vectors for all algorithms. The detailed description of algorithm implementation and how the switching cost is applied to algorithms will be presented in the following sections.

4.3.1 Switching cost calculation

In real calculation of switching cost, the required time (τ) to conduct switching interface command is added. In FreeBSD platform, τ is set to 6000us. Therefore, the switching cost change compared to equation 10 and 11:

$$C_d = A_c - A_d \quad (4.1)$$

$$= A_c - (A_1 + A_2 + B_d \cdot (T - t - \tau)) \quad (4.2)$$

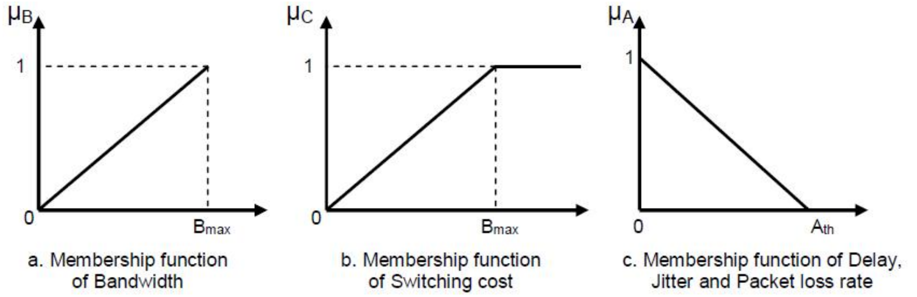

Figure 4.2: Membership functions of Hongwei Liao's algorithm.

Measurement period T is set to 3s. Segment size is set to 576 bytes. IW = 2*576 (bytes) If available bandwidth of current serving path is $bandwidth1$ and available bandwidth and round-trip time of the path i which is not current serving path is $bandwidth2$ and rtt, switching cost U_i of path i is calculated by below sample function $cost\_metric()$(Figure 4.3)

4.3.2 Implementation of Algorithms

Hongwei Liao's and SAW algorithms

These algorithms in iPath system have three major steps:

Normalizing Handoff parameters: In Hongwei Liao's algorithms, end-to-end parameters are compared to maximum value (if retrieved parameters are bandwidth and switching cost) or thresholds values (if retrieved parameters are delay, jitter and packet loss rate) which are set for mobile host. In iPath prototype system, maximum and threshold value are set as below:

- Maximum available bandwidth: $B_{max} = 11Mbps$ (because type two wireless interfaces is 11Mbps type)
- Threshold of packet loss rate are $P_{th} = 0.5$
- Threshold of RTT is $D_{th} = 10000us$
- Threshold of Jitter is $J_{th} = 4000us$

End-to-end parameters are normalized by membership functions which will be described below. Selection possibility of a path is higher when the available bandwidth of that path is high; thus, membership function of available bandwidth of path i $\mu_{B,i}$ is plotted in Figure 4.2(a) and shown in below equation:

$$\mu_{B,i} = \begin{cases} 0 & \text{if } B_i(x) > B_{max} \\ \frac{B(x)}{B_{max}} & \text{if } 0 < B_i(x) < B_{max} \end{cases}$$

As explanation in section 3.3.1, membership function of switching cost of path i is $\mu_{U,i}$ is plotted in 4.2(b) and shown in below equation:

$$\mu_{U,i} = \begin{cases} 1 & \text{if } C_i(x) > B_{max} \\ \frac{C(x)}{B_{max}} & \text{if } 0 < C_i(x) < B_{max} \end{cases}$$

Selection possibility of a path is lower when the delay (RTT), packet loss rate and jitter of that path are high; thus, membership function of delay, packet loss rate and jitter of path i - $\mu_{D,i}$, $\mu_{P,i}$, $\mu_{J,i}$ is plotted in Figure 4.2(c) and shown in below equations:

Membership function of delay (RTT):

$$\mu_{D,i} = \begin{cases} 0 & \text{if } D(x) > D_{th} \\ 1 - \frac{D_i(x)}{D_{th}} & \text{if } 0 < D_i(x) < D_{th} \end{cases}$$

Membership function of Packet loss rate:

$$\mu_{P,i} = \begin{cases} 0 & \text{if } P(x) > P_{th} \\ 1 - \frac{P_i(x)}{P_{th}} & \text{if } 0 < P_i(x) < P_{th} \end{cases}$$

Membership function of Jitter:

$$\mu_{J,i} = \begin{cases} 0 & \text{if } J(x) > J_{th} \\ 1 - \frac{J_i(x)}{J_{th}} & \text{if } 0 < J_i(x) < J_{th} \end{cases}$$

In SAW algorithm, degrees of parameters $\mu_{B,i}$,$\mu_{D,i}$, $\mu_{P,i}$, $\mu_{J,i}$ and $\mu_{U,i}$ of path i are calculated as below: Suppose that B_{max} is the highest value of the measured available bandwidths of all paths; D_{min}, P_{min} and J_{min} are the smallest values of the measured delay, packet loss rate and jitter of all paths respectively. B_i, D_i, P_i, J_i are the measured available bandwidth, delay, packet loss rate and jitter of path i. U_i is the switching cost of to change from serving path to the considered path. Normalizing function of Available bandwidth:

$$\mu_{B,i} = \frac{B_i}{B_{max}} \tag{4.3}$$

Normalizing function of Switching cost:

$$\mu_{U,i} = \frac{U_i}{B_{max}} \tag{4.4}$$

Normalizing function of Delay:

$$\mu_{D,i} = \frac{D_{min}}{D_i} \tag{4.5}$$

Normalizing function of Jitter:

$$\mu_{J,i} = \frac{J_{min}}{J_i} \tag{4.6}$$

Normalizing function of Packet loss rate:

$$\mu_{P,i} = \frac{P_{min}}{P_i} \tag{4.7}$$

Weight vector calculation: The preference on handoff criteria is modeled as weight vector. Weight vector can be assigned by user on criteria (as describe in [40]) or by automatic calculation (as described in [21]). In our system, weight vectors are calculated automatically as described in [21]. Suppose that $\mu_{k,i}$ is the membership degree of metrics k (k can be delay, jitter, packet loss rate and available bandwidth) of path i. There are n paths (n interfaces) in the system.

σ is the standard deviation of $\mu_{k,i}$ and is calculated by equation 4.8

$$\sigma_k = \sqrt{\frac{1}{n-1} \cdot \sum_{i=1}^{n} (\mu_{k,i} - \frac{1}{n} \cdot \sum_{k=1}^{n} \mu_{i,k})^2} \tag{4.8}$$

Then, the weight vectors of available bandwidth w_B, delay w_D, packet loss rate w_P and jitter w_J, are defined as follow:

$$W = (w_B, w_D, w_p, w_J) = (\frac{\sigma_B}{w_B}, \frac{\sigma_D}{w_D}, \frac{\sigma_P}{w_P}, \frac{\sigma_J}{w_J}) \tag{4.9}$$

Calculation of a Path's Vertical Handoff Decision Vector FVHD: In original algorithms, without considering to switching cost, FVHD of the path i is defined as equation 4.10:

$$F_i(x) = w_B \cdot \mu_{B,i} + w_D \cdot \mu_{D,i} + w_P \cdot \mu_{P,i} + w_J \cdot \mu_{J,i} \tag{4.10}$$

In our algorithm, with the consideration to switching cost, FHDV of the path i, which is not the current serving path, is defined as equation 4.11:

$$F_i'(x) = w_B.\mu_{B,i} + w_D.\mu_{D,i} + w_P.\mu_{P,i} + w_J.\mu_{J,i} - w_B.\mu_{U,i} \tag{4.11}$$

The chosen path is the path which has the highest FVHD.

