Interface Selection In Multi-Interface Mobile Devices

Except where reference is made to the work of others, the work described in this thesis is my own or was done in collaboration with my advisory committee. This thesis does not include proprietary or classified information.

Sowmia Devi M.K.

Certificate of Approval:

Thaddeus A. Roppel Associate Professor Electrical and Computer Engineering	Prathima Agrawal, Chair Samuel Ginn Distinguished Professor Electrical and Computer Engineering
Chwan-Hwa "John" Wu Professor Electrical and Computer Engineering	Stephen L. McFarland Dean Graduate School

INTERFACE SELECTION IN MULTI-INTERFACE MOBILE DEVICES

Sowmia Devi M.K.

A Thesis

Submitted to

the Graduate Faculty of

Auburn University

in Partial Fulfillment of the

Requirements for the

Degree of

Master of Science

Auburn, Alabama
August 7, 2006

INTERFACE SELECTION IN MULTI-INTERFACE MOBILE DEVICES

Sowmia Devi M.K.

Signature of Author

Date of Graduation

Vita

Sowmia Devi M.K., daughter of Dr. Krishna Baba and Gnana Kumari was born in Tirupur, India on July 7th 1981. She graduated high school from Sri Sankara Vidhyashramam, Chennai in March 1998. She graduated from Madras University, Chennai with a Bachelor of Engineering degree in July 2002. She began her studies in Electrical and Computer Engineering at Auburn University in August 2003.

Thesis Abstract

Interface Selection In Multi-Interface Mobile Devices

Sowmia Devi M.K.

Master of Science, August 7, 2006
(B.E., Madras University, 2002)

72 Typed Pages

Directed by Prathima Agrawal

This thesis describes interface selection in multi-interface mobile devices. A multi-interface mobile device is defined as a device equipped with more than one network interface to help connect it to an external network such as the Internet. The network could be wired or wireless in nature.

This thesis consists of three important modules. The first, Score Function calculation to determine the best interface for an application at any given point in time. This method takes into consideration both network parameters such as signal strength and cost and client parameter such as power.

The second is a handoff algorithm based on NAT (Network Address Translation) and IP (Internet Protocol) tunneling principles. It helps in the transfer of active sessions from on network interface to another irrespective of whether the two interfaces are similar or not.

The third module is to enable multiple interfaces to work simultaneously.

The interface selection process involving the score function calculation and the simultaneous use of multiple interfaces has been experimentally tested on a test bed successfully. Simulation results of the interface selection algorithm is also presented in this thesis.

ACKNOWLEDGMENTS

I would like to thank my advisor Dr. Prathima Agrawal for her patience, support and guidance in nuturing me into a successful graduate student. She has been a constant source of inspiration throughout the work on my thesis.

I would like to express my gratitude to Dr. James Ellis, my employer, for having helped me through all the adversities until the completion of my degree.

I would also like to thank my family, my father - Dr. Krishna Baba, my mother - Mrs. Gnana Kumari, my sister - Lavanya, my brother-in-law - Roshan and last but not the least my ever loving fiancé - Vivek, for having the confidence in me even when I lacked it. Their love and support has made me the person I am today.

They say “A friend is one who walks in, when the rest of the world walks out”. I would like to thank God for having blessed me with such wonderful friends who have been there for me through this journey of my life. For this, I would like to thank, Rajesh, Madhu, Meetika, Siddhi, Suma and Ambika.

Style manual or journal used Journal of Approximation Theory (together with the style known as “aums”). Bibliograpy follows van Leunen’s *A Handbook for Scholars*.

Computer software used The document preparation package TEX (specifically LATEX) together with the departmental style-file `aums.sty`.

Table of Contents

List of Figures

LIST OF TABLES

CHAPTER 1

INTRODUCTION

The advent of new wireless technologies over the years has enabled the incorporation of more than one network interface in mobile terminals. Advances in semiconductor technology have enabled cost-effective and sophisticated multi-interface mobile devices. A multi-interface mobile device is a mobile terminal equipped with more than one network interface to help it connect to a network (example: the Internet). The device may be equipped with more than one interface of the same kind or a heterogeneous mix of interfaces. The network access technology associated with each of these interfaces may differ in communication cost, bandwidth, data rates and transmission power.

The deployment of more than one network access technology (cellular, wireless local area network etc.) but different service providers in some geographic locations has resulted in wireless overlay network access layers [9]. These sections of overlay networks favor the use of multiple network interfaces simultaneously to ensure better quality of service for the end user [9]. The multiple accesses provided by the overlapping networks also facilitate in the dynamic selection of an interface(s) for an application [7].

1.1 Problem Statement

The objective of the research presented in this thesis is to:

1. Understand the operation of multi-interface mobile devices. To devise a method to efficiently make use of the resources that are provided by them.

2. Develop a mathematical model to choose the best interface for an application at any given instant

in time, taking into consideration the network and client conditions at that instant.

3. Propose a handoff algorithm to seamlessly transfer active sessions from one network interface to another.

4. Enable the simultaneous use of multiple interfaces in multi-interface mobile devices to increase the overall performance of the devices.

1.2 Overview of Results

The main contributions of this thesis have been summarized as follows:

1. A mathematical model to choose the best interface for any given application at any given point in time. This model takes into consideration the different network and client parameters that play a vital role in the performance of the application. The model considers two client parameters: a) battery power consumed by the interface and b) cost of the network access being used. The model also takes into consideration one network parameter, the signal strength for the network access technology being used. In addition, the mathematical model incorporates a weighting system which helps in providing a higher importance to one of the parameters as compared the other two.

2. A handoff algorithm for switching an active application from one interface to another is proposed. The algorithm enables a seamless handoff between interfaces of different types (vertical handoff) under the assumption that the coverage range of different network accesses is overlapping in nature. A loss in connectivity is possible if a vertical handoff is made in the absence of overlapping networks.

3. The problems of unnecessary handoffs and ping-pong effect application alternating between two interfaces are reduced by including a threshold component to the entire model. This ensures the

necessity of a handoff only if the parameter values cross a predetermined threshold value. The sensitivity of the system can be defined by fine-tuning this component.

4. Simultaneous use of multiple interfaces in a multi-interface mobile device has been made possible, the advantages of which have been discussed in detail in chapter 4.

1.3 Thesis Outline

Chapter 2 explains the advantages of multi-interface mobile devices. The interface selection algorithms already existing in the literature have been discussed along with the model developed in this thesis (SAs Interface Selection Algorithm (SAISA)).

Chapter 3 briefly surveys different handoff algorithms. This chapter also proposes a NAT/IP tunneling-based (Network Address Translation) handoff algorithm. The results obtained from simulating SAISA for several scenarios created to emulate various communication and computing situations are also discussed here.

Chapter 4 describes how simultaneous use of multiple interfaces in a multi-interface mobile device helps in improving the performance of applications at the user-end.

Chapter 5 describes the test bed setup and the experiments performed on them.

Chapter 6 concludes with the contributions of this thesis and the scope for future work.

CHAPTER 2

INTERFACE SELECTION FOR MULTI-INTERFACE MOBILE DEVICES

This chapter describes a multi-interface mobile device. An overview of interface selection algorithms in the literature is presented. A new interface selection algorithm named SAISA (Sowmia-Agrawal Interface Selction Algorithm) is proposed. This chapter also presents a simple mathematical model that can be adopted for interface selection in a multi-interface device.

2.1 Multi-interface mobile device and its advantages

A multi-interface mobile device is equipped with more than one network interface for connecting to a variety of networks. For the sake of simplicity, terms "mobile device" and multi-interface mobile devices have been used interchangeably in this thesis. A mobile device with only one network interface, is referred to as an "ordinary mobile device".

Figure 2.1, shows an example of a multi-interface mobile device equipped with several network interfaces namely, cellular, wireless LAN (Local Area Network), infrared and Bluetooth (BT)interfaces. The services supported by these multi-interface mobile devices have been discussed in the IETF mobile IP working group internet draft [1] and in [2]. Some of these services are summarized below:

Load sharing: Multi-interface mobile devices may decrease the end-user delay by helping applications connect to an outside network (for example, the Internet) through more than one network interface device simultaneously. Instead of one network interface device bearing the entire load of all the client applications, the load is shared by all the interfaces present within the range of their respective access networks (simultaneous use of multiple interface is explained in chapter 4). This

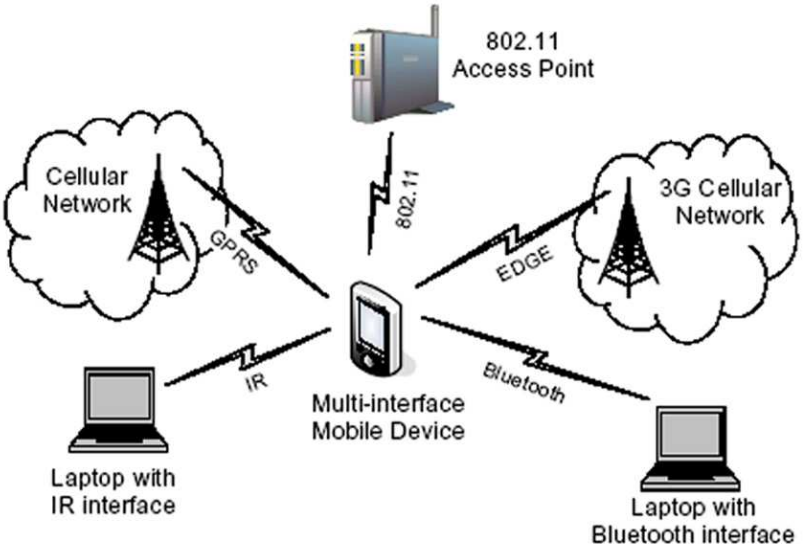


Figure 2.1: Multi-interface mobile device

reduces the wait time for applications and thus reduces the end-user delay.

Pervasive (wide-spread) access: In general, a multi-interface mobile device has more network options to choose from and hence have wide-spread access as compared to a single interface mobile device. The only exception to this is when a single interface mobile device has a network interface with the highest coverage range. A multi-interface device might possess more than one network interface device that is within the range of its respective access network. When these access networks are non-overlapping in nature, the multi-interface device typically has wide-spread access.

Reliability: Multi-interface mobile devices enable more reliable information transfer. Reliable information transfer to a destination can be achieved by employing redundant (more than one interface) access methods (simultaneous use of multiple network interface is explained in chapter 4). Such redundancy in communication will enhance the reliability.

Security: Multi-interface mobile devices promise a better transfer of security sensitive information compared to a single-interface mobile device. The interfaces can be classified based on security issues. Depending on security measured needed by an application, the most secure interface is selected.

Interface selection: Interface selection option is one of the advantages of a multi-interface mobile device. It provides the client with the freedom to choose the interface to be used for an application instead of being forced to use the interface that he has been provided with as in the case of a ordinary mobile device. The preferences for the choice of interface could be user based, application based, network conditions based or client conditions based or a combination of these. This will be explained in detail in section 2.2.2.

The following sections describe criteria for interface selection and the different interface selection algorithms. The best interface is selected from among the active interfaces and those that are appropriate for the application initiated by the client using an interface selection algorithm. This would help towards maximizing the performance of the individual applications. Section 2.2 reviews previous work on interface selection algorithms as well as the SAISA algorithm.

2.2 Interface selection

2.2.1 Literature review of interface selection algorithms

An interface selection algorithm is designed to select the most appropriate interface from among the many available in a mobile terminal. It chooses an optimal network interface device for an application that requires to be networked. In this thesis the term optimal interface has been used to describe a network interface that provides the best possible service out of all the available interfaces for a given application. Both client and network parameters are taken into consideration while selecting an interface. The minimum requirements for a particular application could be user defined or predefined in the algorithm. This section discusses the different interface selection algorithms and their limitations. Reference [5] describes an interface selection procedure. Though it discusses various components and requirements for the interface selection algorithm, details of the mathematical basis for the selection and results are not provided. In [8] the algorithm for interface selection is based on a cost function, $cost_n$. It is a function of the parameters values B_n, P_n, C_n of the network interface n. The equation for the cost function is represented as:

$$Cost_n = f(B_n, P_n, C_n) \tag{2.1}$$

Where,

B_n, is the bandwidth offered by the access network.

