Pro-collaborative mobile systems in next generation IP networks

Author:
Hsieh, Robert Chia-Hung

Publication Date:
2004

DOI:
https://doi.org/10.26190/unsworks/4401

Pro-collaborative Mobile Systems in Next Generation IP Networks

Robert Chia-Hung Hsieh

A thesis submitted in fulfilment

of the requirements for the degree of

Doctor of Philosophy

School of Electrical Engineering and Telecommunications

The University of New South Wales

August 2004

CERTIFICATE OF ORIGINALITY

I hereby declare that this submission is my own work and to the best of my knowledge it contains no materials previously published or written by another person, nor material which to a substantial extent has been accepted for the award of any other degree or diploma at UNSW or any other educational institution, except where due acknowledgement is made in this thesis. Any contribution made to the research by others, with whom I have worked at UNSW or elsewhere, is explicitly acknowledged in the thesis.

I also declare that the intellectual content of this thesis is the product of my own work, except to the extent that assistance from others in the project's design and conception or in style, presentation and linguistic expression is acknowledged.

Preface

This thesis describes the research work accomplished in fulfilment of the requirements for the degree of Doctor of Philosophy at The University of New South Wales, Sydney, Australia. The research was undertaken between the period from March 2001 to November 2003 in the School of Electrical Engineering and Telecommunications.

A fool doth think he is wise, but the wise man knows himself to be a fool.

- William Shakespeare (1564 – 1616), "*As You Like It*", Act 5 Scene 1.

Acknowledgement

PhD was an unforgettable and in many ways a surreal experience. It was inspirational, captivating, sorrowful, joyous and nonetheless extremely eventful. It not only changed the way I have come to understand the things in this world, but also ultimately changed the way I am and what I have become – a thinker and a reasoner. Naturally, I would not have made it in one piece without the wonderful support of my family, friends and mentors.

Here, I would like to, first of all, thank my supervisor, my mentor, Professor Aruna Seneviratne – a guiding light in the wild storm. It never ceases to amaze me how a half hour meeting with him would always follow by/lead to months of hard work. His guidance was extraordinary, direct and sharp to the point. The five months internship at Sprint Advanced Laboratories USA was really an eye opening experience. Thanks Aruna!

Secondly, I would like to thank my fellow colleagues at the MOBQOS group UNSW, Binh Thai, Tim Hu, Sebastien Ardon, Jonathan Chan, Zhe-Guang Zhou, Stephen Wan, Marius Portmann, Woraphon Lilakiatsakun, Pipat Sookavatana, Prawit Chumchu, Apichan Kanjanavapastit, Krit Wongrujira, Sanchai Rattananon, Eranga Perera and Thierry Rakotoarivelo. For the numerous late dinners at the MOBQOS kitchen, golf games and driving ranges, BBQs at Coogee and Aruna's, and most importantly the countless inspirational coffee times. I thank you all!!

My special thanks should go to my parents and sister for their continuous and unconditional support over these past three years. My mate, Liaho, I thank you for the times we spend talking about the philosophy of life. It is always refreshing when things are placed into the 'right' perspective. I would also like to acknowledge CSIRO TIP for funding my research.

Last, but not the least, I would like to thank my fiancée, Alyn, for you make this extraordinary journey complete.

Abstract

Computing system designs of today take on either the interactive or the proactive form. Motivated by the user's desire to make his/her computing experience more intelligent and personalised, the progression from interactive (human-centred) to proactive (human-supervised) is evident. It can be observed that current research mainly emphasises the user as the dominant focus of a user-system interaction. Consider a model that we called the *opponent-process* model. It contains two processes, one representing the user and the other the system, where both processes are capable of *dominating* each other, though working collaboratively towards a predefined task. We argue the necessity to design computing systems which are balanced in this model, such that the system process, at times, becomes the dominant process. We refer to this as the *pro-collaborative* design form.

We dissect mobility into the notion of a *nomadic user* and the notion of a *nomadic system*. The examination into the nomadic user problem space reveals the potential for applying the pro-collaborative approach in optimising handoff management. Significant performance advantages can be obtained with our proposed S-MIP framework, based on the pro-collaborative design, when compared with established handoff latency optimisation schemes. The key differentiator lies in its *indicative* approach in addressing handoff ambiguity. Instead of passively anticipating through prediction as to when a mobile user might cross network boundaries (user-dominant), the system actively indicates to the user *when*, *where* and *how* to handoff (system-dominant). This eliminates the handoff ambiguity.

Regarding the notion of a nomadic system, that is, the ability to move services offered by computing systems to arbitrary points in the Internet, we explore the idea of the dynamic extension of network services to a mobile user on-demand. Based on the pro-collaborative form, we develop the *NET*AMORPHOSE architecture which facilitates such a dynamic service extension. By assuming the proliferation of programmable network switches and computational resources *within* the Internet, we re-examine how 'loose' service agreements between network services providers can be, to achieve such *borderless moving-service* offerings.

The viability of the pro-collaborative form is reflected through our design and implementation of protocols and architectures which address the notion of nomadic user and nomadic system.

Table of Contents

List of Figures

List of Tables

List of Publications

Conference Publications

Hsieh, R., and Seneviratne, A. A Comparison of Mechanisms for Improving Mobile IP Handoff Latency for End-to-End TCP. In *Proceedings of MobiCom*, San Diego, USA, September 2003.

Hsieh, R., Zhou, Z.-G., and Seneviratne, A. S-MIP: A Seamless Handoff Architecture for Mobile IP. In *Proceedings of Infocom*, San Francisco, USA, April 2003.

Hsieh, R., and Seneviratne, A. Dynamic Service Extensibility through Programmable Network in a Mobility Context. In *Proceedings of Pacific-Rim Conference on Multimedia (PCM)*, Hsinchu, Taiwan, December 2002.

Hsieh, R., and Seneviratne, A. Soliman, H., El-Malki, K. Performance Analysis on Hierarchical Mobile IPv6 with Fast-handoff. In *Proceedings of GLOBECOM*, Taipei, Taiwan, November 2002.

Hsieh, R., and Seneviratne, A. Dynamic Service Extensibility through Programmable Network. In *Proceedings of International Conference on Computer Communication (ICCC)*, Mumbai, India, August, 2002.

Hsieh, R., and Seneviratne, A. Performance Analysis of Mobile IP and SLM. In *Proceedings of International Conference on Networks (ICON)*, Bangkok, Thailand, October 2001.

Internet Engineering Task Force drafts

Perera, E., **Hsieh, R.**, and Seneviratne, A. Extended Network Mobility Support. Internet Draft, *Internet Engineering Task Force (IETF)*, draft-perera-nemo-extended-00.txt, July 2003. Work in Progress.

Glossary

AA	Active Application
APR	Application Repository
AS	Autonomous System
CDMA	Code Division Multiple Access
CH	Correspondent Host
CoA	Co-located care-of Address
CP	Control Point
CPG	Control Point Gateway
DEG	Dynamic Extension Gateway
DMZ	Demilitarized Zone
EE	Execution Environment
EEG	Execution Environment Gateway
EEI	Execution Environment Interpreter
EM	Execution Management
FA	Foreign Agent
GPRS	General Packet Radio Service
GPS	Global Position System
GSM	Global System for Mobile communications
HA	Home Agent
HMIPv6	Hierarchical Mobile IP version 6
IETF	Internet Engineering Task Force
IP	Internet Protocol
ISO	International Standard Organisation
ISP	Internet Service Provider
JFWD	Java Forwarding API
JVM	Java Virtual Machine
LAN	Local Area Network
MAP	Mobility Anchor Point
MBAC	Measurement Based Admission Control
MH	Mobile Host

MIP	Mobile IP (version 4)
MIPv6	Mobile IP version 6
MOD	Minimum Overlapping Distance
MRP	Mobility Routing Point
NAR	New Access Router
NS	Network Simulator v2
ORE	Oplet Runtime Environment
OSI	Open System Interconnect
PA	Personal Agent
PAR	Previous Access Router
QoS	Quality-of-Service
SB	Service Bidder
SEM	Service Extension Module
SNMP	Simple Network Management Protocol
SON	Service Overlay Network
SS	Signal Strength
TAP	Tracking Anchor Point
UMTS	Universal Mobile Telecommunications System
WAN	Wide Area Network
WiFi	Wireless Fidelity
WLAN	Wireless Local Area Network

Form is not whatever we could find it. Rather, it existed in a primal realm as a general conception and had neither shape nor dimension. Once seized, form could be particularized through the use of function.

- *Louis Kahn*

The thing always happens that you really believe in; and the belief in a thing makes it happen.

- *Frank Lloyd Wright*

Chapter 1

Introduction

Design is an amalgamation of *form* and *function*. Computing system design is without exception.

Up till the early 1990s, the dominant *form* in computing systems had been *interactive*. Influenced by Licklider's vision on human-computer symbiosis [37] and Weiser's momentous ubiquitous computing ideology [67], the interactive form is human-centred and has been primarily focused on office task automation [59]. From the early to mid 1990s, another form of computing was conceptualised. Described as a paradigm shift, the emerging *proactive* form breaks away from the traditional emphasis on human-centred interaction. Rather, the emphasis is on human-supervised system interaction. Using the interactive form as its basis, the proactive approach creates a one-to-many (user-to-system[1]) interaction relationship, instead of a one-to-one or many-to-one relationship that is typified by the interactive form. The

[1] In this thesis, *computing system* refers to frameworks/architectures which consist of both the users and the computer machineries, while *system* only refers to the computer machineries.

key rationale behind this is that the future computing environment will involve thousands of networked computers per person. The proactive form argues that the degree of human involvement required needs to be reduced, thus the necessity to move away from a one-to-one interactivity and elevates interaction into a supervisory one. In other words, taking the users out of the 'control loop', and placing them 'above' the interaction [59]. *Functions* required to attain the proactive and interactive form are well understood within the research community. For example, IBM's Autonomic Computing [73] suggests the ability of systems to self-monitor, self-heal, self-configure and self-improve (performance), and Intel's Proactive Computing [66] proposes that systems should deal with uncertainty, closing the control loop, and offer personalisation and anticipation.

1.1 Motivation

The concept of 'being proactive' introduces the elements of probability and ambiguity. While it is evident that the potential benefits of a proactive attitude outweigh the penalty incurred, we believe that there exist circumstances where the by-product of being proactive is in fact a hindrance to the overall performance of a computing and/or communication system.

Consider the handoff latency reduction problem in the context of Internet mobility management. Current advance handoff techniques require the *anticipation* of mobile device movements. Therefore, various movement trajectory and location prediction algorithms are devised to support better optimised handoffs. These approaches anticipate the movement of mobile devices based on probabilistic calculations and try to pre-empt likely future access networks, in order to prepare for the arrival of the mobile device (due to handoff) before it gets there. We refer to these approaches as designs which take on a *user-dominant* design emphasis. The general problem is that as far as system is able to anticipate a user/mobile device's movement, the more resource wasteful and cumbersome the system becomes, and as far as the system conserves resources, the less effective the anticipation becomes. Such a seemingly paradoxical relationship suggests that perhaps an alternative or a more suitable *form* is yet to be conceptualised.

Furthermore, consider the often de facto assumption that the design of future computing systems must i) run themselves, ii) adjust to varying circumstances, and ii) anticipate the needs of the users, thereby allowing users to concentrate on what they want to accomplish rather than figuring out how to rig the computing system to get them there [73]. Such a seemingly justifiable and natural assumption has the connotation that the users *know*, a priori, 'the what' and 'the how' in rigging the computer system, and want to avoid being hindered by the operational details. What if the users do not know 'the what' and 'the how' but only know what they *want*? In this case, the users simply want the system to provide an 'advisory' role, not in a 'passive' sense as in simply giving suggestions[2], but 'actively' instructing (step-by-step) them 'what to do' literally. Without advocating purely on artificial intelligence, this requires a specific articulation (or awareness) when designing a computing system to accommodate an interaction style where the computing system is not merely user-dominant (i.e. the system reacting to the actions of the user all the time), but also system-dominant at times.

---

[2] An example of giving advice in a passive sense would be 'suggestions' offered by software wizards which are often employed in modern software applications and operating systems.

User-dominant design emphasis is prevalent in the computing and communication/networking systems of today's Internet. It is easy to understand why this is the case when considering that the guiding principle in Internet architecture is to make the network (system) simple and autonomous, while making the end host (user) complex and rich in functionality [54]. Although there are shifting trends in breaking away from this [12], it is still unclear as to how much 'intelligence' (in the form of software) should be put 'into' the network. It is also unclear that if the network becomes more intelligent, should the user-system interaction still be predominantly user-dominant.

1.2 Challenge

While we enthusiastically support the goals of interactive and proactive computing, we argue the need to explore the possibility of another user-system interaction. For instance, given our previous example on handoff optimisation, an interaction where instead of passively (non-dominant) anticipating through prediction as to when a mobile device/user might handoff, the system actively (dominant) indicates to the mobile device *when*, *where* and *how* to handoff. After all, it is the system which is likely to have a more complete view as to which access network the mobile device should be handed off with to achieve optimal results. At the same time, the initiation of the handoffs is still retained under the control of users, thus forming what can be described as a *mobile device initiated and network determined* handoff mechanism. We term such interaction taking on the *pro-collaborative* form and have devised an *opponent-process* model which attempts to explain and capture its essence. It can be observed that research in interactive and proactive computing mainly emphasises the user as the dominant process. We argue the necessity to design a balanced computing system such that, at times, the system process can also become the dominant process. Moreover, both the user and the system process alternates in taking the dominant role. We believe a re-examination of the issue associated with user-dominant design emphasis is required to either substantiate the necessity for pro-collaborative form or to disprove the need for it.

1.2.1 Pro-collaborative computing design

The pro-collaborative computing form is an exploration of the interaction relationship between the user and the system. Consider an interactive computing *user-system-task* scenario where, given a task, a user uses a system to perform the task. With the proactive form, the system is streamlined to provide the user with a more 'intelligent' interaction experience through techniques such as user anticipation or complexity hiding. With the pro-collaborative form, however, the interaction explicitly includes the *system-user-task* possibility where, given a task, the system indicates to the user 'the what' and 'the how' in accomplishing it.

We construct a simple *opponent-process* model which attempts to capture the new user-system relationship. Consider a model containing two processes where one represents the user and the other represents the system. The idea of the opponent-process is one where the 'user-process' and the 'system-process' compete to dominate the other. That is to say, at times, the user is the dominant process, i.e. directing actions, while the system assumes an ancillary role, i.e. reacts to actions of the dominant process, and at other times, the roles are reversed. Moreover, both processes collaborate together to accomplish a common undertaking or to achieve a predefined goal, regardless of which one is the dominant process. While we are certain that interactive and proactive computing visionaries have touched upon the limitation and implication of user-dominant design emphasis, few researchers have broken such tradition.

In essence, pro-collaborative form is not about advocating artificial intelligence - a lofty goal that will not be attainable in the near future. Rather, it is about looking at computer system interaction from a new perspective and discovering circumstances where the system is in a better position to make decisions or to give instructions than the users. With interactive computing being described as human-centred and proactive computing being described as human-supervised, our pro-collaborative approach can be expressed as human-cofunctioned.

1.2.2 User-oriented vs. User-dominant

At this point, we like to illustrate the subtleties between the idea of the *user-dominant* design emphasis and the idea of *user-oriented* design emphasis. Given any computer system, regardless of the *form* it assumes, it is meant to be interactive and is centred on the user, i.e. the human. It is hard to think otherwise. Indeed, we cannot alter this because the very notion of the creation of computer systems is for the sole purpose (human-centred) of using (interactivity) them. This we refer to as user-oriented design emphasis. On the other hand, we take user-dominant design emphasis as referring to designing user-oriented systems where the users represent the focus or the initiator of interactions within the computing systems. In other words, the user is the intelligent part (the action process), while the system is the less intelligent counter-part and reacts to the actions of the user. Thus, the challenge is for this thesis to illustrate the viability of designing computer systems which are not advocating user-dominance design.

1.2.3 Nomadic User and Nomadic System

We explore the design and application of pro-collaborative form in the realm of ubiquitous and pervasive computing, in particular, within the context of mobility support. We partition mobility into two perspectives: i) the notion of a *nomadic user* and ii) the notion of a *nomadic system.* By nomadic user, we refer to it in a literal sense where a user's physical presence changes with respect to time. Research into this area is well established, some examples include, work on device mobility (IP-layer) [47], [30], [56], [33], higher-level mobility [40], [26], [55], [6], and person/user mobility [60], [79], [53], [2]. By nomadic system, we refer to it as the ability of computer systems that offer network services (e.g. multimedia transformation [24], [42], distributed data storage [34], [52]), to move their services to where the user wants it to be. For example, (home) network services 'move to' a mobile (away) user, as s/he requests the need for them, or scenarios where network services 'come to' a stationary (home) user upon request. Research in this area is gaining popularity, ranging from Service Overlay Networks architectures [71], [8], [50], [63], [82], active/programmable network [17], [28], [36], to service enabled Internet frameworks (vertical and horizontal) [50], [63], as well as Extranet-on-demand [75]. In essence, the goal of this thesis is to present the case *for* pro-collaborative design from the perspective of both a nomadic user and a nomadic system.

1.2.4 Summary

In short, our challenges are as follows:

1.To present an argument *for* the necessity of pro-collaborative design form.

2.To convey the limitations and problems of designing computing and communication systems which emphasise user-dominance.

3.To show the viability and examples of designing pro-collaborative computing and communication systems/protocols within the context of providing mobility support in the ubiquitous and pervasive environment.

1.3 Contribution

This thesis postulates a new form of computing system design, one where the user and the system interchange the dominant role in an interaction within a computing and communication system. We evaluate the new pro-collaborative form through architectures for mobility support, with the goal of working towards the ubiquitous and pervasive computing visions. We partition mobility into nomadic user and nomadic system and examine the application of the new design approach from both perspectives.

With nomadic user, we illustrate an architectural extension for the Mobile IP protocol, which addresses the issues of handoff optimisation, that we call S-MIP (Seamless Handoff Architecture for Mobile IP). We show that the performance gained, as a result of the pro-collaborative design, is significant when compared with existing proposals from the IETF mobileip Working Group. It is possible to achieve a perceived handoff latency of zero, i.e. as if no handoff had occurred from the user's perspective, and a network layer (L3) latency comparable to that of the access layer (L2) handoff delay (in the order of tens of milliseconds). With this, we illustrate the viability of pro-collaborative design.

With nomadic system, we describe a framework for dynamic network services provisioning, termed *NET*AMORPHOSE (NETwork Adaptable and MObility-awaRe Programmable HOrizontal SErvice) architecture, which facilitates the notion of moving network services. We show how network services such as multimedia transformation can be dynamically extended from one's home network to any network where the (away) user is temporality attached to. The willingness for such dynamic service extension requires the re-examination of how 'loosely structured' the service agreements between network operators can be. We hypothesise that if network services are perceived as an abstraction of a piece of software program, then only the abilities to perform computation and store information are required *within* the network, to provide the dynamic extension of network services. We illustrate how pro-collaborative design can be applied in the construction of this architecture with such rudimentary requirements.

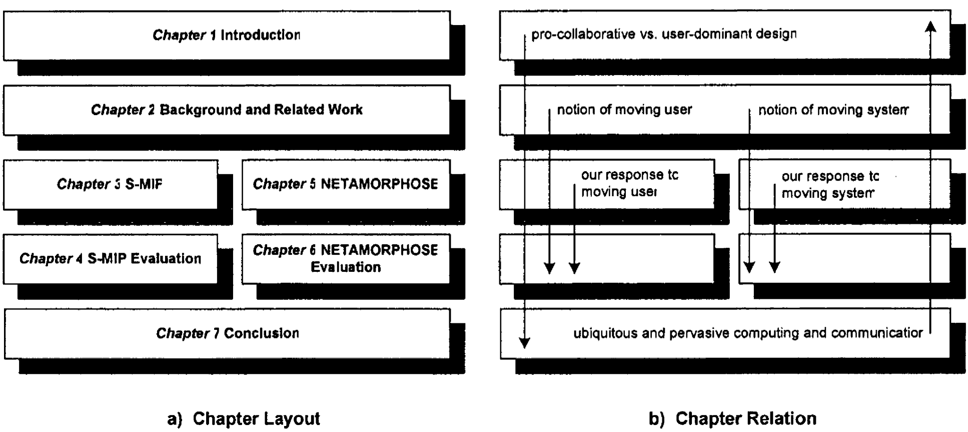


Figure 1-1 Structure of this Thesis

1.4 Organisation of Thesis

The remainder of this thesis presents the pro-collaborative form in designing network computing systems, in particular, mobile networking systems. Figure 1-1 illustrates the structure of the thesis. Chapter 2 motivates our work by presenting a comparison between interactive, proactive and pro-collaborative design approaches and their relationship to current cutting edge researches in mobility management and intelligent service/application architectures. In Chapter 3 and Chapter 4, we present a mobility management scheme that is based on the pro-collaborative form and address the notion of nomadic user. In Chapter 5 and Chapter 6, we illustrate an architecture for dynamic service provisioning which addresses the notion of nomadic system and is also based on the pro-collaborative approach. The evaluation of the pro-collaborative form as well as its comparison to the interactive and the proactive design runs through the entire thesis. Finally, we conclude and outline future research directions in Chapter 7.

As far as the laws of mathematics refer to reality, they are not certain; and as far as they are certain, they do not refer to reality.
- *Albert Einstein*

Everything has its beauty, but not everyone sees it.
- *Confucius*

Chapter 2

Background and Related Work

In this chapter, we first compare the interactive and the proactive computing form with our idea of the pro-collaborative computing form. Our thesis is that, in general, both the interactive and the proactive forms take on the user-dominant design emphasis, and we believe that a balanced approach (between the user and the system) remain unexplored and could potentially be influential. We then survey current research literatures relating to the notion of the nomadic user and the nomadic system. The research literature also accentuates the relationship between these notions and the design forms, i.e. the interactive, the proactive, and the pro-collaborative approach.

2.1 Interactive and Proactive computing

First and foremost, we present our view on what the interactive and the proactive computing forms are, and their relationships to our proposed pro-collaborative approach.

Licklider's vision of Human-computer symbiosis [37] expresses the idea of an interactive and human-centred computing system design. As we have mentioned in the Introduction, we distinguished between user-oriented design emphasis and user-dominant design emphasis, as well as what it meant to be 'human-centred' or 'interactive' in relation to the context of system design. Here, in discussing interactive computing, we specifically refer to the user-dominant aspect where the term is used in a more literal sense, denoting a simple one-to-one or one-to-many interaction between the human and the machine (or the user and the system). The idea is simple. The user initiates an interaction with an action and the system then reacts to that action and returns some results. This is the classic way of interacting with computer system, where the focus is predominantly the automation of (office) tasks [59]. Interactive computing form takes advantage of two functional characteristics of a machine, that is, speed and accuracy.

Proactive computing is about recognising the fact that the interaction with computer systems has gradually shifted towards a 'many machine to one user' relationship. It addresses the issue of enabling the interaction to be more meaningful, intelligent and user-friendly. Its goals are aligned with that of the ubiquitous computing vision described by Weiser a decade ago as well as with the emerging idea of Pervasive computing and communication. For clarity, in this thesis, we refer to ubiquitous computing as having computing power everywhere (extended from what is traditionally known as the distributed systems), whereas pervasive computing and communication as the ability to access computing and communication services anywhere and anytime. Proactive computing inherits the properties of interactive computing, and extends it with two prevailing concepts: i) proactive environment and ii) complexity hiding.

In what follows, we present two main bodies of research work pertaining to proactive computing. Firstly, IBM's Autonomic Computing [73] centres on managing or hiding complexity. It puts forward eight principles of system design. These principles include the ability of systems to self-monitor, self-heal, and self-configure to improve their performances. Moreover, computer systems according to Autonomic Computing should be environmentally aware, guard against attack, use open standards to communicate, and anticipate user actions. Secondly, Intel's Proactive Computing [59] explores the frontiers of computing in the future which overlaps with IBM's Autonomic Computing. Specifically, it investigates intelligent 'proactive' computing environments that anticipate users' needs and act on their behalf. It describes three loci. Firstly, *getting physical* is about connecting networked systems to their physical surroundings. Secondly, *getting real* is about fusing control theory and computer science and responding to external stimuli constantly, e.g. monitoring room temperature changes. Lastly, *getting out* is about placing the human 'above the control loop', that is, moving the user into the management or the supervisory role in relation to computing systems and devices. Proactive Computing also maps out seven specific underlying guiding principles: connecting with the physical world, deep networking, macro-processing, dealing with uncertainty, anticipation, closing the control loop, and making system personal [66].

Examples of Proactive Computing include, but not limited to, the Labscape [4] project and the Personal Server [64] project. In this section, we briefly outline the Labscape project as an example. Under this project, the idea is to record every step taken in performing an experiment in a microbiology laboratory. On a mandate level, this means instrumenting reagents, reaction vessels, test equipments, and the staff, and tracking their relative locations through experiments. The whole experiment can be recorded electronically and the methods and results can be automatically generated in a notebook entry. As a result, no steps can be accidentally lost. Furthermore, expert systems can also be used to examine the data for potential contamination risks or other experimental pitfalls. The electronic record of an experiment also allows speedy error determination and error tracking if this is deemed to be necessary after performing the experiments. In this project, the principles of Autonomic Computing are essential as they serve as the underpinning concepts for the individual systems, and enable individual components to efficiently and reliably cooperate with one another. The project also touches the physical world, requiring real-time responses (getting-physical) and keeping the user out of the computational loop (getting-out) wherever possible.

2.1.1 Comparison to Pro-collaborative form

Pro-collaborative form can be viewed as another extension from the proactive computing form. It describes a single principle, namely, achieving equilibrium in the opponent-process model when designing a computer/communication system, i.e. to achieve a balance between the user-dominant and system-dominant design emphasis. It stresses the necessity to bring the user-system interaction relationship onto a more equal level. It is *not* advocating artificial intelligence, i.e. treating the system as an intelligent human being. But rather, it is about discovering situations where the system should be designed to take on an active role when interacting with users. It is a pattern in design. The key sign which captures the need for the pro-collaborative design is under situations where even though the system can be clearly observed to be in a better position in making a decision, it still operates around anticipating the user's action and subsequently reacting to the user. It should be noted that, without advocating artificial intelligence, the task to be performed is assumed to be defined a priori, for practical purposes, with the pro-collaborative approach. Furthermore, the user should still retain the final decision as to whether to accept the system's decision or not when the system assumes the role of the dominant process. Unlike communication systems such as GSM [43], for example, where the design is strictly system process dominant and that the user has no control (as to what cell to switch to), pro-collaborative approach is less stringent in this regard and offers a 'relief valve' mechanism under situations where the system process malfunctions due to some unforeseen circumstances. Refer to a summary of the different design forms under Table 2-1.

2.2 Notion of Nomadic User

In this section, we discuss the notion of the *nomadic user* in relation to current cutting edge research in device mobility as well as higher-level mobility management

Table 2-1 Comparison of design *Forms*

Form		*Function*	*System's relation to user*	*Goal*
Interactive		classical	Human-centred	Automation (office)
Proactive	- Autonomic	8 principles	Human-supervised	Ubiquitous/Pervasive Computing
	- Proactive	3 Loci & 7 principles		
Pro-collaborative		1 model	Human-cofunctioned	Ubiquitous/Pervasive Computing

schemes. We then discuss and interrelate the architecture (the realisation of design) that governs each of these schemes with the three design forms, which are the interactive, the proactive and the pro-collaborative forms.

In what follows firstly, we outline the differences between the two mobility management perspectives, which are, device mobility and higher-level mobility. We refer to device mobility as mobility management designs that view the physical computing and communication devices as the end point/host/entity for the management of mobility. It has been previously assumed that the reachability of end device equates to reachability to the end user, which clearly reflects the interactive computing form and the one-to-one interaction relationship. This is no longer the case in today's communication environment. In the most general case, IP-layer mobility management schemes are a good actualisation of device mobility, as the delivery of IP packets are identified only by the name (IP address) of the end device.

Higher-level mobility differs from device mobility as it delays the mapping between an end device and the user until deemed necessary. For instance, a user might use two computer communication devices, e.g. a laptop and an Internet-enabled cellular phone. An e-mail message arriving at the laptop, which the user is not attending to, may be re-mapped or re-directed to the cellular phone, which the user is using at that time, with higher-level mobility schemes. These schemes operate at the OSI transport and/or the session layer or higher where the abstraction moves a step closer in representing an actual user/person. As the mobility management schemes move higher within the OSI model, the ability to provide a more flexible abstraction on nomadic users enhances. The defining role played by the end users on the way they interact with computing devices is a living example of the necessity to move from the interactive to the proactive or pro-collaborative computing paradigms.

It should be noted that, in relation to a nomadic user, both the device mobility and the higher-level mobility schemes fall within the same category and are syntactically equivalent, i.e. it centres on the movements of the end host/point representation regardless of whether it is a physical device, a connection, a session or an end user.

2.2.1 Device Mobility

Under this section, we discuss current advanced schemes in providing mobility management at the IP layer. In particular, we emphasise, in greater detail, the handoff management and latency reduction aspects relating to the Mobile IP family of protocols (including both MIP version 4 and MIP version 6). A sound understanding in this area is necessary in grasping the proposed S-MIP protocol which is presented in the next chapter.

In the Internet (IP) environment, when a mobile node/host moves and attaches itself to another network, it needs to obtain a new IP address. This change of IP address means that all existing IP connections to the mobile host need to be terminated and then re-established. This is essential as the IP routing mechanisms rely on the topological information embedded in the IP address to deliver the data to the correct end-point. Mobile IP (MIP) [47] describes a global solution that overcomes this problem through the use of *indirection* provided by a set of network agents. It does not require any modifications to existing routers or end correspondent hosts.

With MIP, each mobile host is identified by an address from its home network regardless of the point of attachment. While a mobile host is away from its home network, it obtains a care-of IP address (CoA) from the visiting (foreign) network/agent and registers it with a home agent within its home network. The home agent intercepts any packets destined to the mobile host and tunnels or explicitly routes (source routing [30]) them to the mobile host's current location. Thus, initiating this indirection requires a timely *address reconfiguration* procedure and a home *network registration* process. The time taken for a mobile host to configure a new network CoA within the visiting network together with the time taken to register with the home agent constitute the (overall) handoff latency.