TOPSIS

Suppose that $\mu_{K,i}$ is the membership degree of metrics K (K can be delay, jitter, packet loss rate and available bandwidth)of path i. There are n paths (n interfaces) in the system. U_i is the switching cost of the path i which is not the current serving path. Normalization function of Available bandwidth of path i:

$$\mu_{B,i} = \frac{B_i - U_i}{\sum_i^n B_i} \tag{4.12}$$

Normalization function of delay, jitter and packet loss of path i is:

$$\mu_{K,i} = \frac{K_i}{\sum_i^n K_i} \tag{4.13}$$

The weight vectors of handoff metrics are calculated as proposed in 4.3.2. The following steps of TOPSIS are implemented as proposed in [40].

4.4 Interface Switching Execution Module

When the best interface is chosen by the Handoff Decision Module, the Interface Switching Execution Module changes the outgoing and incoming interface. In the prototype system of iPath, the transport protocol is supposed to be SCTP; thus, iPath can maintain the transport layer connections against the change of IP address without any support of other mechanisms (such as Mobile IP [28, 16]). IP addresses of the mobile host's interfaces are bound to SCTP association.

Interface Switching Execution Module is implemented by using function *setsockopt*() with the option SCTP_SET_PEER_PRIMARY_ADDR ([34], [36])to change the incoming interface. When the socket is set with this option, correspondent node requests the mobile host to mark the enclosed address (IP address of chosen interface) as the association primary. The following structure is used to make a set peer primary request:

```
struct sctp_setpeerprim {
  sctp_assoc_t sspp_assoc_id;
  struct sockaddr_storage sspp_addr;
};
sspp_addr:  The address to set as primary.
sspp_assoc_id:  This parameter is ignored for one-to-one style
   sockets.  For one-to-many style sockets it identifies the
   association for this request.
```

The sample code to change incoming interface is presented in Figure 4.4.

To change the outgoing path, instead of changing the interface, currently the default gateway of the mobile host is changed by the sample code in figure 4.5. The change of outgoing interface in SCTP will be done in future work.

4.5 Summary

This chapter described in details the implementation of a prototype system, which supposes the transport protocol is SCTP. The Path Property Retrieval Module was implemented by modifying Pathload - an original tool for end-to-end available bandwidth estimation - to measure all required parameters with the lower overhead. The Handoff Decision Module was implemented with three different handoff algorithms: SAW, TOPSIS and Hongwei Liao's algorithms. The switching cost proposed in the previous chapter was also applied to these algorithms. Finally, The Interface Switching Execution Module was implemented by using an option of SCTP socket to change incoming interface of the mobile host and by changing the default gateway to change outgoing path of mobile host.

```
double cost_metric(double bandwidth1, double bandwidth2,
                  double rtt){
  double T1,T2;
  double IW,W, A1, A2, cost;
  double t1,t2;
  int i;
   //available bandwidth ( packets per second) of each path
  T1 = (bandwidth1*1000000/(8*576));
  T2 = (bandwidth2*1000000/(8*576));
  IW = 2;
  //Maximum window size:
  W = 4*rtt*T2/(3*1000000);
  //number of packets transmitted from IW to W
  A1 = 2*IW*((W/IW)-1);
  A2 = 3*pow(W,2)/8;
  //time for cwnd increases from IW to W/2
  if((W/2)>=IW){
    t1=rtt*log(W/(2*IW))/(log(2)*1000000);
    A1 = 2*IW*((W/IW)-1);
  }else{
    t1=0;
    A1=0;
  }
  //time for cwnd increases from W/2 to W
  t2=rtt*W/(2*1000000);
  //Switching cost (byte/s)
  cost = T1*3 - (A1+A2+T2*(3000000-t1-t2-6000)/3000000);
  return(cost);
}
```

Figure 4.3: Sample Code to calculate Switching cost

```
int set_if(int sd,struct sockaddr_in caddr){
  struct sctp_setpeerprim prim;
  struct sockaddr_in *ad;
  socklen_t len=sizeof(struct sctp_setpeerprim);
  int rc;

  memset(&prim,0,len);
  if(memcpy(&prim.sspp_addr,(struct sockaddr_storage *)
            &caddr,sizeof(struct sockaddr_storage))==NULL){
      perror("memcpy");
      exit(1);
  }

  rc=setsockopt(sd,IPPROTO_SCTP,SCTP_SET_PEER_PRIMARY_ADDR,
                &prim,len);
  if(rc<0){
     perror("setsockopt primary");
     exit(1);
  }
  ad = (struct sockaddr_in *)&prim.sspp_addr;
  printf("primary address (choosing interface): s\n",
                      inet_ntoa(ad->sin_addr));
  return(0);

}
```

Figure 4.4: Sample Code to change incoming interface

```
int change_int(struct sockaddr_in intf,
                   struct sockaddr_in c_int){
  char command[512];
  char str1[INET_ADDRSTRLEN], str2[INET_ADDRSTRLEN];

  if(inet_ntop(AF_INET,&intf.sin_addr,str1,
                        INET_ADDRSTRLEN)==NULL){
    perror("inet_ntop chosen interface:");
    exit(-1);
  }

  if(inet_ntop(AF_INET,&c_int.sin_addr,str2,
                          INET_ADDRSTRLEN)==NULL){
    perror("inet_ntop current interface:");
    exit(-1);
  }
   //if interfaces are different, change the default gateway
  if(strcmp(str1,str2)!=0){
    if(strcmp(str1,''133.27.170.219'')==0){
      snprintf(command,512,
            "route -n change default 133.27.170.1");
    }else{ snprintf(command,512,
            "route -n change default 133.27.56.1");
    }
  }
  system(command);
  return(0);
}
```

Figure 4.5: Sample Code to Change default gateway

Chapter 5

Experimental results

This chapter describes the experimental setup and methods to evaluate my proposed system and the application of switching cost to current handoff algorithms. The experimental results will be presented and discussed.

5.1 Experimental Setup

The experimental setup of all experiments is illustrated as in Figure 5.1. We use as the *Mobile Node* (*MN*) an IBM Thinkpad X30 with a 1.2 GHz Pentium 3 and 1,024MB memory, where iPath is implemented. *MN* is equipped with two wireless interfaces: AIRCONECT 11Mbps PCMCIA card and AirStation 11Mbps CF type card. *Correspondent Node* (*CN*) is a Dynabook RX2 with 1.7GHz core 2 duo processor and 1GB memory. Both nodes run FreeBSD 7.0.

We form two end-to-end paths. *Path 1* is the path from interface $wi1$ of *MN* to *CN* through *htwireless* network; IP address of this interface is 133.27.170.219. *Path 2* is the path from the interface $wi2$ of *MN* to *CN* through *SFC* network, IP address of this interface is 133.27.61.205. The name of *CN* is $''hope.ht.sfc.keio.ac.jp''$. A bridge note with Dummynet is located in front of *CN* to change the properties (change RTTs) of both paths for algorithm testing.