P_n, is the power consumed for using the network interface.

C_n, is the cost charged to the client for using the access network.

Reference [7] proposes a Smart Decision (SD) model that employs the score function method to decide the best network interface to use at a given time. This method, which is user interactive, uses a wide variety of parameters, both at the network and client level. The network parameters considered are link capacity (e) and cost (c) of network access. The client parameter considered client power consumption (p) for communication through a particular network interface. These parameters are stored periodically in a configuration file. These are retrieved by a decision module while evaluating the available interfaces for the final selection.

The Smart Decision model computes a score function that represents a score for every interface present on the mobile device. It takes into consideration the network and client parameters to decide the best possible interface. However, the score function calculations and choice of the best interface is made for the system as a whole and not for every application. This implies that all applications irrespective of their individual requirements use the same interface. The equation used to compute the score function was formulated as described below.

The parameters that are chosen, s, p and c are first normalized. Normalization is done so that the sum of the parameters values in different units is meaningful [8]. The normalized functions f_{ei}, f_{ci}, f_{pi} are then multiplied by their respective weights (w_e, w_c, w_p). The sum of the weights is required to be one at all times. The products of the normalized parameter values and their weights are added to give the score function S_i. This score function is calculated for each of the interfaces and the interface with the highest score function is selected for communication. The mathematical model developed is represented as follows:

$$S_i = \sum_{j=1}^{k} w_j f_{ji} \qquad \text{where,} \quad 0 < S_i < 1 \qquad \text{for} \quad \textstyle\sum_{j=1}^{k} w_j = 1 \tag{2.2}$$

$$S_i = w_e f_{ei} + w_c f_{ci} + w_p f_{pi} \tag{2.3}$$

$$\begin{gathered} f_{ei} = \frac{1}{e^{\alpha_i}}, \quad f_{ci} = \frac{e^{\beta_i}}{e^{M}}, \quad f_{pi} = \frac{1}{e^{\gamma_i}} \\ \text{where,} \quad \alpha_i \geq 0, \quad M \geq \beta_i \geq 0, \quad \gamma_i \geq 0 \end{gathered} \tag{2.4}$$

The limitations of this model are as follows: As mentioned earlier, the interface selection algorithm is specific to the system and not application-specific. This is disadvantageous because different applications may have different behavior patterns and requirements. For example, consider an HTTP (HyperText Transfer Protocol) application that consumes more bandwidth than an email application. Consider a scenario where a mobile device is equipped with two interfaces A and B. After calculating the score functions, let interface B is declared as the better of the two choices. At time instant t1, an HTTP application is started which seems to be consuming all the bandwidth. At time instant t2, an email application is also being started. The email application though has been put on hold due to the unavailability of bandwidth and is required to wait for the HTTP application to finish for it to be processed. This is because all applications are required to use the same interface. This not only delays the processing of the second application but also results in wastage of interface B's resources which could have been used to process the email application. Hence, using the only one interface at a time to process all applications is not the best option from the User's point of view. This method also results in unnecessary handoffs and will be explained in detail in Chapter 3.

The most common criterion for interface selection is to use relative signal strength (RSS) as explained in [10]. The advantages of this method are its simplicity and its adaptability to the changing network conditions. This method obtains the real time signal strength values from the base station of the network and chooses the strongest received base station at all times. This method takes only relative signal strength values into consideration. No other parameters are taken into consideration for interface selection.

The disadvantage of this method is that it is also not application specific. Not all applications have the same requirements with respect to network conditions. In addition to the differing needs of applications, users have very diverse priorities. Assume two users namely, User1 and User2. User1 might consider a VoIP (Voice over Internet Protocol) session more important than an ongoing FTP (File Transfer Protocol) session, while User 2 might feel vice versa. Hence personal preferences could also vary. Thus the individual performance and priority of each application is not taken into consideration. Another limitation of this method is that it stimulates unnecessary handoffs and can possibly experience ping pong effect due to the oscillatory nature of RSS values at access network boundaries. These effects will be further elaborated upon in Chapter 3.

2.2.2 SA Interface Selection Algorithm (SAISA)

The algorithm presented in this thesis is named SAISA (Sowmia-Agrawal Interface Selection Algorithm). It consists of three components, namely:

(1) Score Function calculation - a mathematical model explained in section 2.2.3

(2) Simultaneous use of multiple interfaces - explained in sections 4.3

(3) NAT/IPtunnelling-based handoff - explained in section 3.3.2

Interface selection is the process of choosing the best network interface device from amongst all the active and available network interfaces present in a mobile device for every application initiated by the client.

Two approaches to interface selection

(1) Static interface selection is as the selection of an appropriate network interface for an application before the application begins. Figure 2.2 is a flowchart describing the steps involved in static scheduling of applications. Static scheduling of applications on multiple interfaces simultaneously employs the use of Iptables [3] functions that already exist in the Linux (operating system) kernel. This is explained in detail in Chapter 4.

(2) Dynamic interface selection can be described as the selection of the best interface for an application during the initiation of the application and during its execution until its completion. It ensures that the application is using the best interface at all times until the task is complete. Hence, during the execution of an application if the algorithm realizes that the interface being used is no

longer the best, an alternate and appropriate interface is chosen for communication. Dynamic interface selection may become necessary due to the changes in network condition during the execution of an application. An example of the cause of change of network conditions could be user mobility. This would require all active sessions (ongoing applications) to be transferred from the old interface to a new one chosen without losing connectivity. Such handoff plays a major role in dynamic interface selection. This will be described in detail in the following chapter.

As can be seen from Figure 2.2, on initiation of an application the interface selection algorithm examines the configuration file of the application. The preconfigured, preferred list of interfaces is retrieved from the configuration file and checked to see if those interfaces are within the network range of their respective network access technologies. After this process, a list is made of all the interfaces that are available (within network access range).

The configuration files corresponding to each interface is retrieved to extract the values of the parameters under consideration at that moment. These values are then used to calculate the score functions as described in the following section. The highest score function value is obtained after comparing all score functions. The interface with the highest value is used to process the application. In case none of the preferred interfaces are within network range of their respective network accesses, the default interface or any interface that is within network range is used to process the application. The quality of the connection is not guaranteed by the algorithm in this case because the application is not using any of the preferred interfaces.

2.2.3 Score function calculation

In SAISA, a score function is calculated as a function of the network and client parameters as in [7]. In this study, only three parameters have been considered. The score function equation 2.5

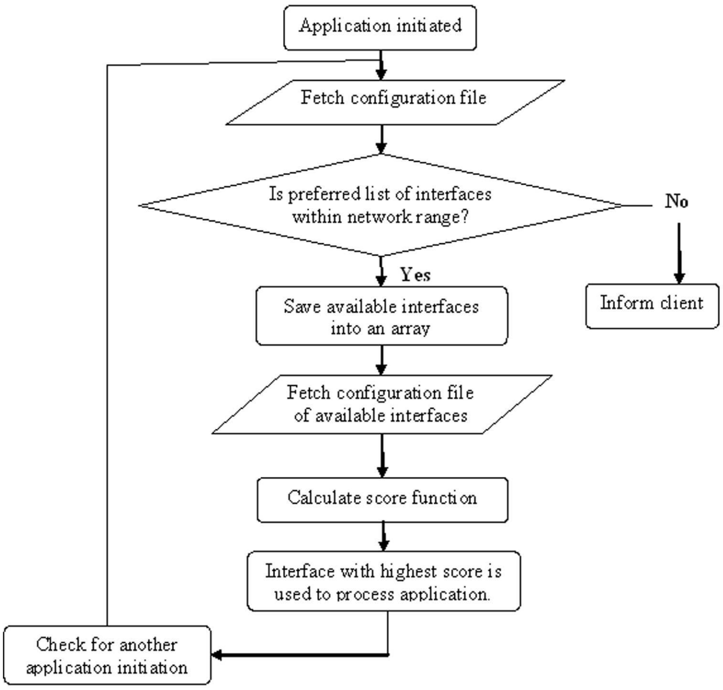

Application initiated
Fetch configuration file
Is preferred list of interfaces within network range?
No
Yes
Inform client
Save available interfaces into an array
Fetch configuration file of available interfaces
Calculate score function
Interface with highest score is used to process application.
Check for another application initiation

Figure 2.2: FlowchartBasic working of SAISA

can be extended for any number of parameters. The parameters considered here are; signal strength (s), cost of using the network access technology (c) and client power consumed for the paticular access technology (p). It is to be noted that a high signal strength value contributes proportionately to the SF (Score Function), whereas cost and power contribute inversely to SF. This is because, an interface having a good signal strength is a better choice, while an interface costing more or consuming more power is a poor choice.

At first the three parameter values are normalized to make the sum of these values in different units meaningful. Equations 2.6, 2.7 and 2.8 represent the normalization equations of the three parameters.

$$SF_i = w_s \times f_{si} + w_p \times f_{pi} + w_c \times f_{ci} \tag{2.5}$$

$$f_{si} = \frac{1}{e^{\frac{\alpha_i}{s}}} \tag{2.6}$$

$$f_{pi} = \frac{1}{e^{\frac{p}{\beta_i}}} \tag{2.7}$$

$$f_{ci} = \frac{1}{e^{\frac{c}{\gamma_i}}} \tag{2.8}$$

Subject to the constraints:

$0 < SF_i < 1$

$\sum_{n=1}^{m} w_n = 1$

where, $\alpha_i \geq 0$, $\beta_i > 0$ and $\gamma_i > 0$

The variables s, p and c represent signal strength in dB, power consumption in mW and cost in /min. The constants α_i, β_i and γ_i are used to normalize the respective parameter values namely s, p and c. The exponential functions have been used to increase the sensitivity of the functions. Finally, they are inversed to bound the functions to a value between zero and one. Thus, denotes the Score function which is a sum of the products of the normalized parameter values and their respective weights keeping the sum of the weights and the score function to remain a value between zero and one. The weights $w_s, w_p and w_c$ represent the priorities assigned by the user to the parameters.

The priorities can be varied by changing the value of the weights attached to each one of them. Depending on the importance of a parameter during the time of calculation of the score function, the value of the weights can be varied. This makes the model increasingly interactive.

2.3 Summary

A brief description of multi-interface mobile devices along with their advantages has been described in this chapter. A detailed explanation of the different interface selection algorithms in the literature, interface selection criteria and SAISA are discussed. SAISA has shown how the resources available with multi-interface mobile devices are effectively used.

The next chapter explains dynamic interface selection which is made possible by incorporating the effect of handoff into the score function module. The topics covered are, handoff, its definition, the types of handoff and various handoff algorithms. A NAT/IP tunneling-based handoff algorithm is also described.

CHAPTER 3

HANDOFF

A variety of devices are available for todays mobile users to network with other hosts. Chapters 2 and 3 have stressed the advantages of a mobile device having multiple interfaces. A clients power to be mobile necessitates networks to support applications while on the move. User mobility thus adds a new challenge of having to maintain the active sessions without a loss in connectivity. With overlapping network access technologies and multi-interface mobile devices this task becomes less complicated.

This chapter will elaborate the topic of dynamic interface selection. The main ingredient required to change the interface selection process from static to dynamic is handoff. The need for dynamic interface selection arises due to the necessity of designing for handing off applications from one interface to another interfce.

3.1 Definition of Handoff

Handoff is defined as the process of transferring the active sessions in a mobile device from one network access point to another. The handoff is considered seamless when the transfer of these active sessions is made from one access point to another without a loss in connectivity.