Handoff latency is the primary cause of packet loss and results in performance degradation, especially in the case of reliable end-to-end communication. As a result, numerous methods of minimising the handoff latency have been proposed in the following research literature. The proposed schemes can be broadly classified into those that operate above the IP layer [6], [7], [15], [16] (details discussed in Section 2.2.2), and those that operate at the IP layer [33], [51], [56], [62]. In general, the solutions that operate at the IP layer are regarded as being more suitable as they do not violate any of the basic Internet design principles [54], [12], and more importantly because they do not require any changes to the protocols at the corresponding hosts. This section focuses on the IP layer solution. Essentially, the IP layer solutions attempt to minimise the registration delay by i) introducing a hierarchical structure, and ii) lowering the address (re)configuration delay through advanced configuration.

2.2.1.1 Hierarchical Structures and Protocols

Hierarchical schemes separate mobility management into micro mobility (intra-domain) and macro mobility (inter-domain). They introduce a Mobility Routing Point (MRP) [18] that separates micro from macro mobility. The MRP entity is normally placed at the edges of a network, above a set of access/edge routers which constitute the MRP's network domain. The MRP intercepts all packets on behalf of the mobile host (MH) it serves and redirects them to the MH. This enables MHs, which move between access networks that are within the same MRP network domain, to register with the MRP, therefore avoiding potential lengthy run-trip delays associated with registration to its home agent. This type of intra-domain mobility is managed by IP Micromobility management protocols, such as the HAWAII [51] and the Cellular IP [62] protocols, whereas inter-domain or macro mobility is almost exclusively managed using Mobile IP.

Within the context of MIPv4, although there are protocols which enable the construction of the hierarchical style in mobility management, as detailed in [51] and [62], there is no standardisation as to how exactly the hierarchical structure is to be formed. However, with MIPv6, a binding mechanism is built into its base protocol. A *binding* is, in general, an association of the MH's home address with its current CoA. This can be achieved by the MH sending the home agent a packet containing a Binding Update (*BU*) message which is in fact, an IPv6 destination option[3]. In response, the home agent replies with the Binding Acknowledgement (*BAck*). This generic indirection mechanism enables MIPv6 qualified hosts to form or to act as

[3] All new messages in MIPv6 are defined as a type of IPv6 destination option. The destination option in IPv6 allows packets to carry the required additional information to be examined/received *only* by the specified destination node.

indirection hosts. Such an inherent property of the MIPv6 protocol can be used to aid the creation of a hierarchical structured framework and most notably, with the Hierarchical Mobile IPv6 mobility management [56] draft from the IETF mobileip Working Group (WG).

In the context of Hierarchical Mobile IPv6 (HMIPv6), the Mobility Routing Point is equivalent to the MAP (Mobility Anchor Point) entity and the protocol for micro and macro mobility is achieved using the features of MIPv6, in particular, the binding mechanism. The MAP in HMIPv6 intercepts all packets on behalf of the MH it serves and tunnels them to the MH's on-link care-of address (LCoA). When a MH moves into a new MAP domain, it acquires a regional address (RCoA) and an on-link[4] (LCoA) address. In the simplistic case, the address of the MAP entity is used as the RCoA while the LCoA address is formed using the Stateless Address Autoconfiguration [61] protocol. Other methods of obtaining the RCoA and LCoA can be found in the HMIPv6 protocol draft. After obtaining these addresses, the MH then sends a *BU* to the MAP which will bind the MH's RCoA to the LCoA. If successful, the MAP will return a *BAck* to the MH indicating a successful binding (registration). In addition, the MH must also register its new RCoA with its home agent by sending another *BU* that specifies the binding between its home address and the RCoA, i.e. the MAP's address. When the MH moves to a new access router within the same MAP domain, it simply acquires an LCoA and updates the MAP.

[4] Three different addressing scopes exist in IPv6. A global address uniquely identifies a node on the Internet. A regional address is a global address that is specific to a particular region/domain on the Internet. An on-link address is an address local to a domain. It is only a unique identifier inside the specific domain and may not be uniquely identified on the Internet.

2.2.1.2 Local Handoff Latency Reduction Protocols

In this section, we discuss techniques used in reducing the address (re)configuration delay in Mobile IP networks. Specifically, we examine and highlight the differences between Low Latency Handoff, Fast-handover, Hierarchical Mobile IPv6 with Fast handover, and Simultaneous Bindings, from the IETF mobileip WG.

Low Latency Address configuration is about configuring an address for the MH in a network that it is likely to move to, *before* it moves. The Low Latency handoff [20] proposal describes two methods of achieving this, namely *Pre-registration* and *Post-registration*. With Pre-registration handoff, the MH is assisted by the network to perform L3 (layer-3) handoff before it completes the L2 (layer-2) handoff. It uses L2 'triggers' which arise as a result of beaconing signals from the network the MH is about to move to, to initiate an IP layer (L3) handoff. This design, however, diverges from the clean separation of L2 and L3 of the base Mobile IPv4 scheme and is not adhering to the End-to-End design principle. With Post-registration handoff, L2 triggers are used to setup a temporary bi-directional tunnel between the oFA (old Foreign Agent) and nFA (new Foreign Agent). This allows the MH to continue using its oFA while performing the registration at the same or later time. A combined method is also possible where, if the Pre-registration does not complete in time, the oFA forwards traffic to the nFA using the Post-registration method in parallel.

Fast-handover [33] is the Low Latency handoff equivalent for the Mobile IPv6 network. It is similar in concept to the combined method as described earlier, and consists of three phases: handover initiation, tunnel establishment and packet forwarding. In its most basic form, the handover initiation is started by the L2 trigger based on certain policy rule (unspecified by IETF at the time of writing). This is done by the MH sending an *RtSolPr* (Router Solicitation Proxy) message to the PAR (Previous Access Router[5]) indicating that it wishes to perform a fast-handover to a new attachment point. The *RtSolPr* contains the IP layer address of the new attachment point which is derived from the NAR's (New Access Router) link-layer beacon messages. The MH will receive, in response, a *PrRtAdv* (Proxy Router Advertisement) message from the PAR, with a set of possible responses indicating that the point of attachment is i) unknown, ii) known but connected through the same access router or iii) known and specifies the network prefix that the MH should use in forming the new CoA. Subsequently, the MH sends an *F-BU* (Fast Binding Update) to the PAR using its newly formed CoA based on the prior *PrRtAdv* response as the last message before the handover is executed. The MH receives an *F-BAck* (Fast Binding Acknowledgment) message either via the PAR or the NAR indicating a successful binding.

The tunnel establishment phase creates a tunnel between the NAR and the PAR. To establish a tunnel, the PAR sends a *HI* (Handover Initiation) message (containing the MH's requesting CoA and the MH's current CoA) to the NAR. In response, the PAR receives a *HAck* (Handover Acknowledgement) message from the NAR. If the new CoA is accepted by the NAR, the PAR sets up a temporary tunnel to the new CoA. Otherwise, the PAR tunnels packets destined for the MH to the NAR, which will take care of forwarding packets to the MH temporarily.

[5] In MIPv6 networks, the old Foreign Agent (oFA) is alternatively termed Previous Access Router (PAR) while the new Foreign Agent (nFA) is termed New Access Router (NAR).

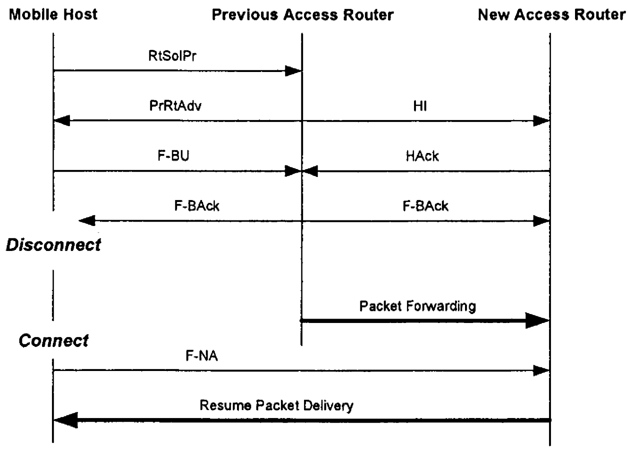


Figure 2-1 Example of Fast-handover protocol Interaction

Finally, the packet forwarding phase is performed to smoothen the handoff until subsequent registration by the MH to the home agent is completed. The PAR interacts with the NAR to facilitate the forwarding of packets between them through the previously established tunnel. The initiation of the forwarding is based on an 'anticipation timing interval' heuristic, that is, the network *anticipates* as to when a MH is likely to handoff and therefore infers the appropriate packet forwarding moment based on the anticipation timing interval. Upon arrival at the new access network, the MH sends the *F-NA* (Fast Neighbour Advertisement) message to initiate the flow of packets (to itself) from the NAR. As observed, the proactive aspects of system design start to emerge with the use of an anticipation timing interval. However, such an anticipation interval is extremely difficult to generalise, and forwarding too early or too late will result in packet losses, negating the purpose of packet forwarding. Figure 2-1 illustrates the basic operation of the Fast-handover protocol.

Hierarchical Mobile IPv6 (HMIPv6) with Fast-handover [56] is another attempt to further reduce the overall handoff latency from what Fast-handover can offer on a stand-alone basis. By combining HMIPv6 with Fast-handover, latency due to i) address configuration and ii) the subsequent home network/agent registration, can both be reduced. The MAP can be viewed as the 'local home agent', and in most cases, is located closer to the MH than the home agent. Therefore, the signalling cost saved is the difference between the roundtrip time of the MH to the MAP, and the roundtrip time between the MH to the home agent, assuming that message processing time within a network node is insignificant in comparison. This combination requires minor modification to the standard HMIPv6 protocol and the Fast-handover protocol, i.e. relocating the forwarding anchor point from the PAR to the MAP as outlined in [56].

An alternative to the packet forwarding scheme has also been proposed, namely, the Simultaneous Bindings scheme [21]. It proposes to reduce packet losses at the MH by n-casting packets for a short period to the MH's current location and to n-other locations where the MH is expected to move to. The n-casting can be carried out by the PAR, the MAP or the HA. The Simultaneous Bindings scheme recognises the problem of not knowing when the MH is likely to move – the timing ambiguity – and attempts to remove it by packet duplication to multiple access networks. In performing the delivery of duplicated packets, this scheme also attempts to address the problem associated with the ping-pong movement of MHs between two access routers. By allowing the possibility of MHs to receive the same packets from multiple access networks, it aims to remove the unnecessary re-configuration of the

MH's CoA during ping-pong movements (rapid back and forth movements between two access routers/points).

2.2.1.3 Summary

As observed throughout our discussion on device mobility, in particular with the family of Mobile IP protocols, the emergence from interactive to proactive form in protocol design is evident. With hierarchical structures such as HMIPv6 resembling the interactive approach (e.g. MH initiating the binding process with MAP/home agent) and the local handoff latency reduction protocols moving toward the proactive approach (e.g. anticipation process), the mobility management schemes were shown to grow in complexity and intelligence. However, with the rising problem of timing ambiguity mainly due to the willingness in creating a more intelligent proactive environment, the necessity for pro-collaborative design pattern becomes apparent.

2.2.2 Higher-level Mobility

At the other end of the mobility spectrum, by higher-level mobility, we refer to mobility management schemes which attempt to provide a superior abstraction model for what the end point of the management of mobility should be. In general, these schemes operate at transport layer and above, including session layer within the OSI model, or even targeting the abstraction level of the actual end user.

Firstly, with mobility management at the transport layer, the key rationale is to provide the communication abstraction at a connection level. As users establish connections to communicate, this approach is a step closer to better user abstraction. The connection abstraction can be connection oriented (e.g. TCP) or connectionless (e.g. UDP). On the other hand, with mobility management at the session layer, the idea is to group connections together as a logical 'session' communication. Thus, a better grasp at representing an end user may be achieved. For instance, a Peer-to-Peer file downloading program may open several simultaneous FTP connections in retrieving a large multimedia file and the logical operation of all these connections at once due to mobility (of the user) is the key rationale behind session mobility. Lastly, with mobility management at the user level, the aim, as described earlier, is to delay the binding between a computer communication device and a user as late as possible. Motivated by the fact that a user may operate multiple computer communication devices at once while most likely using only one particular device at a specific moment in time, it is ideal to be able to (re)direct the incoming communication to the correct end device that the user is currently operating on. In what follows, we step through the higher-level mobility management schemes and highlight the underlying design philosophy of each scheme in relation to the notion of a nomadic user.

2.2.2.1 Transport Oriented

One of the earlier works to manage mobility at the transport layer is the Indirect TCP (I-TCP) [6] protocol. It argues that any interaction from a MH to any fixed network nodes should be partitioned into two separate individual communications. One is between the MH and an intermediate node (wireless side) termed Mobility Support Router, located at the boundaries between the wired and the wireless network mediums. The other is between the Mobility Support Router and the nodes within the fixed network (fixed side). Elegantly, this accommodates special requirements of MHs, such as random wireless error correction, yet offers backward compatibility with that of the existing fixed networks. All the specialised support can be isolated and built within the wireless side of the interaction and more importantly, the fixed side of the network remained unchanged.

The MSOCKS Architecture [40] proposed a proxy architecture where the proxy, placed in between two end hosts, is used to 'fuse' TCP connections together in making it appear as a single TCP connection between the two end hosts. The ability to fuse the TCP connections makes it possible to hide the movements of MH. For instance, as MH moves from one network to another, communication connections are broken as its IP address is changed accordingly. MSOCKS allows the reconnection of the TCP streams due to mobility through the technique called TCP splicing, thereby maintaining an apparent constant 'unbroken' stream of data between the end hosts. It provides a stronger conformance in preserving the End-to-End semantics for mobility management as compared to that of the I-TCP protocol.

Diverging from designs which address the distinct characteristics of the last-hop wireless link (e.g. random wireless errors, round-trip time variations, blackouts, etc.) and being sender centric, more recent schemes like RCP [26] (Reception Control Protocol) argue the viability to place control intelligence in the receiver, i.e. the MH. The RCP protocol is a TCP clone in its general behaviour but allows for better congestion control, loss recovery and power management as compared to the sender-centric approaches, as the MH is able to 'tailor' operations that are best suited to its current communication state. More importantly, by shifting management intelligence to within the MH, RCP allows support for access to heterogeneous wireless networks for MHs equipped with multiple interfaces, e.g., a wireless LAN, à Bluetooth, and an infer-red interfaces.

2.2.2.2 Session Oriented

The Session Layer Mobility (SLM) [35] management extends the IP or transport layer model by introducing a session layer management that operates above TCP (at the transport layer) and is capable of manipulating/switching TCP streams between different connections. Its architectural layout is a proxy based structure similar to that of the MSOCKS protocol. The distinct advantage of a session layer connection abstraction is the ability to apply mobility management to any open data streams, i.e. data streams can be moved in-between different access network types, e.g. from GSM to Internet (IP) or vice versa. Thus, the operations of the SLM protocol can be viewed simply as a concatenation of data stream sockets (UNIX sockets primitives). The location management of SLM is provided through the use of a central database which can be queried in obtaining the location of a MH and conceptually it functions similar to the home agent in Mobile IP or the HLR (Home Location Register) in GSM networks in concept.

The Migrate [55] architecture is an end host centric session management framework. Unlike SLM where the mobility support is provided by a proxy 'within' the network, the end host session management abstraction is achieved with application-level connectivity monitor, policy engine and extensions to current TCP stacks, which includes extra options to facilitate mobility awareness. The novelty of Migrate is its emphasis and recognition that device/terminal mobility and user/personal mobility can be cast as a specific instance of session mobility. Thus, a host movement can be viewed as the simultaneous movement of all sessions terminating at that host and *not* vice versa. For instance, a session can be migrated from device to device, but session migration due to device mobility means that *all* session must be migrated. The location management aspect of Migrate is achieved through modification to existing DNS infrastructure, that is, whenever the MH changes its network attachment point, it updates its new location with the DNS server through a new hostname-to-address mapping.

2.2.2.3 User Oriented

Moving completely out of the OSI reference model, the user or person oriented mobility management schemes mainly deal with mobility management systems that support i) contactability and ii) personalisation. In general, the operation of the mobility management schemes at the user level employs the use of lower layer mobility management protocols (either transport or session layer) in supporting the actual communication. A database request-and-reply model is the prevailing architectural model in designing contactability schemes. The database can be either in a centralised or distributed form. Examples of contactability support schemes include the Universal Personal Telecommunication (UPT) [79] scheme, the Session Initiation Protocol (SIP) [53], the Mobile People Architecture (MPA) [2], the IPMoA (Integrated Personal Mobility Architecture) [60] scheme, and the instant messaging systems such as ICQ [77] and Microsoft Messenger [80]. In particular, with MPA, the concept of a *Person Layer* is introduced. This layer provides a uniform naming scheme to identify a user and allows the use of application-specific addresses such as an e-mail address or even phone numbers, as well as routes communications between these addresses (person-level routing). Moreover, the IPMoA scheme is a realisation of combining contactability and personalisation within a single user-oriented mobility management framework. It enables the person level routing mechanism which is similar to that of MPA, and at the same time, provides personalisation services such as setting user preferences and even content transformation.

2.2.2.4 Summary

As observed from our discussion on higher-level mobility management schemes, there is a distinct trend towards a more 'intelligent' computing and communication environment for the end user. From the simple connection abstraction with the transport layer mobility to the more intelligent session based abstraction, the progression from interactive to proactive computing design is evident. The design for systems which are genuinely proactive can be further accentuated when the abstraction for the management of mobility is actually focused at the end user level, as shown under the person/user mobility schemes.

In relation to the nomadic user, there is little difference whether the mobility management is performed at the higher-level or at the physical device. This is because the nomadic user centres on the movements of the end host/point representation of the user. Therefore, performing mobility management at the higher-level appears to give a more flexible and perhaps a more powerful end-user abstraction.

2.3 Notion of Nomadic System

Within the context of ubiquitous/pervasive environment, by *nomadic system*, we refer to the ability of the 'networked' computer system that offers services on the Internet to *move* its services to where its user wants the services to be. Moreover, the notion of the nomadic system can be viewed from two perspectives. The system/service could 'come to' a stationary user or the system/service could 'move to' or 'follow' a mobile user.

The term *service* is used in a very general sense to mean the provisioning of value-added benefit on top of what the Internet provides currently (i.e. contactability and connectivity). Examples of the types of services we have in mind include distributed content storage, multimedia transformation and dynamic QoS (Quality-of-Service) provisioning. These types of services often require the presence of specialised physical computer systems[6] within an ISP (Internet Service Provider), such as the network operator, for particular services (through the use of these specialised systems) to become available and accessible for the users of the ISP. Therefore, the terms 'system' and 'service' are used interchangeably here as we assume that adding new specialised computer systems within a new network operator means that the services are available within that network.

Consider a content distribution network like Akamai [71]. To enable the operation of its service, specialised Akamai boxes have to be installed or deployed within a local network operator (i.e. an ISP) before the content distribution service can be operational within that local network (for the users of the local network). Assuming that computing resources (computational power and data storage) can be 'rented' on-demand transparently, then the somewhat 'static' nature of the deployment of the Akamai boxes becomes unnecessary. The idea is to have 'virtual' Akamai 'software' boxes which can be set up 'on-the-fly' and deployed on-demand. This captures the essence of the nomadic system, which is, the ability to 'grow' or 'shrink' one's service offering/coverage based on user demand (see Figure 2-2).

[6] In colloquial terms, it is also called a (computer) 'box'.

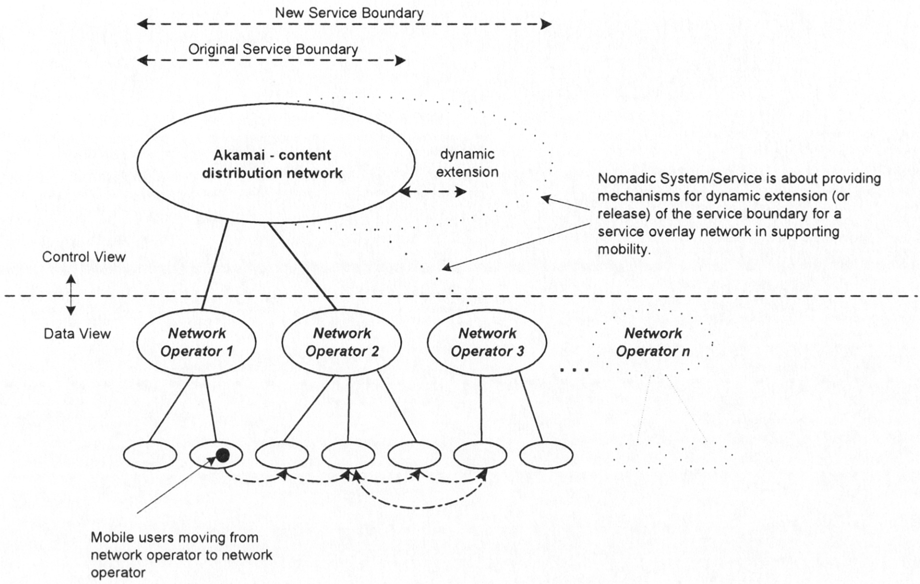


Figure 2-2 An illustration of the Notion of Nomadic System

The emphasis of this thesis is to examine the intersection of the nomadic system and nomadic user, i.e. how to dynamically move the 'home service' of a mobile user (i.e. the service the user used to receive at the home network) when s/he is away from home attached to some distant (foreign) network. Furthermore, we also examine how the nature of nomadic system can be modelled using the pro-collaborative form without resorting to artificial intelligence, as this appears to be unattainable in the foreseeable future.

2.3.1 Concept of Network Service

In understanding the concept of network services (essentially *software* within the network/Internet), firstly, we need to examine the evolution of the Internet architecture over the past decade. The 'old' Internet consists of local networks with local names and switches. The standardisation of the IP (Internet Protocol) enables the linkage of these local networks and provides the means for global

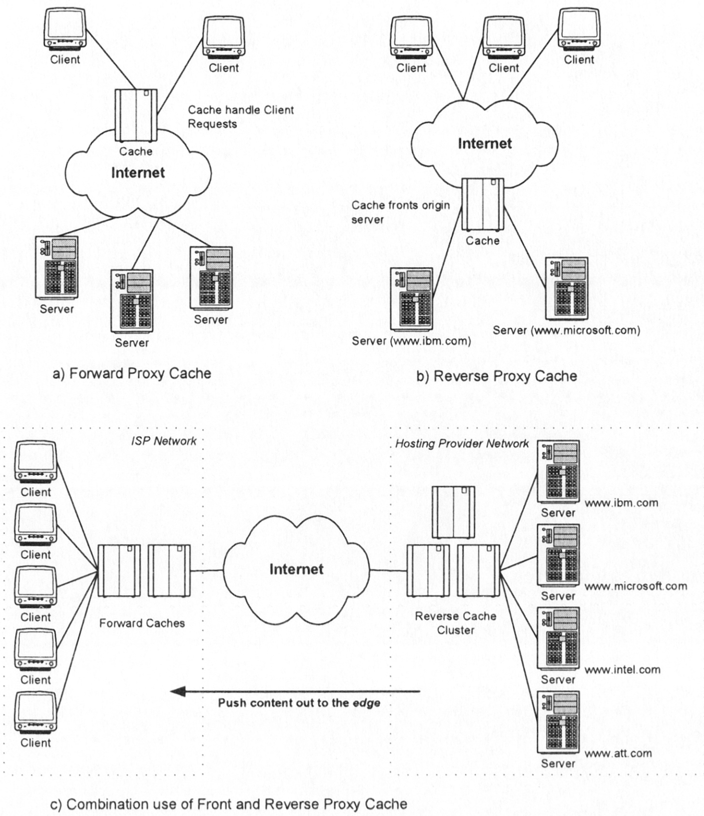


Figure 2-3 Network Service – Software *within* the Network

naming/addressing. Routers are used to connect the local networks (IP names) worldwide and are used to move IP packets. The formation of the Internet is a co-operation (peering) of large network providers. Its design is IP datagram centric which is flexible and easy multiplex (efficient). The system states are kept at the edge of the network and the core is simple and reliable.

The first instance of a 'network service' is the use of caching within the Internet Service Provider. The idea is to keep a local copy of the frequently accessed Internet contents, thus enabling a fast response time, improving bandwidth saving and protecting against burst load, i.e. surge protection against server. This use of the caching technique is generally known as the forward proxy caching (see Figure 2-3a), with the cache being placed in front of the clients. Another type of caching is called reverse proxy caching (see Figure 2-3b), this is the case where the caching is done at the server side. The idea is to place cache cluster at the front of servers to facilitate server-request surge protection by buffering the load for multiple sites, which is analogous to over-draft protection within the banking industry. This enables the availability of bandwidth for unanticipated (request) traffic surge, and allows dynamic allocation of availability bandwidth for big events (e.g. launch of the Soccer World Cup) by simply creating 'more caches' for the same content. It was not long before people recognised the benefit of linking the use of forward and reverse proxy caching (see Figure 2-3c) and hence the emergence of what was known as the content distribution network (e.g. Akamai). The essence of content distribution was the push of content out to the 'edge'. This was the first real acceptance of the idea of the network service and ultimately became widely popular. The combined use of the forward and the reverse proxy caches allows the retrieval of content from the web servers and replicates and distributes to destination caches (forward proxy cache). A by-product of pushing the content to the edge is the possibility of performing information backflow, that is, i) aggregating data about content usage and performance, ii) tracks if service level agreements are being met, and iii) dynamically adjusting content availability based on the backflow information [14].

Another emergence in network service provisioning includes recent research into what is known as edge-based services [8]. Apart from the obvious application such as caching, edge services include applications such as filtering (local content transformation), multimedia transformation, software rental or even data storage.

Often, the network services are user-centric and user-paid, and exploit architectural properties such as high local bandwidth, wide-area bandwidth efficiency, fast response time and integrated localised content. The general structure of edge services can be defined by four key logical architectural entities. There are, firstly, a '*service application*' entity that operates data flows between a service provider application and a service consumer application. A '*service dispatcher*' entity that invokes service applications based on a '*service policy*' entity which specifies usability and constraints. Lastly, a 'service flow' that represents a cooperative undertaking between service applications, service dispatchers and service consumers. Service applications and dispatchers group themselves in defining the *effective service boundary* [8].

In general, we refer to the above schemes which use proxy; content distribution and edge services, as Service Overlay Networks (SON). We consider the service boundary as 'closed' within a SON, i.e. it is unable to 'open' dynamically and provide the services to any network 'on-the-fly' without the pre-arrangement or pre-installation of service specific computing hardware.

2.3.2 Dynamic Network Services Provisioning (DNSP)

In this section, we discuss the provisioning of dynamic network services strictly within the scope of Service Overlay Networks.

It is generally considered that a large number of service overlay networks, each offering different and competitive services, will be available on the 'new' Internet in the foreseeable future, thereby envisaging the notion of a 'service-enabled' Internet [31], [50], where the Internet is a market place capable of offering services tailored to individual user needs. This service-enabled Internet provides methods and systems which lie within a logical 'service layer' to facilitate the payment transaction, the construction and the use of these services. Whilst the exact structure and meaning of what this service layer should be remains an open question, we present a view of what we believe a service layer should consist of in the next section.

Within this service-enabled Internet, for a mobile user to obtain a certain set of network services (e.g. media transformation and filtering) after detaching from a network and moving to a new network, s/he can acquire the same (set of) services through the service layer from a new set of network service providers associated with the new network operator. These sets of network service providers are likely to be different from the previous network service providers. What we are interested in as detailed within this thesis, is the possibility of providing a method to the network service provider to dynamically grow or shrink its service boundary (to and from different underlying network operators) to accommodate the mobility of (its) users/customers, and as a result, allow the possibility of the mobile user to simply maintain its current association with the existing network service provider.

In other words, we are interested in examining how loose service agreements between network operators[7] can be, in order to enable the dynamic deployment of network services by the network service provider. A network service, in its most basic form, can be simply considered as software programs. Thus, we postulate that given i) the availability of computing resources (computational process power and

[7] We refer to a network operator as the operator of the physical network, e.g. ISP, while a network service provider is a provider of network services. A network operator can also be a network service provider. A network service provider is generally considered as a virtual network operator who does not necessarily have to own any physical equipment or infrastructure.

data storage), ii) the possibility of renting these resources, and iii) the ability to install and program network services on the rented resources[8] within a network, a truly dynamic provisioning of network services is achievable. Naturally, one of the most important issues that need to be addressed, in this type of system modelling, is the security consideration. We formulate a security model that addresses this in Chapter 5.

As a contextual comparison, we refer to applications like Extranet-on-demand as being 'rigid' in service level agreement. This is because the dynamic extension of intranet-like services is normally directed to one's own organisation (that is geographically far apart) or one's business partners which is restricted to potentially a small set of networks and require pre-agreements and/or arrangements. The essence of DNSP is the ability to achieve essentially the same thing but without the pre-agreements and/or arrangements, given that computing resources are accessible and can be rented on-demand. The economical motivation behind this structure is to enable 'pure' network service providers to enter the service market place at a lower cost than the traditional model (i.e. renting resources instead of owning), where the service providers often need to own substantial infrastructure or equipments before hand. This decoupling of the network operator and the network service provider is a major and necessary shift within the new Internet milieu [31].

[8] It should be noted that the computing resources may be provided by the network operators themselves or third-party service providers which are co-located with the network operators and have a pre-agreement with the network operators.

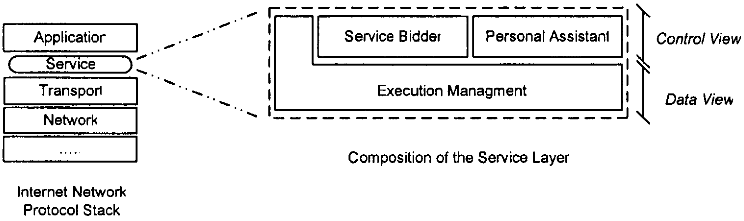


Figure 2-4 The Service Layer

2.3.3 Deconstruction of DNSP – An Abstract View

In what follows, we present our view of a service layer composition (Figure 2-4), followed by a theoretical example of the dynamical provisioning of a network service.