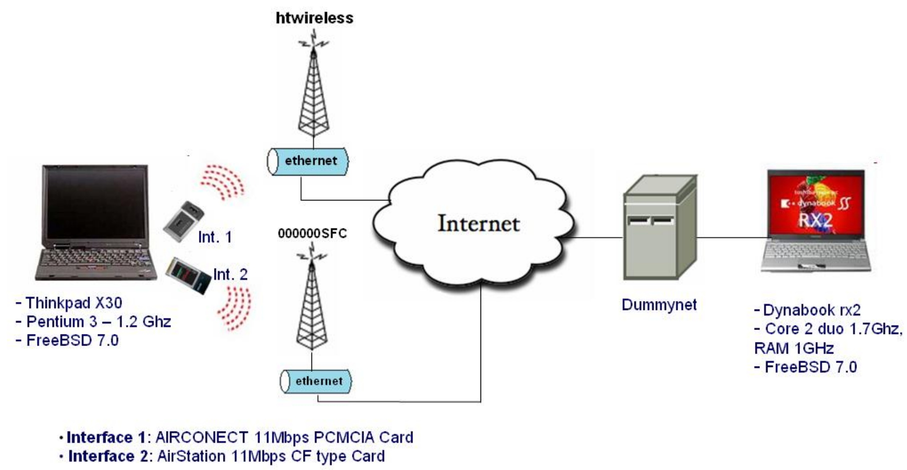

Figure 5.1: Testing environment.

5.2 Evaluation results

5.2.1 Evaluation Methods

The iPath system performance is captured with the following four evaluations:

Accuracy of Path Property Retrieval Module

To prove the effectiveness and accuracy of Path Property Retrieval Module, the below comparisons are done:

- Evaluate the available bandwidth variance of Pathload in measurements which are different in time.
- Compare between available bandwidth measurement results of modified Pathload and real transmission speeds.
- Compare between RTT measurement results of modified Pathload and ICMP echo.

This evaluation will prove the accuracy and improvement of modified Pathload.

Accuracy of handoff decision algorithms and effectiveness of switching Cost

This evaluation will confirm if handoff algorithms without switching cost or with switching cost are more accurate in choosing the best performance paths and in which network situation switching cost is the most effective. It is also used to check if handoff decision based on end-to-end path properties can considerably improve network performance.

Evaluation 1: We created a sctp sender in correspondent node and a sctp receiver in mobile hosts, the sender sends 400kbytes data to the receiver. Dummynet is used to change the RTT and packet loss rate of paths. We then measure the transmission time of data in chosen paths by: case 1. Algorithms which are not applied switching cost; case 2. algorithms which are applied switching cost. The right decision is the decision which chooses the higher performance paths between two paths in selection list. The number of right decisions are compared between case 1 and case 2.

Evaluation 2: To investigate in which network situation switching cost is effective, properties of paths are changed to create testing cases:

- Calculated performances of two paths are similar
- The current serving path has worse performance compared to the other path
- The current serving path has better performance compared to the other path.

Chosen interfaces of algorithms (Hongwei Liao's, SAW and TOPSIS) without and with switching cost are evaluated and compared.

Operation of Interface Switching Module

This evaluation confirms if the Interface Switching Module operates accurately. An SCTP association is created between *MN* and *CN*; IP addresses of two interfaces $wi1$ and $wi2$ are bound to this association and $wi1$ is chosen as incoming interface.

CN continuously sends SCTP data to *MN*. We use *tcpdump* software with *sctp* option to capture SCTP packets which are transmitted to two interfaces.

Operation time of iPath's Modules

In this evalaation, necessary time for each module to complete its task are estimated.

5.2.2 Evaluation results

Accuracy of Path Property Retrieval Module

Figure 5.2 illustrates the variances of available bandwith which are measured by Pathload compared to real transmission speeds when measurements are different in time. The graph shows that if measurement time is higher than 7 seconds, the variance is high when the measurement time is high. This is because the network condition change quickly, especially in wireless environment, and the measurement is not fast enough for the changing. The highest accurarcy measurement is achieved when measurement time is around 7 seconds. The variance then raises when the measurement time becomes small. Pathload has totally wrong estimation when measurement time is less than 1.8 seconds. In our cases, we need to use Pathload in a mobile system, Pathload is adjust to measure within 1.8 seconds.

Figure 5.3 presents that modified Pathload gives approximate results to real transmission speeds in 10 cases. The measurements of ICMP Echo and modified Pathload also give the similar outcome (Figure 5.4)

These results show that, the parameters measured by Path Property Retrieval Module are reliable for handoff decision.

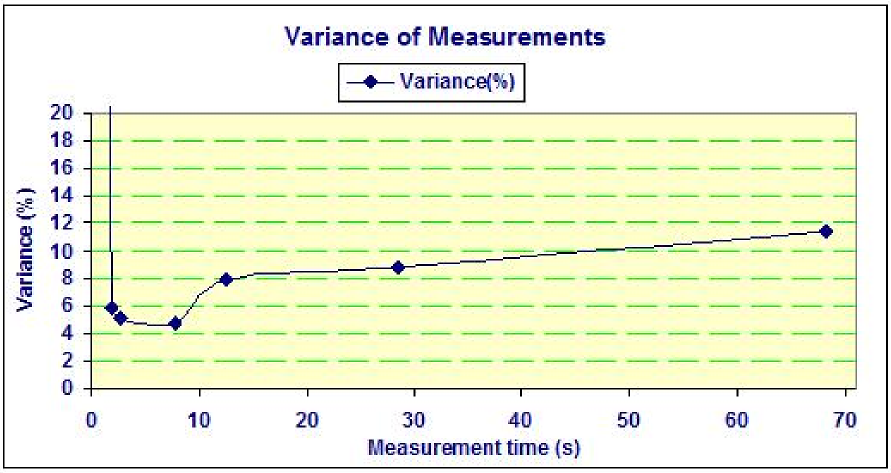

Figure 5.2: Available bandwidth measurement variance of Pathload according to measurement time.

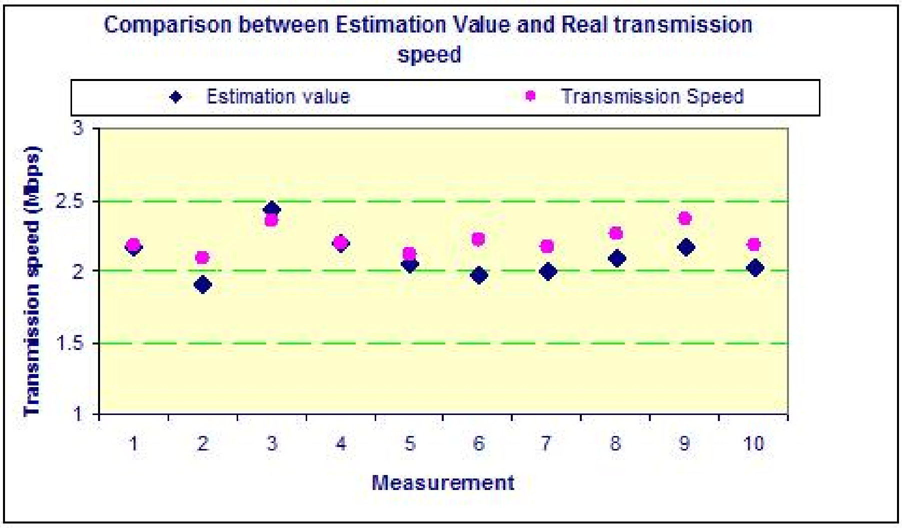

Figure 5.3: Comparison between measured available bandwidth by modified Pathload and real transmission speed.

