

Stochastic Control of Inter-Switch Handoff and Location Update in Wireless Cellular Networks

by

WAI-SHUEN VINCENT WONG

M.A.Sc., University of Waterloo, 1996

B.Sc., University of Manitoba, 1994

**A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY**

in

**THE FACULTY OF GRADUATE STUDIES
DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING**

**We accept this thesis as conforming
to the required standard**

THE UNIVERSITY OF BRITISH COLUMBIA

April 2000

© W. S. Vincent Wong, 2000

In presenting this thesis in partial fulfilment of the requirements for an advanced degree at the University of British Columbia, I agree that the Library shall make it freely available for reference and study. I further agree that permission for extensive copying of this thesis for scholarly purposes may be granted by the head of my department or by his or her representatives. It is understood that copying or publication of this thesis for financial gain shall not be allowed without my written permission.

Department of ELECTRICAL & COMPUTER ENGINEERING
The University of British Columbia
Vancouver, Canada

Date 31ST MARCH, 2000

Abstract

One of the issues in mobility management is to support handoff. When the mobile user moves from one location to another, the network should ensure that all ongoing connections are rerouted to another access point in a seamless manner. Part of our work focuses on connection rerouting due to inter-switch handoff in wireless ATM networks. Although fast local connection rerouting minimizes handoff delay, the end-to-end path after rerouting may become “suboptimal”, which implies an inefficient use of network resources. Path optimization may be necessary afterwards. Our research begins with the following question: “How often should path optimization be performed?” To this end, we propose three path optimization schemes (namely: exponential, periodic, and Bernoulli), which are simple to implement. Closed-form solutions of the optimal operating point are derived for each scheme.

We further investigate this problem and propose a stochastic model to determine the optimal time to initiate path optimization. Link cost and signaling cost functions are introduced to capture the trade-off between the network resources utilized by a connection and the signaling and processing load incurred on the network. Results indicate that the optimal policy derived from our model has a better performance compared to other heuristics.

Another issue in mobility management is to track the location of the users between call arrivals. Although it has been shown that the distance-based location update algorithm has a better performance than the movement and timer based schemes, the determination of the optimal distance threshold is often based on certain unrealistic assumptions. We propose a stochastic model to study the distance-based update algorithm. Our model is applicable to arbitrary cell topologies and the cell residence time can follow general distributions. We consider Markovian movement patterns in which the probability that the

mobile user moves to a particular neighboring cell can depend on the location of the current cell or a list of cells recently visited. Results indicate that the distance thresholds determined from our model have a better performance than those derived from a hexagonal cell configuration with random walk movement pattern.

Table of Contents

Abstract	ii
List of Figures	viii
List of Tables	xi
Acknowledgements	xii
Chapter 1 Introduction	1
1.1 Related Work on Inter-Switch Handoff	3
1.1.1 Virtual Connection Tree	4
1.1.2 Multicast-Based Rerouting	4
1.1.3 Path Extension	4
1.1.4 Path Rerouting	5
1.1.5 Two-Phase Handoff Scheme.....	6
1.2 Related Work on Location Update.....	8
1.2.1 Location Area (LA)-based.....	9
1.2.2 Selective LA Update.....	9
1.2.3 Profile-based	10
1.2.4 Movement-based.....	10
1.2.5 Timer-based	11
1.2.6 Distance-based.....	11
1.2.7 Predictive Distance-based.....	12
1.2.8 State-based	13
1.2.9 LeZi Update	13

1.3	Related Work on Terminal Paging	14
1.3.1	Blanket Polling	14
1.3.2	Shortest-Distance-First	15
1.3.3	Sequential Paging based on User's Location Probability	16
1.3.4	Velocity Paging	17
1.4	Motivations and Objectives	17
1.5	Contributions of the Thesis	20
1.6	Organization of the Thesis	21
Chapter 2	Path Optimization for Inter-Switch Handoff: Heuristics	23
2.1	Introduction	23
2.2	Path Optimization Schemes	24
2.2.1	QoS-based	24
2.2.2	Network-based	24
2.2.3	Timer-based	25
2.2.4	Handoff-based	25
2.3	Analytical Model	26
2.3.1	Exponential Path Optimization Scheme	28
2.3.2	Periodic Path Optimization Scheme	29
2.3.3	Bernoulli Path Optimization Scheme	30
2.4	Simulation Model	31
2.5	Results and Discussions	34
2.6	Summary	39
Chapter 3	Path Optimization for Inter-Switch Handoff: Optimal Policy	46