3.2 Types of handoff

A handoff can be broadly classified as a) hard handoffs and b) soft handoffs. Handoffs can also be categorized based on whether the decision is made by the mobile device or the network. From the decision process point of view handoffs can be of three types, namely a) network-controlled

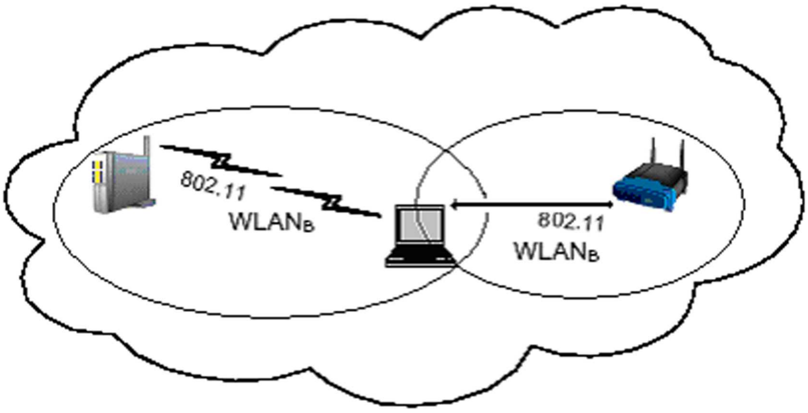


Figure 3.1: Hard handoff - Before handoff

handoff, b) mobile-assisted handoff and c) mobile-controlled handoff [25]. Handoffs can also be described based on the type of network access technologies involved as a) horizontal handoff b) vertical handoff. The following sections describe the different types of handoffs in detail.

3.2.1 Hard and Soft handoffs

Hard handoff: A hard handoff can be defined as a transfer of active sessions from one access point to another where the mobile device is connected to only one access point at any time. A hard handoff can also be described as "break before make" [25]. There exists a period of no connectivity when the mobile device gets disconnected from the old network and before getting connected to the new network. This handoff method is avoided as much as possible to ensure that the user is provided with good quality of service.

Soft handoff: Soft handoff is defined as a transfer of an active session from one access point to another without a loss in connectivity. A soft handoff can also be viewed as a "make before

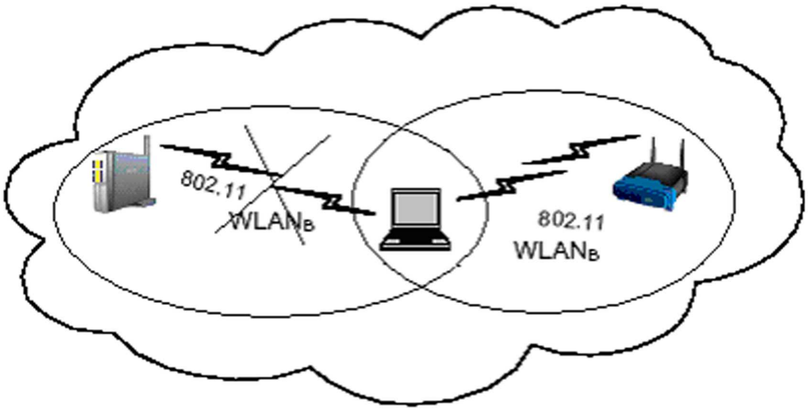


Figure 3.2: Hard handoff - After handoff

break" handoff. The mobile device makes its connection with the new access point before getting disconnected from the old access point. This can be easily achieved in case of overlay access networks where more than one network access technologies are active [9].

3.2.2 Network-controlled, Mobile-assisted and Mobile-controlled handoffs

Network-controlled handoff: In network-controlled handoffs, the handoff decision is made by the network based on the measurements (example: signal strength) of the mobile device. The base station makes the handoff decision based on the network information collected from all the mobile devices under its coverage area.

Mobile-assisted handoff: Here the mobile device makes the measurements while the network makes the decision for handing off the active sessions.

Mobile-controlled handoff: In mobile-controlled handoffs, the mobile devices make the handoff decision. The mobile device makes both the measurements and the decision of the handoff.

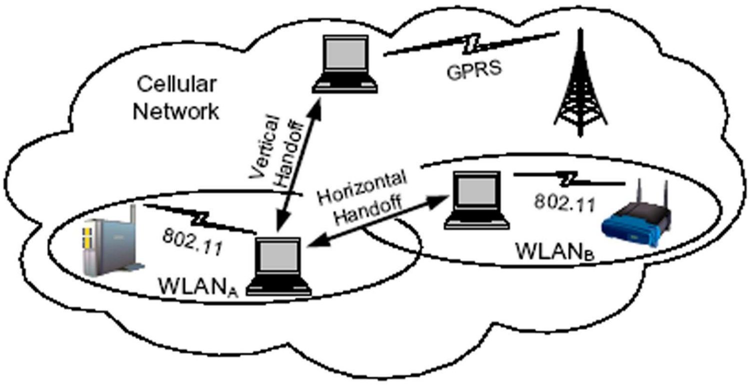


Figure 3.3: Horizontal and Vertical handoff

3.2.3 Horizontal and Vertical handoff

Figure 3.3 shows a vertical handoff between dissimilar networks (wireless LAN and cellular) and a horizontal handoff among two similar networks (WLANs).

Horizontal handoff: A handoff is said to be horizontal when an active session is transferred from one to another interface of the same type. For example, when a mobile client moves from one WLAN network to another as can be seen from Figure 3.3, the active sessions are transferred between similar interface types. This is called horizontal handoff.

Vertical handoff: If the active sessions are transferred from one network access technology to another, the handoff is said to be a vertical handoff [9]. For example, when a mobile client moves from a WLAN network to a cellular network, as can be seen from Figure 3.3, the active sessions are transferred between two dissimilar networks. This is called vertical handoff.

3.3 Handoff algorithms

The following section briefs on the different handoff algorithms available in the literature. A NAT-based handoff algorithm has also been proposed in this chapter.

3.3.1 Literature review

An implementation of a vertical handoff system [9] for mobile users in wireless overlay networks Networks - a hierarchical structure of roomsize, building-size, and wide area data networks. This helps in providing the user with the best possible connectivity for as long as possible with as minimal a handoff delay as possible. It is also claimed that this vertical handoff system has reduced the handoff latency time considerably.

Reference [10] provides an overview of different handoff techniques and the current trends in handoff research. Techniques like the relative signal strength method, relative signal strength with threshold, relative signal strength with hysteresis and relative signal strength with threshold and with hysteresis have been described.

An intra-domain multicast based mobility architecture for providing efficient and smooth handoff, while being able to co-exist and inter-operate with other technologies has been described in [12]. NS-2 simulations in this paper have shown that the scheme provides a handoff performance that is 1.23 to 3 times better than other schemes depending on the network topology and mobility of the clients.

Reference [13] is a proposal by the IEEE 802.21 working group to support handoffs for both mobile and stationary users. The working group is creating a standard named Media Independent Handover (MIH), the scope of which is to develop a specification that provides link layer intelligence and other related network information to the upper layers to optimize handovers between heterogeneous media. This includes links specified by 3GPP (3rd Generation Partnership Project) [41], 3GPP2 (3rd Generation Partnership Project 2) [42] and both wired and wireless media in the IEEE 802 family of specifications.

A fully operational vertical handoff system has been designed, implemented and evaluated in [14]. Reference [14] describes how the coverage area of WLAN (Wireless Local Area Network) networks is very less compared to that of cellular networks and how the system OMNICON, helps mobile clients to fall back on cellular networks when the WLAN signal is unavailable. Reference [17] describes a hierarchical communication subsystem using existing radio technologies. The system optimizing the power consumption of a mobile device during the wireless device discovery and connection setup process.

3.3.2 NAT/IPtunneling based handoff algorithm

The algorithm proposed in this thesis is similar to the handoff mechanism used in [7]. It is based on Network Address Translation (NAT). The NAT is a function that belongs to the IPtables module in the Linux kernel. It is also a table similar to that of the Mangle and Filter tables discussed in section 4.3.2. It is used to map one IP address, known publicly, to one or more IP addresses known privately. Some of the most common reasons for using NAT are that they help in reducing the scarcity of IP addresses behind a LAN, security and sharing internet connections.

Section 4.4 describes dynamic scheduling of packets requiring handoff procedures that ensure the best interface being used by an application at any given point of time both at the start and during the execution of the application. This transfer of interfaces during the execution of applications needs to be quick and seamless. Multi-interface mobile devices and wireless overlay networks help in achieving this to a certain extent.

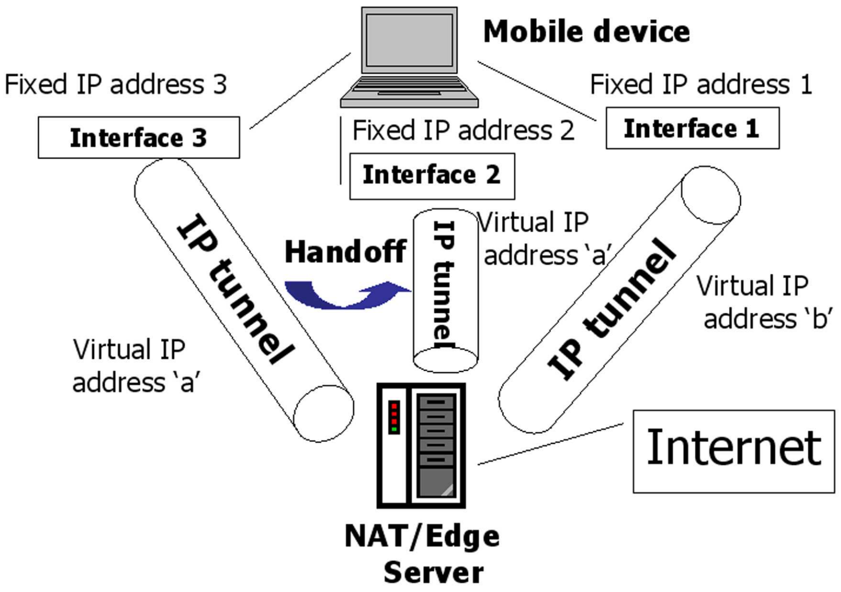


Figure 3.4: NAT/IP tunelling based handoff architecture

Handoff in the algorithm proposed in this thesis essentially implies the transfer of active sessions from one interface in a mobile device to another interface in the same device. The interfaces could be of the same or of different types. When the interfaces involved in a handoff are of different types (vertical handoff), making the handoff seamless becomes less complicated when the assumption of wireless overlay networks is made.

The NAT/IP tunneling-based handoff architecture requires the use of a NAT server for handoff at the edge of the mobile network as shown in Figure 3.4. The reason for this being, in normal handoff scenarios as explained in some of the above literature, a handoff is usually necessitated by the mobility of the client from one coverage area to another. Hence, the handoff is between two base stations not necessarily of the same kind. Also, at any given point, only one interface is being used for communication. However, in practice, more than one interface can be active at a given time. The

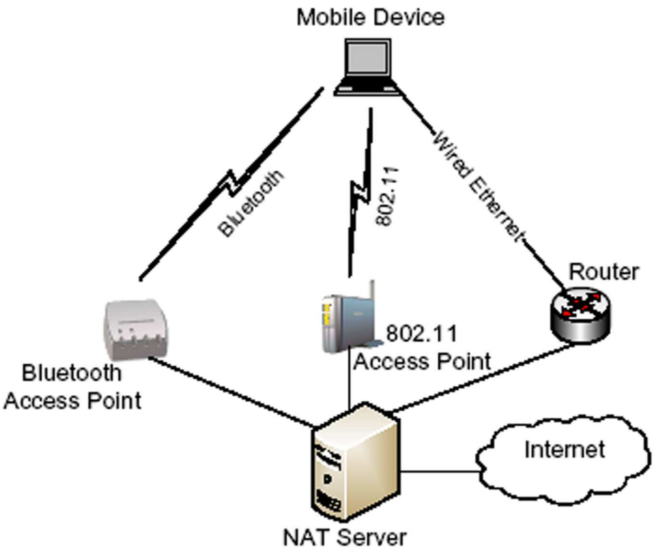


Figure 3.5: Multiple communication paths - NAT

reason for handoff need not necessarily be mobility of the client device in our case. Handoff could take place because of parameter priority change (example: signal strength or cost of connection) or because of change in network or client parameter values. In addition to these reasons, the need for a NAT server is essential when the interface to be used by both the outgoing and incoming traffic is not an obvious choice. It is rather calculated and specified by the interface selection algorithm from among several.

The NAT server thus acts as a gateway between the network to which the client belongs and other networks. The NAT serves monitors, filters and helps packets to leave and enter the correct client via the interface of its choice.