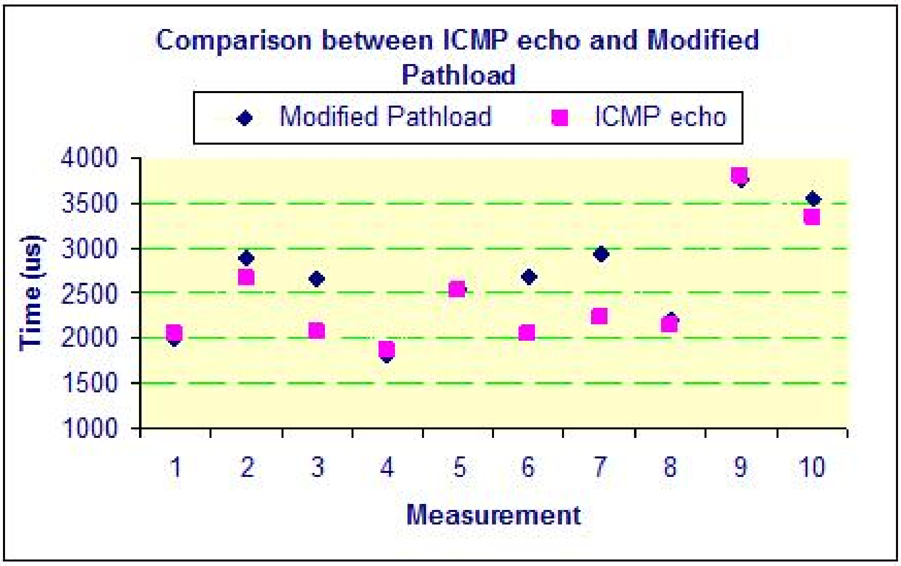

Figure 5.4: Measured RTTs by Modified Pathload and ICMP echo.

Accuracy of Handoff Decision Algorithms and effectiveness of Switching Cost

Evaluation 1: We created 15 testing cases by adjusting RTTs of paths (paths through *htwirekess* and *SFC*). Switching cost is applied to all three algorithms. Table 5.1 shows that the numbers of right decisions are higher in all three algorithms when they are applied switching cost. It proves that switching cost improves the accuracy of all considered handoff decision algorithms.

We explain here an example how the system selects the high-performance path. Table 5.2 shows parameters of paths through *htwireless* and *SFC*. All the algorithms choose path through *htwirless* as path which has higher performance while path through *SFC* is current serving path. We measure transmission time of application if only path through *SFC* is used and if the switching to *htwireless* occurs when 200 kbyte data is sent.

Figure 5.5 illustrates that if only path through *SFC* is used, transmission time is 1981375us; if the switching occurs as the recommendation by handoff

Table 5.1: Number of right decisions by algorithms without and with switching cost

Algorithms	Hongwei Liao's		SAW		TOPSIS	
apply cost?	no cost	with cost	no cost	with cost	no cost	with cost
Right decisions	10	13	7	10	9	10

decision module, the transmission time is 1354629us. It means that the transmission time is improved by 31.63%.

Table 5.2: Example: Estimation of paths' parameters and chosen path by algorithms

Network	packet loss rate	jitter (us)	bandwidth (Mbps)	delay (us)	Hongwei	SAW	TOPSIS
htwireless (chosen path)	0	406.5	2.30	3198	0.542	0.505	1
SFC (current path)	0	2701	2.35	18465	0.385	0.138	0

Evaluation 2: We created 3 network cases in which the relativeness of current serving path and the other path in selection list changes. By this experiment, we check in which network situation, switching cost is the most effective.

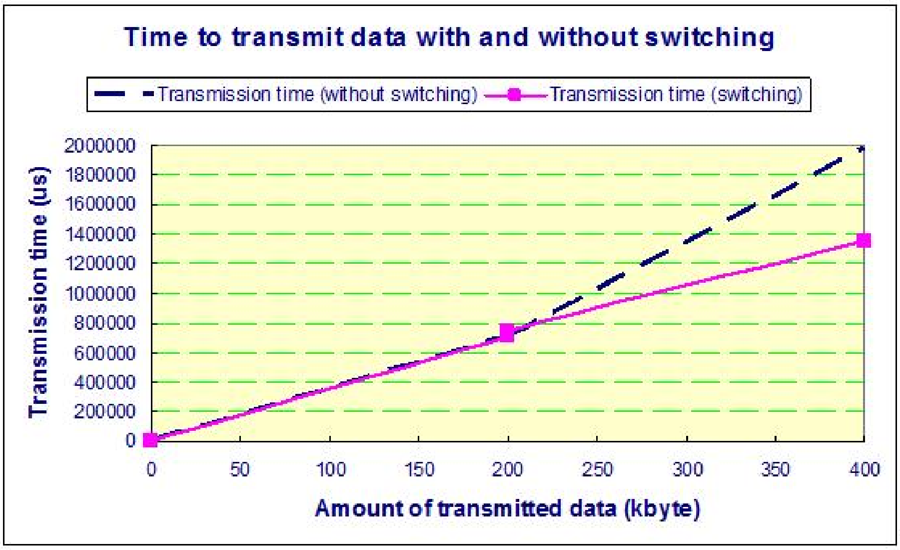

Figure 5.5: Transmission time in current serving path and in chosen path.

Case 1 - current serving path has higher calculated performance: We added 5ms delay between *htwireless* and *CN*, 3ms delay between *SFC* and *CN* using Dummynet.

htwireless is the current serving network and path through *SFC* is a path in selection list. *htwireless* has worse delay but higher bandwidth, lower jitter than those of *SFC* (table 5.3). The vertical handoff decision values (FVHDs) of interfaces which are calculated with and without switching cost by algorithms are presented in Table 5.4.

Table 5.3: Measured handoff parameters: Case 1

Network	packet loss rate	jitter (us)	bandwidth (Mbps)	delay (us)	cost (byte/s)
htwireless	0	491	2.36	5826	0
SFC	0	1235	1.59	5689	1189.53

Table 5.4: Vertical Handoff Decision Values: case 1

Algorithms	Hongwei Liao's		SAW		TOPSIS	
Network	*htwireless*	*SFC*	*htwireless*	*SFC*	*htwireless*	*SFC*
without cost	0.661	0.514	0.975	0.482	0.568	0.432
with cost	0.661	0.471	0.975	0.217	1	0.014

htwireless is chosen in all algorithms, even when the switching cost is adopted. The switching cost has no effect to current serving network (*htwireless*) but it makes the FHDV of (*SFC*) become smaller, the distance of value between two interfaces become larger. Current interface is definitely chosen.