We believe that the service layer should i) determine the end points for the communication session and their capabilities and whether any special services are required to facilitate a communication session, ii) be responsible for locating the required network service providers in accordance with the user requirements if special services are required, iii) setup the service, and iv) create, monitor and maintain the session. With this in mind, we define the service layer as consisting of two separate *views* (or planes), that is, the control view and the data view. In terms of the control view, a logical *Personal Assistant* entity is responsible to perform activities such as locating the remote user(s)/application(s), determining the characteristics of the networks and devices being used, recognising user preferences, and deciding if network service providers are required for a communication session. The logical *Service Bidder* entity is responsible for activities that relates to the discovery of the network service providers, the coordination of service bidding and the booking of resources, if required. The logical Execution Management entity spans across both the control and data views. From the control view, it is responsible for the management, operation and maintenance of a network service. From the data view, it is responsible for the provision of computational power and data storage. These resources are used for the execution of network service specific modules. In other words, this equates to the execution of the *service application* (the network service specific module) for a *service flow* (session), as viewed from the perspective of the edge-based services.

The issue of automating the service composition (i.e. more than one network service is required and would therefore need to be concatenated) is outside the scope of this thesis. We argue that the composition process can never be fully automated in the near future (i.e. heavy reliance on artificial intelligence), rather, the process must require human intervention to be practical. Hence, we assume that the Service Bidder provides mechanisms for the users to become involved in the decision-making process. This alone raises the issue of *not* designing a DSNP framework that is predominantly system-dominant. Apart from the obvious technical challenges involving artificial intelligence and the complexity of a potential infinite number of permutable compositions of services, it would simply be a bad system design not to involve or inform the stake holder of the system, i.e. the user, in the decision-making process[9]. At the same time, referring to examples given on the nomadic user, it is also not ideal to design a DSNP framework that is predominantly user-dominant. The sheer anticipation of what network services a user may want to use or compose is already an overwhelming task without the augmentation of the consideration of user

[9] For a less complex system design, e.g. one with a well predefined goal such as the GSM network with the goal of voice roaming service, it may be conceivable or even advantageous to design a system that is system-dominant.

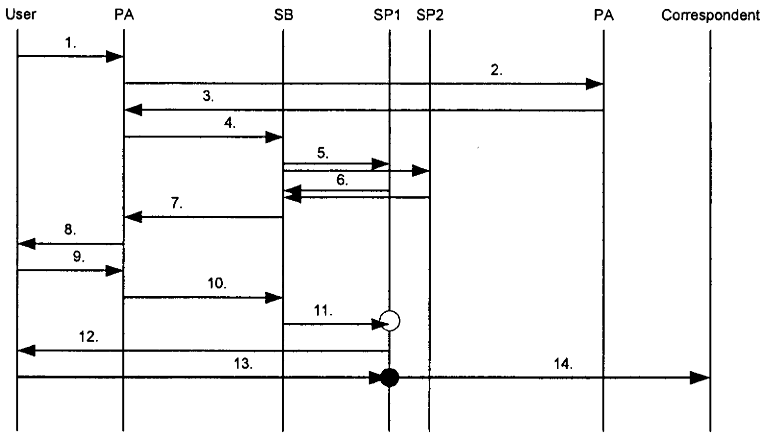


Figure 2-5 Network Service Creation Example (Inter-personal)

mobility. Rather, the best compromise is perhaps a user-system-cofunctioned design, i.e. the pro-collaborative form, where the user initiates a 'wanting' request and the system provides the best 'indicative' response(s) capable of fulfilling the request at a specific point in time.

Figure 2-5 illustrates a simple and generic example of the dynamic provisioning of network service. When a (mobile) user initiates a communication session with a correspondent (1), the Personal Assistant (PA) locates the remote correspondent application/user (2,3) and exchanges device capabilities and user preferences. The PA then decides whether any network services are required for the session. If required, it requests the appropriate network services from the Service Bidder (SB) (4). The SB locates network service providers who can provide the required services and then performs (multiple) tentative booking for resource hiring (5). Admission control is then performed at the network Service Provider (SP) to determine whether enough resources can be secured to accommodate the request on its execution environment. Upon receiving positive replies from the SPs (6), the available list of network service options are presented to the user (7, 8). The user then decides and/or composes which one to use and the chosen network service is conveyed to the service provider (9, 10, 11[10]). The SP 'installs' the network service specific module that is required to provide the network service. Once the service is installed, the user/application is notified (12) of the instantiation and the session commences (13, 14). The above is an example for the inter-personal communication. For a client-server type of communication, step 2 and 3 may be discarded.

2.3.4 Summary

In quantifying the notion of nomadic systems, firstly, we discussed the emergence of network services, followed by an examination of the necessity and the motivation behind the ability to dynamically provision network services within the service-enabled Internet. We then presented our view of what a service layer might consist of and discussed briefly how DNSP could be achieved, given the service layer, as well as how the pro-collaborative form may be applied in designing a DNSP framework.

2.4 Summary - Towards a new Internet

In this chapter, we presented a new perspective on Internet mobility, where the idea of mobility could be viewed from two different perspectives, i.e. the notion of a nomadic user and the notion of a nomadic system. The notion of the nomadic user is, in essence, an abstraction of indirections. The notion of nomadic systems, on the

[10] The empty circle denotes the provisioning of the service while the filled circle denotes the running of the service.

other hand, is about the mobility, or more specifically the reachability and availability, of network services. We envisage that the new Internet will facilitate both the notion of a nomadic user and the notion of a nomadic system, as well as a combination of these two perspectives to achieve a fully dynamic and intelligent Internet milieu. We postulate that in order to achieve these, the normal approach/view of the conventional design practice (the interactive and proactive form) might need to be adjusted to include a new design form that we termed pro-collaborative, where the computer and communication systems are designed to emphasise a user-system interaction based on the idea of human-cofunctioning. In other words, an interactive environment where both the system and the user take turns in leading a collaboration effort that is deemed to be most beneficial for the overall functioning of the computer and communication systems.

It is possible to fail in many ways... while to succeed is possible only in one way.

- Aristotle

Chapter 3

S-MIP Architecture

Current network system design approach to mobility management has been *interactive* or *proactive* in accommodating user mobility. The Seamless handoff architecture for Mobile IP (S-MIP) is an attempt to move away from the pure interactive and/or the proactive approaches, which are often user-dominant in nature. It is based on the pro-collaborative form, thus creating a mobility management scheme that is 'balanced' between the interaction of the user and the system in achieving an optimal management scheme. It is our response to the notion of the nomadic user, more specifically, it offers an IP layer solution for seamless handoff management.

3.1 Indicative Mobility Management

The traditional Mobile IP schemes [47], [30], can be described as *passively* 'interactive' to mobile users/hosts, from the perspective of the network. Here the network reacts to actions of mobile hosts, that is, a user-dominant emphasis. For instance, the handoff management is purely triggered by the mobile host moving out of a network coverage area. Furthermore, the states of the protocol are kept at the edge, i.e. at the mobile host, home agent and/or foreign agents. Recent advances in optimising the performance of Mobile IP brought about several Mobile IP extension schemes [56], [33], and location/trajectory/path prediction schemes [1], [38], [23]. These can be described as *passively* 'proactive'. Here, the network proactively anticipates actions which may be taken by the mobile hosts, yet still emphasising user-dominance, i.e. passively reacts subsequently to actions initiated by the mobile host. Essentially, in improving handoff performance, the design of these schemes centres on the network formulating a best guess at the mobile host's next movement, i.e. the new attachment point. However, there is only certain level of guarantee that the mobile host will indeed move to the anticipated networks. It is unclear as to the potential benefits as weighted against the possible penalty incurred.

In contrast, we see no apparent disadvantages for the network not being *active* in handoff decision and management. This means giving handoff instructions to the mobile hosts for a handover to the new attachment point, as well as being active in preparing and coordinating these handoff instructions. We term this approach Indicative Mobility Management, an application of the pro-collaborative form in the context of IP mobility management. In this chapter, we present S-MIP, a protocol that is designed based on the concept of Indicative Mobility Management.

3.2 System Model and Objectives

In response to great demands for wireless computing in public spaces (i.e. airports and conventional halls), as well as the rising popularity for real-time bounded services requiring high bandwidth, current advanced mobility management schemes, such as IETF mobileip WG's HMIPv6 and HMIPv6 with Fast-handover

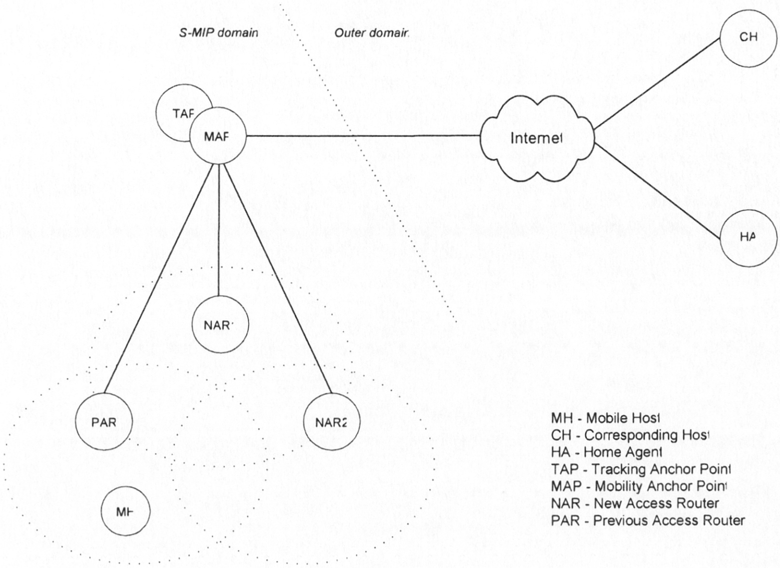


Figure 3-1 S-MIP Architecture

[56], exhibit microcellular-like system architecture. We model S-MIP as an extension to the HMIPv6 with Fast-handover scheme. Therefore, similar to HMIPv6, our system model (Figure 3-1) includes the home agent (HA), the corresponding hosts (CH), and consists of a collection of access routers (base stations), connecting to intermediate routers or switches. These are in turn interconnected over a backbone network or to the Internet. Likewise, the interconnecting routing node linking together access routers is referred to as the MAP (Mobility Anchor Point) entity. The New Access Router (NAR) is the potential next attachment point while the Previous Access Router (PAR) is the current attachment point for the mobile host (MH). We introduce a new logical entity called TAP (Tracking Anchor Point) which is similar to the MAP in its operational scope and acts as the coordination point for the location tracking and the movement patterning functionalities.

We assume that the user movements are typically within a single administration domain (i.e. within one MAP domain) and the speed of the movements are slow at 1 meter per second (approximating human walking speed). Recent measurement based study on wireless (WiFi) data networks indicates that user mobility is mostly localised [58]. It also shows that for those users who move frequently, locations that they visit are usually concentrated, i.e. users tend to stay in a neighbourhood. However, sometimes big 'jumps' are possible, though after such move, users tend to stay in a new location/neighbourhood for a while. Thus with S-MIP, we simply assume that the domain or the operational setting is within a large open-space indoor environment, having similar layout characteristics as airports and conventional halls.

The following lists our main objectives for the design of S-MIP:

- Easy Deployment: S-MIP must be non-obtrusive on existing mobility management schemes and offers as an add-on option to existing schemes.
- Minimal Signalling Overhead: Within the resource scarce wireless segment of the network, the signalling system of S-MIP must incur overhead no greater than the HMIPv6 with Fast-handover protocol. Additional overhead at the wired segment must be maintained at a minimal.
- Extremely Low handoff Latency: S-MIP is to achieve seamless connectivity at network layer (IP), where the mobile host is to experience, ideally, no loss of packets during handoffs.
- Ping-pong Consideration: S-MIP is to address the undesirable degradation of transport layer communication and handoff performance due to the ping-pong effect.

3.3 The protocol

The design of S-MIP can be illustrated by examining it at two parts. In the first part (this section), we assume an indicative mechanism is available within the network, which instructs *how*, *when* and *where* a handoff should take place for a mobile host. Given such an assumption, we describe how the mobility management protocol unfolds from the network's perspective as well as from the mobile host's perspective. In the second part (Section 3.3), we illustrate how this indicative mechanism can be constructed. We then discuss the new handoff algorithms which arise from the indicative mechanism.

In what follows, we briefly give the overview of the indicative mechanism in reference to the operation of the S-MIP protocol which is the main focus of this section. Essentially, S-MIP can be described as having a *mobile host initiated but network determined* handoff management scheme. This allows the mobile host, which has the best knowledge regarding its intention and current location to initiate

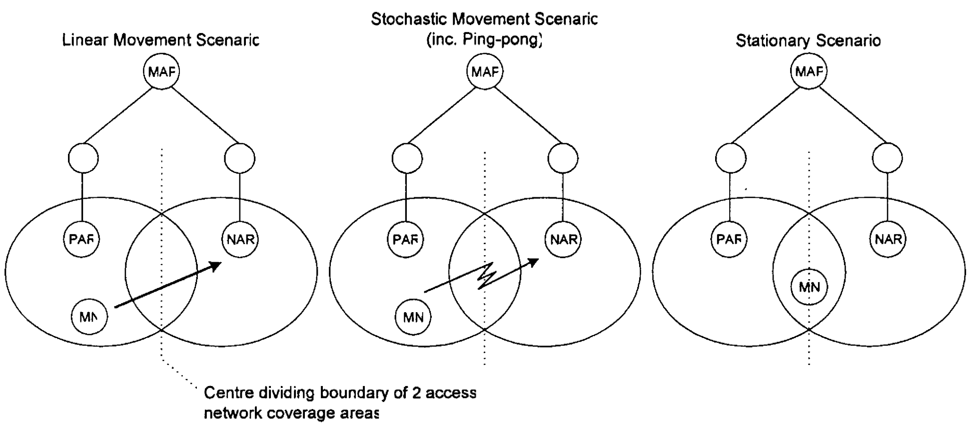


Figure 3-2 Movement Types

the handoff, yet empowers the network to unambiguously determine and instruct which network a mobile host should handoff with. The decision as to which network to handover is formulated from the movement tracking mechanism, based on periodic synchronised location feedback information. It aims to distinguish the following three conditions/indications, namely, i) is the MH currently moving linearly, ii) is the MH currently moving stochastically (including ping-pong movements), or iii) is the MH stationary near the centre boundary between two access network coverage areas (see Figure 3-2). The advantage of determining these movement conditions is that different handoff strategies can be applied to different conditions with respect to the nature of a handoff.

3.3.1 Signalling Messages and Packet Simulcast

By using the HMIPv6 with Fast-handover as the base protocol, six new additional messages are introduced with the S-MIP protocol. These are:

- Current Tracking Status (*CTS*) message from MH to TAP via AR. It contains the location tracking information.
- Carrying Load Status (*CLS*) message from AR to TAP. It contains information regarding the number of MH an AR is currently associated with.
- Handoff Decision (*HD*) message from TAP to AR. It contains the handoff decision of TAP, namely, which AR a MH should handoff to.
- Handoff Notification (*HN*) message from PAR to MH. It contains the indication from PAR directing precisely which NAR the MH should handover to. The PAR derives the content of *HN* message from the received *HD* message.

- Simulcast (*Scast*) message from PAR to MAP. The *Scast* message triggers the start of the Synchronised-Packet-Simulcast (SPS) process at the MAP.
- Simulcast Off (*Soff*) message from NAR to MAP. This message terminates the SPS process.

In addition, the Router Advertisement message is modified to include the TAP reply option, analogous to that of the MAP discovery option [56].

The packet simulcast mechanism is to remove packet loss/out of order phenomena at the NAR during the handoff process. We classify packet losses as either being due to the loss between i) the MAP and the ARs (segment-packet loss)

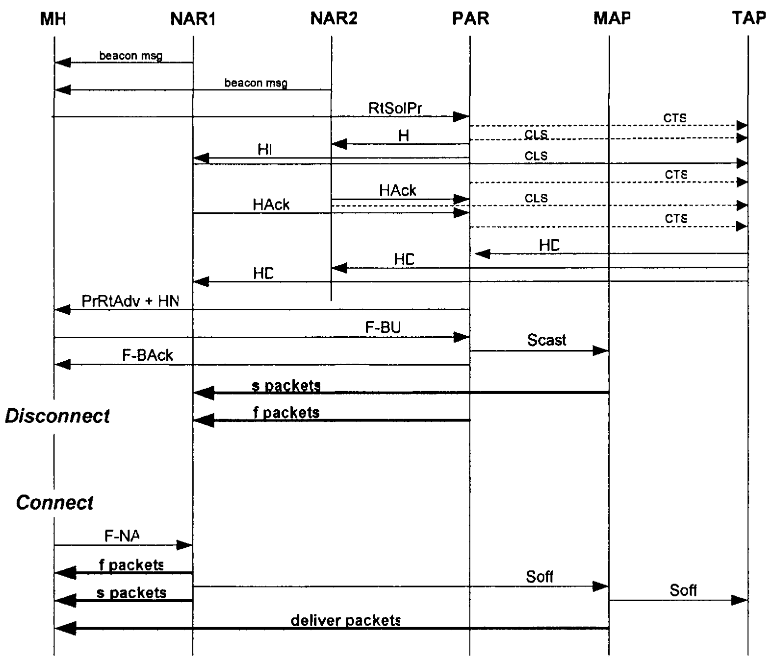


Figure 3-3 Linear Movement Operation Scenario

or ii) the last AR and the MH (edge-packet loss). The former occurs due to the non-deterministic nature of handoff timing and location, as well as the subsequent switching of the data stream at the MAP (after receiving the MAP binding update), while the latter occurs due to mobility of a MH and the possible wireless transmission errors. We believe edge-packet loss can be minimised by keeping the anchor point for the forwarding mechanism[11] as close to the MH as possible, hence in S-MIP, we locate it at the AR that bridges the wireless network and the wired network. On the other hand, segment-packet loss can be minimised via our devised Synchronised-Packet-Simulcast (SPS) mechanism. We detail the operation of the SPS mechanism through the operational scenarios described next.

3.3.2 Operational Scenarios

We illustrate the operation of S-MIP by describing the behaviours of entities involved in the exchange of the newly introduced S-MIP signalling messages and the HMIPv6 with Fast-handover messages[12]. As can be seen in Figure 3-3, upon receiving beacon advertisement messages from the newly discovered ARs, the MH initiates a *provisional* handoff by sending the *RtSolPr* (Router Solicitation Proxy) message to the PAR. This indicates that the MH desires to proceed with possible handoffs to new attachment points. Once received the *RtSolPr* message from the MH, the PAR sends *HI* (Handoff Initiation) messages to all potential NARs (e.g. NAR1 and NAR2) identified by the MH through the *RtSolPr* message. These *HI* messages contain the requested care-of address (CoA) on the new access network and the CoA used at the current access network. All identified NARs respond to the *HI* message with a *HAck* (Handover Acknowledgement) message either accepting or rejecting the

[11] Part of the soft-handover procedure outline in HMIPv6 scheme.
[12] Refer to Chapter 2 Section 2.2.1.2 for detail.

new CoA. Similar to HMIPv6 with Fast-handover scheme [56], if the new CoA is accepted by the NAR then the PAR sets up a temporary tunnel to the new CoA. Otherwise, the PAR tunnels packets destined for the MH to the NAR, which take care of forwarding packets to the MH temporarily. In response to the *RtSolPr* message, the MH receives a *PrRtAdv* (Proxy Router Advertisement) message from the PAR, with a set of possible replies, identical to what is specified in the Fast-handover protocol [33].

Meanwhile, the ARs send *CLS* messages to the TAP (Tracking Anchor Point) periodically as a reply to the modified Router Advertisement message with the TAP reply option (approximately once every 3 seconds, defined by the MinRtrAdvInterval in [44]). This synchronises the timing of the *CLS* message from individual AR to the TAP. The *CLS* message indicates how many MHs are associated with a particular AR. It must be noted that *CLS* messaging is optional. It is for the TAP to provide load balancing decision if implemented. Contrarily, a *CTS* message is generated by the MH every time it receives a L2 (layer 2) beacon advertisement from the ARs, which can be several times a second. The *CTS* message is sent back to the ARs via piggybacking as part of L2 messaging. Each *CTS* message contains the signal strength of the detectable AR and the respective identifier (Id) of the AR. The signal strength and AR Ids represents the basic unit of data which forms the MH's location tracking information. The current AR only forwards this tracking information (*CTS*) every second to the TAP, and generally stops forwarding upon the reception of the *HD* message from the TAP, unless otherwise specified. This two stage process for the generation of *CTS* message is to avoid explicit signalling which wastes precious wireless network resources. The MH may choose to send the *CTS* message to other ARs, other than the current AR, if the connection to the current AR is poor (judged by signal strength quality). In this case, the duplicated *CTS* messages arrive at the TAP are simply discarded.

3.3.2.1 Moving Linearly

After analysing the *CTS* and/or *CLS* messages, including tracking the MH's movement for a short period (minimum of 3 seconds, see Section 3.4), the TAP sends *HD* messages to all participating ARs for the specific MH requesting the handoff. In turn, the PAR sends a *HN* message together with the *PrRtAdv* message to the MH. Assuming that the MH is determined to be moving in a linear fashion, in this case, the *HD* message contains information on which AR the MH is to be handed off to. The ARs which are not selected for handover by the TAP are notified to discontinue from further participation in this handoff process in their *HD* message respectively.

The MH sends a *F-BU* (Fast Binding Update) message to the PAR after receiving the *HN* message. This *F-BU* message binds MH's on-link[13] address to the newly formed CoA, and the mechanism in forming the new CoA (by the MH) is described in [61]. When the PAR receives this *F-BU*, it sends the *Scast* message to the MAP, initiating the simulcast of data packets, hence the SPS process begins. Every subsequent data packets from the correspondent hosts arriving after the reception of the *Scast* message, at the MAP, are duplicated and send to both the PAR and the NAR simultaneously. These packets are marked with an *S* bit, as an option parameter,

[13] As a reminder, three different addressing scopes exist in IPv6. A global address uniquely identifies a node on the Internet. A regional address is a global address that is specific to a particular region/domain on the Internet. An on-link address is an address local to a domain. It is only a unique identifier inside the specific domain and my not be uniquely identified on the Internet.

in the IP header. Contrarily to [56], i.e. by sending the *F-BU* to the PAR, only one *F-BAck* (Fast Binding Acknowledgement) is sent by the PAR to the MH (still connected via the PAR) as a reply to the MH's *F-BU* message. This serves as the precise *indication* that the MH is to handoff at this moment in time. Thus the *actual* handoff begins and the MH *must not* handoff before receiving this indication. Here, the network is no longer ambiguous[14] as to *where* and *when* a MH might handoff to an AR, as well as actively indicates *how* to handoff with the *F-BAck* message.

The NAR maintains two distinct buffers within the S-MIP architecture. The f-buffer contains packets forwarded from the PAR (f-packets) while the s-buffer contains packets that are marked with the *S* bit (s-packets). The NAR start delivering buffered packets to the MH once it receives the *F-NA* (Fast Neighbour Advertisement) message from the MH, signifying that the MH has arrived at its network. The PAR attempts to transmit the f-buffer and empties it before beginning to transmit from the s-buffer. Meanwhile, at the PAR, it only forwards those packets, which do not have the *S* bit marked to the NAR, and all packets are *not* sent on its wireless channel, during the SPS process. After the f-buffer has been emptied, the NAR sends the *Soff* message to the MAP indicating the termination of the packet simulcasting, that is, the ending of the SPS process.

Upon receiving the *Soff* message from the NAR, the MAP performs the binding update, associating the new (on-link) address of the MH with its (regional care-of) address. The MAP also forwards the *Soff* message to the TAP, signifying the completion of the handoff process. It must be noted that the TAP will not allow MHs

---

[14] The ambiguity exists in the HMIPv6 with Fast-handoff scheme, shown clearly with the sending of multiple F-*BAck* messages to all potential ARs.

to perform another handoff before the current handoff has been completed. As can be observed, the SPS mechanism consists of a coarse-grained packet (re)sequencing scheme. Since IP does not provide explicit packet sequencing mechanism, this separation between the two types of packets is necessary to minimise the problematic out of sequence packets from occurring at the network (IP) layer. It must also be noted that SPS is different from the concept of bicasting/n-casting [21]. The SPS mechanism is a structured approach in managing local retransmission and buffering as compared to simply duplicate packets to multiple 'likely' destinations.

3.3.2.2 Other Movements

For scenarios where the MH is moving stochastically (including ping-pong movements), the handoff procedure is the same as described previously for the linear case, until the beginning of the SPS process. In this mode of operation, the *HD* message will inform potential ARs to be in the anticipation-mode. Even though a MH might no longer wish to be associated with an AR, the AR still maintains the MH's binding, in preparation for the returning of the MH (ping-pong effect due to physical movements). This avoids the unnecessary re-setup overhead. The subsequent delivery of the HN message to the MH, from the PAR, will indicate that the MH is able to switch network freely, using the *F-NA* message, once the signal strength has decreased to a certain predefined threshold. The TAP may send further *HD* message to any of the participating ARs in cases where it determines that they are no longer required in the anticipation-mode. Meanwhile, the MAP simulcasts packets to *all* potential NAR identified by the TAP (indicated via the *Scast* message). Until further *HD* messages from the TAP to any of the participating ARs, the packet simulcast will continue even if the f-buffer of the NAR has been emptied.

For scenarios where the MH is determined to be in the stationary state near the centre boundary between two access network coverage areas, the *HD* message from the TAP will instruct the action of multiple bindings as potential ping-pong effect due to rapid signal variation may occur extremely frequently. The handoff procedure is similar to the stochastic case, except that the MH now binds simultaneously to more than one AR by using multiple CoAs (technique described in [30]).

3.4 Movement Tracking and Handoff

Movement tracking is a two-phase process consists of location tracking and followed by movement pattern detection. It determines the MH's movement type, linear, stationary or stochastic. More significantly, this is achieved by using only readily available infrastructure, i.e. access routers, in performing the location tracking. No additional reference points, pre-measurements or supplementary hardware devices (e.g. GPS) or infrastructure, such as [65], [5] and [49], are required. We assume the use of 802.11 (WiFi) access network technology, and a minimum of three access routers (base stations) in the S-MIP architecture. The tracking process is selective, i.e. it only executes continuously once a MH enters within predefined zones, where movement pattern detection is required for the purpose of the handoff management.

3.4.1 Coverage Areas and Zone Definitions

The coverage area of 802.11 can be defined in terms of Signal Strength (SS). Spatial/temporal and/or noise/fading issues are not considered in S-MIP. Briefly, when a MH is near to an AR, the SS value from the AR is strong, the link quality is high, and the probability of loosing packet is very low, usually zero [70]. When the

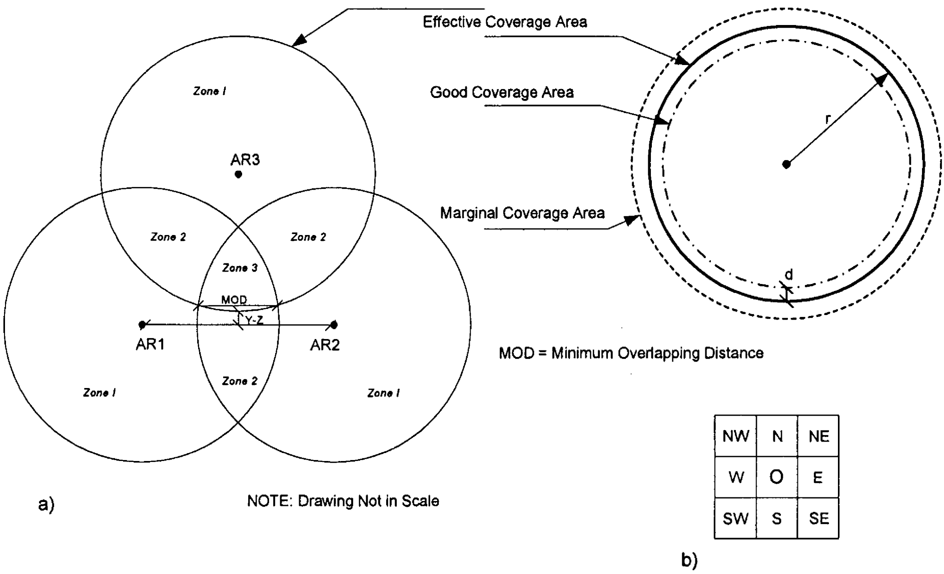


Figure 3-4 Coverage Area and Movement Direction Models

MH moves further away, the SS decreases. This can be modelled using a negative log function curve. In S-MIP, the coverage areas of each AR is divided into three different areas, shown in Figure 3-4a. First, referred to as the *effective* coverage area, has high signal strength and results in no packet loss in transmission. The second is called the *marginal* coverage area. It corresponds to the region outside the effective area where a low percentage of packet loss (below 5%) is maintained. The third is referred to as the *good* coverage area which lies within the effective coverage area. In zoning definition, for any overlapping set of three ARs, we specify that the good coverage area intersects at a single point (in theory). Hence, the three different zones formed are: *Zone 1*, where the MH can only receive signals from only one AR, *Zone 2*, where the MH is able to receive from two different ARs, and *Zone 3*, where the MH can receive from all three ARs.

We introduce a concept called the *Minimum Overlapping Distance* (MOD). It can be expressed as a function over the radius of the effective coverage area, *r*, and the overlapping distance, *d*, between any two ARs (see Appendix A). Here the overlapping distance, *d*, is defined as the difference in radius between the good and effective coverage areas (see Figure 3-4a). It is known that the attenuation factor due to human obstruction or device orientation is 6.4dB and 9.0dB respectively [70]. Therefore, assuming that attenuation can be approximated by a negative log function, 6.4 dB equates to at least 5 meters in distance and thus a *d* value, i.e. the radius difference of 2.5 meters. Consequently, we assume that *d* is at least 2.5 meters in the S-MIP architecture. A *d* value of 2.5 meters and *r* value of 25 meters will result in a MOD of 7 meters, see Appendix C for detail calculation. Furthermore, by varying the value of *r* does not change the MOD value between two ARs, i.e. given typical WiFi coverage ranges from r = 25 meters to r = 50 meters, the difference in MOD is only 0.3 meters[15]. Thus, given our previous assumption of slow average movement speed of 1 meter/second, even with the worst case scenario, where the MH is moving linearly across the MOD, the MH should still be located near the centre of Zone 2. Strategically, this is the best point in making handoff decisions. A minimum of three samples can still be gathered, since we are sampling location tracking information once a second, and is thus sufficient in movement patterning (describe in Section 3.4.3).