Figure 3.5 shows a multi-interface mobile device equipped with three interfaces namely; Bluetooth, Wireless LAN and Ethernet (LAN). The three interfaces are connected to their respective access points which are connected to the edge router at the periphery of the Internet. The NAT server is the first point of contact for any packet entering the network and the last point of contact in the client network for any packet leaving the network. This way both the outbound and their respective inbound packets can be controlled.

The NAT based handoff algorithm is client-controlled. This is because the client measures the client as well as network parameter values (example: signal strength) and lets the algorithm decide if the interface being used is the best. If it is not the best, the handoff decision is selected by the algorithm at the client and hence affects client-controlled handoff. The handoff could be horizontal or vertical depending on the interfaces the client is equipped with.

The function of a NAT server is to keep an updated routing table. The routing table includes the source and destination machine information of every packet that leaves the network. This information is then used when there is a return packet trying to enter the client network. The NAT server does a reverse lookup and sends the packet back to the correct machine. This procedure is used in the NAT based handoff algorithm described in this section.

The multi-interface mobile device depicted in Figure 3.5 is equipped with three interfaces each of which has a fixed IP address of its own. This IP address is used to make a physical connection with any other host or server. Virtual IP addresses are also assigned to each of the interfaces by the algorithm. A virtual IP address is defined as an IP address that is shared between multiple domain names or servers. It helps in eliminating host dependency upon individual network interfaces [26]. Thus every interface has two IP addresses: one, the fixed IP address and the other, the virtual IP address.

The IP tunneling method is employed in the NAT based handoff algorithm. An IP tunnel is made between every interface and the NAT server, the NAT server being on one end of the tunnel while the client serving as the other. All application layer communications are bound to the tunnel and since virtual IP addresses are being used, all packets will be sent to these addresses making them less dependent on the interface. When there is a need for interface switching or handoff, the mobile host is responsible for switching the underlying physical connection of the virtual tunnel to the new interface and the NAT server is intimated of the change. This basically means that the actual IP address changes while the virtual IP address remains the same. This information is then updated at the routing table in the NAT server. The packets that are sent thereafter would be delivered to the same virtual address and to the same client but through a different interface. This is because the virtual IP address is now bound to a different interface. All encapsulated packets are sent using the UDP protocol, thus not requiring the tunnel to be reset after the handoff. This helps keeping the TCP (Transfer Control Protocol) connections intact.

The handoff procedure described here is seamless under the assumption of a permanently present wireless overlay network. The interfaces in the mobile client that are within their coverage range are always kept alive. The always enabled interfaces in addition to the presence of wireless overlay networks ensures minimum handoff delays, thus making the handoff seamless. In the absence of a wireless overlay network, there could be a loss in connectivity.

3.4 Simulation results

The mathematical model described in section 2.2.3 was simulated in MATLAB (MATrix LABoratory - a data manipulation software package) to understand its behavior under different scenarios. In the following scenarios, the values for the three parameters (signal strength, power and cost) under consideration have been chosen specifically to emulate the various network and client conditions that arise during the operation of a network. Two interfaces were used in the simulations. The same scenarios were used in the test bed with two WLAN interfaces. There is agreement between the two results. Five scenarios have been considered:

Scenario 1 - Avoid unnecessary handoffs

Scenario 2 - Avoid Ping Pong effect

Scenario 3 - Give priority to signal strength

Scenario 4 - Give priority to power

Scenario 5 - Give priority to cost

The simulated outputs for scenarios 1 and 2 display four curves. The two blue curves represent the score function values of the two interfaces while the two green curves are the interfaces chosen for that time instant by SAISA and the algorithm in [7]. The rest of the output graphs display three details.

3.4.1 Scenario 1: Unnecessary handoffs

This scenario represents normal behavior of the network accesses. Equal weights are to all parameters with no preference to any particular one. Thus at any given instant, the interface that is being used is the best for that moment and with respect to that application.

Figure 3.6 shows the values of the input parameters that have been chosen as functions of time. The respective output using the SAISA is shown in Figure 3.7. The output for the same input values using the algorithm in [7] can also be seen in the figure.

On observing the graph in Figure 3.7, it can be seen that the interface selection algorithm in [7] switches interfaces twice. A switch is made from interface two to interface one at time instant two and another switch is made back to interface two at time instant 5. This interface switching is caused by the change in signal strength values between the two interfaces, as cost and power of

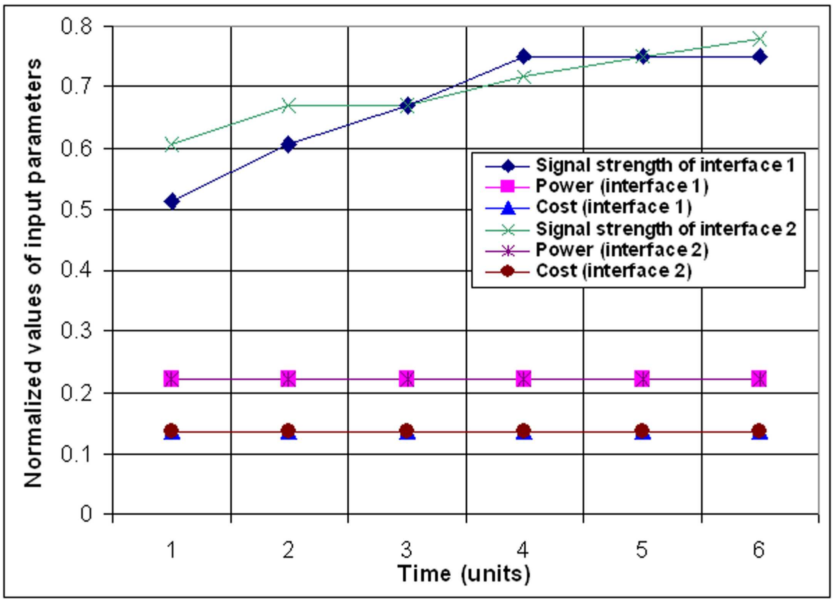


Figure 3.6: Variation of normalized parameter values as functions of time - Scenario 1

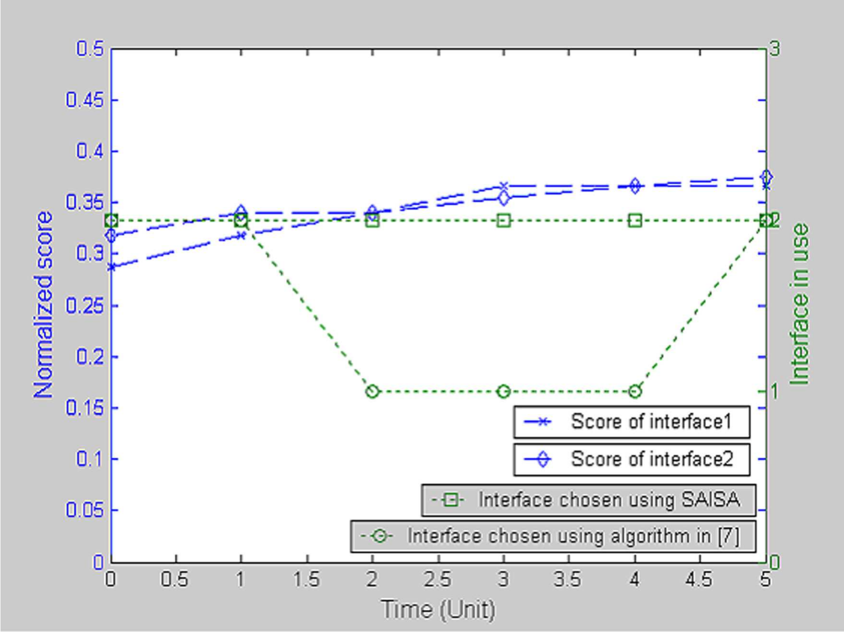


Figure 3.7: Interface selection using SAISA and algorithm in [7]

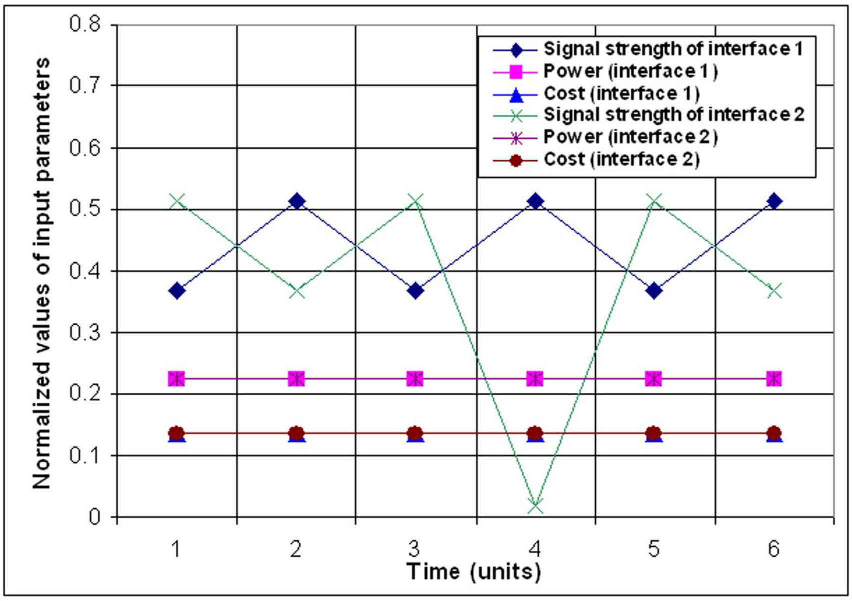


Figure 3.8: Variation of normalized parameter values as functions of time - Scenario 2

the two interfaces are maintained the same. SAISA, on the other hand does not switch interfaces even once. The reason being that the change in signal strength value is not higher than the threshold value that has been set. This feature of the SAISA has helped in avoiding unnecessary handoffs. A threshold value is set for every parameter value. Unless the change in the parameter value is below the threshold value, a handoff decision is not triggered. The threshold value can be user defined or predefined. The value of threshold will determine how senstive the client needs to be towards a change in the parameter values.

3.4.2 Scenario 2: Ping Pong effect

The ping pong effect is a phenomenon observed when a mobile device is at the boundary of the coverage area. This occurs due to the fluctuation of signal strength values. Equal weights have

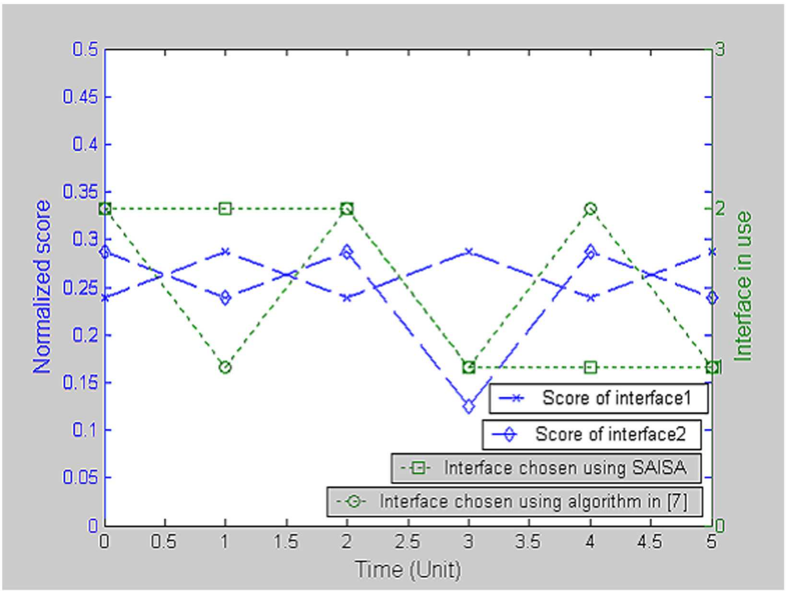


Figure 3.9: Choice of interfaces while using SAISA and algorithm in [7]

been assigned to the three parameters. As can be seen in Figure 3.8 the values of signal strength is fluctuated to emulate the ping pong effect. The choice of interface on simulating the algorithm in [7] can be seen to fluctuate with signal strength. The choice results in the application switching to the network interface with the higher signal strength value. Figure 3.9 shows a handoff at each one of the time instants. Thus, SAISA avoids the ping pong effect comfortably by switching interfaces only at time instant three due to the change in signal strength being higher than the threshold value. The choice of interfaces while using SAISA can also be observed in Figure 3.9.