Case 2 - two paths have similar calculated performance: We added 10ms delay between *htwireless* and *CN*, 3ms delay between *SFC* and *CN* using Dummynet.

Table 5.5 shows the measured parameters in case 2, in which *SFC* is the current serving network. *SFC* is worse in jitter, available bandwidth but much better in delay than those of *htwireless*.

Table 5.5: Measured handoff parameters: Case 2

Network	packet loss rate	jitter (us)	bandwidth (Mbps)	delay (us)	cost (byte/s)
htwireless	0	738.75	2.33	13083	670
SFC	0	869.75	1.87	7413	0

As shown in Table 5.6, TOPSIS chooses *SFC* in three situations. Hongwei Liao's and SAW have different choice from TOPSIS's when switching cost is not considered. Without switching cost, Hongwei Liao's and SAW algorithms choose to change to *htwireless* network while with switching cost, all the algorithms choose to stay in current network *SFC*. In this case, switching cost prevent system from switching interface because two networks have similar network performance calculation. It can be explained that because two networks are not much different in performance, TOPSIS and other algorithms choose different networks when switching cost is not considered.

Table 5.6: Vertical Handoff Decision Values: case 2

Algorithms	Hongwei Liao's		SAW		TOPSIS	
Network	*htwireless*	*SFC*	*htwireless*	*SFC*	*htwireless*	*SFC*
without cost	0.107	0.098	0.366	0.455	0.97	1
with cost	0.094	0.098	0.334	0.455	0.67	1

Case 3 - current serving path has lower calculated performance: We added 3ms delay between *htwireless* and *CN*, 3ms delay between *SFC* and *CN* using Dummynet.

Measured parameters in case 3 are presented in Table 5.7.

Table 5.7: Measured handoff parameters: Case 3

Network	packet loss rate	jitter (us)	bandwidth (Mbps)	delay (us)	cost (byte/s)
htwireless	0	248.5	2.51	4777	701.87
SFC	0	642.5	1.92	5698	0

SFC is current serving network but *htwireless* is better in all handoff metrics than *SFC*. Without switching cost, *htwireless* is chosen in all algorithms. When switching cost is considered, FHDVs of *htwireless* is reduced but still higher than FHDVs of *SFC* and chosen interface is the interface that connects to *htwireless* in all algorithms.

These results prove that switching cost is effective to avoid unnecessary handoffs in case networks are not clearly different in network performance calculation, especially with Hongwei Liao's algorithm, in which FHDVs of interfaces are close.

Table 5.8: Vertical Handoff Decision Values: case 3

Algorithms	Hongwei Liao's		SAW		TOPSIS	
Network	*htwireless*	*SFC*	*htwireless*	*SFC*	*htwireless*	*SFC*
without cost	0.428	0.377	0.84	0.413	1	0.117
with cost	0.407	0.377	0.74	0.413	0.997	0.314

Operation of Interface Switching Module

The result of this evaluation is shown in Figure 5.6.

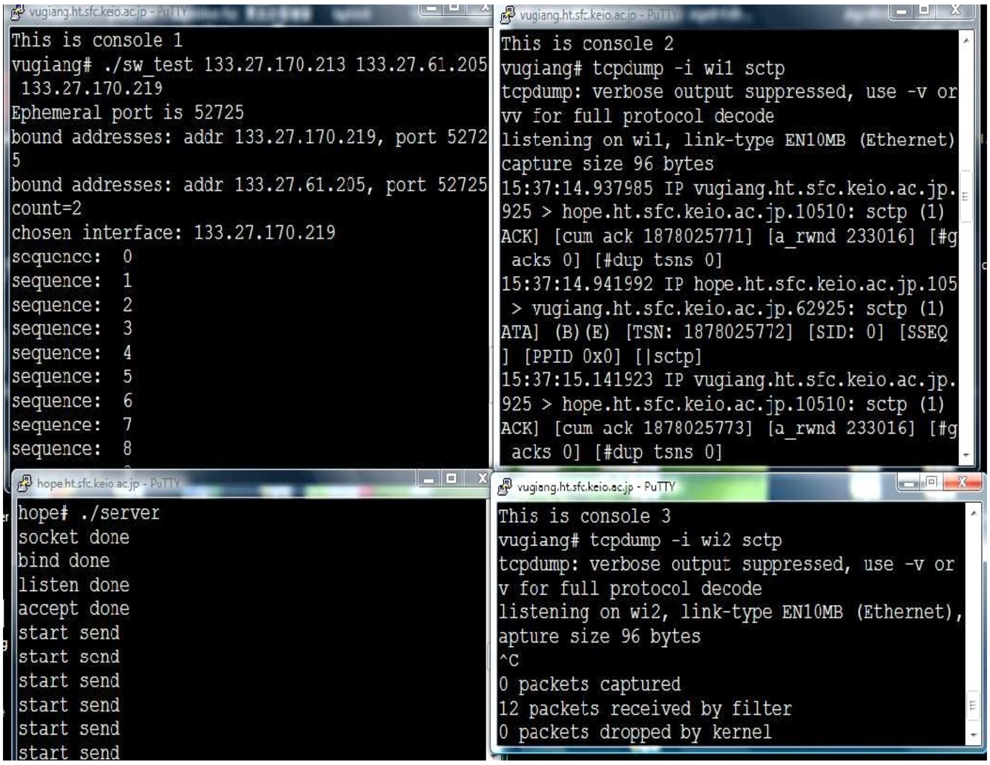

Figure 5.6: Incoming SCTP packets in two interfaces of *MN*.

In the console of *CN* (*hope.ht.sfc.keio.ac.jp*), we run an SCTP application that sends data to *MN*. The receiver application was run in console 1. This receiver application set $wi1$, which has IP address as 133.27.170.219, primary address. The console 2 and console 3 show the results of *tcpdump* running. When *tcpdump* was run to check the SCTP flow in console 2, many SCTP data packets from *CN* to *MN* were captured. Meanwhile, no SCTP packet was captured in $wi2$ (illustrated in console 3).

This result confirms that the Interface Switching Execution Module used only $wi1$ for receiving data although both IP addresses were bound to the SCTP association.

Operation time of iPath's modules

In this experiment, each result is the average of 10 measurements. As in Table 5.9, the calculation time of all agorithms in Handoff Decision Module are lower than 50us; neccessary time for Interface Switching Module to complete its task is about 16ms. The module which takes the most time of the system is Path Property Retrieval Module; it takes around 1.8 seconds to measure all required paramters. With the assumption that mobile host is moving slowly at walking speed, the operation time of current iPath's modules are enough efficient.