3.4.2 Location Tracking

Location tracking begins as a MH enters Zone 2 or Zone 3. In identifying the location of a MH, we use the triangulation calculation method, based on technique

[15] In fact, in our simulation (next chapter), we set *r* to 40 meters to validate MOD's impartiality from the *r* value.

described in [70], which is accurate to within 1 to 2 meters, requiring only 3 different SS values from each of the ARs.

In Zone 2, a MH receives two SS signals from two different ARs, i.e. AR1 and AR2 in Figure 3-4a. Thus location tracking mechanism is activated upon the TAP receiving the first *CTS* message that contains the SS values of the two access routers from the MH. The MH updates the TAP with the CTS message every second. In this zone, a specific location cannot be determined precisely because two effective signal strength distance circle intersect at two points. It is not possible to differentiate one from another without an auxiliary reference. Nevertheless, the 'top' portion of the combined coverage area is likely to be partially covered by AR3. Therefore, if the MH is located near this area, three SS values can still be obtained. In cases where a MH is located near the centre of Zone 2, the exact location can also still be determined using Marginal Coverage Area Inference calculations. Given a r value of 25 meters and d value of 2 meters, the required overlapping distance (Y-Z in Figure 3-4a) by the marginal coverage is approximately 9.5 meters (see Appendix B). Given that marginal coverage is at least equal to that of the effective radius, i.e. 25 meters (assuming SS exhibits negative log function behaviour), it is more than sufficient in covering 9.5 meters. Hence the marginal coverage crosses the centre of Zone 2 as defined in Figure 3-4a as line AR1-AR2. Thus, there are no 'dead corners' inside Zone 2, and location tracking is achievable anywhere inside Zone 2, with a worst case scenario of 95% accuracy, as the marginal coverage is defined at a loss of 5% or less.

3.4.3 Movement Pattern Detection

The accuracy of movement pattern detection depends on the sampling period and the MH moving speed.

To reiterate, in the S-MIP architecture, it is assumed that the MH moves with a slow average speed of 1 meter/second, and the sampling period is defined to be one per second which begins with the reception of the first *CTS* message (send by MH) to the TAP entity. Movement directions are based on the matrix shown in Figure 3-4b. The centre, O, indicates the current location of a MH and the remaining eight squares illustrate the next movement direction of the MH in terms of North(N), South (S), East (E), West (W), North East (NE), South East (SE), South West (SW), and North West (NW). The direction is calculated by firstly determining the location position using SS values and then the differential of the location position at two different time intervals. Once the direction of movement is known, the movement patterning can be established by looking at the history of a series of directions in which the MH has moved. For example, a set of repeated direction indicates linear movement, while a set of repeated location indicates non movement (stationary), that is, the MH is not deviating from position O. To determine if the MH is stationary near the boundary of two overlapping coverage area, apart from determining to be stationary, the SS values from both ARs must be similar. Anything that is not linear or stationary can be considered to represent the stochastic movement. Under these conditions, a minimum of only three samples can suffice in establishing the movement pattern for the worst case scenario, travel linearly over MOD, where the potential period for sampling procedure is the shortest (see Appendix C).

In must be noted that movement pattern detection mechanism in the S-MIP architecture maintains a 'soft-state', meaning, the history of movement direction stored is merely part of the handoff process, and is discarded after the handoff. This is critical when considering network scalability and performance issues, particular for the TAP entity. The states are kept 'alive' by the MH's periodic *CTS* messaging during a handoff process, analogous to the approach taken by Mobile IP in its registration management scheme.

3.5 Discussion

In what follows, we discuss outstanding issues which we deliberately omit for clarity of presentation of the S-MIP architecture previously.

3.5.1 Issues with L2 triggers

Current advanced Mobility Management schemes [33], [56] assume the initiation of handoff process, by the MH, is activated through what is known as the L2-trigger. The argument being that, as part of the L2 handoff mechanism, inter-layer messaging, i.e. L2 trigger can be used to pre-empt higher layer (L3) to also start preparing for or even perform a handoff. There are two issues with regards to the nature of L2 handoff which we would like to address. Firstly, can the L2 trigger be considered as an example of the system dominant design? We argue that it is simply a reactive 'add-on' mechanism to the L2 handoff mechanism. If it is to be considered as being system dominant, it would have to include 'instructions' for L3 handoff, and not just an indication to prepare for potential L3 handoffs. If the L2 trigger is redesigned to be system dominant, it would still be undesirable because of inter-layer communication and complexity problems.

Secondly, we are skeptical in the scalability, pertaining to access network dependency, of such scheme given that the MH needs to understand explicitly the L2 triggering mechanism in use – the End-to-End argument has been violated with this approach. While we concur with the merit of pre-empting handoff setup at L3 in gaining better performance, we believe such indication mechanism is better achieved with the S-MIP example, that is, preserving the horizontal independency through the use of location tracking at selective periods during a handoff. With the S-MIP architecture, interfaces for location tracking are defined from the network's perspective, meaning, what is required by the higher layer (L3) and what needs to be provided by the access layer (L2) are identified. Thus the architecture decouples access network technology from higher layer protocols, and making it more scalable over different types of access network. Although we have demonstrated a simplistic model on how location tracking can be achieved in large free-space environments, more sophisticated modelling and location identification mechanism can be 'plugged-in' to facilitate the tracking infrastructure, as long as it satisfies the AR's requirement in *CTS* message generation.

3.5.2 Optimisation Techniques of S-MIP Revisited

Unlike schemes such as the HMIPv6 with Fast-handover, the PAR does not forward any packets (in general) onto the wireless channel during the SPS process. In this section, we detail the rationale behind this.

Suppose the following (see Figure 3-5), during a linear handoff, the PAR forwards packets which are not marked with an *S* bit in the IP option parameter to the NAR (f-packets), as well as sending a copy (f-packets) onto the wireless channel during the SPS process. The forwarding of packets onto the PAR's wireless channel is to allow the MH to still receive these packets, in case it does not switch networks immediately [56]. Therefore the worst case scenario is one where the MH receives all the f-packets while still in the access network of the PAR, but again receives all the forwarded f-packets (from PAR to NAR), when arriving at the NAR's access network. In the case of reliable end-to-end communication, received packets that are out of sequence (in reality duplicated packets) are interpreted as errors. In terms of TCP, duplicated acknowledgements will be sent by the receiver causing source throttling and maybe even retransmission. A simple solution is simply *not* to send the f-packets onto the wireless channel.

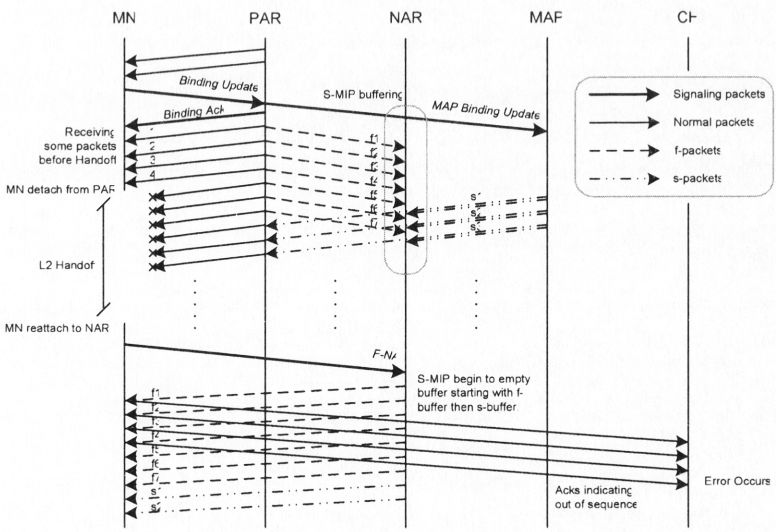


Figure 3-5 Problem of Packet Forwarding onto current access network

In other words, upon receiving the *F-BAck* message from the PAR, the MH must immediately switch to the NAR. Furthermore, the PAR must immediately forward packets to the NAR after receiving the *F-BU*. Therefore, the PAR only needs to send the *F-BAck* message to its own access network. Thus the possibility of receiving duplicated packets is reduced to zero. Indeed, unlike schemes such as HMIPv6 with Fast-handover or Simultaneous Bindings, once a handoff decision has been received by the MH, the handoff must take place in S-MIP.

Consider the case for stochastic handoff, where the SPS is forwarding packets marked with *S* bit to multiple ARs, and the MH is allowed to switch to any of the AR if the conditions deem to be necessary, i.e. weak signal strength with the current AR. In this case, packet duplication is formidable. Thus an IP packet 'filtering' mechanism at the NAR must be considered. This is to prevent receiving duplicated packets from different NARs. Essentially, a matching algorithm that compares IP packets within the s-buffer and the f-buffer at the NAR is required to discard any identical packets inside the s-buffer. One way to proceed in such comparison is to examine the 16 bit IP identification, fragment offset and flag (used in conjunction with the fragment offset) fields in the IP packet header. In this way, one can be sure that even if IP packets have been fragmented on the traversing path, they can be uniquely identified and therefore be compared and discarded where appropriate. The implementation details are out of the scope of this thesis.

3.5.3 Scope of TAP domain

A reasonable scope of a TAP's domain in the S-MIP architecture is one that exemplifies conventional halls or airport terminals, that is, single organisation/authoritative body operating within a confined and isolate physical environment. If a MH moves out of a domain, it cannot expect the availability of the S-MIP service. However, if a single domain requires more than one TAP to sustain the location tracking ability, then the problem of synchronising all the TAP to a common view, or the problem of handoff between the TAP sub domains, will be similar to what HMIPv6 faces with MH moving in-between the MAP domains. These issues are out of the scope of this thesis.

3.5.4 S-MIP Architectural Validity

The S-MIP Architecture is based on two fundamental observations. The first is that all f-packets are unlikely to be received at the NAR before any s-packets are received. The second is that the PAR is unlikely to start forwarding f-packets to the NAR before the MH switches to the new access network. A formal justification of these observations is given in Appendix D. These observations imply that it is necessary to provide a mechanism for packet re-sequencing and buffering, i.e. the SPS mechanism.

3.5.5 Pro-collaborative Nature

In designing the nature of the handoff management scheme, four different possible variations can be conceived (see Table 3-1). They are i) network initiated network determined handoff, ii) network initiated mobile (host) determined handoff, iii) mobile initiated mobile determined handoff, and iv) mobile initiated network determined handoff.

First, let's decipher the nomenclature surrounding these. The term 'mobile' is analogues to the end user while the term 'network' is analogous to the computer and/or communication systems. The term 'determined' has the connotation that the entity, either network or mobile, is being active, or even proactive, in performing the handoff decision while the other counterpart assuming the reactive role. Thus, for both *network initiated mobile determined* and *mobile initiated mobile determined* cases, it is clear that the user is being the active part, as well as taking on the dominant role in an interaction. For the *network initiated network determined* case, it is also clear that the system is the dominant process. However, it is with the subtlety for the *mobile initiated network determined* case, the interaction becomes an interesting one. While network determined suggests system dominant design, without the trigger or the initiation process of the user, the interaction is incomplete. This, as mentioned previously, is referred to as the Indicative mobility management approach, and requires the co-functioning of both the user and the system in completing an interaction.

In reference to design forms, from the perspective of the network, both *network initiated network determined* approach and *mobile initiated mobile determined* approach illustrate the interactive design form, with the emphasis being system dominant for the first case and user dominant for the second case. Being 'network

Table 3-1 Design Approach Deconstructed

Design Variations	*Example*	*Design Emphasis*	*Design Form*
Network Init. Network Det.	GSM	System Dominant	Interactive/Proactive
Network Init. Mobile Det.	H/F MIP*	User Dominant	Proactive
Mobile Init. Mobile Det.	MIP	User Dominant	Interactive
Mobile Init. Network Det.	S-MIP	Balanced	Pro-collaborative

* H/F MIP = Hierarchical Mobile IP and Hierarchical Mobile IP with Fast-handover

initiated' in the case of *network initiated network determined* approach also suggests proactive design. However, this is framework specific. On the other hand, for the *network initiated mobile determined* approach, we categorise it under the general concept of proactive design form as in order for network to initiate an interaction, certain level of intelligence must be inherent. Finally, the *mobile initiated network determined* approach exemplifies the principle of pro-collaborative form.

Examples for user dominant designs, i.e. *mobile initiated mobile determined* approach and *network initiated mobile determined* approach include, respectively, standard Mobile IP and Hierarchical Mobile IP or Fast-handover schemes. Example for the case where the handoff is *network initiated network determined* would be GSM. Lastly, S-MIP is an example of the *mobile initiated network determined* approach.

What if the *network determined* aspect suddenly failed, i.e. while assuming the dominant role, the system fail to respond? The worst case scenario conceivable is merely to revert back to the *mobile initiated mobile determined* user-dominant situation. In other words, the system moves from the pro-collaborative form back to the proactive form. In the context of S-MIP, this means that the handoff optimisation would drop from seamless handoff to what is offered by the usual Hierarchical Mobile IPv6 with Fast-handover, if the TAP ceases to function. Such non-obtrusive nature of the S-MIP scheme on current protocol also means that, if the mobile user moved out of the 'S-MIP enabled' access network, s/he would simply resume the use of the normal HMIPv6 with Fast-handover or just HMIPv6 protocol, depending on the access network configuration.

3.6 Summary

In this chapter, we presented in detail a system architecture that is based on the pro-collaborative design form. In keeping with the theme of the notion of a nomadic user, we focus on an IP layer solution that offers seamless connectivity during a handoff. Our approach, termed Indicative Mobility Management, is one where it requires the collaboration of the network system and the mobile user in handoff management. A complementing location tracking mechanism, assisting in 3 different handoff strategies, is also proposed.

In the next chapter we compare the handoff performance of the S-MIP architecture with various leading IETF mobileip WG proposals. In particular, we examine the use of TCP as the target transport protocol for this research.

Truth is generally the best vindication against slander.

- Abraham Lincoln

Chapter 4

S-MIP Performance Evaluation

Handoff latency results in packet losses and severe End-to-End TCP performance degradation as TCP, perceiving these losses as congestion, causes source throttling or retransmission. In order to mitigate these effects, various Mobile IP(v6) extensions have been designed to augment the base Mobile IP with hierarchical registration management, address pre-fetching and local retransmission mechanisms.

In this chapter, we comprehensively evaluated the impact of layer-3 handoff latency on End-to-End TCP for the S-MIP architecture. For reference, we also compare S-MIP with various leading IETF Mobile IP(v6) extension proposals. In all, five frameworks are compared with the base Mobile IPv6 framework, namely, i) S-MIP, ii) Hierarchical Mobile IPv6, iii) Hierarchical Mobile IPv6 with Fast-handover, iv) (Flat) Mobile IPv6 with Fast-handover, and v) Simultaneous Bindings. We propose an evaluation model examining the effect of linear and ping-pong movements on the handoff latency and the TCP goodput, for all above frameworks. All performance evaluations are simulation based.

4.1 S-MIP Signalling Cost Analysis

Before presenting the performance comparison of all the frameworks mentioned, we first give a theoretical account on the signalling cost of the S-MIP architecture. The signalling cost associated with S-MIP can be divided into two parts. The first part is associated with the setting up of the handoff while the second part is associated with movement tracking, in particular, the *CLS* (Current Load Status) and *CTS* (Current Tracking Status) messaging mechanism.

Firstly, as can be seen from the handoff example in the previous chapter, the functional role of the *RtSolPr* (Router Solicitation Proxy), the *HI* (Handover Initiation), and the *HAck* (Handover Acknowledgement) messages remain unchanged from the Hierarchical Mobile IPv6 with Fast-handover scheme (see Figure 3-3). The *PrRtAdv* message needs extension to contain the *HN* (Handoff Notification) information, though this only incurs processing overhead, therefore is negligible. Thus, the S-MIP architecture has an equal handoff setup signalling cost compared with the Hierarchical Mobile IPv6 with Fast-handover scheme.

Secondly, with *CLS* messaging, additional signalling is introduced as the access routers need to reply to the modified Router Advertisement with the TAP option, approximately once every 3 seconds (defined by the MinRtrAdvInterval in [44]). With the *CTS* messaging, additional signalling is also introduced since the mobile host is required to send the *CTS* messages to the TAP via the current access router. However, in the wireless segment of the S-MIP architecture, there is *no* additional signalling overhead when compared with Hierarchical Mobile IPv6 with Fast-handover. The two stage process in *CTS* message generation prevents explicit signalling by the mobile host. The tracking information (SS values and AR Id) is sent as piggybacks via the access layer (L2) messaging. Therefore, the 'real' overhead occurs once a second between the access router(s) and the TAP (Tracking Anchor Point), which nevertheless only takes place during the handoff period. By restricting the signalling overhead within the wired segment, S-MIP minimises the overall system performance penalty. After all, the wired segment of the network, in comparison to the wireless, has far more and cheaper resources.

4.2 Simulation Model

The goal of our simulation is to examine the effectiveness of the S-MIP architecture, in comparison to various Mobile IP extension schemes, on L3 (IP layer) handoff latency reduction over reliable end-to-end communication (i.e. TCP). In particular, we are interested in examining the bulk data flow rather than the interactive data flow scenario, since bulk data flow is more prone to disruption during a handoff. The security aspect, such as AAA (Authentication, Authorisation and Accounting) [72] and encryption methods [41], [48], is out of the scope of this comparison.

4.2.1 Implementation Details

This section describes the protocols we have implemented, as the extensions to the Network Simulator version 2 (*ns-2*) [85], in facilitating the performance comparison. These protocols include Mobile IPv6 (MIPv6) [30], Hierarchical Mobile IPv6 (HMIPv6) [56], Fast-handover [33], Simultaneous Bindings [21] and S-MIP. The base *ns* distribution ns-allinone2.1b7a was selected as our simulation platform. It was patched with the '*ns* wireless extension' module [68], to allow basic

Mobile IP(v4) protocol operation. It was then further extended with our set of protocol implementations.

For the Mobile IPv6 protocol suite, we did not implement a complete set of proper IPv6 features as this is unnecessary for our simulation purposes[16]. In respect to our simulation, the key differentiator between Mobile IPv4 and Mobile IPv6 is the Binding Cache Management. This includes the IP Destination Option, which is necessary to support the Home Address Option. We implemented these on top of the existing registration, packet encapsulation/decapsulation mechanisms on the *ns* wireless extension module. No security mechanism was implemented for the binding cache update.

For Hierarchical Mobile IPv6 protocol suite, a MAP (Mobility Anchor Point) Agent was implemented to provide the MAP registration functionalities. The MAP agent only supports the use of its address as the RCoA (Regional Care-of Address) for the mobile host (MH). A simplified MAP discovery mechanism was created to enable the MAP's RCoA to be discovered by the MHs. The Mobile IP router advertisement (beacon) message was also modified to include the MAP advertisement option. The schematic of a MAP Node is depicted in Figure 4-1.

For the Fast-handover protocol suite, we made a slight modification to the Node entity in *ns* to facilitate the use of the encapsulator/decapsulator provided by the wireless extension module. This enables tunnel mechanism (IP encapsulation in IP [46]) to be setup between all types of node in *ns*, namely, wired, wireless and hybrid wired-wireless nodes (i.e. the BaseStation Node entity). Further, the messages

[16] It must be noted that we did not implement, in full, all the protocols mentioned in this chapter. Rather, we only implemented what is necessary for our simulation purposes.

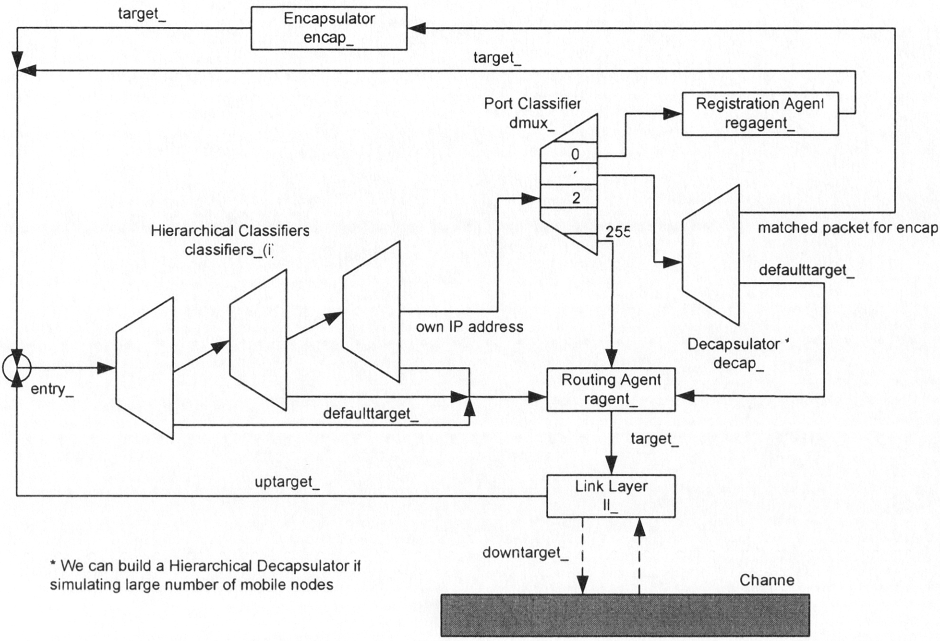


Figure 4-1 Schematic of a Mobility Anchor Point (MAP) Node

required by the Fast-handover, that is, the *PrRtAdv*, the *RtSolPr*, the *HI*, the *HAck*, the *F-BU*, the *F-BAck* and the *F-NA* messages were also added. The BaseStation Node in *ns-2* was further modified to handle these messages.

For the Simultaneous Bindings protocol suite, we are only interested in the case where the MH binds with the MAP entity. Other cases, i.e. binding with the HA (Home Agent) or the PAR (Previous Access Router), are not considered in our comparison. Therefore, we implemented purely the operations required for the MAP binding case. The bicasting/n-casting mechanism was added to the MAP Node.

For the S-MIP protocol suite, we implemented the TAP Agent which collects the MH position information and makes handoff decisions based on movement pattern

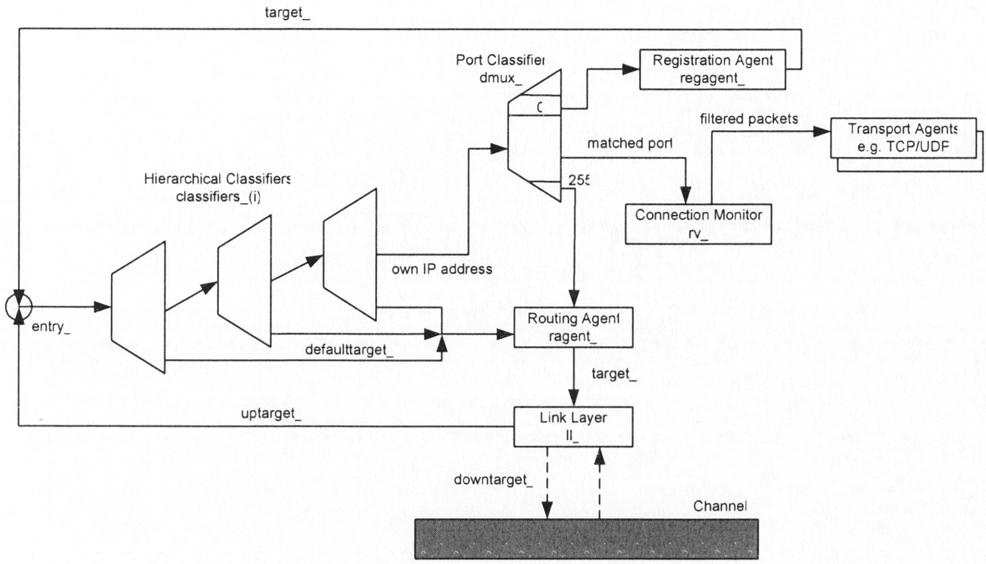


Figure 4-2 Schematic of Mobile Node

identification. The positioning in this case can be easily calculated using the Node class's getLoc() support (not necessarily with the signal strength value). Also, all new messages associated with the S-MIP protocol were implemented in full. The support for packet simulcasting (SPS) was also included, consisting of the s-buffer, the f-buffer, and the forwarding mechanisms.

Apart from these protocol extension implantations, two other important modifications were necessary on our *ns-2* simulation platform. Firstly, the WaveLan implementation of the current *ns-2*, was conceived through the Monarch project [81], and was mainly developed for the simulation of wireless ad-hoc networks (broadcast mode). What is required for our purpose is the support for the 'infrastructure mode'. While a new implementation of the complete 802.11b/g standard [78] would be ideal, we opted for a simpler 'emulative' solution. This is made possible by a new CMon

(Connection Monitor) entity. It is inserted between the port classifier (dumx_) and the receiving agent(s) (e.g. TCP/UDP) connecting to specific port(s) of the port classifier (dmux_), see Figure 4-2. The CMon controls the MH's reception of packets of a communication flow, e.g. TCP. When the MH starts a L2 handoff to the new access network, CMon is set to drop any received packets until the L2 handoff is complete. With this, we are able to treat the receptions of control messages, (e.g. periodic beacons from ARs which arrive at the registration agent (regagent_) by passing the CMon) as if operating in the periodic channel scanning mode. On the other hand, the CMon emulates the channel change, which occurs when the MH is switching between two access networks, by blocking the agents from receiving any packets during this period. We assume that adjacent access networks use different frequency channels. Another advantage of using the CMon is the flexibility to 'identify' communication flows. For instance, if operating in the route optimised mode where the MH is required to perform a binding update to the corresponding node, the MH is able to query the CMon about the identity of any corresponding node per communication flow. See Figure 4-2 for the schematic of the Mobile Node.

Secondly, we implemented an additional handoff algorithm/strategy, that is, 'midway handoff', to facilitate the comparison experiments. Since we know that for S-MIP, a MH is likely to perform the handoff around the centre dividing boundary between two access networks, the purpose of the midway handoff algorithm is to ensure that all other framework handoff near the time vicinity of S-MIP. This is to provide a more accurate and consistent comparison. The midway handoff is defined as such that a MH must handoff to an access router 'closer' to itself, determined via

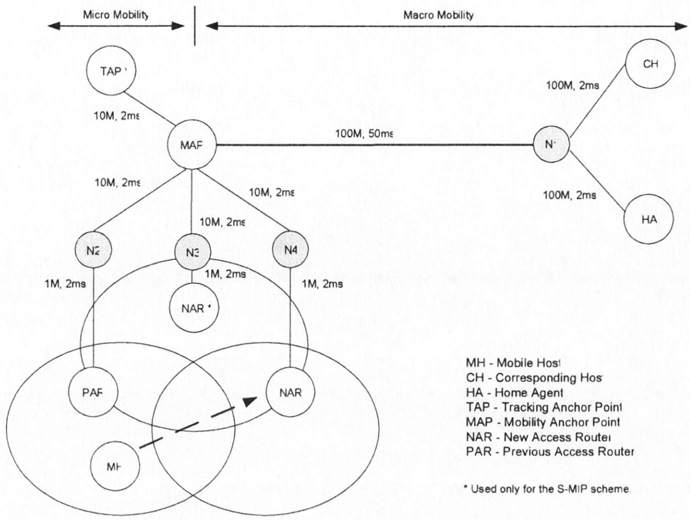


Figure 4-3 Simulation Network Topology

the reception of the beacon message. This handoff strategy is especially tailored for the purpose of the ping-pong movement experiments.

4.2.2 Simulation Scenarios

Figure 4-3 illustrates the network topology used for the simulation experiments. This topology reflects the setup of an open space local environment (where the MH is situated) connecting to a distant home network. Both Corresponding Host (CH) and the Home Agent (HA) are connected to an (dummy) intermediate node (N1) with 2 milliseconds (ms) delay, 100 Megabits/s (M/s) links. The link between N1 and the MAP is a 100M/s link with 50ms delay. This simulates the distant home network (macro mobility). Below the MAP is considered to be the 'local' network (micro mobility). The MAP is connected to 3 intermediate nodes (N2, N3, N4) with 2ms delay, 10M/s links. The N2 is connected with the PAR while others to NARs, all with 2ms delay, 1M/s links. All links use the RED (Random Early Detection) queue, except links from the intermediate nodes to the ARs (both NAR and PAR), which are

Droptail (FIFO) queues. The access routers are set to be 70 meters apart with free space environment in between. This reduces the complexity of results analysis, as we only need to consider signal interference. We assume the use of 802.11 as our access technology and the effective coverage area is set to 40 meters in radius. An *ns* TCP source (Tahao) agent is attached to the CH and an *ns* TCP sink agent is attached at the MH. The TCP packet size is set to 512 bytes and the window size is 32. A bulk data transfer application is attached on the established TCP link that transfers packets from the CH to the MH, 5 seconds after the start of the simulation. The total duration for the simulation is 80 seconds.

We model two movement scenarios, namely, the MH moving linearly between two access networks and the MH moving in a back and forth (ping-pong) fashion near the midway between two access network coverage areas. For the linear case, the MH starts moving towards the NAR from the PAR 10 seconds into the simulation, at the speed of 1 meter/second (approximating human walking speed). It stops at the NAR when simulation time reaches 80 seconds. The corresponding handoff algorithm used in this scenario is the 'priority handoff'[17], where the MH switches from one AR to another AR if the priority (contained in the beacon message) of the new AR is higher than that of the current one. We favour this handoff strategy over midway handoff to minimise unexpected interferences and to maintain the consistency as to when a handoff will take place, for comparison purposes. We set the beacon priority of the NAR to be higher than the PAR, hence the handoff should occur when the MH receives the first few beacon messages from the NAR at around 40 seconds into the simulation. For the ping-pong case, we define the movement of

[17] The priority handoff is included with the *ns* wireless extension module.

the MH to be similar to that of the linear case, except that at 46 seconds into the simulation (just past midway), the MH reverses its direction of movement back towards the PAR. It reverses its direction every 2 seconds thereafter, until finally heading towards the NAR again, 52 seconds into the simulation. (See the stochastic scenario in Figure 3-2 to visualise this ping-pong movement.) The corresponding handoff algorithm used in this scenario is the 'midway handoff' algorithm. Finally, in all simulations, we consider the simplistic case with only one MH initiating one single connection at a time.