3.4.3 Scenario 3: Signal Strength Priority

This scenario shows the behavior of the algorithm when a higher priority is given to one parameter. The value of w_s, the weight attached to signal strength is set to 0.6, while w_c and w_p,

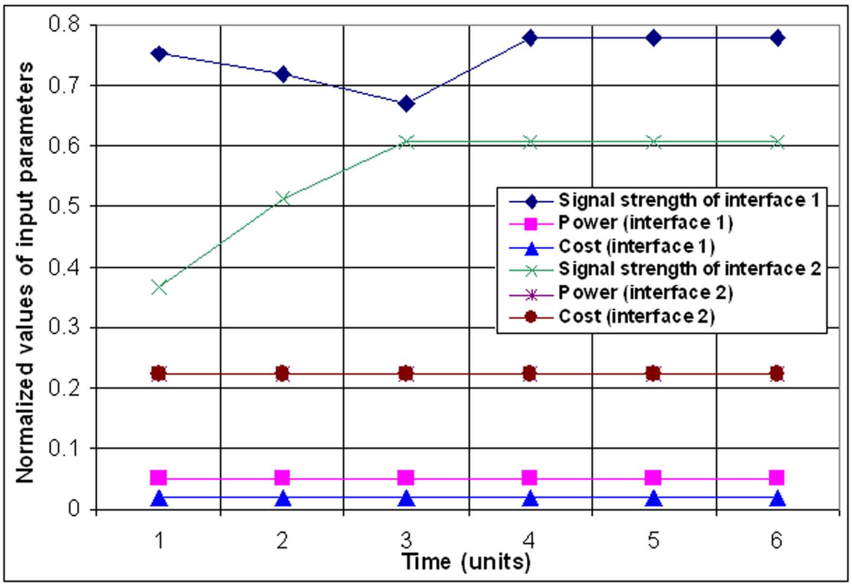


Figure 3.10: Variation of normalized parameter values as functions of time - Scenario 3

the weights of cost and power are set to 0.2 and 0.2 each. Figure 3.10 shows the values of signal strength, cost and power of the two interfaces. Interface 1 is given a signal strength characteristic better than that of interface 2. The cost and power characteristics of interface 2 is better than that of interface 1.

The same input values are also simulated assigning equal weights to all three parameters. w_s, w_p and w_c are set to 0.33 each. SAISA chooses interface 1 at all time instants as can be seen from Figure 3.11. This is because interface 1 possesses better signal strength characteristics. It is also to be noted that the score function of interface 2 at time instant three, is higher than that of interface 1. A switch in the interface is not made because of the threshold condition not having been satisfied. Figure 3.12 shows the choice of interfaces when equal priority is given to the three

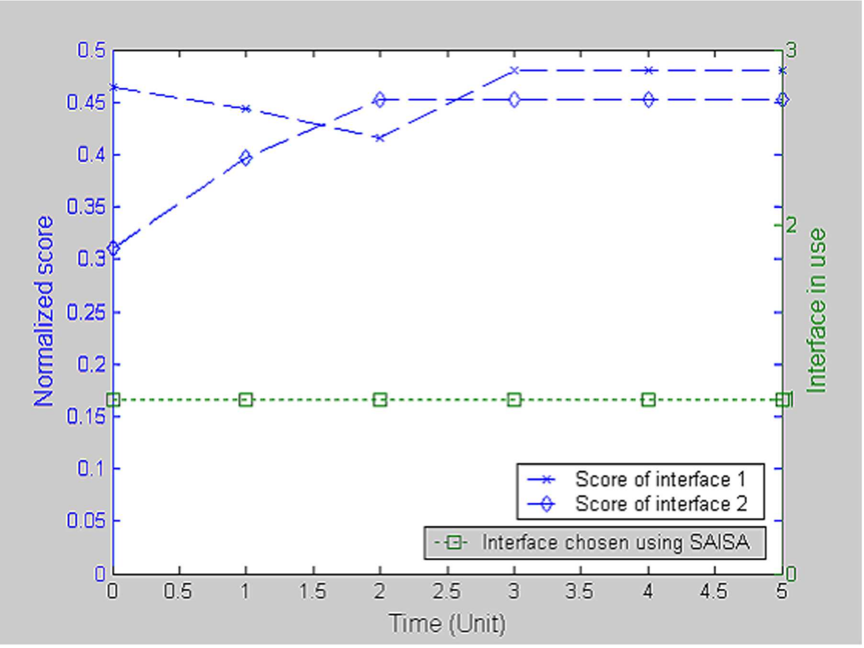


Figure 3.11: Choice of interfaces by SAISA for $w_s = 0.6, w_c = 0.2$ and $w_p = 0.2$

paramters. Interface 1 seems to be the choice only at time instant 1. The choice then shifts to interface 2 until time instant 5.

The major disadvantage in assigning one parameter with a higher priority than the other parameters is the tradeoffs. In this scenario, the client would have to end up paying much more and losing battery power more rapidly than while staying with the choices made by equally prioritizing the parameters. The extent to which a tradeoff is to be made depends on how heavily one of the parameters is being prioritized. The tradeoff information could prove to be useful if the client could be warned with the loss he would have to incur in terms of power and cost. It would help the client in making a decision.

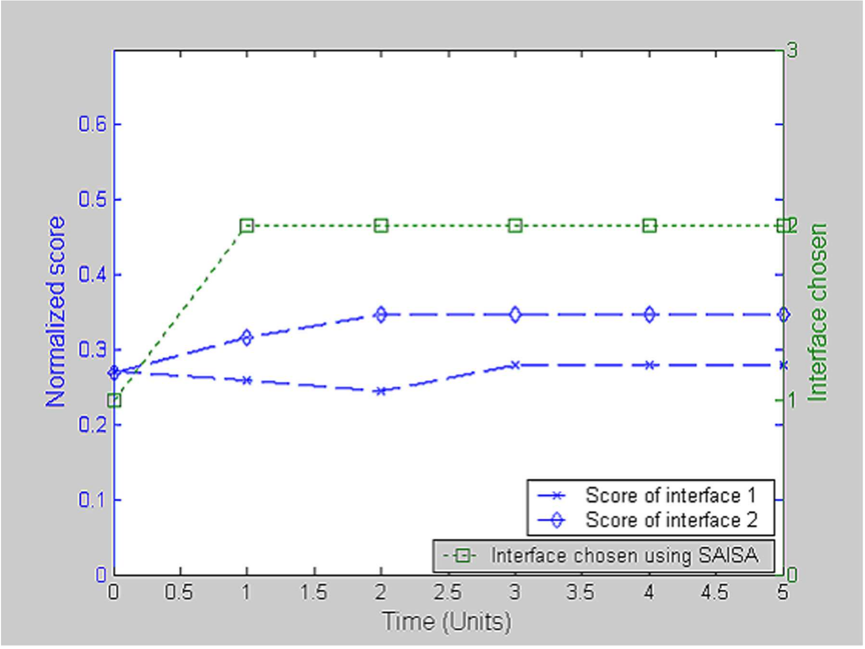


Figure 3.12: Choice of interfaces by SAISA for $w_s = 0.33, w_c = 0.33$ and $w_p = 0.33$

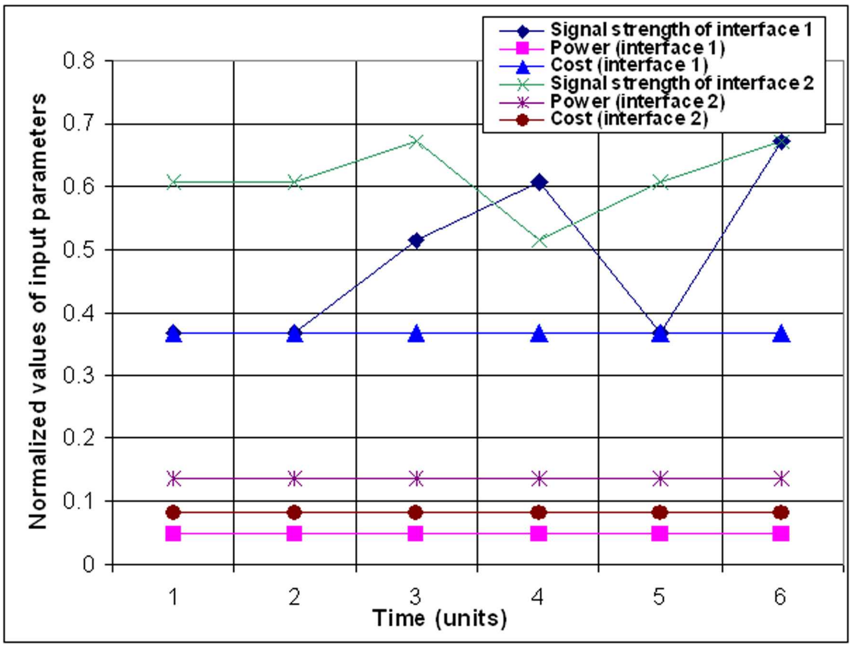


Figure 3.13: Variation of normalized parameter values as functions of time - Scenario 4

3.4.4 Scenario 4: Power priority

A power priority is required when conserving battery power of the client is a major concern. Thus, the power criteria is given high importance in this case $w_p = 0.6$). The other two parameters are given an equal priority ($w_s = w_c = 0.2$) each. Figure 3.13 shows the input values of the two interfaces considered. Interface 1 is seen to have better power conservation characteristics as compared to interface 2 while the former's cost and signal strength characteristics is worse than that of interface 2. Figures 3.14 and 3.15 are the outputs when the values in Figure 3.13 are simulated with the value of weights in the first case being $w_p = 0.6, w_c = 0.2$ and $w_s = 0.2$. The value of weights attached to the parameters in the second case is $w_s = w_c = w_p = 0.33$. It is observed that interface 1 is the constant choice when prioritized, due to its superior power characteristics. The

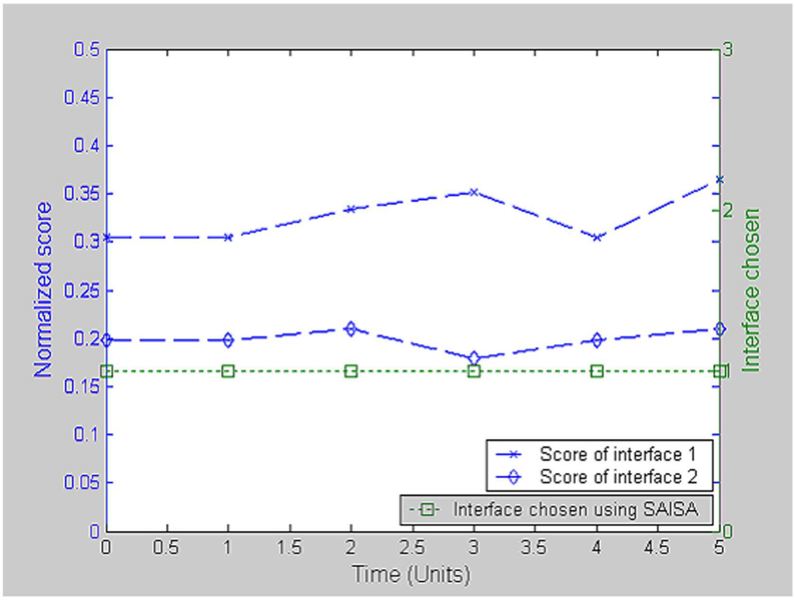


Figure 3.14: Choice of interfaces by SAISA for $w_p = 0.6, w_s = 0.2$ and $w_c = 0.2$

choice of interfaces on assigning equal priorities to the parameters shows the extra cost incurred along with the tradeoff in signal strength made due to the power constraint.