Table 5.9: Operation time of iPath's modules

	Handoff Decision Module			
Path Property Retrieval Module (time/interface)	Hongwei	SAW	TOPSIS	Interface Switching Module
1.8s	43.6us	31.0us	17.9us	16393us

5.3 Summary

This chapter described the testing environment and evaluated the performance of the prototype system in different network situations. Firstly, the accuracy and effectiveness of Path Property Retrieval Module was evaluated. The results show that this module gives reliable estimations in a small time measurement (about 1.8s for measuring all parameters of one path). Secondly, the effectiveness of handoff decision module was tested. The results prove that handoff algorithms with switching cost have higher accuracy than the algorithms without switching cost. This switching cost is proved to prevent unnecessary handoffs in case networks are not clearly different in network performance calculation and is especially effective in case of Hongwei Liao's algorithm, in which FHDVs of interfaces are close. By this evaluation, we also proved that interface switching based on end-to-end path properties can significantly improve network performance (in our example: 30%). Thirdly, operation of Interface Switching Execution Module was checked. This module is proved to be able to switch to desired incoming interface. The evalation of operation time of iPath's modules shows that Path Property Retrieval Module has the highest operation time but iPath can still work well with mobile hosts which are moving at walking speed. The next chapter presents the related work of our research.

Chapter 6

Related Work

This chapter positions my research with respect to the related work on vertical handoff system for heterogeneous wireless networks. Existing work on network selection schemes can be classified into two groups: research on new metrics to trigger handoff in the end-system and research on network selection algorithms to select the best network more appropriately. In this chapter, we will present survey on listed issues.

6.1 Metrics for handoff

This section is divided into 2 sub-sections: Handoff metrics and Handoff metrics measurement methods.

6.1.1 Handoff metrics

This part will review all the possible handoff metrics in proposed handoff systems. In order to decide if the end-system executes handoff, many handoff metrics have been proposed. We made a survey on handoff metrics of current handoff system from the year 2004 to 2009. The result is presented on table 6.1. Handoff metrics in most systems are RSS and related metrics, such as Signal-to-Noise ratio [8, 23, 19, 22, 3, 18, 39], which are link-layer parameters and represent the physical transmission quality of the networks. The reason why these parameters are chosen is that handoff happens when current serving network becomes unreachable or its signal quality falls below threshold to satisfy the communication quality. Some recent systems [8, 23, 19, 7, 3, 18] propose new metrics, such as cost for ISPs, user preference, system power consumption and link-layer bandwidth or available bandwidth. However, none of them adopts end-to-end parameters as the metrics for handoff decision.

Table 6.1: Survey on handoff metrics of current vertical handoff decision schemes

System	RSS related metrics	Power usage	Channel related metric	User Priority	Cost for ISP
[20]		×	×		
[8]	×		×		
[39]	×				
[19]	×		×	×	
[23]	×		×	×	×
[18]	×		×		
[7]			×		×
[22]	×				
[3]	×			×	

In [41], handoff metrics can be classified into groups:

- **Service type**: Different types of services require various combinations of reliability, latency and data rate.
- **Monetary Cost**: A major consideration to users, as different networks may employ different billing strategies that may affect the users' choice to handoff.
- **Network Conditions**: Network-related parameters, such as traffic, available bandwidth, network latency, and congestion (packet loss)

may need to be considered for effective network usage. Use of network information in the choice to handoff can also be useful for load balancing across different networks, possibly relieving congestion in certain systems.

- **System Performance**: To guarantee the system performance, a variety of parameters can be employed in the handoff decision, such as the channel propagation characteristics, path loss, interchannel interference, signal-to-noise ratio (SNR), and the bit error rate (BER). In addition, battery power may be another crucial actor for certain users. When the battery level is low, the user may choose to switch to a network with lower power requirements, such as an ad hoc Bluetooth network.

- **Mobile Terminal Conditions**: MT condition includes dynamic factors such as velocity, moving pattern, moving histories and location information.

- **User Preferences**: User preference can be added to cater to special requests for users that favor one type.

However, to use all those parameters as handoff metrics in one system is almost impossible because it takes much time for information retrieval. The parameters which are used for handoff should be based on the purpose of the system (for example, optimize system performance, network performance).

6.1.2 Handoff metrics measurement

It is always difficult to know the characteristics of an end-to-end path before actually using it.

Among required parameters, end-to-end available bandwidth measurement might be the most difficult task. Many bandwidth estimation tools (ABETs) have been emerged to measure available bandwidth such as Pathload [14], PathChirp [30], Spruce [37], Cprobe [5]. The paper [32] made an empirical evaluation of above tools; the result shows that each tool has its benefits and drawbacks.

The research [13] gives a definition for available bandwidth of a network path P. The end-to-end available bandwidth is defined as the maximum rate that the path can provide to flow, without reducing the rate of the rest traffic in P (a network path was defined as a sequence of store-and-forward links that transfer packets from a sender to a receiver).

A popular method to measure RTT is ICMP echo. In this method, ICMP echo packets are sent between two end systems to measure the round trip time. In some papers [4, 15], RTT is measured by passive measurement methodology. For every data packet, the sequence number and timestamp are recorded and RTT sample is the difference in the two timestamp. This RTT sample is then smoothed using Jacobsons Algorithm [4] to generate RTT estimate for a stream. Jitter is a measure of RTT variability [4]. Jitter can be calculated by the equations from RTT sample and RTT_t at time t. There are several works on packet loss rate measurement such as Queen [38], RON [2] and King [10].

It is obvious that methods to measure each end-to-end path property have been proposed, but there is no mentioned method can measure all properties. This is a challenge for the work on handoff based on end-to-end parameters.

6.2 Network selection algorithms

In conventional system, handoff metrics is only one or two parameters (e.g., RSS and Bit-error-rate); therefore, handoff algorithms compare received parameters with their thresholds then choose the network which has higher values, such as in cellular network case. In future systems, more metrics will be used for handoff thus the algorithm will become more complex. Fuzzy logic based algorithms, such as Hongwei Liao's algorithm [21] and multiple decision making algorithms, (e.g., SAW, TOPSIS, Maxmin method [40],MEW [27]) have been mentioned as they can handle complicated handoff decision.

In SAW and Hongwei Liao's algorithms, the overall score of a candidate network is determined by the weighted sum of all the attribute values. The score of each candidate network i is obtained by adding the normalized contributions from each metric r_{ij} multiplied by the importance weight assigned w_j of metric j. The selected network A_{SAW} is:

$$A_{SAW} = arg \max_{i \in M} \sum_{j=1}^{N} w_j \cdot r_{ij}$$

In TOPSIS, the selected candidate network is the one which is the closest to the ideal solution (and the farthest from the worst case solution). The ideal solution is obtained by using the best values for each metric. Let c_i i denote the relative closeness (or similarity) of the candidate network i to the ideal solution. The selected network A_{TOP} is:

$$A_{TOP} = arg \max_{i \in M} c_i \tag{6.1}$$

The Multiplicative Exponent Weighting (MEW) is another method. The vertical handoff decision problem can be expressed as a matrix form, where each row i corresponds to the candidate network i and each column j corresponds to an attribute (e.g., bandwidth, delay). The score Si of network i is determined by the weighted product of the attributes (or metrics):

$$A_{MEW} = \prod_{j=1}^{N} x_{ij}^{w_j} \tag{6.2}$$

where x_{ij} denotes attribute j of candidate network i, w_j denotes the weight of attribute j. There are some methods to calculate the weight vectors for these algorithms [21], AHP [27] and fuzzy logic [40].

These algorithms, however, do not consider switching cost of handoff. Hence, in unstable wireless networks, redundant inefficient handoffs could occur, reducing transport layer performance. Discovering the switching cost for handoff algorithms, which gives current serving network higher priority compared to other networks and avoid ineffective handoffs, is necessary.