4.3 Experimental Results

The experiments were separated into two groups, namely, the linear movement experiments and the ping-pong movement experiments. We first simulated all the frameworks mentioned, namely, MIPv6, MIPv6 (flat) with Fast-handover, HMIPv6, HMIPv6 with Fast-handover, Simultaneous Bindings, and S-MIP, with the linear movement scenario. Following this, we simulated all frameworks with the ping-pong movement scenario. We set the L2 handoff time to 20ms and address resolution time to 100ms for all experiments[18]. The address resolution time here refers to the time taken for a MH to obtain a new care-of address. We measured the TCP goodput, delay and the *cwnd* (TCP congestion window) values for all simulations.

4.3.1 Hierarchical Mobile IPv6

This section describes the handoff latency comparison between HMIPv6 and MIPv6. In Figure 4-4, the *source_send_mip* and the *source_send_hmip* curves (the

[18] Currently, we lack statistical L2 handoff values to formulate a L2-handoff delay distribution. Therefore, we are unable to provide a more realistic simulation except to fix the L2 handoff time. Same applies to address resolution time.

lower two curves) illustrate the handoff delay for the MIPv6 and the HMIPv6 respectively from the perspective of source's (CH) sending sequence. In the MIPv6 case, the time between the first retransmitted packet sent by the source and the last time this packet was sent is labelled with *a*, in Figure 4-4. It is approximately 747ms. This must *not* be used as a measure for the overall handoff delay. We need to add the time required for this packet to reach the MH. At least an additional 60ms is necessary (five 2ms delay link segments, one 50ms link plus wireless link delay, see Figure 4-3), as we are operating in non route-optimised mode, where packets from the CH are sent to the HA first before being routed to the MH. The total delay is in fact 814ms. In this chapter, we use this duration as a standard measure for all frameworks analysed. We refer to this duration as delay *D*.

In the HMIPv6 case, label *b* depicts the L2 handoff duration (happening at the MH) while label *c* depicts the address resolution period. Subsequently, the MH performs the MAP binding update (label *d*). After the successful binding, the MH receives out of sequence packets, and thus sends acknowledgement (*ack*) messages, containing the expected sequence number, back to the CH. This period is marked with label e, and consists of the time for *ack* messages to traverse from the MH to the MAP, the time from the MAP to the HA, and the time from the HA to the CH. After the CH receives three such *ack* messages, retransmission starts. Note that the TCP enters slow start due to duplicated *ack* messages received previously. The *D* value for this case is 326ms.

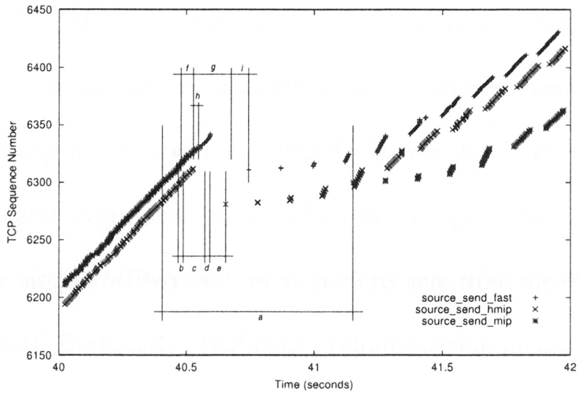


Figure 4-4 Comparison between MIPv6, HMIPv6 and MIPv6 (flat) with Fast-handover

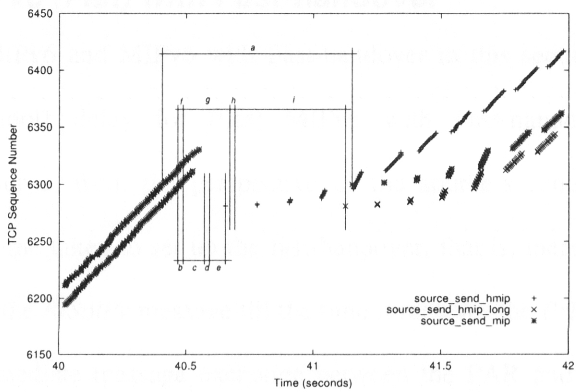


Figure 4-5 Comparison between MIPv6, HMIPv6 and HMIPv6 with longer address resolution

Figure 4-5 illustrates the effect that the address resolution time has on the handoff latency. The *source_send_mip*, the *source_send_hmip*, and the *source_send_hmip_long* curves illustrate the handoff delay for MIPv6, HMIPv6 with 100ms address resolution delay and HMIPv6 with 200ms address resolution delay respectively. Labels *a* through *e* are the same as in Figure 4-4. Label *g* depicts the 200ms address resolution period, while labels *f* and *h* represent the L2 handoff and the MAP binding update. Compared with the approximately 60ms delay for the previous case (label *e* in Figure 4-4), there is an approximately 450ms delay period (label *i* in Figure 4-5) before the retransmission starts. This is because the delay (mainly due to address resolution) in obtaining a care-of-address means that the MH is unable to receive any packets, which eventually drives the TCP sources (CH) to reach zero window and stop transmitting. When this happens, the performance is similar to that of the simple MIPv6 case, as all packets sent from the offered window are lost due to handoff. Eventually, TCP enters retransmission mode, similar to that of the MIPv6 scheme.

4.3.2 Mobile IPv6 (Flat) with Fast-handover

We compare MIPv6 and MIPv6 with Fast-handover in this section. Figure 4-4 illustrates the handoff delay for (flat) MIPv6 with Fast-handover with the *source_send_fast* curve, from the perspective of the source's sending sequence. Label *f* shows the time taken to set up the fast-handover, that is, measured from the time the MH sends the *RtSolPr* message till the time it receives the *PrRtAdv* message, including the intermediate message exchange between the PAR and the NAR (*HI* and *HAck* messages). As can be observed in Figure 4-4, at around the 40.5 second mark, the *source_send_fast* sends a little longer than the *source_send_mip* before being interrupted by the handoff. This is because the setting up of fast-handover does not prevent TCP from continuing packet transfer. A TCP flow only starts to be disrupted once entering the binding process. The binding update (with HA) duration is depicted using label *g*, while the L2 handoff is through label *h*. We switch network (L2 handoff) immediately after sending the binding update, for all schemes employing the Fast-handover mechanism, in consideration of consistency and easy comparison purposes. This is because that the double *F-BAck* response (one to the

NAR and one to the PAR) make the reception of the *F-BAck* message ambiguous, if not controlled.

Similar to the MIPv6 case (label *e*), label *i* depicts the delay of the *ack* message traversing from the MH to the CH. The binding update delay is proportional to the round-trip delay between the CH and the MH (traversing via the HA in our case). This is the weakness of operating Fast-handover in the flat Mobile IP environment. Even through that the forwarding mechanism is delivering packets from the PAR to the NAR, the MH is unable to receive these until the binding update is complete, signified by the sending of the *F-NA* message to activate the data flow at the NAR. The *D* value for this case is 358ms.

4.3.3 Hierarchical Mobile IPv6 with Fast-handover

This section details the handoff behaviour for the HMIPv6 with Fast-handover scheme. In prior work [27], we illustrated the performance of a simple superimposition of HMIPv6 scheme with the Fast-handover protocol. It showed that superimposition results in a complex behaviour. In contrast, we illustrate the 'integrated' approach, outlined in the Appendix section of [56] here. The main difference being that the forwarding mechanism (part of the Fast-handover) is anchored at the MAP, instead of at the PAR. The 'integrated' approach is an optimised solution in comparison to that of the simple superimposition. Essentially, two notable events occur as denoted with box A and box B in Figure 4-6a. Firstly, packet loss happens when L2 handoff is initiated (box A). Secondly, the MH starts receiving out of sequence packets after switching to the new access network (box B).

Figure 4-6a shows the details of handoff from the TCP receiver's (sink) perspective. The receiver's receiving sequence is denoted by the *sink_recv* (data) curve while the sending sequence is denoted by the *sink_send* (ack) curve. Likewise, Figure 4-6b shows the details of the handoff from the TCP sender's (source) perspective. The sender's sending sequence and the receiving sequence are depicted by the *source_send* (data) and the *source_recv* (ack) curves respectively. The label *a* in Figure 4-6a depicts the time vicinity where the first instance of disruption in the TCP communication occurs, namely the L2 handoff. Some packets are lost as a result. Due to the Fast-handover forwarding setup between the PAR and the NAR, the MH is able to receive the forwarded packets (through the use of the *F-NA* message) at the new access network from the NAR shortly after arriving, as shown near label *b*. However, these packets contain out of order sequence numbers (higher than expected). As a result, an *ack* reply, containing the expected sequence number, is sent for each of the (out of sequence) data packet received (label *c*). This is reflected in the sender's receiving sequence (label *g*). These duplicated *acks* trigger the TCP sender to enter the congestion mode and slow start occurs (around label *h*).

The TCP slow start process continues until around label *d*. By this point in time, the 'missing' packets have all been received by the MH. Due to the accumulative acknowledgement, the MH's window moves a whole segment and 'opens' abruptly. This is reflected in the TCP sender near label *i*, as well as the corresponding *cwnd* curve shown at around 41.2 second (label *m*). However, packets sent previously by the sender (label *j*) still arrive at the receiver (label *e*). These are repeatedly acknowledged with the same expected sequence number (label *f*) by the receiver. Eventually, these *ack* messages arrive at the sender, but are discarded since the

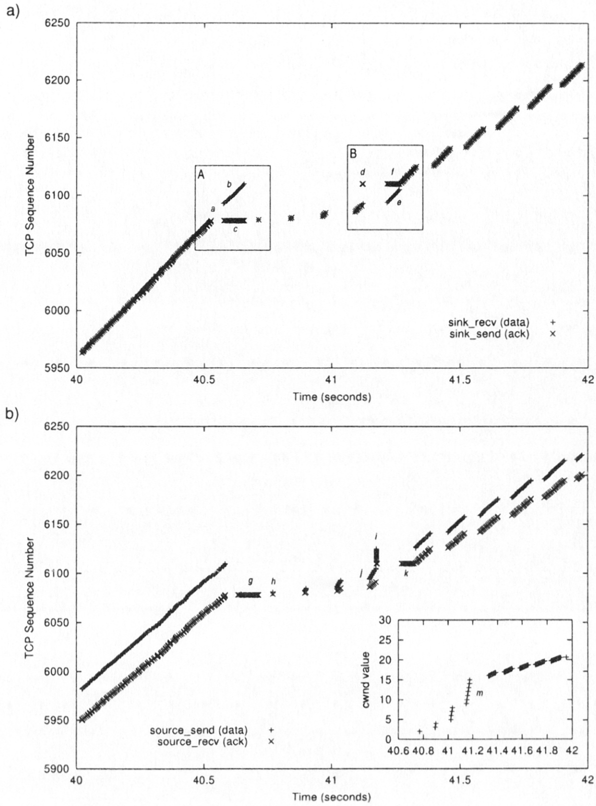


Figure 4-6 Handoff Behaviour for HMIPv6 with Fast-handover in detail

sequence numbers have already been acknowledged. The communication returns to normal operation after this. The *D* value for this case is 270ms.

4.3.4 Simultaneous Bindings and S-MIP

We compare the Simultaneous Bindings scheme and the S-MIP scheme in this section. A hypothetical result is also shown for comparison purposes where we deliberately created the situation where out of sequence (duplicated) packets are present at the NAR. This is represented by the *smip_nonop* (non optimised) curve,

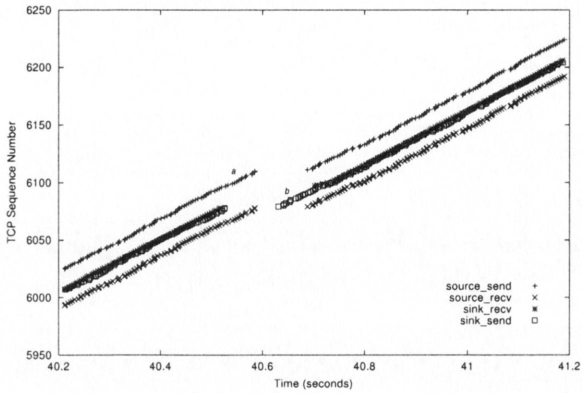


Figure 4-7 Handoff Behaviour for S-MIP in detail

and typifies situations such as sending s-packets to multiple NARs, during ping-pong movements.

As can be seen from the *source_send*, the *source_recv*, the *sink_recv*, and the *sink_send* curves in Figure 4-7, the S-MIP scheme performs exceedingly well, considering that the overall handoff perceived by the sender is a mere 100ms in duration approximately (*source_send* curve). There is no apparent packet loss perceived by the sender and therefore no congestion or flow control is required. The *D* value cannot be calculated since there are no packets being retransmitted. The emptying of the s-buffer and the f-buffer corresponds to *ack* message responses near label *b* (*sink_send* curve) in Figure 4-7. In other words, these acknowledgements, for data packets sent by the source (around label *a*), are simply delayed slightly in reply, due to the handoff. Subsequently, everything returns to normal as if no handoff had taken place.

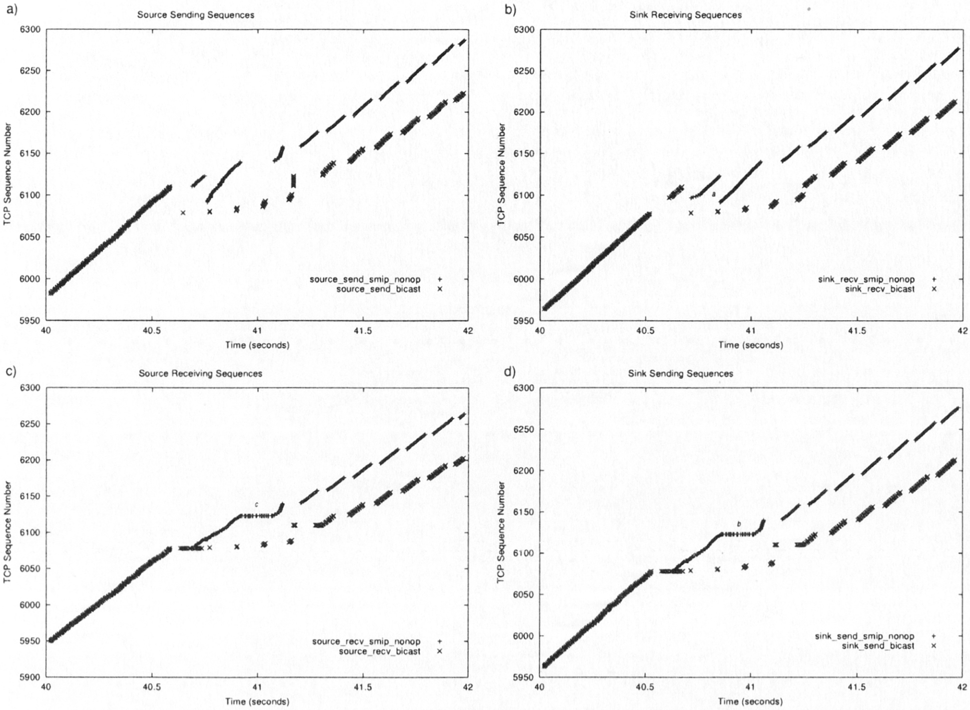


Figure 4-8 Handoff Behaviour for Simultaneous Bindings and S-MIP non optimised in detail

In contrast, the hypothetical worse case scenario for the non optimised S-MIP case is shown as the smip_*nonop* curves in Figure 4-8a to Figure 4-8d, where packet out of sequence takes place at the NAR. The *source_send_smip_nonop*, *sink_recv_smip_nonop*, *source_recv_smip_nonop*, and *sink_send_smip_nonop* curves illustrate this case. A few things can be noted in this scenario. Due to the coarse grain buffering mechanism of S-MIP, this is a case where no packets are lost during the handoff. However, we deliberately induce duplicated packets within the f-buffer and the s-buffer of the NAR, to cause out of sequence behaviour. This illustrates the effect of receiving out of sequence packets by the MH and how it may upset the TCP flow within the S-MIP architecture. Nevertheless, compared to packet loss, the penalty is not as severe in our experimental context, because of the accumulative acknowledgement mechanism. The two key handoff behaviours are, firstly, the receiving of the duplicated packets by the MH (label *a* in Figure 4-8b), and secondly, these duplicated packets cause the MH to send duplicated *ack* messages (label *b*). Fortunately, these have already been acknowledged previously, therefore are discarded by the CH (label *c*).

The handoff behaviour for Simultaneous Bindings is similar to that of the HMIPv6 with Fast-handover case, as can be seen from the *source_send_bicast* (Figure 4-8a), the *sink_recv_bicast* (Figure 4-8b), the *source_recv_bicast* (Figure 4-8c), and the *sink_send_bicast* (Figure 4-8d) curves, when compared with Figure 4-6. This is because the benefit of bicasting/n-casting have been 'eroded' away, as the MH is traversing linearly and switching network immediately after sending the binding update to the MAP. For the Simultaneous Bindings case, as the MH is switching immediately, the only subtle difference between it and the HMIPv6 with Fast-handover case, is that some packets 'inside' the link segment between the MAP and the PAR might be lost as a result of the network switch. However, since the packet forwarding delay between the PAR and the NAR is much smaller than the time taken to complete a L2 handoff, even if these packets are forwarded from the PAR to the NAR (as in the case for HMIPv6 with Fast-handover), it would nevertheless be dropped by the NAR, since the MH is unlikely to have completed the L2 handoff.

Even if the MH is to switch slightly later, in order to eliminate segment packet loss, some packet losses will still take place due to the L2 handoff, similar to that of the HMIPv6 with Fast-handover case. Although Simultaneous Bindings might 'cover-up' the timing ambiguity (refer to Chapter 2), we argue that the performance improvement is minimal, as packet loss is a substantial penalty. In comparison, as shown in Figure 4-8a to Figure 4-8d with the *smip_nonop* curves, the severity of packet loss is much greater than that of packets out of sequence. For this reason, the performance for Simultaneous Bindings and HMIPv6 with Fast-handover would be similar. The *D* value is 268ms for the Simultaneous Bindings scheme.

4.3.5 Ping-pong Movement Results

In this section, we examine the effect of ping-pong movement for all the frameworks which we have examined thus far. We have defined in the previous section how the ping-pong scenario is simulated (see Figure 3-2 and Section 4.2.2). Figure 4-9 depicts the handoff behaviour of ping-pong movement for MIPv6, MIPv6 (flat) with Fast-handover and HMIPv6 schemes with curves *source_send_mip_pp*, *source_send_fast_pp*, and *source_send_hmip_pp* respectively. Similarly, Figure 4-10 illustrates the handoff behaviour for HMIPv6 with Fast-handover, Simultaneous Bindings, S-MIP, and S-MIP non optimised schemes with *source_send_hmipfast_pp*, *source_send_bicast_pp*, *source_send_smip_pp*, and *source_send_smip_nonop_pp* curves respectively. (Note that we are only presenting from the perspective of source's sending sequence.) As can be seen in Figure 4-9, the MIPv6 completely breaks down during the ping-pong movement. The MIPv6 with Fast-handover and the HMIPv6 schemes are also notably disrupted with distinguishable breaks in the communication flow. Figure 4-10 shows that the HMIPv6 with Fast-handover and the Simultaneous Bindings schemes are affected to a lesser extent. However, severe throttling (i.e. the decrease in gradient) is still notable for both cases. Remarkably,

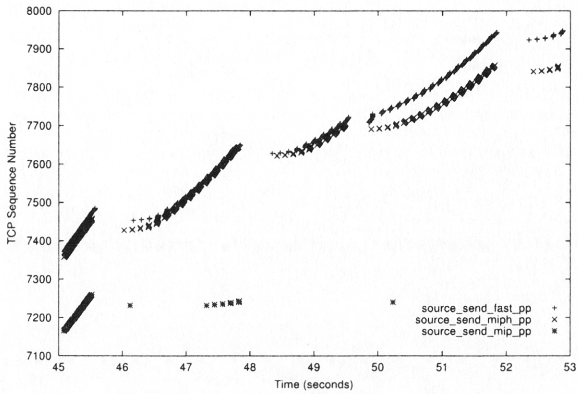


Figure 4-9 Ping-pong Behaviour for MIPv6, HMIPv6 and MIPv6 (flat) with Fast-handover

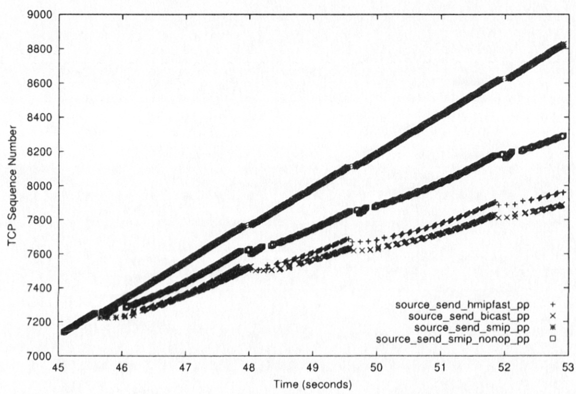


Figure 4-10 Ping-pong Behaviour for HMIPv6 with Fast-handover, Simultaneous Bindings, S-MIP non optimised and S-MIP

the S-MIP cases illustrate excellent resilience to ping-pong movements. The communication remains virtually unaffected even for the non optimised case.

Essentially, the ping-pong movement can be perceived as multiple individual linear handoffs. The ping-pong movement defined in the experiment is too 'short' for the MIPv6 scheme to complete a single linear handoff therefore the scheme performs poorly. Except for the S-MIP architecture, if we shorten the period in-between the switching back and forward (towards the *D* value), eventually all schemes will break

Table 4-1 Performance Matrix

Frameworks	*Avg. Goodput (Kbytes/second)*		*Time to transfer 6.5M file (seconds)*	
	Linear	*Pingpong*	*Linear*	*Pingpong*
MIPv6	100.847	83.820	66.001	79.408
MIPv6 + Fast'H	101.213	90.240	65.762	73.759
HMIPv6	101.520	91.587	65.563	72.674
HMIPv6 + Fast'H	101.593	93.867	65.516	70.909
Bicast/n-cast	101.580	92.713	65.524	71.051
SMIP (nonop)	102.007	91.113	65.127	68.538
SMIP	103.106	102.120	64.554	65.178
No Handoff	103.293		64.438	

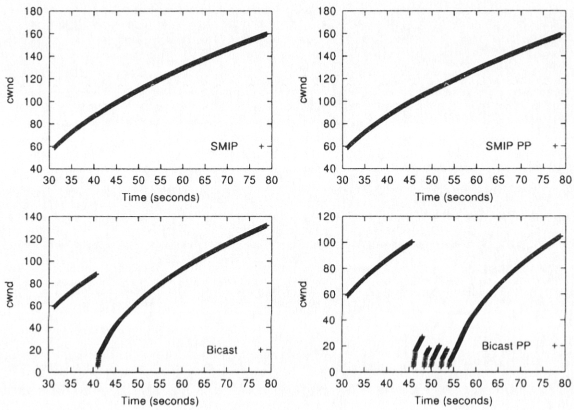


Figure 4-11 A Comparison on *cwnd* values over time for S-MIP and Simultaneous Bindings.

down (smaller than D value), similar to that of the MIPv6 case. The S-MIP guards against such break down as another handoff requested by the MH will not be allowed by the TAP until the current one is completed.

4.4 Discussion

4.4.1 Performance Comparison

Table 4-1 illustrates the relative goodput for all frameworks in comparison to that of a scenario where the MH is stationary near the PAR, i.e. no handoff takes place. We measure the time taken for each framework to transfer a 6.5 megabytes (M) file, with the linear and the ping-pong movement scenarios. As shown in Table 4-1, with the 'perfect' condition (no handoff), it takes 64.438s to transfer 6.5M file from the CH to the MH with the TCP goodput of 103.293 kilobytes/second (K/s). The S-MIP framework compares well with the no handoff case, achieving a goodput of 103.106 K/s for linear movement and requires slightly longer transfer time of 64.554s. What makes S-MIP stand out is its performance during the ping-pong movement scenario, where the goodput is still maintained at over 100K/s, at 102.120K/s, and the transfer time at 65.178s. The closest goodput is at 93.867 K/s in comparison, showing a significant reduction in performance. The worst performer is naturally the MIPv6 scheme, taking almost the full 80s to transfer the data in the ping-pong scenario, while achieving a 66.001s transfer time (still the worst) for the linear case. The spread in time for the linear scenario between the best and the worst framework is 1.447 seconds. However, this variation is amplified for the ping-pong scenario to 14.23 seconds, an increase of almost 10 folds. In conjunction to Table 4-1, we illustrate the TCP congestion window (*cwnd* value) plotted against time for the S-MIP framework in Figure 4-11. As can be observed, for both the linear (handoff at around t = 40s) and ping-pong movement scenarios (handoff between t = 45s and t = 55s), the *cwnd* values for S-MIP (SMIP and SMIP PP) do not drop significantly

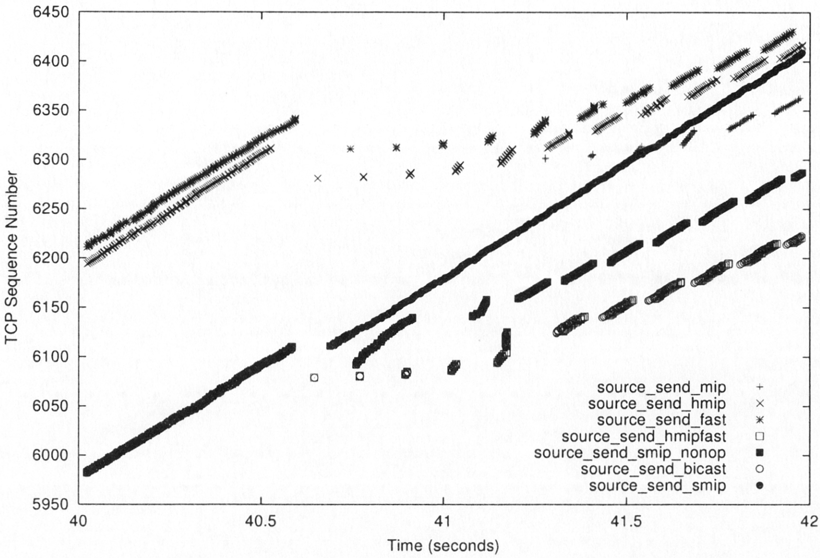


Figure 4-12 A Comparison between all the Schemes

unlike the case for the Simultaneous Bindings scheme (Bicast and Bicast PP), which typifies the *cwnd* behaviour for all other schemes.

Figure 4-12 illustrates the handoff delay (linear) for all frameworks mentioned in this chapter, from the perspective of the TCP source's sending sequence. We transpose curves for HMIPv6 with Fast-handover, Simultaneous Bindings, SMIP non optimised and S-MIP, in order to view clearly the relationship among themselves, as well as between them and those which possess much longer delay, i.e. MIP, HMIPv6 and MIPv6 with Fast-handover schemes. With our experimental setup, the order of the most effective latency reduction scheme to the least effective is: SMIP, SMIP non optimised, Simultaneous Bindings (D = 268ms), HMIPv6 with Fast-handover (D = 270ms), HMIPv6 (D = 326ms), MIPv6 (flat) with Fast-handover (D = 358ms), and lastly MIPv6 (D = 814ms). We rank Simultaneous Bindings higher than that of the HMIPv6 with Fast-handover because we believe it can potentially offer better results, even though in our simulation scenario, the two are comparable. The performance of HMIPv6 and MIPv6 (flat) with Fast-handover are very much dependent on the experimental topology layout. Therefore, the ordering between the two is only applicable to this series of simulation experiments.

It is well understood that, if applying the normal Mobile IP handoff algorithm, the overall handoff latency is at least 2 seconds, as the mobile host is required to wait for 2 beacon misses before initiating a handoff. In this chapter, we look at an optimisation of this, where the mobile host initiates a handoff when it receives an unseen beacon from a new access router[19]. With this, the handoff latency values ranges from approximately 800ms (for the MIPv6 case) to approximately 250ms (for the Simultaneous Bindings case), a significant reduction from 2 seconds. More impressively, with the S-MIP scheme, although the actual handoff delay is around 100ms, the latency perceived by the sender and the receiver is zero, i.e. as if no handoff has ever occurred.

4.4.2 Performance Summary

The performance of the L3 handoff latency reduction for the HMIPv6 scheme is bounded by the address resolution time (A), the one-way trip time (T) from the CH to the MH, and the size of the sliding window (W). If time A is greater than the time for the sender to send W packets to a receiver located T seconds away, then the performance degrades to that of the MIPv6 case. For the (flat) MIPv6 with Fast-handover case, its performance is affected predominantly by the round-trip time between the CH and the MH (the forwarding delay from the PAR to the NAR is very

---

[19] This is only suitable for linear movement or very fast MH movement, i.e. the MH is within a moving vehicle.

small in comparison). This time determines the number of packet errors (loss, if the NAR not buffering, duplicated *ack*, if the NAR is buffering) that occur at the MH. A recovery process, i.e. slow start, is almost inevitable, unless no more than 3 errors are detected.

For HMIPv6 with Fast-handover and the Simultaneous Bindings schemes, the key deficiency is the packet loss due to the L2 handoff, as these two schemes do not explicitly buffer packets during a handoff, triggering congestion control mechanisms. In comparison with this, the subsequent packets out of sequence have a lesser impact on the handoff latency. Even through this should have also triggered the congestion control, due to the setup of the forwarding scheme and the accumulative acknowledgement at the sender, the activation of the congestion control mechanism is avoided. Furthermore, initiating a handoff or packet forwarding based on a pre-determined or on-the-fly anticipation timing interval, as suggested/hinted in [33], [56], [20] is undesirable. It is most likely to be sub-optimal, unless the anticipation takes into consideration not just the time MH is likely to switch network, but also the movement direction of the MH.