3.4.5 Scenario 5: Cost priority

The last scenario described here is when a high priority is given to the cost of using a particular access network. A weight of 0.6 is given to the cost parameter while signal strength and power share an equal value of 0.2 each. The input values in Figure 3.16 are chosen such that the least expensive interface consumes more power and has a worse signal strength quality than the other interface. Such a choice has been made to display the adverse effects of giving a high priority to cost. This can be seen in Figures 3.17 and 3.18. The output seen in Figure 3.17 is observed when a

Figure 3.15: Choice of interfaces by SAISA for $w_s = 0.33, w_c = 0.33$ and $w_p = 0.33$

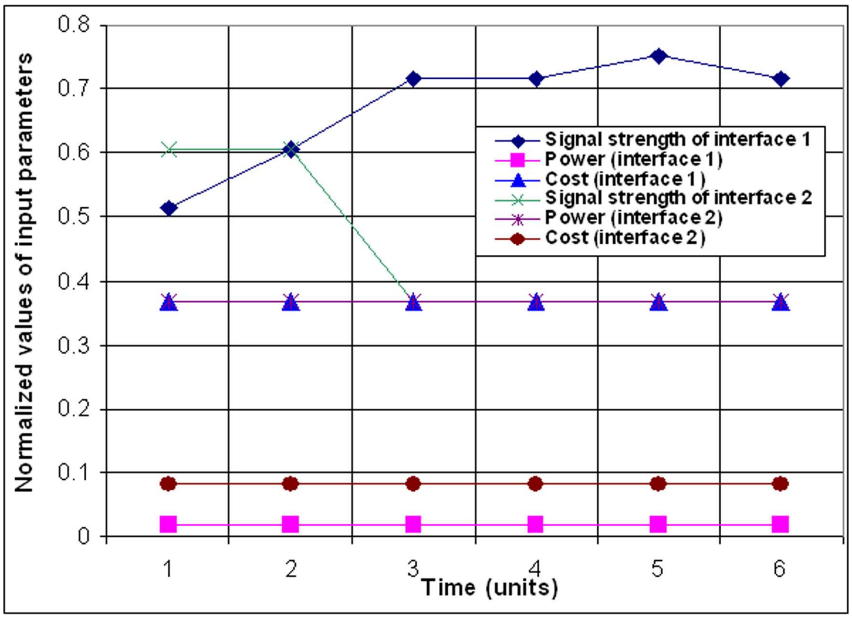


Figure 3.16: Variation of normalized parameter values as functions of time - Scenario 5

higher priority is given to power while Figure 3.18 is the simulated output for the same parameter values, but with equal weights.

3.5 Summary

The various types of handoffs along with different handoff algorithms have been discussed in this chapter. The NAT/IP tunnelling-based handoff algorithm has also been described.

The effectiveness of SAISA in avoiding both unnecessary handoffs and diminishing the ping pong effect has been demonstrated. Further, the ineffectiveness of the algorithm described in [7] is also seen. SAISA has also displayed its flexibility in choosing interfaces. It offers user interactivity when client parameters such as power and cost are considered. On the other hand, the disadvantages

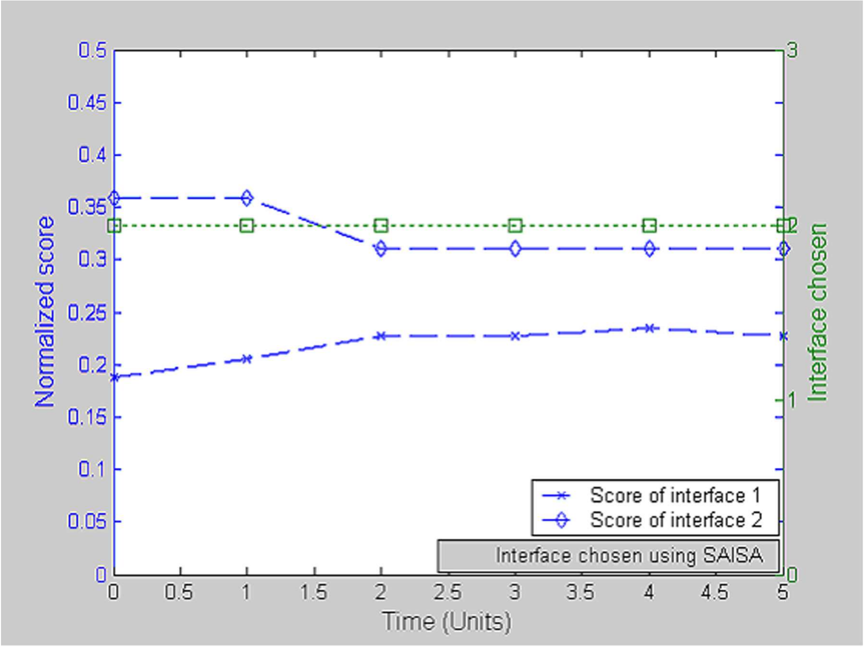


Figure 3.17: Choice of interfaces by SAISA for $w_c = 0.6, w_s = 0.2$ and $w_p = 0.2$

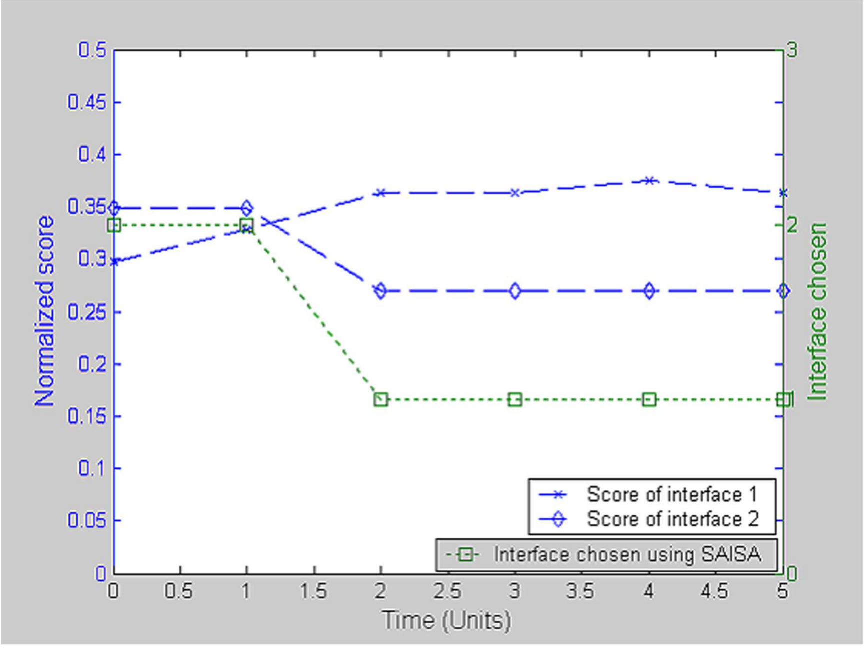


Figure 3.18: Choice of interfaces by SAISA for $w_s = 0.33, w_c = 0.33$ and $w_p = 0.33$

the user could face when very high priorities are set for any of the parameters are also demonstrated. Chapter 4 gives the details of the test bed designed and implemented for the research work in this thesis. It also discusses and summarizes the experiments conducted on the test bed along with their respective results.

CHAPTER 4

SIMULTANEOUS USE OF MULTIPLE INTERFACES

With the emergence of new wireless technologies incorporation of more than one network interface device in mobile devices has become common. As was mentioned earlier, the presence of overlapping network access technologies enables mobile devices to use more than one interface simultaneously. Such usage enhances the utilization of a mobile device in executing multiple applications concurrently .

Wireless technologies differ in their attributes, such as, communication cost, bandwidth, data rates, transmission power etc. Restricting the use to one interface at a time decreases the flexibility of the end-user to avail the resources. Using multiple interfaces simultaneously aids in making use of these resources effectively. The next section discusses some of the advantages of simultaneous use of multiple interfaces [1].

4.1 Advantages of simultaneous use of multiple interfaces

Bandwidth aggregation: Simultaneous use of interfaces permits bandwidth aggregation. The bandwidth offered by all the interfaces present in the mobile device could be aggregated to process high bandwidth demanding applications. It thus helps in providing better quality of service.

Mobility Support: The presence of overlapping network access technologies along with the presence of multiple interfaces in mobile devices aids in reducing handoff delay by keeping another interface device always alive.

Resource sharing: In ad hoc networks, nodes are connected using their local interfaces (LAN - 802.11, Bluetooth etc). Some of these nodes could be equipped with wide area network connections whose bandwidth resources could be shared by all nodes to connect to the internet.

4.2 Literature review for simultaneous use of multiple interfaces

Reference [1] describes an algorithm, Earliest Delivery First (EDF) to ensure proper utilization of the resources of each of the interfaces present in the mobile terminal. This is accomplished by partitioning the traffic and scheduling the packets generated to different interfaces simultaneously. The partitioning of the applications waiting to be processed (traffic) is done based on the estimated delivery time of each packet on the individual interfaces. The interface that requires the least time to deliver a packet to its destination is then used. This paper uses Mobile IP protocol [20] to support mobility and simultaneous use of interfaces.

4.3 Basics of IPtables

To enable multiple interfaces to work simultaneously we have employed the use of IPtables - a module present in the Linux operating system kernel. Multiple interfaces are used simultaneously by forcing certain fields in the IPtables module in a predertermined way. This procedure is explained in section 4.3.4. The next few sections explain the basics of the IPtable's functioning.

4.3.1 TCP/IP

The IPtables work in the TCP/IP layer of the protocol stack. The TCP/IP reference model constitutes five layers [22]. Table 4.1 indicates the different layers and the common protocols used in the stack. As can be seen, the TCP and IP work at the transport and network layers respectively.

Since it is these two layers that are involved with the working of the Iptables, the functions of only these layers are described below:

Internet/Network Layer: The internet/network layer is the linchpin that holds the whole architecture together [22]. The main function of the internet layer is to allow hosts to inject packets into any network and help them reach their destination. The internet layer helps in packet routing.

Table 4.1: TCP/IP Architecture

Layer Number	Layer Name	Example of protocols used
5	Application	– FTP, SMTP
4	Transport	– TCP, UDP
3	Internet/Network	– IP
2	Link	– MAC
1	Physical	– LAN

Transport layer: The layer above the internet layer is called the transport layer. The two protocols used in this layer are TCP and UDP (User datagram protocol) [22]. The main function of this layer is to allow two hosts to carry on a conversation. The TCP is a connection oriented protocol that ensures that the information is not only carried to the destination, but is also delivered in the correct order and accurately. On the other hand, the UDP is a connectionless protocol also known as the best effort protocol that concentrates more on transporting the information to the destination at the right time rather than transporting it accurately.

4.3.2 Iptables

Iptables is a module already present in the Linux kernel. The Iptables functions are used by packet filters to enable simultaneous use of multiple interfaces in this thesis. Packet filters or firewalls, in general, are agents employed to filter packets entering a network to ensure that they are not spurious or unwanted.This module is made of three individual tables namely:

(1) Network Address Translation (NAT) table

(2) Mangle table and

(3) Filter table.

These tables are basically made up of chains. The chains are made of rules that are either predefined in the system or defined by the user. The rules are used to decide the fate of every packet that traverses the rules. The Filter table and Mangle table are discussed here while NAT will be discussed in detail in the next chapter.

The Filter table is the simplest in terms of its operations. It is used to "drop" or "accept" packets by taking a look at its contents and matching them with the rules specified in the table.

We have used the Mangle table to enable the use of multiple interfaces simultaneously. Its function is to manipulate the Type of Service (TOS) and Time to Live (TTL) fields in the IP header of the packets.

The TOS field is used to set and/or change the Type of Service field in the IP header while the TTL field defines the amount of time or the number of hops through which the packet would stay alive in the network before reaching its destination. Apart from these two fields present in the IP header, a third field called MARK can also be manipulated in the Mangle table. It is a special field present only in the Linux kernel and is used for purposes like traffic shaping and filtering. The Mangle table consists of five chains: prerouting, postrouting, output, input and forward.