6.3 Summary

This chapter reviewed existing research on vertical handoff systems. We presented these works by dividing them into two groups: studies that focus on handoff metrics; and studies on network selection algorithms. Though these studies have several advantages, we indicated there are some problems which prevent mobile host from connecting to the high performance network anytime, anywhere. The next chapter presents future work on our research and concludes this thesis.

Chapter 7

Future Work and Conclusion

This chapter describes the summary of this thesis. Firstly, we will conclude this thesis. After that, we will state the future directions of our research.

7.1 Conclusion

In this thesis, we presented iPath, which allows users to maintain the highest performance end-to-end path by network interface selection based on end-to-end path properties (available bandwidth, packet loss rate, delay and jitter). iPath is also distinguished from other systems, because it adopts "switching cost" for network interface switching, which is calculated based on congestion control behavior of transport protocols.

Our experiments prove that our end-to-end measurement operates effectively. Path Property Retrieval module provides reliable path parameters as the metric for handoff, the switching cost implemented in SAW, Hongwei Liao's and TOPSIS algorithms with our designed weight vector also prevents ineffective handoff when access networks are similar in performance, especially in Hongwei Liao's algorithm. The evaluation results also show that interface switching based on end-to-end path properties can significantly improve the network performance.

7.2 Future Work

This section suggests the future directions of my research. My future work will focus on issues: wide range evaluations, implementation of iPath in all transport protocols, improving accuracy and overhead of Path Property Retrieval Module and handoff strategies considering moving speeds of mobile host.

Evaluation: More evaluation should be done to ensure that the handoff which is triggered based on end-to-end parameters definitely improve the network performance compared to current handoff schemes.

The current testing environment of iPath is quite limited. We will evaluate operation of iPath in different real situations.

Completing iPath implementation in all transport protocols: A prototype system of iPath is now implemented with the supposition that transport protocol is SCTP. My desire is that iPath can run with all transport layer protocol; therefore, one of the most important future works is implementing in other transport protocol such as TCP, UDP and DCCP [17].

Improving accuracy and overhead of Path Property Retrieval Module: In iPath, end-to-end parameters are retrieved by active measurement, thus an amount of packets are injected to networks. In future, we intend to use other methods to retrieve required parameters.

Some methods have been considered. For example, Site Multihoming by IPv6 intermediation SHIM6 [26] is an end-to-end multi-homing scheme located between the network and transport layer. From this layer, we desire to get all required parameter.

Handoff strategies considering moving speed of mobile host: As mentioned in section 3.1, an ideal handoff system has different handoff strategies based on moving speed of mobile host, available wireless coverage and available wireless media. iPath is currently designed for the mobile hosts moving slowly (users stay or walk in an area) between networks. To collect end-to-end path properties takes time; therefore, if users are moving by vehicles, by the time iPath collects enough end-to-end parameters, users might move to another area and handoff decision might be not accurate.

We desire to design iPath to work well with all moving speeds of mobile hosts. The speed of mobile host will be inferred by GPS or RSS (comparing positions of mobile hosts after a certain time). This relates to localization challenge.

Acknowledgement

This thesis is the accomplishment of the two-year study and research that have been done since I came to Japan. By that time, I have worked with a great number of people, who considerably contributed to my research and the completion of this thesis in different ways. It is a pleasure to convey thankfulness to them all in my humble acknowledgement.

In the first place, I would like to thank HEDPSI project for giving me the chance to come to this country, meet many respectable people and research in the first-rate education environment.

The sincerest gratitude goes to my supervisor, Professor Hideyuki Tokuda, for his guidance and advice from the very early stage of this research. Above all and the most needed, he provided me unflinching encouragement and support in various ways.

I also thank members of the thesis committee, Associate Professor Huroyuki Kusumoto and Assistant Professor Rodney Van Meter III. I am thankful that in the midst of their busy schedules, they accepted to be members of reading committee. Their recommendations are valuable for my research.

I would like to extend my appreciation to members of the Ubiquitous Computing and Networking Laboratory; Associate Professor Kazunori Takashio, Assistant Professor Jin Nakazawa, Ms. Michiko Nitta, Dr. Jun'ichi Yura, Dr. Soko Aoki, Dr. Masaki Ito, Dr. Aida Hiroto, graduate and undergraduate students, for their devoted instructors and support.

I am grateful to the professors and teachers in Cyber Informatics (CI) program, Graduate School of Media and Governance. Their well-established and intellectual lectures broaden my knowledge on computer science.

I warmly thank Hiroshi Sakakibara, PhD student, for his valuable advice and friendly, patient help from the very first day in Keio University. His extensive discussions around my work have been very helpful for this study. I thankfully acknowledge Michio Honda, PhD student, for his supervision and important suggestions to this thesis. Many thanks go in particular to Takuro Yonezawa, PhD candidate, and other Move!group's members; they have made my life in Japan colorful, enjoyable and to me, that is a warm friendship.

Vietnamese friends in Keio University have inspired my life during these two years. I sincerely thank them for being here with me through all the good and the bad time. I am grateful to JICE's members, who are our very first Japanese friends. Without their help, we would have not known how to manage our lives sometimes.

My deepest thank goes to my family and my beloved ones for their understanding and persistent confidence in me despite the distance between us. This thesis is simply impossible without them.

Finally, I would like to thank everybody who is important to the completion of this thesis, as well as expressing my apology that I could not mention personally one by one.

Vu Thi Huong Giang

February 11, 2010

Bibliography

[1] A User description. *Technical document Erricson AB 1994-2007.*

[2] David Andersen, Haris Blakrishnan, Frans Kaashoek, Robert Morris. Resilient Overlay Networks. *ACM SIGOPS Symposium on Operating Systems Principles Review*,volume 35, pages 131–135, 2001

[3] Paola Bellavista, Antonio Corradi and Luca Foschini. Context-aware handoff middleware for transparent service continuity in wireless networks. *Pervasive and Mobile Computing*, Volume 3, Issue 4, August 2007.

[4] Jason But, Urs Keller and Grenville Armitage. Passive TCP Stream Estimation of RTT and Jitter Parameters. In *the 30th IEEE Local Computer Networks*, Australia, 2005.

[5] Robert Leston Carter and Mark Edward Crovella. Measuring bottleneck link speed in Packet-Switched Networks. *Performance Evaluation*, volume 27-28, pages 297–318, October 1996.

[6] Bin Bin Chen and Mun Choon Chan. Resource Management in Heterogeneous Wireless Networks with Overlapping Coverage. In *First International Conference on Communication System Software and Middleware Comsware*, New Delhi, India, 2006

[7] Ling Jyh Chen, Tony Sun, Benny Chen, Venkatesh Rajendran and Mario Gerla. A smart decision for vertical handoff. In *Proceedings 4th*

ANWIRE International Workshop on Wireless Internet and Reconfigurability, Athens, Greece, 2004.

[8] Z. Dai; R. Fracchia; J. Gosteau; P. Pellati; G. Vivier. Vertical handover criteria and algorithm in IEEE 802.11 and 802.16 hybrid networks. In *IEEE conference on Communications (ICC'08)*, 2008

[9] Sally Floyd, Mark Handley and Jitendra Padhye, A comparison of Equation-based and AIMD Congestion control. In 2000.