The S-MIP architecture is a novel scheme that addresses all of the short comings described prior. It buffers packets explicitly at the NAR and provides an explicit anticipation method through network determined handoff, calculated passively using movement tracking and the MH position identification (via triangular signal strength evaluation). Although, the previously outlined MH position identification scheme (Chapter 3) for open space environment is rather primitive, S-MIP is structured in a way where more sophisticated positioning scheme maybe developed and 'plugged-in' to the framework, allowing it to operate in different physical environments. This is possible since the TAP messaging (the trigger) mechanism is access network independent.

4.5 Concluding Remarks on S-MIP

The analysis of the various handoff latency reduction schemes examined in this chapter showed that it is possible to significantly reduce the latency perceived by a mobile host during a handoff. This is especially true for the S-MIP case, achieving packet lossless handovers, where the mobile host and the correspondent host are unaware of the handoff which is taking place.

Not knowing *when* a mobile host will initiate a handoff and *how* it is likely to move after a handoff will inevitably result in packet losses. With packet forwarding (as the case for Fast-handover) or packet n-casting (Simultaneous Bindings), these losses may be reduced. However, this is at the expense of duplicated packets and/or delay in receiving packets. The S-MIP framework specifies, with reasonable accuracy, *when* a mobile host should switch and *how* it should switch, through passive movement tracking. It also specifies different techniques in optimising a handoff by categorising movement patterns of mobile hosts. *How* a mobile host should switch network is critical when considering stochastic/ping-pong movements. The lack of proper management leads to unnecessarily high setup cost, and may results in severe disruption if the switching frequency is shorter than the minimal time required for the L2 handoff.

The S-MIP architecture illustrates an application of the pro-collaborative approach in designing communication system and protocols. Based on what we called an 'Indicative' style in mobility management, the performance gain is substantial in comparison to other leading IETF handoff optimisation schemes. The S-MIP architecture is our response to the notion of nomadic user, examining at how to provide a seamless handoff environment at the network IP layer. This concludes our discussions on the notion of nomadic user and the S-MIP architecture.

Be not afraid of growing slowly; be afraid only of standing still.

- Chinese Proverb

Chapter 5

*NET*AMORPHOSE

We believe that one of the key open challenges for the 'new' Internet is the support for dynamic network service extensibility in achieving the availability of services anywhere and anytime for those users who are highly mobile in nature. We explore the notion of the nomadic system (services) and, in particular, the intersection of it with the notion of the nomadic user. An obvious example would be how to dynamically move the service (e.g. media transformation) that a mobile user uses to receive at his/her home network when s/he moves outside the home network. Another example would be how the network service providers dynamically extend their services, to follow a nomadic user/customer to arbitrary networks, thereby preserving the customer relationship by allowing a perceived continuum of services. Looking from a different perspective would be to consider a hypothetical architecture that allows 'virtual' network service providers, that own little or no physical network infrastructure, to provide, create and maintain network services over the Internet, by 'renting' resources from existing network operators.

5.1 Goals and Assumptions

The *NET*AMORPHOSE (NETwork-Adaptable and MObility-awaRe Programmable HOrizontal SErvice) Architecture is our response to offering the possibility of computing and/or communication systems, that provide network services, to move the services around the Internet, and to maintain a perceived continuum in network services offerings. It is based on the pro-collaborative form in design principle and the goal is to illustrate the principle and the viability of the notion of the nomadic system through architectural design.

We make three pivotal assumptions in designing *NET*AMORPHOSE within the 'new' Internet environment. Firstly, we assume that a service-enabled Internet model [31], [50] will be widely accepted and that the concept of a 'service layer' will also be adopted. This means a decoupling of the physical network operators from the network service providers, and the formation of a horizontally layered computer and communication system structure. Secondly, we assume that the idea of renting resources will be prevalent due to the service-enabled Internet. We refer to 'resource renting' as a procedure where a *service originator*[20] is able to rent from a *service host* computational resources and data storage for an agreed duration at a negotiated price. Thirdly, we assume a ubiquitous deployment of programmable networking devices capable of manipulating communication flows based on rules and agreements, and that these devices are modifiable by trusted entities. In other words, we assume programmability within the network and, to a certain extent, a 'software-enabled' Internet core.

[20] The *service originator* network is a network where the dynamic extension of network service is originated from, while the *service host* network is a network where the service extension is hosted.

5.2 The Design

The *NET*AMORPHOSE Architecture consists of three core functional areas, namely, security protection, resource renting and network service installation. Each of these areas is mapped to a functional model. Firstly, the *Security Trust Model* addresses the issue of security protection and intrusion prevention which are the key elements for the correct functioning of the *NET*AMORPHOSE architecture. The dynamic extension of network services into foreign networks from a secured (home) network poses itself as the foremost consideration for security design. The model aims to protect the service originator network and the foreign service host network as well as their respective users from breaches and unauthorised access. Secondly, the *Resource Renting Model* describes the mechanisms that enable the service originator network to rent computational resources from the service host network. Thirdly, the *Programmable Network Model* defines the protocols for the installation, the maintenance and the teardown of network services on the rented resources. In what follows, we elaborate on each of these areas and its associated models.

5.2.1 Security Trust Model

Figure 5-1 illustrates the generic trust model that depicts our security assumptions within the *NET*AMORHPOSE architecture. A network firewall is located at the service host network. It interfaces three different networks and is placed on the data path that intersects all three networks. One network interface is connected to the public (dirty/unsecured) side of the network. The other network interface is connected to the private (clean/secured) side of the network that attaches to resources requiring protection. The third network interface is connected to the

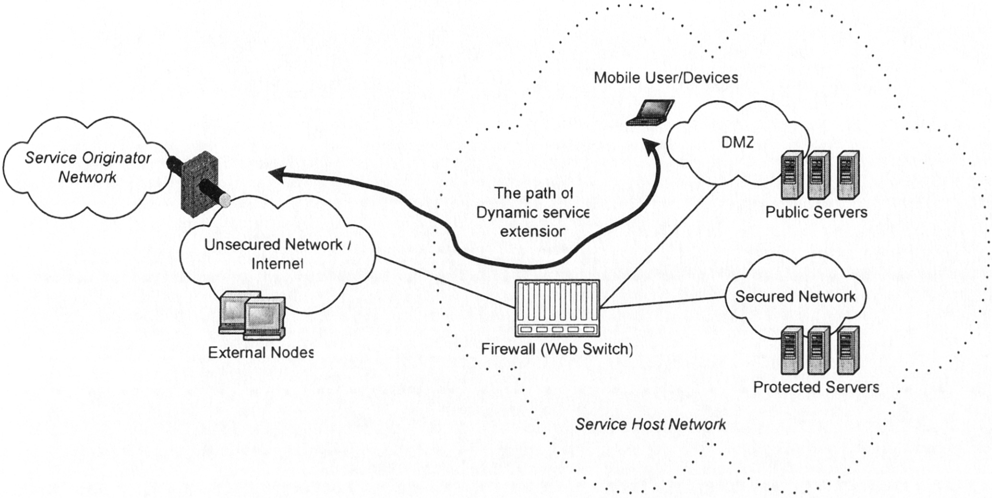


Figure 5-1 Security Model

demilitarized zone (DMZ), which is more secure than the dirty side of the firewall but less secure than the clean side of the firewall. An example usage of the DMZ is to locate web servers within this zone, where the public access must be allowed yet provides a certain level of protection.

We can safely assume that temporary network and network resource access, for a visiting mobile user attaching to the service host network, are most likely to be restricted within the DMZ. It is unlikely that a visiting mobile user would be granted access to the secured side of the network unless prior agreements have been formed. Therefore, in extending dynamic network services, a temporary security 'pinhole' at the firewall of the service host network is required to be opened for the network services from the service originator network to be 'pushed' inside the DMZ. Naturally, a corresponding opening of the security pinhole at the service originator's network firewall is also necessary. Assuming that the network services are operating at the secured side of the service originator's network, the network of the service originator is actually more susceptible to possible malicious attacks than the service host network. This is because the pinhole allows direct access within the secured side of the service originator's network. Thus, the temporary pinhole must be guarded heavily to prevent attacks initiated from within the service host network. These attacks include man-in-the-middle, communication hijacking, passive wiretapping, impersonation and denial-of-service. In the event where the security has been compromised within the service host network, only the DMZ will be affected, while the secured side of the network remains unaffected. Also, as strategies for disaster recovery and network reconstruction are well understood within the security community, it should not pose as a severe threat.

An example of how the security pinhole can be implemented is through the use of programmable firewall configuration. Essentially, by changing the firewall rules and filters, to allow only specific IP packet flow to go through, during the set up stage of the dynamic extension process creates this pinhole effect. Naturally, packet signature check is still required. Therefore, the placement, the availability and the programmability of a configurable firewall form the focal points for the provision, the execution and the resource renting activities.

5.2.2 Programmable Network Model

The *NET*AMORPHOSE architecture can be viewed from two perspectives, that is, *control* and *data*. Within *control*, we can subdivide it into *monitor* and *management*. Thus, the architecture is comprised of three different layers or planes, namely, the Data plane, the Monitoring plane and the Management plane, as illustrated in Figure 5-2.

The Data plane within the *NET*AMORPHOSE architecture consists of a confederation of network domains or Autonomous Systems (AS) that offers, as a collective whole, specific sets of network services. From the perspective of mobile users, this collection of ASs forms a service coverage referred to as the Home Service Network. Also, an AS is the 'home' for some mobile users where the particular AS is referred to as the Home Network for the mobile users. Both the Home Network and the Home Service Network can be referred to what we previously described as the service originator network. The network that a mobile user is visiting, but does not belong to the confederation of ASs, is simply referred to as the Foreign Network. However, it is referred to as the Ephemeral Service Network once the services from the Home Service Network have been extended to the Foreign Network. It can also be referred to what we previously described as the service host network.

The Monitor plane consists of control nodes, named Control Points (CP), where each CP is associated with one or more ASs[21]. A CP is responsible for the localised resource management, policing and monitoring of an AS. It functions as the logical connecting point and superimposes on top of the Data plane in forming a virtual network service provider. The CP, together with the Control Point Gateway (CPG), maps out the Service Control Overlay Network. It should be noted that the architecture and design for such a overlay is beyond the scope of this thesis. The Control Point Gateway (CPG) provides means for new ASs or Foreign Networks to join the existing Service Control Overlay Network, in other words, the fusion of the Foreign Network with the existing Home Service Network or the transformation of

[21] There could be more than one CP per AS for redundancy or performance reasons.

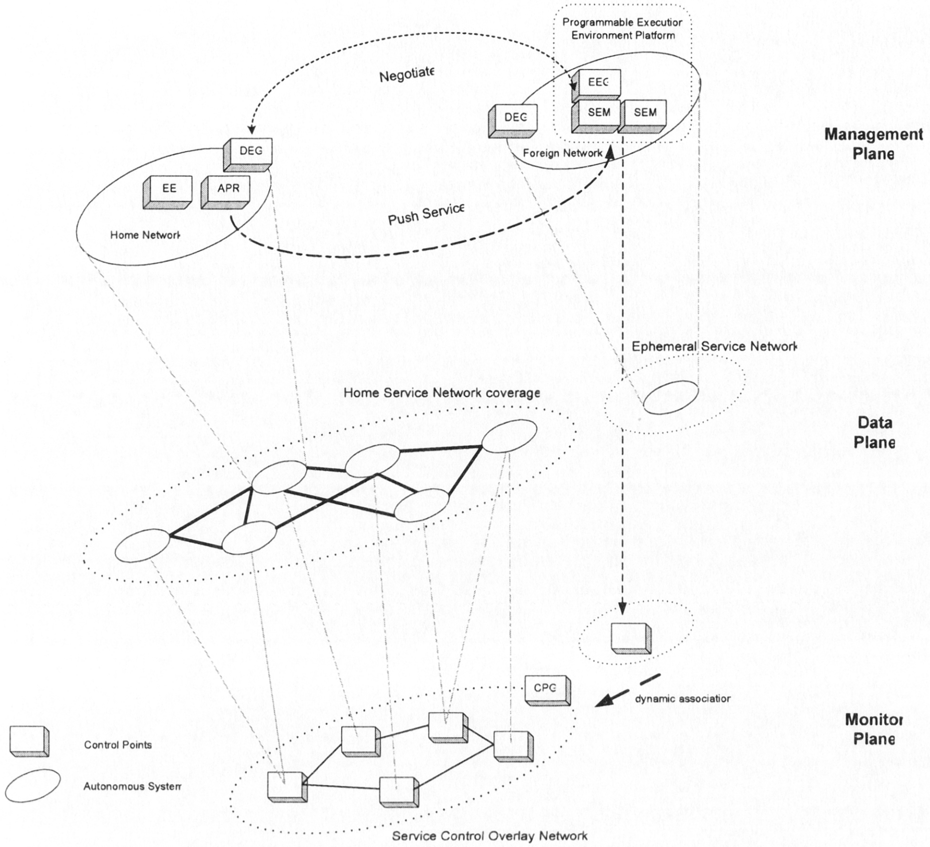


Figure 5-2 *NET*AMORPHOSE Architecture

the Foreign Network into the Ephemeral Service Network. The CPG(s) are scattered throughout the Home Service Network and are provided by individual ASs.

The Management plane consists of several control entities for the provision of the dynamic extension of network services. The Dynamic network service Extension Gateway (DEG) provides the admission control for the new service host network and the upload of the Service Extension Module (SEM) from the Application Repository (APR) of the service originator network to the service host network. The SEM can be viewed as a software-based Control Point. Thus, the DEG manages the mapping/association of the self deployed software-based Control Point (at the Foreign Network) to the existing Service Control Overlay Network. The association is established by the service originator network deploying the SEM into the execution environment of the service host network through negotiations with the Execution Environment Gateway (EEG) of the service host network. We refer to the execution environment, in a general sense, as a programmable hybrid software-hardware platform that has the capability of running software programs. Given such a definition of the execution environment, the SEM can therefore also be perceived as the actual network service application which is responsible to perform the network services, such as, data transformation and filtering.

The realisation of the execution environment is captured with the Programmable Execution Environment Platform. It houses the EEG where its primary function is the management of the available resources within the execution environment platform. In general, the EEG negotiates with the DEG of a service originator network to provide the computational resources for the dynamic deployment of the SEM. The Programmable Execution Environment Platform is co-located within the firewall of the service host network. (The next chapter will elaborate on the implementation issues regarding the programmable platform.) The final control entity in this discussion is the Execution Environment Interpreter (EEI). Analogues to a language interpreter, its role is to facilitate the communication between the DEG and the EEG in performing resource renting, i.e. the allocation of computing resources within the execution environment, in the events of a negotiation protocol mismatch.

It should be noted that whilst the teardown and maintenance of the network service is not the focus of this thesis and these remain as open research issues to be addressed, we intent to discuss the relevance of these issues with reference to our framework. Within the context of *NET*AMORPHOSE, the SEM would be responsible for the maintenance of the communication session (at the Ephemeral Network), as well as traffic or activity monitoring if applicable. We envision that the SEM can be persistent where it may keep operational states within the programmable execution environment, or it may be ephemeral and terminates after execution, depending on the session or the requirements of the network services. Through SNMP (Simple Network Management Protocol) [57] or specialised network device API within the programmable execution environment, the SEM would be capable of performing monitoring and accounting tasks and facilitating the proper use of the 'rented' resources. On a coarser timescale, the DEG may need to re-negotiate with the EEG if the current resource is insufficient or the level of service extension is no longer appropriate. Regarding service teardown, it would be triggered by the mobile user or the PA. The DEG would notify the associated SEM to enter into a 'destruction mode', and the cost of service with the service host network and eventually with the mobile user is settled.

5.2.3 Resource Renting Model

The idea of a software-enabled programmable network leads quite naturally to the concept of computational resource renting including CPU processing power and temporal data storage. In viewing the Internet as a large software-based computing and communication system, and assuming that the computational resources within the network (i.e. the execution environments) are ubiquitous, the resource renting model is constructed to facilitate the 'renting-out' of these resources, from the service host network's perspective. To reiterate, we assume that an ample 'rentable' resource at the core of the network can be co-located with network firewalls, core/edge routers and switches. Therefore, the aim is to provide the renting capability of these resources independent of the underlying hardware technologies. As mentioned previously, the Execution Environment Interpreter (EEI) is used to facilitate the communication between the DEG and the EEG. The EEI maintains a list of known renting protocols. In cases where the renting protocol is unknown to the EEI, it queries a central 'renting protocol database' for the rules of the unseen renting protocol, mimicking the traditional database request-and-reply model[22]. We argue that service host networks will actively update the database with its renting protocol as it is in their best interest to 'sell' the available computational resources.

The other aspect of the renting model is the cost settlement after renting-out the computational resources. We specify that the cost settlement must be between institutional bodies, i.e. service originator network and service host network, and not between a mobile user and the service host network. This should enforce transaction accountability and reliability. Furthermore, the service originator network can subsequently charge the mobile user in one aggregated transaction, at a later stage, under situations where the mobile user traverses through multiple service host networks. It must be noted that the resource renting model assumes the existence of network connectivity, that is, conventional Service Level Agreements between the various network operators are present a priori. Therefore, the model can be perceived

[22] Under a high traffic demand scenario, such a database might be duplicated and distributed to provide high availability. It should be noted that here are well-known commercial products that support this.

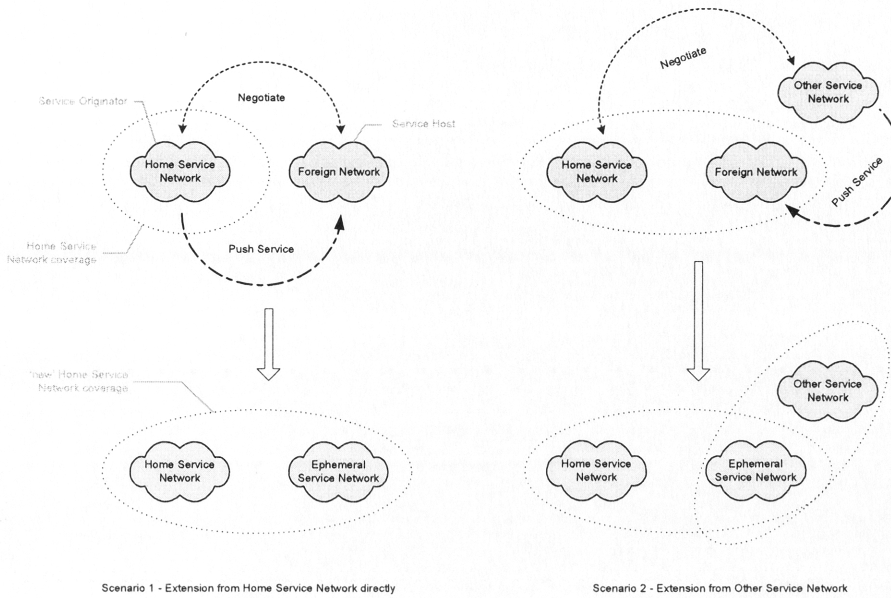


Figure 5-3 The concept of Dynamic Service Extension

as an overlay agreement, i.e. an agreement level that is predominantly associated with the provisioning of temporal computational resources for the dynamic extension and execution of value-added network services.

5.3 Dynamic Network Service Creation Examples

In accompany the discussion on the design of the *NET*AMORPHOSE architecture, Figure 5-3 illustrates an overview of the operations for the dynamic provisioning of network services. Two scenarios are presented. The first scenario is one where a mobile user has just re-attached to a new Foreign Network after leaving his/her Home Service Network, and has gained network connectivity at the DMZ within the Foreign Network. We show that the home network service extends dynamically to the domain of the Foreign Network, i.e. the Home Service Network (service originator network) negotiates with the Foreign Network (service host network) in extending its network service. With the second scenario, we assume that the Home Service Network is unable to extend the network service to the Foreign Network. Therefore, in this case, we show how equivalent network services (or other new services) offered by other network service provider is able to be deployed onto the foreign network in compensation for the inability of the Home Service Network.

Our scenarios assume the existence of the service layer within the Internet as outlined in Section 2.3.3. Briefly, the service layer contains three logical entities, which are the Personal Assistance (PA), the Service Bidder (SB) and the Execution Management (EM). The PA is a network-self of the mobile user and mediates with the service originator network as well as with the mobile user's correspondent(s). The SB is responsible for activities relating to the discovery of network services and their providers. The EM is responsible for the provision of the computational resources as well as the control, the operation and the maintenance of network services. Naturally, the focus of the *NETA*MROPHOSE architecture is on the Execution Management entity within the service layer. For clarity purposes, we assume that the mobile user wishes to maintain a media transformation network service as s/he is using a handheld device with limited display capabilities. We illustrate the two scenarios with the client-server communication style[23].

[23] An inter-personal communication style would simply require an additional step on top of the client-server example, namely, the inter-PA communication. The inter-PA communication determines correspondent location and negotiates end-terminal capabilities. This occurs before the client-server communication begins.

Under the first scenario (Figure 5-4), after the mobile user has re-attached to the Foreign Network, s/he chooses to maintain its current level of service and subsequently informs the PA (1). The PA relays the network service request to the mobile user's Home Network DEG (2). In turn, the DEG initiates a negotiation with the EEG in securing computation and storage resources located at some execution environment within the Foreign Network where the mobile user is currently residing (3, 4). The negotiation outcome is conveyed to the user (5, 6) and the user accepts the service extension by replying to the PA (7), which in turn notifies the DEG (8). Subsequently, the DEG deploys, from a co-located APR (omitted in the diagram), the SEM into the execution environment located at the Foreign Network (9). This SEM functions not only as the Control Point for the new Ephemeral Service Network, but also as a media transformation service application. The SEM notifies the DEG after the successful uploading (10) which in turn notifies the PA (11). The PA then issues a 'join Service Control Overlay Network' message (12). The SEM firstly contacts the CPG (13) which then confirms the arrival of a new Ephemeral Service Network with other CPs within the Home Service Network (14). The successful joint operation is then conveyed to the SEM (15) and triggers the SEM to inform and instruct the mobile user via the PA (16, 17) regarding how to route data to itself to obtain media transformation service. From this point onwards, the service extension is completed and the data is routed via the SEM (18, 19). After the media service is no longer needed the total cost of the service is firstly charged to the Home Network, i.e. the service originator network, and subsequently to the mobile user (20, 21).

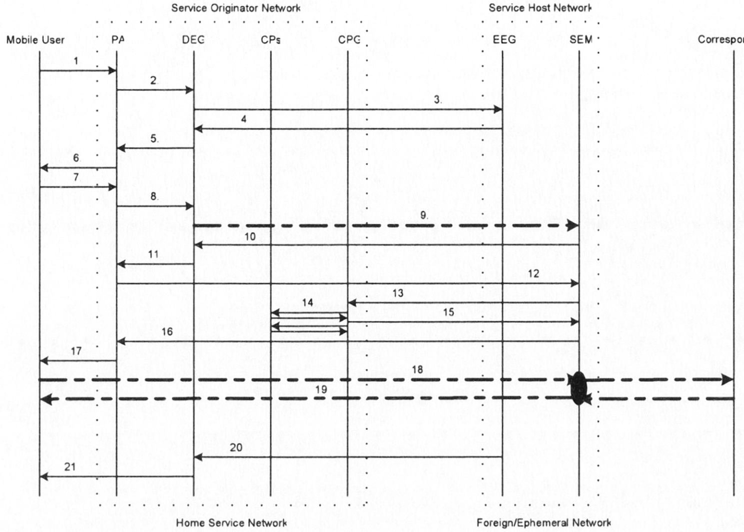


Figure 5-4 Extension from Home Service Network directly (Scenario 1)

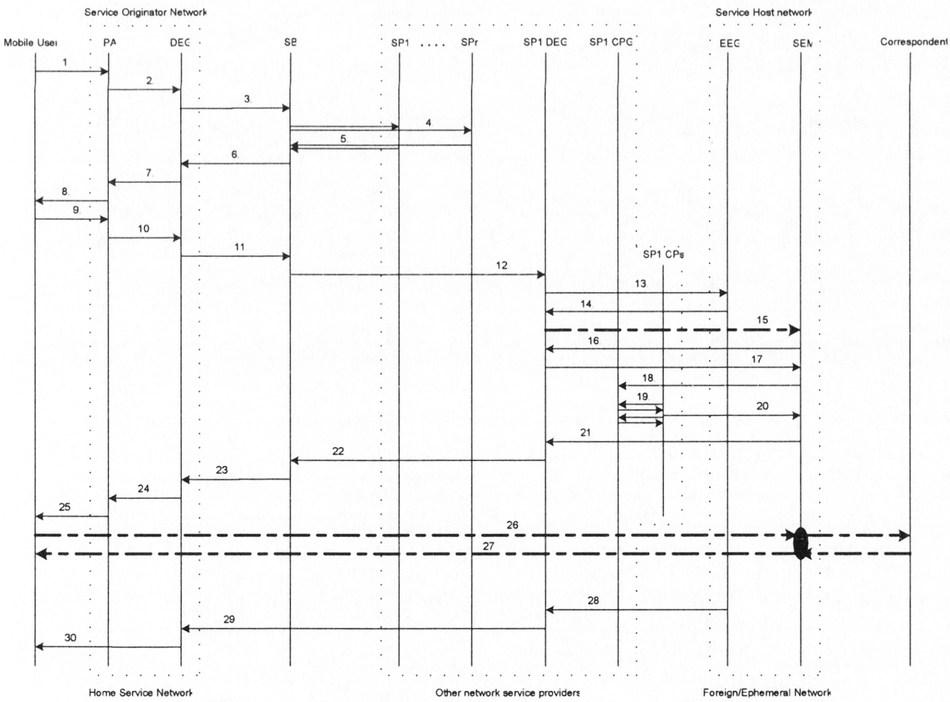


Figure 5-5 Extension from another Service Network (Scenario 2)

The second scenario (Figure 5-5) deals with the extension of network services from a third party network service provider and exemplifies the flexibility of a service-enabled 'new' Internet. Essentially, this requires an additional step compared with the previous scenario, i.e. the service bidding process. Similar to the previous example, after the mobile user has re-attached to the foreign network, s/he chooses to maintain its current level of service. However, this time, s/he also requires an additional service which is not offered by the Home Service Network. Let's assume that the additional service required another media transformation codec.

In what follows, we describe how the additional service can be dynamically extended. Firstly, the mobile user informs the PA about the need for the additional service (1). The PA then relays the network service requests to the mobile user's Home Network DEG (2). In this case, the Home Network DEG sends the request to the Service Bidder (SB) (3). The SB communicates/polls its known set of Service Providers (SP) in finding out who can fulfil the required service (4, 5)[24]. A list of possible SPs who can 'extend' their service boundary to fulfil the request is conveyed to the mobile user via the DEG and PA (6, 7, 8). Subsequently, the mobile user decides which SP to use and the choice is relayed to the SB (9, 10, 11) via the DEG. The SB then contacts the chosen SP's DEG (in this case SP1) (12) and initiates, on behalf of the DEG, the dynamic extension of services from SP1 to the Ephemeral Service Network where the mobile user is currently residing. The SP1 DEG negotiates with the EEG of the Ephemeral Service Network in securing computation and storage resources (13, 14). The SP1 DEG then deploys its SEM to the Ephemeral Network on behalf of the Home Network DEG (15). The SEM notifies the SP1 DEG

[24] Note that this process is a service discovery problem. It is out of the scope of our discussion in this thesis.

after the successful uploading (16). The SP1 DEG issues the 'joint Service Control Overlay Network' message (17), thereby tying the Ephemeral Network to its Service Network. The SEM then establishes contacts with SP1 CPG (18) which confirms the arrival of the new Ephemeral Service Network with other CPs within the Service Network (19). The successful joint operation is conveyed to the SEM (20), which in turn triggers the SEM to inform and instruct the mobile user via the SP1 DEG, SB, Home Network DEG and PA (21, 22, 23, 24, 25), regarding how to route data to itself to obtain media transformation service. From this point onwards, the service extension is completed and the data is routed via the SEM (26, 27). After the media service is no longer needed, the total cost is charged from the Service Host network to the SP1's Service Network, which in turn charges the Service Originator Network (28, 29). The Service Originator Network (which is the Home Network for the mobile user) then charges the user at a later stage through existing contractual payment mechanisms established between themselves (30).

5.4 Pro-collaborative Nature

Similar to the discussion in Section 3.5.5, the design of a complex dynamic network service provisioning framework, such as the case for *NET*AMORPHOSE, can be drawn from three main architectural and hence usage emphases. Aligning our discussion to the user-system design emphasis, the architecture can be oriented towards a system-dominant, a user-dominant or a balanced approach as shown graphically in Figure 5-6. The system architecture, in its entirety, is comprised of an edge (user / user's end device) component and the core (system / network) component. The design choices are essentially about how *dominant* each component

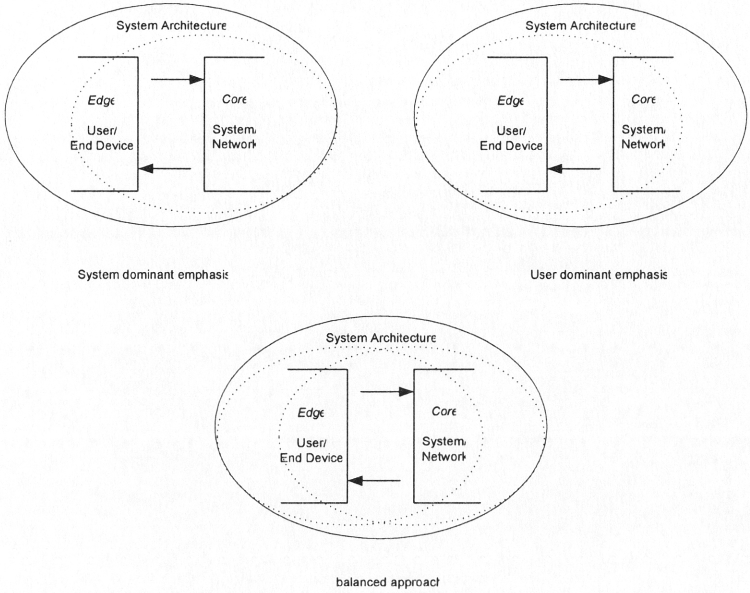


Figure 5-6 User-System design emphasis variations

has upon its counterpart in an interaction, where the aim of the interaction is to satisfy a goal set out by the user.