The set of rules defined in a chain are examined before a packet is allowed to traverse the chain. Every chain has a specific purpose and the chain is applied to groups of packets depending on the type of packets. For example, an input packet will have to conform to the rules in the input chain. Just as an input chain carries rules for input packets, the output chain is used to modify packets

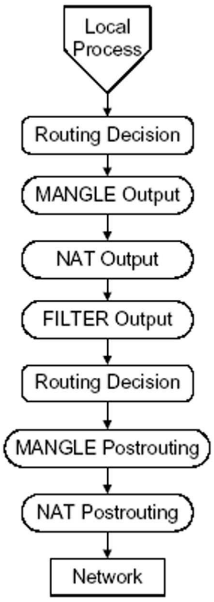


Figure 4.1: Route traversed by an outgoing packet from the device to a network via a chosen interface

generated locally. This chain enables simultaneous use of multiple interfaces. It is to be noted that while assigning interfaces to applications, only outgoing packets can be controlled. If incoming packets also need to be controlled and assigned interfaces, a server is required to control them. This will be discussed in detail in the next chapter. Figure 4.1 is a flowchart representation of the route that a locally generated packet takes before it leaves the client via an interface. All outgoing packets from a device pass through the output chains of the Mangle, NAT and Filter tables before the routing decision is made. The rules defined in these chains aid the client in choosing the interface for a particular application. Once the decision of which interface is to be used for the application is made, the packets then traverse the postrouting chains of the Mangle and the NAT tables to make any changes to the packet headers if required. The packets from the application are then dispatched using that interface. The packet is finally sent out to the network en route its destination.

4.3.3 Type of Service (TOS)

The field used for static scheduling of the packets is TOS. This field was developed for the purpose of distinguishing between different types of packets by the routers. This helps in providing varying services depending on the TOS value of the packet. Current routers though ignore this field [22]. The field has undergone some small changes to accommodate differentiated services [24]. The TOS field consists of 8 bits, the value may be 0-255, or in hex 0x00-0xFF [23]. The default value on most packets is zero. Table 4.2 shows some of the TOS values in decimal and hexadecimal. The significance of the field remains the same i.e. ignored by network routers. Thus, the SAISA algorithm exploits this unused characteristics of the TOS field within the client and before the application packets enter the network. In the algorithm described in this thesis, SAISA, the TOS field is used only by and within the clients and with outbound packets and hence does not pose a threat of being ignored by the routers.

Table 4.2: Type of Service Values

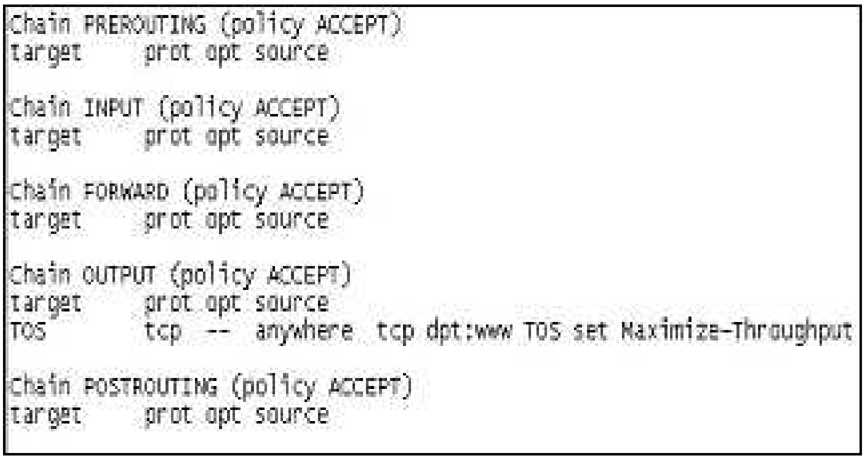
TOS (Dec)	TOS (Hex)	Named Values
16	0x10	– Maximize-Delay
8	0x08	– Maximize-Throughput
4	0x04	– Maximize-Reliability
2	0x02	– Maximize-Cost
0	0x00	– Normal Service

4.3.4 Simultaneous enabling of multiple interfaces

On the initiation of an application, generally the transport protocol uses the TOS value that is assigned by the application [23]. This value chosen by the application can be changed using the functions in the IPtables module.

The "–set-tos" function is a part of the IPtables module. This function is used to assign a TOS value to a packet, provided the rest of the attributes defined in the function match with those of the packet. An "ip route add" command is then used to specify the interface that the packets with the assigned TOS value are required to use. The interface specified in the ip route add command is set as the default for these packets. The IP address of the interface is specified in the route add command and a subsequent entry is made into the IP routing table. This enables all marked packets to exit through the interface as specified in the routing table.

For example, consider two applications, HTTP and FTP. To channelize the packets of the two application types through different interfaces, say eth0 and eth1 respectively the following steps need to be taken:



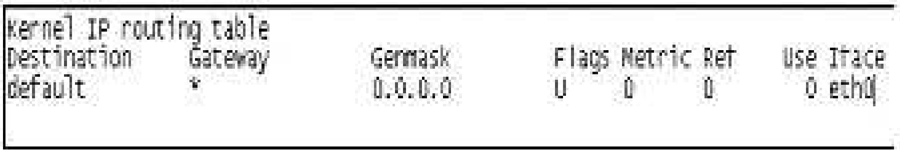
```
Chain PREROUTING (policy ACCEPT)
target     prot opt source

Chain INPUT (policy ACCEPT)
target     prot opt source

Chain FORWARD (policy ACCEPT)
target     prot opt source

Chain OUTPUT (policy ACCEPT)
target     prot opt source
TOS        tcp  --  anywhere  tcp dpt:www TOS set Maximize-Throughput

Chain POSTROUTING (policy ACCEPT)
target     prot opt source
```



```
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
default         *               0.0.0.0         U     0      0        0 eth0
```

Figure 4.3: Screenshot of the routing table

(1) Mark all HTTP packets with any TOS value, in this case 0x08 is chosen (named value: Maximize-Throughput as in Table 4.2) at the output chain of the Mangle table. Figure 4.2 is a screenshot of the Mangle table showing the tos entry that has been made.

(2) Now an entry in the ip routing table is made such that all packets marked with a TOS value of 0x08 are use interface eth0 whose IP address could be anything, in this case say xxx.xxx.xxx.xxx as the default. This can be seen in the screenshot in Figure 4.3

The screenshots were generated by viewing the Mangle and ip route tables after making the entries as described in steps 1 and 2 above. The same procedure is followed for any application layer protocol as explained above. FTP (File Transfer Protocol) packets can also be marked with a TOS value, but a value different from that of the HTTP packets since it is required for them to use different interfaces to display simultaneous use of multiple interfaces. Thus, a value of 0x02 can be assigned at the output chain of the Mangle table to the FTP packets. An entry at the routing table can be made to add an interface eth1 as the default interface to be used by all packets marked with a TOS value of 0x02. If an FTP session is started, one would notice that all the FTP packets use the interface eth1. If an HTTP application is started simultaneously it can be observed that the http packets use interface eth0 while the FTP packets at the same time are using eth1. Thus simultaneous use of two interfaces; eth0 and eth1, has been accomplished for two different applications HTTP and FTP respectively.

A major disadvantage of this method is that only outgoing packets can be controlled. The incoming packets would automatically choose the default device mentioned in the routing table. For the outgoing packets, a TOS value is chosen by the application which will be used by the transport layer protocols. It can then be manipulated to a different value using the –set-tos command if required. For the incoming packets on the other hand, the TOS value would be unknown (they need not necessarily use the same TOS value as the outgoing packets) and therefore cannot be manipulated by the client. One of the solutions to this problem is to use the NAT (Network Address Translation) server seen in Figure 3.3 to control the incoming packets.

Another drawback is that, at the time of writing, the TOS field was designed to take any of the five values as shown in Table 4.2. The number of interfaces that can be operated simultaneously based on the TOS value is thus limited to five. On the contrary, this value is quite practical for today since very few devices are equipped with more than five interfaces.

4.4 Summary

This chapter has given a detailed explanation on the advantages of simultaneous use of multiple interfaces along with discussing the how-to of employing simultaneous use of interfaces.

CHAPTER 5

EXPERIMENTS PERFORMED ON A TEST BED

Experiments were conducted using a test bed to verify some of the ideas generated in this thesis. This chapter summarizes the experimentally observed results.

The test bed network consisted of an IBM ThinkPad R52 (the client) equipped with three interfaces. The built in Ethernet and Bluetooth interfaces were used for understanding the behavior of simultaneous activation of multiple interfaces. All three interfaces namely, Bluetooth, Ethernet and WLAN were used for studying the behavior of static interface selection. The Bluetooth and Ethernet interfaces were both built in to the laptop. For WLAN, a network card was used. The technical specifications of all the equipment used are as follows: Client: IBM ThinkPad R52, 1.6GHz Intel Pentium M processor Interfaces: Bluetooth and Ethernet were built in to the laptop. WLAN network card: Linksys Access point: AXIS 9010 for Bluetooth Router: Microsoft wireless router, MN-700 The test bed is shown in Figure 5.1. All three interfaces in this test bed are treated as sources of connection to the internet.

5.1 Interfaces used

5.1.1 Bluetooth (IEEE 802.15)

The project Bluetooth was named after Harald Blataand II who had unified Denmark and Norway [22]. It is a cable replacement technology. Its advantages are a convenient and untethered means to interconnect a variety of electronic gadgets. It does not require direct line of sight like Infrared technology. Bluetooth operates in the 2.45GHz ISM band with a data rate of up to 721kbps.

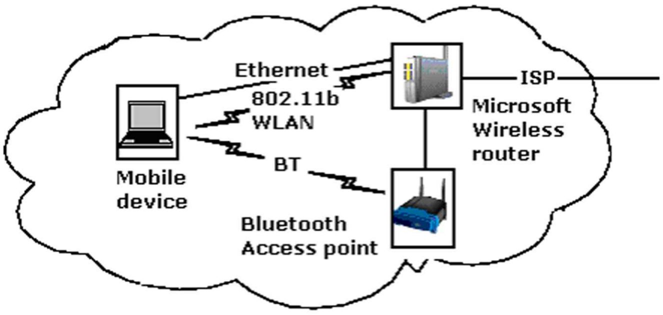


Figure 5.1: Test bed setup

It operates within a range of 10-100 meters, with the range depending on power levels of the participating Bluetooth gadgets. The basic unit of a Bluetooth unit is the piconet which is a network of not more than eight Bluetooth gadgets. The eight nodes comprise of one master node and seven active slaves. Interconnected piconets are called scatternets. There can be up to 255 parked nodes in the piconet apart from the seven active nodes. A node is said to be in a parked state when it is operating under reduced power levels to prevent battery draining. The nodes in parked state can only respond to an activation or beacon signal from the master.

5.1.2 WLAN (IEEE 802.11)

The wireless local area network consists of mobile users who transmit and receive packets to and from an access point within the radius of a few tens or hundreds of meters [22]. The access point is a base station that is connected to the wired network. It behaves as the gateway between the wired and wireless networks. WLANs come in several standards. The most common among them being 802.11a, b and g. 802.11a uses Orthogonal Frequency Division Multiplexing to deliver up to 54Mbps in the wider 5GHz ISM band. 802.11b uses High Rate Direct Sequence Spread Spectrum

(HR-DSSS) and supports data rates of 1, 2, 5.5 and 11Mbps in the 2.4GHz frequency range. 802.11g is the enhanced version on 802.11b technology. It works in the same band as 802.11b but provides a theoretical data rate of 54Mbps. It uses the same modulation technique as 802.11a.

5.1.3 Ethernet (IEEE 802.3)

It was invented in the mid-1970s by Bob Metcalfe and David Boggs. In spite of having had considerable competition from other Local Area Network (LAN) technologies like Token Ring, Asynchronous Transfer Mode (ATM) and Fiber Distributed Data Interface (FDDI), Ethernet has strived to evolve as a dominant LAN technology. Some of the reasons for its dominance in the market are its inexpensive hardware requirements along with its simple, efficient and decentralized access control technology. Some of the features of Ethernet are that it uses the CSMA/CD (Carrier Sense Multiple Access/Collision Detection) protocol at the MAC (Medium access control) layer. It yields a variety of data rates of 10Mbps, 100Mbps, 1Gbps and 10Gbps [22]. The communication channels that could be used are coaxial cable, twisted pair copper wires or fiber optics in bus or star topologies.