[10] Krishna P. Gummadi, Stefan Saroiu, Stevan D. Gribble. King: Estimating Latency between Arbitrary Internet End Hosts. In *ACM SIGCOMM Internet Measurement Workshop*, 2002.

[11] Michio Honda, Jin Nakazawa, Yoshifumi Nishida, Masahiro Kozuka and Hideyuki Tokuda. A Connectivity-Driven Retransmission Scheme Based On Transport Layer Readdressing. In *The 28th IEEE International Conference on Distributed Computing System*, Beijing, China, 2008.

[12] Ninging Hu and Peter Steenkiste. Estimating Available Bandwidth Using Packet Pair Probing. In *Technical report* 2002.

[13] Manish Jain and Constantinos Dovrolis. End-to-End available bandwidth: Measurement Methodology, Dynamics, and Relation with TCP throughput. In *Proceedings ACM SIGCOMM*, pages 295–308, August 2002.

[14] Manish Jain and Constantinos Dovrolis. Pathload: A Measurement Tool for End-to-end Available Bandwidth. In *Proceedings of Passive and Active Measurements (PAM) Workshop*, pages 14–25,2002.

[15] Hao Jiang and Constantinos Dovrolis. Passive estimation of TCP round-trip times. *ACM SIGCOMM Computer Communication Review*, volume 32, pages 75–88, July 2002.

[16] David B. Johnson, Charles Perkin and Jari Arkko. Mobility Support in IPv6. *RFC3775*, June 2004.

[17] Eddie Kohler, Mark Handley and Sally Floyd. Designing DCCP: congestion control without reliability. In *ACM SIGCOMM Computer Communication Review*, pages 27–38, 2006.

[18] Ewa Kozlowska et al. Optimization of handover mechanism in 802.16e using Fuzzy Logic. In *Personal Wireless Communications, (Boston: Springer)*, pages 115–122, 2007.

[19] Cheng Wei Lee, Li Ming Chen, Meng Chang Chen, Yeali Sunny Sun. A framework of handoffs in Wireless Overlay Network based on Mobile IPv6. In *IEEE Journal on Selected Areas in Communications*, volume 23, pages 2118-2128, 2005.

[20] Su Kyoung Lee, Kotikalapudi Sriram, Kyungsoo Kim, Yoon Hyuk Kim and Nada Golmie. Vertical Handoff Decision Algorithms for Providing Optimized performance in Heterogeneous Wireless Networks. In *IEEE Transaction on Vehicular Technology*, January 2009.

[21] Hongwei Liao, Ling Tie and Zhao Du. Vertical Handover Decision Algorithm based on Fuzzy Control Theory. In *Proceedings of the First International Multi-Symposiums on Computer and Computational Sciences (IMSCCS'06)*, pages 309–313, 2006.

[22] Min Liu, Zhong Cheng Li, Xiao Bing Guo, E. Dutkiewicz and Ming Hui Wang. SAVA: A novel self-adaptive vertical handoff algorithm for heterogeneous wireless networks. In *IEEE Conference on Global Telecommunications (GLOBECOM'06)*, pages 1–5, 2006 .

[23] Jiping Lv, Yuanchen Ma and Satoshi Yoshizawa. Intelligent Seamless Vertical Handoff algorithm for the next generation wireless networks. In *Proceedings of the 1st international conference on MOBILe Wireless MiddleWARE, Operating Systems, and Applications Mobilware*, volume 278, 2008.

[24] Ed. Manner and M. Kojo. Mobility Related Terminology. *RFC 3753*, June 2004.

[25] R. Moskowitz and P. Nikander. Host Identity Protocol (HIP) Architecture. *RFC 4423*, May 2006.

[26] E. Nordmark and M. Bagnulo. Shim6: Level 3 Multihoming Shim Protocol for IPv6 *RFC5533*, April 1999.

[27] Enrique Stevens Navarro and Vincent W.S Wong. Comparison between Vertical Handoff Decision Algorithms for Heterogeneous Wireless Networks. In *Vehicular Technology Conference (VTC'06)*, 2006.

[28] Charles Perkins. IP mobility support. *RFC2002*, October 1996.

[29] M. Allman, V. Paxon and W. Stevens. TCP congestion control. *RFC2581*, April 1999.

[30] Vinay J. Ribeiro, Rudolf H. Riedi, Richard G. Baraniuk, Jiri Navratil, and Les Cottrell. pathChirp: Efficient Available Bandwidth estimation for network path. In *Proceedings of Passive and Active Measurement Workshop*, 2003.

[31] Simon Schutz, Lars Eggert, Stefan Schmid and Marcus Brunner. Protocol enhancements for intermittently connected hosts. *ACM SIGCOMM Computer Communication Review*, pages 5–18, volume 35, 2005.

[32] A. Shriram and J. Kaur. Empirical Evaluation of Techniques for Measuring available bandwidth. In *The 26th IEEE International Conference on Computer Communication (INFOCOM)*, 2007.

[33] Anita Singhrova and Nupur Parakash. A review of vertical handoff decision algorithm in Heterogeneous network. In *Proceedings of the 4th international conference on mobile technology, applications, and systems and the 1st international symposium on Computer human interaction in mobile technology (Mobility'07)*, 2007.

[34] R. Stewart, K. Poon, M. Tuexen, V. Yasevich and P. Lei. Sockets API Extensions for Stream Control Transmission Protocol. *draft-ietf-tsvwg-sctpsocket-19.txt*, August 20, 2009.

[35] R. Stewart, Q. Xie, K. Morneault, C. Sharp, H. Schwarzbauer, T. Taylor, I. Rytina, M. Kalla, L. Zhang and V. Paxson. Stream Control Transmission Protocol. *RFC 2960*, October 2000.

[36] R. Stewart, Q. Xie, M. Tuexen, S. Maruyama, M. Kozuka. Stream Control Transmission Protocol (SCTP) Dynamic Address Reconfiguration. *RFC 5061*, September 2007.

[37] Jacob Strauss, Dina Katabi, and Frans Kaashoek. A measurement study of available bandwidth estimation tools. In *Proceedings ACM SIGCOMM Internet measurement*, pages 39-44, 2003.

[38] Angela Wang, Cheng Huang, Jin Li and Keith W. Ross. Queen: Estimating Packet Loss Rate between Arbitrary Internet Hosts. In *The 10th Passive and Active Measurement Conference*, pages 57–66, April 2009.

[39] Kemeng Yang, I. Gondal, Bin Qiu and L.S. Dooley. Combined SINR based Vertical Handoff Algorithm for Next Generation Heterogeneous Wireless Networks. In *Proceedings IEEE Global Conference (GLOBECOM'07)*, 2007.

[40] Wenhui Zhang. Handover Decision Using Fuzzy MADM in Heterogeneous Networks. In *IEEE Conference on Wireless Communications and Networkings (WCNC'04)*, volume 2, pages 653–658, Atlanta, GA, 2004.

[41] Fang Zhu and J. McNair. Optimization for Vertical Handoff Decision Algorithms. In *IEEE Wireless Communications and Networkings Conference (WCNC'04)*, 2004.

[42] A Wikipedia. http://en.wikipedia.org/wiki/