By the system-dominant approach, we mean a design where the computing/networking system both initiates and determines the actions to be taken in an interaction with a user. The user is totally passive and maybe even unaware of the interaction in this scenario[25]. Take service provisioning within *NETA*MORHPOSE as an example. The system would need to actively or even proactively i) monitor user behaviour, ii) take initiation in provisioning the services including service compositions if necessary, and iii) inform, instruct and manage the user's end devices in the interaction. Apart from the obvious technical challenges relating to

[25] An example would be how the GSM network handles the cell switching of mobile handsets.

artificial intelligence, the key drawback with such a design approach is the lack of involvement of the user (a key stakeholder in the whole exercise) in the decision making process. As the complexity of the interaction increases, user involvement ought to be an integral part of the system architecture. There is no reason to leave the user completely 'outside the loop' even at a supervisory level, yet attempts to satisfy the user's needs. This is especially true as the goal(s) for complex interaction is often ambiguous or multi-dimensional.

By the user-dominant approach, we mean a design where the user both initiates and determines the actions to be taken in an interaction with a computing/networking system. Similarly, take service provisioning as an example. The user would initiate (by instructing the system) the provision of a service and hence would expect the availability of the requested service. Such is the conventional design approach for most Internet applications or protocol architecture where the 'intelligence' is placed at the end system/user. The advantage of this approach is that the core/system can easily be kept simple and is tremendously robust. However, in the context of a service oriented system, this approach may not be ideal. Firstly, it is difficult to operate such a system efficiently in terms of utilising the computation, storage, and bandwidth resources, considering that the core does not seem to have active influence in the decision making process. Secondly, as a direct result, it is also difficult to guarantee the delivery of a particular service, and therefore, it is also difficult to set a convincing pricing scheme with such a design. But more importantly, if the interaction is to be elevated from merely interactive to proactive (e.g. anticipation), then the complexity of the dynamic service extension would equate to, what we suspect, an inverse productivity gain, that is, the degree of system anticipation, in situation where the objective/goal is unknown a priori , must grow exponentially.

By the balanced approach, we mean the pro-collaborative design approach where the interaction is a collaborative effort with the dominating emphasis that alternates between the user and the system (or edge and core). Take service provisioning within *NET*AMORPHOSE for instance, the user would initiate a 'wanting' request (steps 7, 8 in Figure 5-4 and steps 9~12 in Figure 5-5) for a particular service and the system would react to the request by setting up the required service. Then, conversely, the system would initiate an 'instructing' sequence (steps 16, 17 in Figure 5-4 and steps 21~25 in Figure 5-5), to indicate/instruct to the user the best method/approach with which to receive the requested service. However, there are a few points to note. This is not interactive computing per se since the origin of interactivity interchanges between the user and the system. In the classic interactive or proactive fashion where it is predominantly emphasising dominant user process, the assumption would be that the user knows about the 'details' of setting up the dynamic service extension. Thus, the system would 'inform', instead of instructing, the user of the successful setup of a network service. While this approach locates the intelligence or gives flexibility to the user in his/her decision making, it also means that somehow the user must know 'the where' and 'the how', a priori, with regards to directing the data to the newly installed SEM.

5.4 Enabler for Virtual Network Service Providers

Apart from addressing the dynamic extension of services, the *NET*AMORPHOSE architecture serves as a platform or an enabler for the realisation of 'pure virtual' service providers. The Resource Renting Model within *NET*AMORPHOSE implies that no ownership of any physical infrastructure or equipments is necessary. Therefore, from the business perspective, it means that (virtual) service provider is able to enter the service market place at a much lower cost than what is required traditionally. For instance, at one extreme, it may be conceivable that a service overlay network is formed purely by rented infrastructure (computational and data resources) on a per customer use basis. This means that service provider is able to focus entirely on providing and developing value-added services – a true service-oriented and horizontally layered approach. The Programmable Network Model within *NET*AMORPHOSE, in particular the programmable execution environment, specifies the way in which the renting and the deployment of the services belong to a virtual service provider is achieved. The details regarding the formation/construction of such a virtual service provider is out of the scope of this thesis.

5.5 Summary

The provision of programmable and customisable network services within the emerging service-oriented Internet environment has attracted tremendous interest over the recent years. We have presented a view of the dynamic extensible architecture that supports an ad-hoc-like service overlay network establishment.

While current service overlay network provide application enhancing services, our framework complements it with the possibility of dynamically extending their services to arbitrary networks. The proposed architecture addresses the security, resource hiring and service programmability issues surrounding network service extensibility.

In this chapter, we presented in detail an architectural framework that is derived from the pro-collaborative design form. Under the notion of a nomadic system, we have addressed the problem of dynamic network service extension, in particular, the service provisioning aspect of the dynamic network service extension. While the individual components of the architecture are not new, the novelty of our work lies in the entire architectural design.

In the next chapter, we describe the prototype design of *NET*AMORPHOSE and present an examination on system scalability.

Opportunity is missed by most people because it is dressed in overalls and looks like work.

- Thomas A. Edison

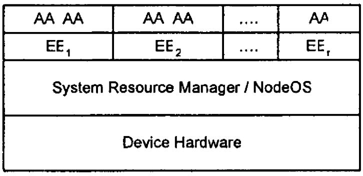
Chapter 6

*NET*AMORPHOSE Scalability Analysis

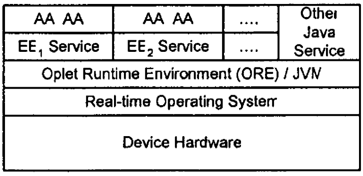
In this chapter, we outline a prototype design of the programmable execution environment for the *NET*AMORPHOSE architecture. The programmable execution environment is the key enabler for the dynamic provision of network services. We show how the programmable execution environment can be implemented. Given the novelty of the framework, there currently exists no related work that can be used for a direct performance comparison. Instead, we present a scalability analysis of the system to provide a 'feel' as to the feasibility of the proposed execution environment and the *NET*AMORPHOSE Architecture.

6.1 The Realisation: A Demonstrative Prototype Design of the Programmable Execution Environment

Recent advances in Active/Programmable Networking [17], [28], [36] provide an ideal foundation for the design of a programmable execution environment for the *NET*AMROPHOSE architecture. The characteristics of the Active Network Reference Model [17] are well suited to our requirements of the programmable



a) Active Network Reference Model



b) Oplet Runtime Environment Model

Figure 6-1 Programmable Network Reference Models

execution environment. Figure 6-1a illustrates schematically this reference model. The bottom layer is the System Resource Manager which provides mechanisms to prevent one execution environment (EE) from interfering with another. These mechanisms include limiting resources to each EE as well as dispatching packets to the correct EE. An EE accepts programs and packets that it deems valid. The Active Applications (AA) are the actual active custom software programs. We have chosen an implementation of this reference mode, namely, the Oplet Runtime Environment (ORE), developed by OpenetLab, as our base system framework (Figure 6-1b).

The ORE supports the dynamic injection of customised software services (oplets) into network devices and provides secure downloading, installation, and safe execution of these software services on the network device. Oplets are self-contained downloadable units that encapsulate one or more services, service attributes, authentication information, and resource requirements specifications [36]. Possible services that can be provided using oplets include monitoring, routing, diagnostic, data (multimedia) transformation, and other user specified functions that are within the limits of the oplet's operational scope. For example, oplets can access specific SNMP MIB (Simple Network Management Protocol Management Information Base) variables on network devices through the Java MIB API or use the Java SNMP API to facilitate certain execution decision making processes. The ORE provides a means to download the oplet, manage the oplet lifecycle, maintain a repository of active services, and track dependencies between the oplets and services. The software aspect of our implementation design for the programmable execution environment consists of a Java Virtual Machine (JVM) with the ORE running on top. As the ORE and its services are constrained to run in the JVM, the system stability of the core network device operation is not affected. Furthermore, the underlying JVM, which hosts the ORE, may also be modified to facilitate accounting tasks such as tracking the computational and memory consumptions.

We choose the Web switch as the hardware platform for the support of the programmable execution environment. The Web switch is also known as the Layer-7 switch. This is a special class of high-end switches that include a wirespeed ASIC-based packet forwarding hardware, servicing normal Layers 2/3 switching, and a programmable component with the flexibility to perform a variety of Layers 4-7 switching services. The combination of Nortel Passport 8600 series switch and the Alteon Web Switching Module are used as the base hardware configuration. With this setup, two working planes are introduced. The forwarding plane along the data path is implemented at each port using the WebIC modules. The WebIC is a network processing ASIC that combines a L2 packet engine and two RISC processors onto a single chip. The packet engine in each WebIC switches Layers 2/3 packets in hardware while the network processors also support Layers 3-7[26] switching in software if required. Up to a maximum of 20 RISC network processors can be combined together, forming parallel switching operations inside the multi-gigabit switch fabric. In this setup, every network processor across all switch ports can process traffic simultaneously regardless of the physical ports through which session traffic traverses through [84]. In addition, a JVM can be housed in the ROM image inside the 8600 series of switches, which in turn are capable of supporting the operation of ORE. The oplet uses the Java Forwarding API (JFWD) [36] to instruct the forwarding engine regarding the handling of packets by the network switch. The JFWD is a platform independent set of interfaces (developed in conjunction with the ORE) through which application programs can control the forwarding engines of heterogeneous network devices, i.e. the Nortel 8600 series of switches.

Figure 6-2 depicts the design of the programmable execution environment prototype. The control plane houses the JVM, ORE, and network applications that make up the execution environment. The control plane services can be further divided into two service planes, namely, control and data. The 'control plane control services' deal with network management issues such as altering the forwarding behaviour by modifying the forwarding priority along the data path, while the 'control plane data services' deal with data transformations, which means the cutting through of the data path and taking in and processing particular packets before forwarding them. Take dynamic service provisioning as an example. The extension request would arrive at the EEG (Execution Environment Gateway) initially (1) through normal traffic forwarding, at a particular port listening to the request. After

[26] Software-based Layer 3 switching includes activities such as complex packet filtering and manipulation.

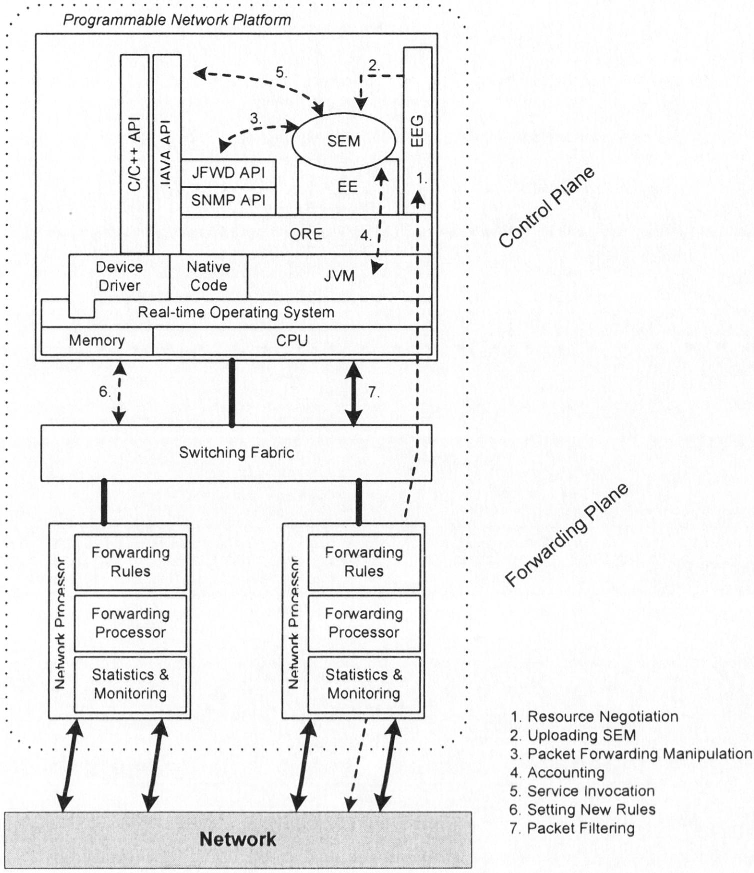


Figure 6-2 Programmable Execution Environment Design Schematic

successful resource negotiation, the SEM (Service Extension Module), in the form of an oplet, is uploaded to the EE (Execution Environment) inside the ORE (2). The SEM is able to gain access to the switch's forwarding engine via the JFWD API (3), the accounting information regarding resource usage directly from the JVM (4), as well as a set of JAVA API, in assisting the operation of the services that it is providing (5). These activities result in the setting of new rules for packet filtering (6) or simply the packet filtering process itself (7).

Relating back to the design of the *NET*AMORPHOSE architecture, the choice of using the Web/Layers 3-7 switches as the base system platform can be justified as follows. Strategically, Layers 3-7 switches represent the ideal platform for the integration of the management and the control functionalities. Through the provision of the common connectivity fabric for all network devices, while front-ending all servers and their applications, these devices are located in the perfect spot for administrators to exert control functionalities and traffic classifications. Therefore, from the point of view of network security, Web switches can be configured to function as sophisticated parallel firewalls through the use of Layer 3-7 switching programmability. This potentially equates to higher scalability and eliminates the single point of failure that typifies ordinary firewall. The combined use of the web switch (Nortel 8600 and Alteon) and the Oplet Runtime Environment forms the ideal programmable execution platform. We believe that Layer 4-7 oriented Web switches will become increasingly prevalent within the Internet core infrastructure and eventually widely accepted, and become a standard choice even within the edge network infrastructures.

6.2 Scalability Analysis

One critical issue that we ought to consider is how well *NET*AMORPHOSE scales once the dynamic service extensions are established rather than the time it takes for a dynamic extension to setup a service. It is evident that the setup time of the dynamic service extension (in the order of minutes or hours) will be much less than the current practice, which is the manual deployment (in the order of days or months) of service infrastructure. The focus of the scalability analysis is not about finding out when the system (ORE) crashes, but rather, when the service that the system is providing becomes non-acceptable. Here, we refer to non-acceptable as having the system throughput decreased to half of what it can offer at a maximum capability.

6.2.1 Experimental Setup

The idea of the analysis is to construct the experiments in such a way that given any number of oplets that are concurrently being executed inside the ORE (system load), the maximum number of oplets that can be additionally loaded per second (application throughput) is measurable. The delays in loading oplets were also being recorded to reflect the minimum 'wait' that users perceive for each invocation of the dynamic extension session. Also, the dynamic extension session may last for varying periods of time and there can be a certain number of paths in-service at any given time. This number, averaged over time, is the measure of the system load, and therefore an indication of the system scalability. Thus, in our experiments, we fixed the system load for the duration of an experiment and varied it across different service runs. Two parameters were measured: i) throughput for dynamic service extension installation and ii) service installation latency.

For the programmable execution environment at the hardware level, we configured two different system setups. With the first setup, the entire programmable execution environment was hosted on the Nortel Passport 8600 switch and the JVM was co-located within the ROM inside the switch firmware where the ORE is housed. In this case, the computational resource (CPU) was provided by the standard control-module within the switch. With the second setup, we wanted to examine the case where the programmable execution environment is hosted with an Alteon iSD configuration, where a cluster of Linux boxes are connected to the Alteon switch to enhance the processing level of the Layer 3-7 switching[27]. As we were unable to gain access to the clustering equipments, our setup consisted of only one high end Linux box with P3-750MHz CPU processing power and 256 Megabytes of system memory. The JVM and ORE were both housed within this Linux box. We assumed that the performance result obtained through the use of a single computation node could be approximated to derive the performance of using a cluster.

In addition to the programmable execution environment, we implemented the corresponding DEG (Dynamic Extension Gateway) and APR (Application Repository). We configured a PC box emulating a user's dynamic extension requests. It executed a scheduler and a service listener program. These programs operate as follows. The scheduler sends oplets to the ORE while the listener listens for messages sending back from the installed oplets. The scheduler sends, at a given duration, as fast and as many oplets as possible to the ORE, while the listener determines precisely how many oplets are successfully installed in the same period. We define a successful installation of an oplet on the ORE as when the listener receives an acknowledgement message from oplets that had been sent earlier.

We implemented three different prototype oplets to illustrate the following conceivable usage of the oplet as the SEM (Service Extension Module) for a dynamic network service extension. Firstly, the *SetFilter Oplet* is designed to manipulate the filter setting on the Nortel 8600 switch. The use of this type of oplet represents the setup of the 'pinhole' within the *NET*AMORPHOSE security model.

[27] We assume that the ORE will be fully integrated and ported to the Nortel Alteon platform in the near future.

Essentially, this oplet, which is housed within the control plane of the programmable execution environment, issues a filter change command to the packet filter engine to alter the switch's filtering behaviour. This activity consumes near-zero computational resources, and requires no communication outside the programmable execution environment or the switch. Hence, we used the performance of the SetFilter oplet as our benchmark for other types of oplets. Secondly, the *Messaging Oplet* is designed to send dummy control messages (uniform distribution) to the service initiator (i.e. the listener program). The operation of this oplet is to emulate the communication of the CP (Control Point) entity to maintain the operation of the Service Control Overlay Network. This oplet is designed to consume light resource yet generating heavy communication. Lastly, the *Transcoding Oplet* is designed to be essentially a Messaging Oplet with built-in media (picture) transformation capabilities, including GIF to JPEG and JPEG to GIF. With this oplets, both heavy communication and heavy computational resource are evident. This is a coarse representation of the functionality of the SEM. The relative performances of these oplets serve as an indication for the feasibility of using the Nortel Passport, Alteon and ORE technologies as an integrated solution for the implementation of the programmable execution environment within the *NET*AMORPHOSE architecture.

6.2.2 Results

The results shown in Table 6-1 are based on the average of ten independent experimental runs with identical starting conditions. Figure 6-3a and Figure 6-3b show the experimental results for the Passport 8600 setup while Figure 6-3c and Figure 6-3d show the experimental results for the Linux box setup. We define zero-load as instances where there are no oplets running on the ORE when uploading a

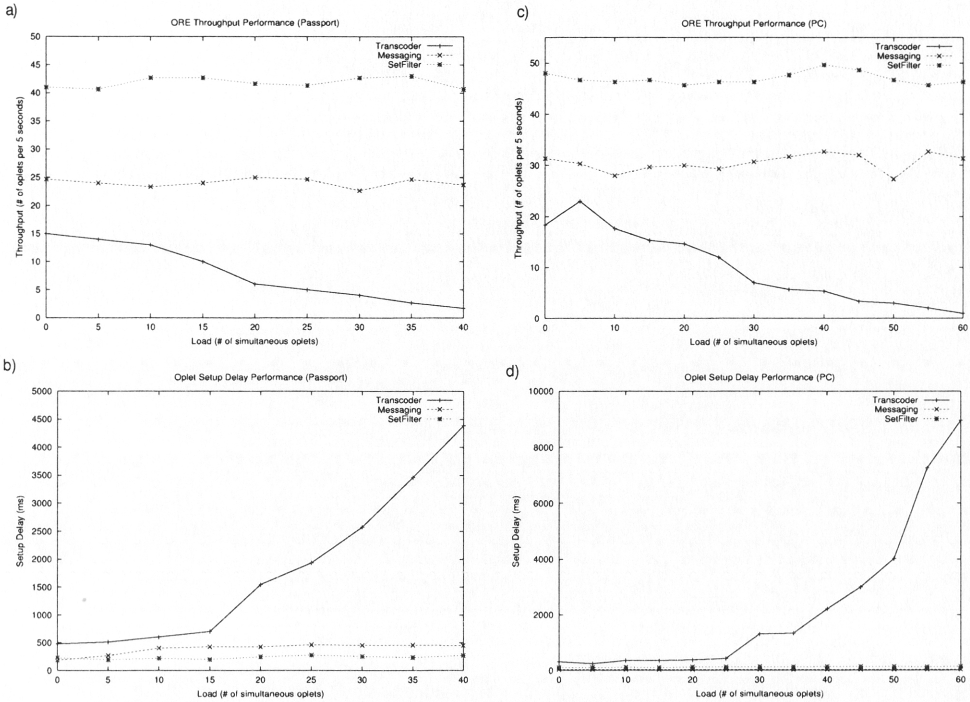


Figure 6-3 Scalability Analysis Results

new oplet (i.e. SEM). We refer to a session as the establishment (including uploading of oplets), operation, and maintenance of a service.

Under the Passport 8600 setup, an average load of at least 15~18 sessions can be supported before the (application loading) throughput drops to half of its value under zero-load, for the resource intensive Transcoding Oplet experiments (see Transcoder curve in Figure 6-3a). This number is much higher for Messaging Oplet (300 ~350 sessions) and even higher still for the SetFilter Oplet (500 ~ 550 sessions) experiments. These are not shown directly on Figure 6-3a due to the scaling of the graphs. However, a generally constant horizontal slope for the Messaging and SetFilter curves, for system load up to 40, suggests that these oplets do not consume much computational and/or data storage resources. As for the oplet installation delay analysis, we can observe a sudden increase in delay after the load passes 15 concurrent Transcoding Oplets running in the system as shown in Figure 6-3b. This equates well with the throughput analysis, as we can observe the sharp dip at approximately the same point in Figure 6-3a. Likewise, similar behaviour can be observed with the SetFilter and the Messaging Oplet experiments for both the throughput and the delay analysis for the Passport 8600 setup.

For the Linux box setup, a similar pattern can be observed although the throughput is notably higher as the system can sustain higher processing loads. As expected, the processing capability of the control-module card/blade inside the Passport 8600 is not designed for complex computation tasks. At 40 concurrent Transcoding Oplets, the Passport 8600 exhibits a delay of approximately 4500ms in creating and installing an oplet, while the corresponding Linux box takes half the time at approximately 2200ms as shown in Figure 6-3b and Figure 6-3d respectively. Scalability wise, the Linux box setup is reasonable, i.e. 26 sessions per machine and 2.4 creations per second (12 sessions divided by 5 seconds block[28]). In other words, to handle a throughput of 300 dynamic network service extensions per second for the Transcoding Oplets, we need a cluster of approximately 125 Linux boxes (assuming linear performance scalability) while 3250 service extensions can be in progress concurrently. Likewise, to achieve a throughput of 300 dynamic service extensions per second for the SetFilter Oplet or the Messaging Oplet, a cluster of at least 62 or 100 Linux boxes are necessary, while 51000 or 65625 extensions can be in progress concurrently (see Table 6-1).

---

[28] We measure in blocks of 5 seconds because a 1 second measurement is too fine grained for Transcoding Oplet experiments.

We use the classical two-way voice call model to examine the user base performance of our dynamic service extension. We believe the servicing time, in our case, should vary in the same order of magnitude to that of the voice model which would give us a quick and approximate feel of the potential performance. Since the nature of mobility means that users are able to move readily from one access network to another access network, it is therefore conceivable that some dynamic extensions would be very short. The following determines the maximum number of users that can be supported, i.e. the user base. Given an average call lasts for time t and the arrival rate of R, the average number of calls at any given time is $L = t * R$. At a steady state, L must be greater than R for the system to function correctly. In the context of two-way communication calls, statistics [39] showed that at a given busy hour, R equals to 2.8 call sessions/hour/user * N and t is 2.6 minutes, where N is the

Table 6-1 Summary of the Analysis Results

System Setup	Passport SetFilter[29]	Passport Messg	Passport T'coding	Linux SetFilter	Linux Messg	Linux T'coding
Throughput @ 50 %	21	12	8	24	15	12
Load (L)	550	350	18	1050	510	26
Rate (R)	3.53	2.24	0.12	6.73	3.27	0.17
Number of users supported (N)	4532	2884	148	8653	4203	214
Number of machine for 300 sessions/sec	71	125	188	63	100	125
Possible concurrent session	39285	43750	3375	65625	51000	3250

[29] The maximum number of configurable filter settings on the Passport 8600 is 376. In some experiments, repeated filter configurations are necessary.

number of users in the system. From Figure 6-3a, it can be observed that the throughput falls to half, at around 1.4 session/second (which is 7 session/5 seconds through extrapolation) when L is 18 (i.e. 18 concurrent Transcoding Oplets can be running at the background). Since $R = L / t = 18 / (2.6 * 60) = 0.12$ session/second, thus the system can handle $N = 0.12 / (2.8*60*60) = 148$ users. In other words, this equates to a user community of approximately 140, which can easily be supported by our system that consists of a Passport 8600 switch, to service a reasonably complicated operation. Table 6-1 presents a summary of all the experiments.

6.3 Summary

In this Chapter, we presented the prototype design of the programmable execution environment for the *NET*AMORPHOSE architecture. We leveraged our implementation design from the Active Network research field. We have constructed, configured and implemented the programmable execution environment, as well as all the associated entities required for the execution of the dynamic extension of services. We were able to show a preliminary scalability analysis of the *NET*AMORPHOSE architecture based on a few mock-up test scenarios. Our results to date indicated that our system design is viable.

The *NET*AMORPHOSE architecture is our response to the notion of the nomadic system through an examination of the dynamic provision of a seamless network service environment in the Internet. This concludes our discussions on the notion of nomadic system and the *NET*AMORPHOSE architecture.

Everything you can imagine is real.

- Pablo Picasso

Chapter 7

Conclusions

This chapter concludes the thesis by summarising the work it described, and noting areas in which future work is required.

7.1 Thesis Summary

This thesis has addressed issues of mobility in the Internet environment and a new design emphasis that we termed the *pro-collaborative* approach. The focus of this thesis is to illustrate the elements of the pro-collaborative design through discussion and formulation of architectures and protocols relating to mobility in the Internet. *Chapter 1* began by motivating the need for pro-collaborative design. It argued that current design approaches used in designing mobility system are unsatisfactory at times and proposed that successful mobility system design requires not only consideration at the user-dominant design emphasis but also the system-dominant design emphasis. It concluded by outlining the new emphasis, namely, the pro-collaborative approach, as a useful complementary alternative for designing computing and communication systems, in particular, mobility oriented systems.

Chapter 2 then considered the background and the related works to the issue of mobility in the Internet. It first presented the relationship between the pro-collaborative design and the interactive and proactive design paradigms. It then categorised mobility into the notion of *nomadic user* and the notion of *nomadic system*. With nomadic user, the chapter referred to it in a literal sense where a user's physical presence changes with respect to time. The relevant literatures were reviewed, including a discussion on device mobility and higher level mobility, namely, OSI session layer and above. With nomadic system, the chapter referred to it as the ability of computing and communication systems that offer value-added services to move their services to arbitrary locations within the Internet. The current state of research on service overlay networks were reviewed, the notion of service as pieces of software was discussed, and the idea of a service-enabled horizontally layered Internet was outlined. It argued that the ability to dynamically extend service coverage of service overlay networks would indeed satisfy the notion of nomadic systems. Finally, this chapter noted the context of the work described in this thesis, in terms of the structure of the network and the assumptions made about it.

The bulk of the contribution of this thesis was reported in the next four chapters. *Chapter 3* addressed a specific protocol design within the context of nomadic user, more specifically, it addressed the issue of handoff latency in Mobile IP networks. A new protocol (S-MIP) design aimed at providing seamless handoff for Mobile IP version 6 enabled heterogeneous networks was proposed. Its main novelty was the introduction of a number of handoff pre-emptive trigger based on location identification and movement tracking techniques. These triggers can be fed-into and/or used in conjunction with a series of proposed local handoff reduction techniques to achieve seamless handoff. The chapter then finished with a discussion on the choice of design approach considered, in relation to the handoff latency reduction problem, and presented a case for using the pro-collaborative approach.

Chapter 4 continued from the design of S-MIP and looked at the performance aspect of the proposed protocol. Firstly, the signalling cost analysis of S-MIP was presented. Secondly, a simulation model was proposed to compare the handoff latency reduction performance between S-MIP and five leading handoff latency reduction protocols outlined by the Internet Engineering Task Force (IETF). The UCB/LBNL/VINT Network Simulator version 2, *ns-2*, was chosen as the simulation tool. Various modifications and extensions were made to facilitate the simulation experiments. Finally, the results of the simulations were presented and it was shown that S-MIP out-performed all other protocols proposed by IETF in handoff latency reduction.

Chapter 5 proposed a new service architecture for the dynamic provisioning of services within the context of a nomadic system. The proposed architecture (*NET*AMORPHOSE) examined the issue of dynamic network service provisioning from three key perspectives, namely, security, programmability and resource hiring. It then went on to focus on the design of what was called the programmable execution environment, which was the key enabler in the dynamic network service provisioning. The chapter finished with a discussion on the design emphasis taken, which was the pro-collaborative approach.

Chapter 6 continued from the design of *NET*AMORPHOSE and examined the scalability of the proposed programmable execution environment. A prototype design and implementation, based on Active/Programmable Network concepts, of the programmable execution environment was described. Finally, several demonstrative experiments were carried out to illustrative the scalability of the proposed implementation of the programmable execution environment within the *NET*AMORPHOSE architecture.

7.2 Conclusion

To conclude, it is the hypothesis of this thesis that the design for mobility system in the Internet required a new form of user-system design emphasis. Although the computing and communication system are ultimately user-oriented in design goal, we argue that it is unsatisfactory to design systems which are predominantly either user-dominant or system-dominant in nature. We presented a new approach, *pro-collaborative* design, that is aimed at balancing such skewed emphasis and brings the interaction between the user and system to a co-functioned type, i.e. the dominating role alternates between the user and the system (edge and core) throughout an interaction.

In summary, this thesis has presented and evaluated, through protocol and architecture designs of mobility systems, the viability of pro-collaborative emphasis in mobile system design. It illustrated the potential of the pro-collaborative approach by firstly being performance oriented and has examined the gain in handoff latency reduction through the S-MIP protocol design. Subsequently, the focus was shifted to become architectural design oriented through the examination of the placement of system autonomy and intelligence within the *NET*AMORPHOSE architecture.

7.3 Future Directions

This section notes areas where further work is required and future directions that related work could take. Leaving aside the issues of implementing the protocols or engineering the prototypes described in this thesis before deployment could occur, there are a number of areas where further work is required.