5.2 Experiments conducted

The experiments conducted on the test bed were to ensure that the simultaneous use of interfaces and the SAISA gave the results that were claimed. Two experiments were conducted: one, test the simultaneous use of multiple interfaces and two, observe SAISA operation.

5.2.1 Experiment 1 - simultaneous use of interfaces

The experiment conducted for simultaneous use of interfaces was as follows. The application used for the experiment was HTTP. At first the outbound HTTP packets going to two different destination IP addresses (destination IP addresses corresponding to two World Wide Web addresses or URLs(Uniform Resource Locator)) were marked with two different TOS values as described in section 4.3. Two routing table entries were made to allow the packets with the varying TOS values to take two different network interfaces. The TCP dump data was then recorded while the two browsers were enabled and the destination URLs were accessed. It was seen that the outbound packets to the two destinations used the network interfaces as mentioned in the routing table while the return packets used the default network interface that was mentioned in the routing table. This was because a NAT server was not installed. Hence, the inbound return packets were not being controlled to use the same path as its corresponding outbound packets.

5.2.2 Experiment 2 - execution of SAISA

The second experiment implemented SAISA. A file was maintained to list the preferred network interfaces for every type of application. The parameter values, such as signal strength, cost and power of all the network interfaces, were stored in text files and accessed by the algorithm periodically at every t interval of time. The preferred interfaces for the HTTP application were specified to be Ethernet,WLAN and BT. When the HTTP session was initiated, the algorithm retrieved the configuration file to read the preferred list. It then checked to see if the interfaces were available. On observing the availability of both the interfaces, the configuration files of the interfaces were retrieved to read the parameter values. The score function was then calculated for each one of these interfaces. The interface with the highest score function was then chosen and entered into the routing table as the default interface for all outbound packets belonging to the HTTP application. The outgoing packets were marked with a TOS value before final transmission. This experiment was then repeated for different parameter values emulating different scenarios as was discussed in section 3.4. It was observed that both the experimental and simulation results tallied. The above two experiments have thus tested the simultaneous use of interfaces and the interface selection procedure (Static). The experimental results for the latter also tallied with the simulation results.

5.3 Summary

The experiments that were performed on the test bed setup were described in this chapter.

Chapter 6

Conclusion

The main contributions of this thesis are the use if multiple network interfaces simultaneously and optimal selection of an interface for communicating data pertaining to an application. A Linux kernel based method has been exploited to facilitate the best interface selection solution. A mathematical framework is developed, for understanding and manipulating various parameters such as signal strength, to enhance the utilization of the mobile device while providing a higher quality of service to the applications. The interface selection process along with the score function calculation has been simulated to compare and contrast the qualities of SAISA with previously conceived algorithms. The advantages of prioritizing parameters based on client and network conditions have also been discussed. A NAT-based handoff procedure has been proposed to enable dynamic scheduling of packets supporting both horizontal and vertical handoffs seamlessly. This method does not require any modification to the existing infrastructure. In addition to the existing infrastructure, it requires the installation of NAT servers.

6.1 Future work

SAISA has been experimentally tested with three interfaces. The number of interfaces used could be scaled up to test the scalability of the algorithm.

Only three parameters have been considered in this paper. The system can be modified for larger number of parameters to observe the change in handoff performance.

The NAT-based handoff algorithm needs to be experimentally tested along with the rest of the model.

The simultaneous use of multiple interfaces along with dynamic scheduling of packets opens the area of parallel processing to be explored in future research.

BIBLIOGRAPHY

[1] Nicolas Montavont, Ryuji Wakikawa, Thierry Ernst and Thomas Noel, “Problem Statement for Multihomed Mobile Nodes,” IETF Mobile IP Working Group Internet Draft, Oct., 2003

[2] Kameswari Chebrolu and Ramesh Rao, “Bandwidth Aggregation for Real-Time Applications in Heterogeneous Networks,” IEEE Transactions on Mobile Computing, vol. 5, no. 4, pp. 388-403, Apr., 2006.

[3] Oskar Andreasson, “Iptables Tutorial 1.2.0,” http://iptables-tutorial.frozentux.net/iptables-tutorial.html

[4] Vikrant Kaulgud and Sakib Mondal, “Exploiting Multihoming for Low Latency Handoff in Heterogeneous Networks.” ConTEL 2005, Proceedings of the 8th International Conference, Vol. 1, pp. 49-55, Jun., 2005.

[5] Jukka Ylitalo, Tony Jokikyyny, Tero Kauppinen, Antti J. Tuominen, Jaakko Laine, “Dynamic Network Interface Selection in Multihomed Mobile Hosts,” System Sciences, Proceedings of the 36th Annual Hawaii International Conference, pp. 315, Jan., 2003.

[6] Francoise Andre, Jean-Marie Bonnin, Bruno Deniaud, Karine Guillouard, Nicolas Montavont, Thomas Noel and Lucian Suciu, “Optimized Support of Multiple Wireless Interfaces within an IPv6 End-Terminal,” Proceedings of the Smart Objects Conference (SOC 2003), May, 2003.

[7] Ling-Jyh Chen, Tony Sun, Benny Chen, Venkatesh Rajendran and Mario Gerla, “A Smart Decision Model for Vertical Handoff,” ANWIRE International Workshop, 2004.

[8] Helen Wang, Randy Katz, and Jochen Giese, “Policy-Enabled Handoffs across Heterogeneous Wireless Networks,” Proceedings of Second IEEE Workshop on Mobile Computing Systems and Applications (WMCSA ’99), pp. 51-60, Feb., 1999.

[9] Mark Stemm and Randy Katz, “Vertical Handoffs in Wireless Overlay Networks,” Proceedings of ACM Mobile Networking (MONET), Special Issue on Mobile Networking in the Internet, Vol, 3 , Issue 4, pp. 335-350, Summer, 1998.

[10] Gregory Pollini, “Trends in Handover Design,” Proceedings of IEEE Communications Magazine, Vol. 34, Issue 3, pp. 82-90, Mar., 1996.

[11] Kjeld Borch Egevang and Paul Francis, “The IP Network Address Translator,” RFC 1631, May, 1994.

[12] Ahmed Helmy and Muhammed Jaseemuddin, “Efficient Micro-Mobility using Intra-domain Multicast-based Mechanisms,” ACM SIGCOMM Computer Communication Review, Vol. 32 , Issue 5, pp. 61-72, Nov., 2002.

[13] Adina Burian, “Media Independent Handover 802.21,” http://www.ieee802.org/21/

[14] Frank A. Zdarsky and Jens B. Schmitt, “Handover in Mobile Communication Networks: Who is in Control Anyway?,” In Proceedings of the 30th IEEE EUROMICRO Conference, pp. 205212, Sep., 2004.

[15] Srikant Sharma, Inho Baek, Yuvrajsinh Dodia and Tzi-cker Chiueh, “Omnicon: A Mobile IP-based Vertical Handoff System for Wireless LAN and GPRS Links,” Proceedings of the 33rd International Conference on Parallel Processing Workshops (ICPP 2004 Workshops), pp. 15-18, Aug., 2004.

[16] Eric van den Berg, Sunil Madhani, Tao Zhang, “Real-Time Comparison of Quality of Interfaces (QoI) by Mobile Devices over Heterogeneous Radio Networks,” First International Conference on Quality of Service in Heterogeneous Wired/Wireless Networks (QSHINE’04), pp. 206-215, Oct., 2004.

[17] Trevor Pering, Vijay Raghunathan, and Roy Want, “Exploiting radio hierarchies for power-efficient wireless device discovery and connection setup,” Proceedings of the 18th International Conference on VLSI Design held jointly with 4th International Conference on Embedded Systems Design (VLSID’05), pp. 774-779, Jan., 2005.

[18] David Famolari, Praveen Gopalakrishnan and Kamesh Medepalli, “A method for simultaneous transmission across multiple interfaces: Send on next available interface,” Private Communication.

[19] www.wikipedia.com

[20] Charles Perkins, “IP mobility support for IPv4,” RFC 3220, Jan., 2002.

[21] Robert Woodburn “IP tunneling,” RFC 1241, Jul., 1991.

[22] Andrew Tanenbaum, “Computer Networks,” Publisher: Prentice Hall, Fourth edition, Aug., 2002.

[23] P. Almquist, “Type of Service in the Internet Protocol Suite,” RFC 1349, Jul., 1992.

[24] S. Blake, David Black, M. Carlson, E. Davies, Z. Wang, W. Weiss, “An Architecture for Differentiated Services,” RFC 2475, Dec., 1998.

[25] Qing-An Zeng and Dharma Agrawal, “Handbook of Wireless Networks and Mobile Computing,” Publishers: John Wiley and Sons, Feb., 2002.

[26] http://www.webopedia.com/TERM/V/virtual-IP-address.html

[27] Qian Zhang, Chuanxiong Guo, Zihua Guo and Wenwu Zhu "Efficient mobility management for vertical handoff between WWAN and WLAN," IEEE Communications Magazine, vol. 41, no. 11, pp. 102-108, Nov., 2003.

[28] ChingYao Huang, Ming Yuan Tsai and Joe Huang "Multi-slope quality based handoff controls in third generation CDMA wireless systems," Oct., 2003.

[29] M. Buddhikot, G. Chandranmenon, S. Han, Y.W. Lee, S. Miller, L. Salgarelli, "Integration of 802.11 and Third generation wireless data networks", In Proceedings of the IEEE INFOCOM, Apr., 2003.

[30] Stefan Aust, Daniel Proetel, Nikolaus Fikouras, Carmelita Gorg, "Policy based mobile IP handoff decision (POLIMAND) using generic link layer information", The fifth IEEE International Conference on Mobile and Wireless Communication Networks (MWCN 2003), Oct., 2003.

[31] Eugene Shih, Paramvir Bahl, Michael Sinclair, "Wake on wireless: An energy driven energy saving strategy for battery operated devices," In Proceedings of the 8th Annual International Conference on Mobile Computing and Networking (MOBICOM '02), Jul., 2002.

[32] Hyosoon Park, Sunghoon Yoon, Taehyoun Kim , Jungshin Park, Misun Do, and Jaiyong Lee, "Vertical handoff procedure and algorithm between IEEE802.11 WLAN and CDMA cellular network," Mobile Communications, 7th CDMA International Conference, 2002.

[33] Mohammed Atiquzzaman, Shaojian Fu, William Ivancic, "TraSH-SN: A transport layer seamleass handoff scheme for space networks," NASA Earth Science Technology Conference (ESTC), Jun., 2004.

[34] Jean Tourrilhes and Venky Krishnan, "Co-Link configuration : Using wireless diversity for more than just connectivity," IEEE Wireless Communications and Networking Conference (WCNC 2003), pp. 1090-1095, Mar., 2003

[35] Hakima Chaouchi and Munaretto fonseca Anelise "Adaptive QoS Management for IEEE 802.11 future Wireless ISPs", ACM Wireless Networks, WiNET, Special issue on Ad Hoc networking, Vol. 10, no. 4, Jul., 2004

[36] Simon Xin Cheng, The intra-domain WLAN handoff for real-time applications and its implementation suggestion, School of computing science, Simon Fraser University, unpublished, 2004.

[37] Tao Zhang, Byungsuk Kim, Prathima Agrawal, "Methods for supporting seamless multi-interface IP mobility," Private communication.

[38] Christer hlund and Arkady Zaslavsky, "Multihoming in Mobile IP," Proceedings of 6th IEEE International Conference on High Speed Networks and Multimedia Communications (HSNMC'03), Jul., 2003.

[39] Qi Wang and Mosa Ali Abu-Rgheff, "A Multi-layer mobility management architecture using cross-layer signalling interactions," Mobile communications reseach group, School of Computing, communications and Electronics, The University of Plymouth.

[40] Kapil Sood, Emily Qi and Vivek gupta, "Seamless platform mobility across wireless networks", http://www.intel.com/technology/magazine/communications/mobility-on-wireless-0905.htm.

[41] 3rd Generation Partnership Project, http://www.3gpp.org/

[42] 3rd Generation Partnership Project 2, http://www.3gpp2.org/