Pertaining to S-MIP is the inclusion of security considerations to prevent man-in-the-middle, hijacking, passive wiretapping, impersonation and denial-of-service attacks. This involves an examination on the integration of IPsec Encapsulating Security Payload (ESP) [32] to protect the control traffic including Binding Update and Acknowledgement messages, Return Routability messages (Home Test Init and Home Test) [3], S-MIP related control messages and ICMPv6 [19] messages. Full protection against replay attack must also be considered through the use of IKE (Internet Key Exchange) [25].

Furthermore, there are three principal pieces of additional work arising from Chapter 3 and Chapter 4. The first is the need for a more rigorous scalability analysis which examines the messaging cost and operational implication in an environment involving hundreds of mobile hosts. The second is the need for a finer-grained and more 'realistic' location tracking mechanism which would enable S-MIP to operate under complex environments involving varying physical barriers and boundaries. The final piece of additional work is the need to determine an algorithmic solution for an optimal hierarchy in a multi-level hierarchical network layout pertaining to the placement of the TAP entity within the S-MIP enabled network.

Immediate future research directions relating to S-MIP includes the integration with Quality-of-Service (QoS) provisioning and the consideration for seamless vertical handoff support. Possible QoS provisioning methods or classes which may be considered include Measurement Based Admission Control (MBAC) [22], [29] (including end-point admission control [13]), Integrated Services [69], and Differentiated Services [11], [45], or a hybrid of the last two methods. We believe that the MBAC type of provisioning technique is most suitable for integration with the S-MIP protocol. Some promising results have been obtained which examined the suitability of combining end-point admission control with hierarchical Mobile IP networks [9], [10]. Regarding the consideration for seamless vertical handoff, future research areas include the examination of unifying different access networking technology, though the use of IP (Internet Protocol), including GPRS [76], CDMA [74], UMTS [86] 3G, Wireless LAN [78], and LAN. A broader framework, that considers 'seamless inter-network' handover where vastly different link characteristics are expected, is still largely an unchartered area of research.

Pertaining to *NET*AMORPHOSE, a variety of issues still remains for further work. Firstly, multiple instances of the Service Extension Module located at the service host network could potentially cooperate together through aggregation to serve larger multiple user groups belonging to the same service originator network. This would be an optimisation problem similar to that of the multi-level hierarchal network layout of TAP problem. Secondly, the architecture and protocol relating to dynamic network service teardown and maintenance within *NET*AMORPHOSE remain unaddressed and would need further investigation. Thirdly, an improved security scheme that specifically caters for the protection of the service originator network, during the dynamic extension of network service, would be necessary. It remains unclear, security wise, as how to deal with this aspect. Furthermore, failure recovery protocol also needs to be devised. It should be noted that, at the current state, the failure of SEM or the programmable execution environment hardware during a session would require the dynamic network service extension to be re-provisioned from scratch. Lastly, if the user is multiple 'network operator hops' away from the home service network, issues regarding dynamic provision of QoS-centric services, spanning wide area networks, would remain as a challenge.

Finally, the direction for future research includes a broader framework that captures what is required for the 'service layer' in facilitating a 'new' service-enabled Internet. This clearly remains as the open research problem. Issues to be considered includes, but are not limited to, i) the examination into infrastructure for service brokering and service discovery, in relation to the dynamic extension of services, ii) the design for a resource renting/negotiation protocol, iii) the concept of bidding services from multiple service providers and the idea of service subcontracting, and iv) an examination into the billing and transaction infrastructure that supports a cost-effective monetary settlement, including micro-payments technologies and mobile e-Commerce oriented payment methods [83]. Also, an examination into the possibility of creating a fully fledged service enabled Internet infrastructure, where the (application) service providers would be able to own little or no physical infrastructure, is a direction of great importance for future research.

This thesis has focused only on the mobile networking aspect of system design that dealt with the pro-collaborative design emphasis. We believe that the pro-collaborative approach is as universal in application as the interactive and the proactive counterpart. An examination into non mobile networking related fields is required to further strengthen and illustrate the need for the pro-collaborative approach.

Appendix

A. Minimum Overlapping Distance

Problem.

Show the minimum distance of intersection of three access routers (AR) in terms of the effective coverage radius, *r*, and the overlapping radius, *d*.

Assumption.

1. The centre of the three ARs form an equilateral triangle ABC, with three good coverage circles intersect at point S as shown in Figure A-1a.
2. The coverage area of an AR is a pure circle with a radius *r*.
3. The overlapped radius, *d*, is less than half of the effective radius, *r*, i.e. $d < r/2$.

Solution.

Suppose the radius of these three ARs overlapped by an amount of *d*, then the intersection of the effective coverage circles forms an overlapping area with three vertexes, A', B' and C', as shown in Figure A-1a.

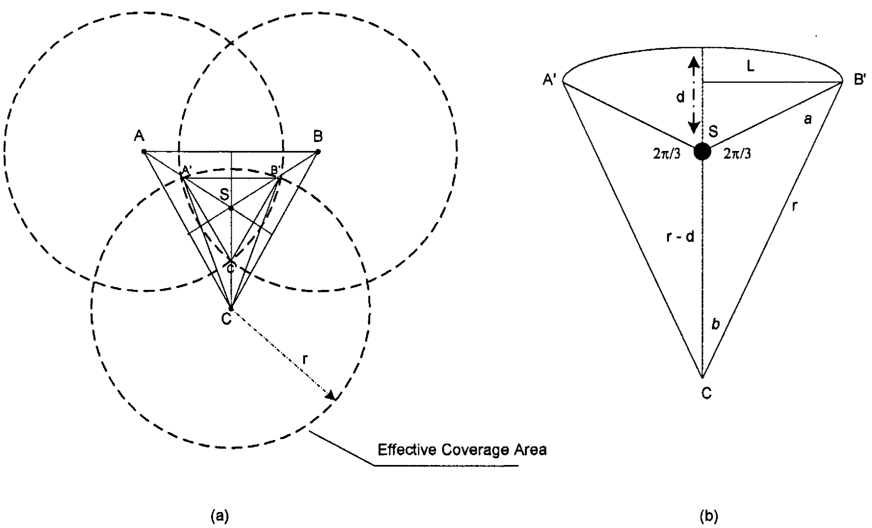


Figure A-1 Minimum Overlapping Distance

Since the overlapped radius is the same for all three ARs, the overlapping area forms an internal equilateral triangle ΔA'B'C', which is similar to ΔABC. Point S is also the centre of ΔA'B'C'. We need to show that the distance, *A'B'*, in terms of *d* and radius, *r*.

Since ΔA'B'C' is an equilateral triangle, so ∠A'SB' is $2\pi/3$. We enlarge ΔA'B'C as shown in Figure A-1b, the distance of *A'B'* is given by the following. By sine rule:

$$\frac{\sin\left(\frac{2\pi}{3}\right)}{r} = \frac{\sin(a)}{r-d}$$

$$\therefore a = \sin^{-1}\left[\frac{\sqrt{3}}{2}\left(1-\frac{d}{r}\right)\right] \quad \text{and} \quad b = \frac{\pi}{3} - a$$

Thus, $L = r \cdot \sin(b)$

$$\therefore A'B' = 2L = 2r \cdot \sin\left\{\frac{\pi}{3} - \sin^{-1}\left[\frac{\sqrt{3}}{2}\left(1-\frac{d}{r}\right)\right]\right\}$$

Therefore, for typical r = 25m, if we assume that d = 1m, then *A'B'* = 3.28m, if d = 2m then *A'B'* = 6.25m.

B. Overlapping Distance of the Marginal Coverage

Problem.

We want to compute the overlapping distance of two effective circles, i.e. the distance of *YC'*, *ZC'* and *YZ* in Figure A-2, and show that the distance is short enough for the marginal coverage area to cover.

Solution.

Since three internal (good coverage) circles with radius *r-d* intersect at centre point S, and the centres of these three circles form an equilateral triangle ΔABC,

$$\therefore XS = r - d \Rightarrow YS = \frac{r-d}{2}$$

$$\therefore YZ = YS - d = \frac{r-d}{2} - d = \frac{r-3d}{2}$$

From previous computations, we know that *A'B'* = 2*L* and ∠A'SB' is $2\pi/3$. For an equilateral triangle,

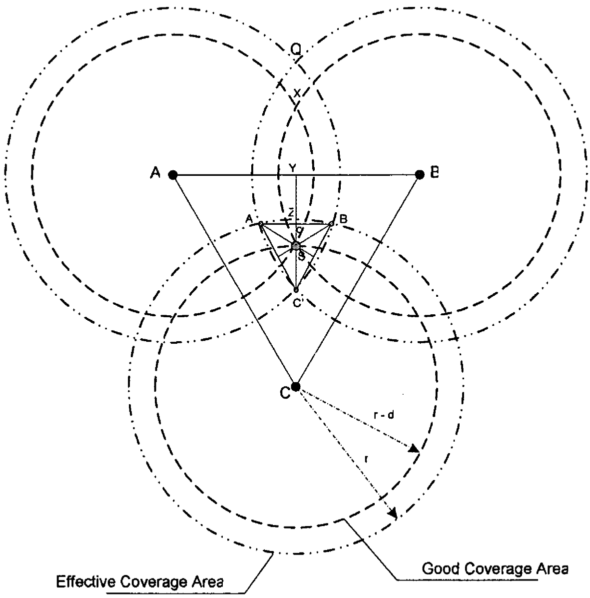


Figure A-2 Overlapping Distance of the Marginal Coverage Area

$$\angle ZSB' = \frac{\pi}{3} \text{ and } SC' = SB'$$

$$\sin(\angle ZSB') = \frac{\frac{1}{2}A'B'}{SB'} = \frac{L}{SC'} \Rightarrow SC' = \frac{2\sqrt{3}}{3}L$$

$$YC' = YS + SC' = \frac{r-d}{2} + \frac{2\sqrt{3}}{3}L$$

$$\therefore QC' = 2YC' = r - d + \frac{4\sqrt{3}}{3}L \qquad \text{and}$$

$$ZC' = d + SC' = d + \frac{2\sqrt{3}}{3}L$$

Using the data from Appendix A,

For r = 25m and d = 1m,

YC' = 13.89m, ZC' = 2.89m and YZ = 11m

For r = 25m and d = 2m,

YC' = 15.1m, ZC' = 5.6m and YZ = 9.5m

A distance of 9.5m for YZ is short compared to the distance that the marginal coverage can provide. Marginal coverage is at least equal to that of the effective radius, namely, 25m, when assuming SS exhibits negative log function behaviour.

C. Minimum Period for Movement Pattern Detection

Problem.

Find out the relationship between the minimal overlapping distance, *A'B'* and the coverage radius, *r*.

Solution.

From the solution in Appendix A, we can see that the distance *A'B'* is in terms of *r*, and *d* for $d < r/2$. Also, we know that the signal attenuation due to human body is 6.4dB and due to orientation of mobile device receiver is 9dB on average [70]. These equates to a difference in distance greater than 5m. Hence, the radius difference of the good coverage circle (internal circle in Figure A-2) and the effective coverage circle (external circle in Figure A-2) for a single AR must be greater than 2m (i.e. $d >$ 2m). However, in order to maximise the efficiency of the AR, it is necessary to make the coverage area as large as possible. This means that, we want to minimise the radius difference, *d*. Therefore, it is reasonable to keep *d* as a constant and find out the relationship between *A'B'* and *r*.

Now suppose *d* is a constant and let $d = 2$, therefore $r >$ 4m, as $d < r/2$. Figure A-3 shows the relationship of *r* and *A'B'*, varies from $r =$ 4m to $r =$ 200m. As we can see from the curve, when *r* is increased up to about 50m, the distance of *A'B'* tends to a constant level, which is about 7m. And the difference of *A'B'* between $r =$ 25m

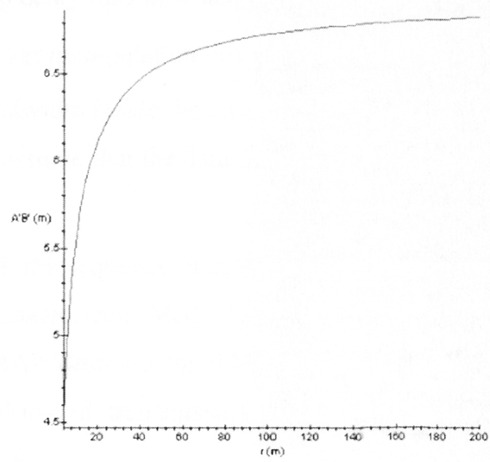


Figure A-3 Minimum Overlapping Distance (A'B') versus Radius (r)

and r = 50m is only 0.3m. Therefore, it is reasonable to assume that the distance $A'B'$, in our context, is about 7m. The distance $A'B'$ converges to a constant when r is greater than 20m. This computation shows that even through typical 802.11 coverage radius, r, varies from 25m to 50m, the minimal overlapping distance, $A'B'$, does not change significantly, i.e., within 1m. Therefore, the assumption of 7m for $A'B'$ is valid. The movement speed in the indoor environment is relatively slow and we assumed a constant speed of 1m/s. In this case, if the location sampling period is 1 sample/s, then after 3 samples, the mobile device is still located near the centre of the two overlapping zone, i.e. at most half way in between $A'B'$. Strategically, this is the best point in making the handoff decision. Hence, we show that in the worst case scenario, i.e. linear movement across the minimum overlapping distance path, 3 samples as a minimal are still sufficient for the movement pattern detection.

D. S-MIP Architectural Validity

Problem.

Show that packet out of order must occur within the hierarchical MIPv6 architecture at the NAR.

Assumption.

1. Assume that $T_{in} < D_h$ where T_{in} is the inter-arrival time between 2 packets, and D_h is the hop delay for *Path1* and *Path2*. See Figure A-4.
2. Since the packet manipulation processing time $T_{process} << D_h$, we ignore $T_{process}$.
3. Since the bandwidth inside the core network is greater than that of at the edge network, we assume that the data rate in *Path3* is not less than *Path1* or *Path2*.

Solution.

Let packet with the sequence number n be the first forwarded packet from PAR back to MAP originated from MAP. Let T be the MAP arrival time for the packet originated from MAP destined for PAR, but forward from PAR to NAR through MAP. For an end-to-end transmission using transport protocol with sequencing mechanism, i.e. TCP,

$$\forall i,j \in S : i < j \Leftrightarrow t_i < t_j$$

where S is the set of sequence number and t is the packet arrival time

$\because$ $T_{in} < D_h$ and the time interval for n travelling from MAP to PAR and return to MAP is $2D_h$

$\therefore$ $\exists$ m packets in Path1 and Path2 for some $m \geq 2$

Case 1: n and $n+m+1$ arriving at MAP simultaneously

If MAP process packet n first

$\because$ $m \geq 2$

$\therefore$ some packets must exist in either Path1 or Path2, so that, $t_{n+m+1} < T_{n+m}$

$\Rightarrow T_n < t_{n+m+1} < T_{n+m}$ i.e. packet with sequence number $n+m+1$ is in between n and $n+m$, showing out of sequence

If MAP process packet $n+m+1$ first

$\therefore$ $t_{n+m+1} < T_n$

We need to prove that $t_{n+m+1} < T_n < T_{n+1} < \ldots < T_{n+m} < t_{n+m+2}$ cannot exist.

Suppose $n+m+2$ has the longest delay which is D_h^- but packet $n+m$ is still in Path1. Since $T_{in} < D_h$ and packet $n+m+1$ just arrived at MAP so that packet $n+m$ does not reach PAR yet.

$\Rightarrow T_n < t_{n+m+2} < T_{n+m}$

Case 2: n arrived at MAP but $n+m+1$ still in the distance

Similar to second scenario in Case 1, $n+m$ is still in Path1,

$\therefore$ $T_n < t_{n+m+1} < T_{n+m}$

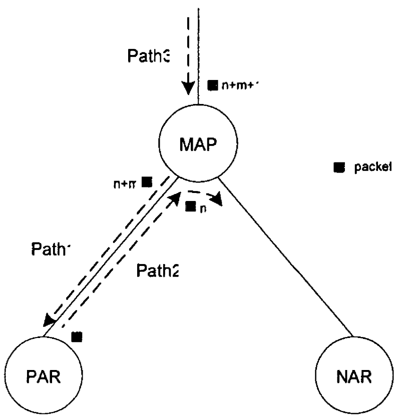


Figure A-4 Model of Packet Out-of-order

References

[1] Aljadhai, A., and Znati, T. F. A Framework for Call Admission Control and QoS Support in Wireless Environments. In *Proceedings of Infocom*, 1999.

[2] Appenzeller, G. et al. The Mobile People Architecture. *ACM Mobile Computing and Communications Review*, July 1999.

[3] Arkko, J., Devarapalli, V., and Dupont, F. Using IPsec to Protect Mobile IPv6 Signalling between Mobile Nodes and Home Agents. Internet Draft, IETF, June 2003. Work in Progress.

[4] Arnstein, L. et al. Labscape: A Smart Environment for the Cell Biology Laboratory. *IEEE Pervasive Computing*, Vol. 1, No. 3, July-September 2002.

[5] Bahl, P., and Padmanabhan, V. N. RADAR: An in-building RF-based user location and tracking system. In *Proceedings of Infocom*, Tel-Aviv, Israel, March 2000.

[6] Bakre, A., and Badrinath, B. R. I-TCP: Indirect TCP for Mobile Hosts. In *Proceedings of ICDCS*, May 1995.

[7] Balakrishnan, H., Seshan, S., and Katz, R. H. Improving Reliable Transport and Handoff Performance in Cellular Wireless Networks. *ACM Wireless Networks*, Vol. 1, No. 4, December 1995.

[8] Barbir, A. An Architecture for Open Pluggable Edge Services (OPES). Internet Draft, IETF, November 2002. Work in Progress.

[9] Bianchi, G., Blefari-Melazzi, N., Femminella, M., and Pugini, F. Joint Support of QoS and Mobility in a Stateless IP Environment. In *Proceedings of GLOBECOM*, San Antonio, USA, November 2001.

[10] Bianchi, G., Blefari-Melazzi, N., Femminella, M., and Pugini, F. Performance Evaluation of a Measurement-Based Algorithm for Distributed Admission Control in a DiffServ Framework. In *Proceedings of GLOBECOM*, San Antonio, Texas, USA, November 2001.

[11] Blake, S. et al., An Architecture for differentiated services. RFC 2485, IETF, 1998.

[12] Blumenthal, M. S., and Clark, D. D. Rethinking the design of the Internet: The end to end argument vs. the brave new world. *ACM Transactions on Internet Technology*, Vol. 1, No., 1, August 2001.

[13] Breslau, L., Knightly, E. W., Shenker, S., Stoica, I., and Zhang, H. Endpoint Admission Control: Architectural Issues and Performance. In *Proceedings of SIGCOMM*, Stockholm, Sweden, 2000.

[14] Brewer, E. A. Software in the Network: What happened and Where to Go. Keynote address at *Infocom*, March 2003.

[15] Caceres, R., and Iftode, L. Improving the Performance of Reliable Transport Protocols in Mobile Computing Environments. *IEEE Journal on Selected Areas in Communications (JSAC)*, Vol. 13, No. 5, June 1995.

[16] Caceres, R., and Padmanabhan, V. N. Fast and Scalable Handoffs in Wireless Internetworks. In *Proceedings of MobiCom*, 1996.

[17] Calvert, K. ed. Architecture Framework for Active Networks. AN draft, AN Architecture Working Group, 1998.

[18] Campbell, A. T. et al. Comparison of IP Micro-Mobility Protocols. *IEEE Wireless Communications Magazine*, Vol. 9, No. 1, February 2002.

[19] Conta, A., and Deering, S. Internet Control Message Protocol for the Internet Protocol Version 6 Specification. RFC 1885, IETF, December 1995.

[20] El-Malki, K. (eds.). Low Latency Handoffs in Mobile IPv4. Internet Draft, IETF, June 2003. Work in Progress.

[21] El-Malki, K., and Soliman, H. Simultaneous Bindings for Mobile IPv6. Internet Draft, IETF, June 2002.

[22] Gibbens, R., and Kelly, F. Measurement-based Connection Admission Control. In *Proceedings of the International Teletraffic Congress*, Washington, DC, USA, 1997.

[23] Hadjiefthymiades, S., Papayiannis, S., and Merakos, L. Using Path Prediction to Improve TCP Performance in Wireless/Mobile Communications. *IEEE Communications Magazine*, Vol. 40, No. 8, August 2002.

[24] Han, R., and Bhagwat, P. Dynamic Adaptation In an Image Transcoding Proxy For Mobile Web Browsing. *IEEE Personal Communications Magazine*, Vol. 5, No. 6, December 1998.

[25] Harkins, D., and Carrel, D. The Internet Key Exchange (IKE). RFC 2409, IETF, November 1998.

[26] Hsieh, H.-Y., Kim, K.-H., Zhu, Y., and Sivakumar, R. A Receiver-Centric Transport Protocol for Mobile Hosts with Heterogeneous Wireless Interfaces. In *Proceedings of MobiCom*, San Diego, USA, September 2003.

[27] Hsieh, R., Seneviratne, A., Soliman, H., and El-Malki, K. Performance Analysis on Hierarchical Mobile IPv6 with Fast-handoff. In *Proceedings of GLOBECOM*, Taipei, Taiwan, November 2002.

[28] Jaeger, R., Duncan, R., Travostine, F., Lavian, T., and Hollingsworth, J. An Active Network Services Architecture for Routers with Silicon-Based Forwarding Engines. In *Proceedings of LANMAN*, 1999.

[29] Jamin, S., Shenker, S., and Danzig, P. Comparison of Measurement-based Admission Control Algorithms for Controlled-Load Service. In *Proceedings of INFOCOM*, April 1997.

[30] Johnson, D., Perkins, C., and Arkko, J. Mobility Support in IPv6. Internet Draft, IETF, June 2003. Work in Progress.

[31] Katz, R. H. The Services-Enabled Internet: Implications for Mobile Wireless Networks. Keynote address at *INET*, Stockholm, 2001.

[32] Kent, S., and Atkinson R. IP Encapsulating Security Payload (ESP). RFC 2460, IETF, November 1998.

[33] Koodli, R. (eds.). Fast Handovers for Mobile IPv6. Internet Draft, IETF, March 2003. Work in Progress.

[34] Kubiatowicz, J. et al. OceanStore: An Architecture for Global-Scale Persistent Storage. In *Proceedings of ACM ASPLOS*, November 2000.

[35] Landfeldt, B., Larsson, T., Ismailov, Y., and Seneviratne, A. SLM, a framework for session layer mobility management. In *Proceedings of International Conference on Computer Communications and Network (ICCCN)*, Natick, Massachusetts, USA, October 1999.

[36] Lavian, T. et al. Enabling Active Flow Manipulation in Silicon-based Network Forwarding Engines. *IEEE Journal of Communications and Networks*, March 2001.

[37] Licklider, J. C. R. Man-computer symbiosis. *IRE Transaction of Human Factor in Electronics*, Issue 1, Vol. 1, March 1960.

[38] Liu, T., Bahl, P., and Chlamtac, I. Mobility Modeling, Location Tracking, and Trajectory Prediction in Wireless ATM Networks. *IEEE Journal on Selected Areas in Communications (JSAC)*, Vol. 16, No. 6, August 1998.

[39] Lo, C. N., Wolff, R. S., Bernhardt, R.C. An Estimate of Network Database Transaction Volume to Support Universal Personal Communication Services. In *Proceedings of ICUPC*, 1992.

[40] Maltz, D. A., Bhagwat, P. MSOCKS: An Architecture for Transport Layer Mobility. In *Proceedings of Infocom*, 1998.

[41] Mitton, D. et al. Authentication, Authorization, and Accounting: Protocol Evaluation. RFC (Informational) 3127, IETF, June 2001.

[42] Mohan, R., Smith, J., and Li, C.-S. Adapting Multimedia Internet Content For Universal Access. *IEEE Transaction on Multimedia*, Vol. 1, No. 1, March 1999.

[43] Mouly, M., and Pautet, M.-B. The GSM System for Mobile Communications. Published by the authors, 1992.

[44] Narten, T., Nordmark, E., and Simpson, W. Neighbor Discovery for IP Version 6 (IPv6). RFC 2461, IETF, December 1998.

[45] Nichols, K., Jacobson, V., and Zhang, L. Two-bit differentiated services architecture for the Internet. RFC 2638, IETF, 1999.

[46] Perkins, C. E. Encapsulation within IP. RFC 2003, IETF, October 1996.

[47] Perkins, C. E. IP Mobility support for IPv4. RFC 3344, IETF, August 2002.

[48] Perkins, C. E., and Calhoun, P. R. AAA Registration Keys for Mobile IP. Internet Draft, IETF, June 2003. Work in Progress.

[49] Priyantha, N. B., Miu, A., Balakrishnan, H., and Teller, S. The Cricket compass for context-aware mobile applications. In *Proceedings of MobiCom*, Rome, Italy, July 2001.

[50] Raman, B. et al. The SAHARA model for Service Composition across Multiple Providers. Invited paper for *International Conference on Pervasive Computing*, 2002.

[51] Ramjee, R. et al. HAWAII: A Domain-Based Approach for Supporting Mobility in Wide-area Wireless Networks. In *Proceedings of ICNP*, 1999.

[52] Rhea, S. et al. Maintenance-Free Global Data Storage. In *IEEE Internet Computing*, Vol. 5, No. 5, September/October 2001.

[53] Rosenberg, J. et al. SIP: Session Initiation Protocol. RFC 3261, IETF, June 2002.

[54] Saltzer, J. H., Reed, D. P., and Clark, D. D. End-to-End Arguments in System Design. *ACM Transactions on Computer Systems*, Vol. 2, No. 4, November 1984.

[55] Snoeren, M. A. C. A Session-Based Architecture for Internet Mobility. PhD Dissertation, MIT 2003.

[56] Soliman, H., Castelluccia, C., Malki, K., and Bellier, L. Hierarchical Mobile IPv6 mobility management. Internet Draft, IETF, June 2003. Work in Progress.

[57] Stallings, W. SNMP, SNMPv2, SNMPv3, and RMON1 and 2. 3rd Edition, Addison Wesley, 1999.

[58] Tang, D., and Baker, M. Analysis of a Metropolitan-Area Wireless Network. In *Proceedings of MobiCom*, Seattle, USA, 1999.

[59] Tennenhouse, D. L. Proactive Computing. *Communications of ACM*, Issue 43, No. 5, May 2000.

[60] Thai, B., Wan, R., Rakotoarivelo, T., and Seneviratne, A. Integrated Personal Mobility Architecture: A Complete Personal Mobility Solution. *ACM Mobile Networks and Applications (MONET)* Special Issue: Personal Environment Mobility in Multi-Provider and Multi-Segment Network, Vol. 8, Issue 1, 2003.

[61] Thomson, S., and Narten, T. IPv6 stateless address autoconfiguration. RFC 2462, IETF, December 1998.

[62] Valko, A. et al. Design and Performance of Cellular Access Networks. *IEEE Personal Communications*, Special issue on IP-Based Mobile Telecommunications Networks, 2000.

[63] Wang, H. et al. ICEBERG: An Internet-core Network Architecture for Integrated Communications. *IEEE Personal Communications*, Special Issue on IP-based Mobile Telecommunication Networks, 2000.

[64] Want, R. et al. The Personal Server: Changing the Way We Think about Ubiquitous Computing. In *Proceedings of International Conference on Ubiquitous Computing (Ubicomp)*, Gotebory, Sweden, September 2002.

[65] Want, R., Hopper, A., Falcao, V., and Gibbons, J. The Active Badge location system. *ACM Transactions on Information Systems*, Vol. 10, Issue 1, 1992.

[66] Want, R., Pering, T., and Tennenhouse, D. L. Comparing autonomic and proactive computing. *IBM Systems Journal*, Vol. 41, No. 1, 2003.

[67] Weiser, M. The Computer of the 21st Century. *Scientific American*, Issue 256, No. 3, September 1991.

[68] Widmer, J. Extension to the ns Network Simulator. http://www.informatik.uni-mannheim.de/~widmer.

[69] Zhang, L., Deering, S., Estrin, D., Shenker, S., and Zappala, D. RSVP: A New Resource ReSerVation Protocol. *IEEE Network*, Vol. 7, Issue 5, September 1993.

[70] Zhou, Z.-G., Chen, R., Chumchu, P., and Seneviratne, A. A Software Based Indoor Relative Location Management System. In *Proceedings of Wireless and Optical Communications*, Canada, 2002.

[71] Akamai website. http://www.akamai.com.

[72] Authentication, Authorization and Accounting (AAA). IETF Working Group Charter. http://www.ietf.org/html.charters/aaa-charter.html.

[73] Autonomic Computing: IBM's Perspective on the State of Information Technology. Whitepaper, IBM. http://www.ibm.com/research/autonomic.

[74] Code Division Multiple Access (CDMA), CDMA Development Group. http://www.cdg.org.

[75] Extranet On-Demand project. CeNTIE Technology project. http://www.centie.org.

[76] General Packet Radio Service (GPRS), GPRS Forum. http://www.gsmworld.com/technology/gprs/index.shtml.

[77] ICQ website. http://www.icq.com.

[78] IEEE 802.11 Wireless Local Area Networks. http://grouper.ieee.org/groups/802/11.

[79] ITU-T Recommendation F.850. Principles of Universal Personal Telecommunication (UPT). ITU, March 1993.

[80] Microsoft Messenger homepage. http://messenger.msn.com.

[81] Monarch project. Rice University. http://www.monarch.cs.rice.edu.

[82] Open Pluggable Edge Services (opes). IETF Working Group Charter. http://www.ietf.org/html.charters/opes-charter.html.

[83] Pay for Parking with your Telstra mobile phone. Telstra m-Commerce. http://www.telstra.com.au/mobilenet/cur_prom/dialpark.htm.

[84] Scaling Next Generation Web Infrastructure with Content-Intelligent Switching. Nortel Networks, White Paper. http://www.nortelnetworks.com/products/alteon.

[85] The UCB/LBNL/VINT Network Simulator – ns (version 2). http://www.isi.edu/nsnam.

[86] Universal Mobile Telecommunications System (UMTS), UMTS Forum. http://www.umts-forum.org.