3.1	Introduction.....	46
3.2	Model Formulation	47
3.2.1	Semi-Markov Decision Process Model	47
3.2.2	State Transition Probability Function	52
3.2.3	Cost Functions	54
3.3	Optimality Equations	56
3.3.1	Value Iteration Algorithm.....	57
3.3.2	Structure of the Optimal Policy	59
3.4	Implementation Aspects.....	59
3.5	Model Extensions.....	60
3.5.1	Extension to Mobile-to-Mobile Connection	61
3.5.2	Extension to QoS Constraints	62
3.6	Numerical Results and Discussions	63
3.6.1	Simulation Model	64
3.6.2	Results.....	65
3.7	Summary	68
Chapter 4	Distance-based Location Update	75
4.1	Introduction.....	75
4.2	Model Formulation	76
4.3	Distance-based Location Update Algorithm.....	80
4.3.1	Linear Cell Configuration.....	80
4.3.2	Hexagonal Cell Configuration	85
4.3.3	Arbitrary Cell Topology	88

4.4	Numerical Results and Discussions	91
4.4.1	Paging Delay Constraints.....	91
4.4.2	Cell Residence Time Distributions	93
4.4.3	Cell Topologies.....	95
4.4.4	Performance Comparisons.....	96
4.4.5	Sensitivity Analysis	98
4.5	Summary	98
Chapter 5	Conclusions	108
5.1	Summary	108
5.2	Further work.....	111
	Bibliography	113
	Appendix A. Expected Cost for the Exponential Path Optimization Scheme	122
	Appendix B. Expected Cost for the Periodic Path Optimization Scheme	126
	Appendix C. Expected Cost for the Bernoulli Path Optimization Scheme	129
	Appendix D. Semi-Markov Decision Processes	132
	Glossary of Acronyms	141

List of Figures

Figure 1.1	(a) Path extension scheme, (b) Path rerouting scheme.	5
Figure 1.2	Two-phase handoff protocol.	7
Figure 1.3	Classification of different paging schemes.	15
Figure 1.4	Hexagonal cell configuration.	16
Figure 1.5	Structure of the thesis.	21
Figure 2.1	A 20-node random graph with node degree of 3; minimum distance between any two nodes is 15.	33
Figure 2.2	Exponential path optimization scheme.	41
Figure 2.3	Periodic path optimization scheme.	41
Figure 2.4	Bernoulli path optimization scheme.	42
Figure 2.5	Analytical and simulation results of the periodic path optimization scheme.	42
Figure 2.6	Analytical and simulation results of the exponential path optimization scheme.	43
Figure 2.7	Analytical and simulation results of the Bernoulli path optimization scheme.	43
Figure 2.8	Comparisons between different path optimization schemes.	44
Figure 2.9	Variation of the average call termination time.	44
Figure 2.10	Variation of the time between inter-switch handoff.	45
Figure 3.1	Timing diagram.	48
Figure 3.2	Expected total cost versus link cost rate	70
Figure 3.3	Expected number of path optimizations versus link cost rate	70
Figure 3.4	Expected total cost versus inter-switch handoff rate	71

Figure 3.5	Expected number of path optimizations versus inter-switch handoff rate . 71
Figure 3.6	Expected total cost versus call termination rate 72
Figure 3.7	Expected number of path optimizations versus call termination rate . . 72
Figure 3.8	Minimum expected total cost of the optimal policy versus inter-switch handoff rate for various distributions..... 73
Figure 3.9	Minimum expected total cost versus link cost rate for various inter- switch handoff rate distributions..... 73
Figure 3.10	Variation of the average call termination time. 74
Figure 3.11	Variation of the average time between inter-switch handoffs..... 74
Figure 4.1	Timing diagram..... 76
Figure 4.2	One-dimensional linear model. 80
Figure 4.3	Hexagonal cell configuration. 85
Figure 4.4	Update boundary derived from an arbitrary cell configuration. Location update is performed whenever the mobile terminal moves to any cells labeled with "B". 90
Figure 4.5	Expected total cost versus call arrival rate under various delay constraints..... 100
Figure 4.6	Optimal distance threshold versus call arrival rate under various delay constraints..... 100
Figure 4.7	Expected total cost versus cell crossing rate under various delay constraints..... 101
Figure 4.8	Optimal distance threshold versus cell crossing rate under various delay constraints..... 101
Figure 4.9	Expected total cost versus location update cost under various delay constraints..... 102
Figure 4.10	Optimal distance threshold versus location update cost under various delay constraints. 102

Figure 4.11	Optimal distance threshold versus cell crossing rate under various cell residence time distributions and delay constraints: (a) maximum paging delay constraint = 1, (b) maximum paging delay constraint = 2, (c) no paging delay constraint.	103
Figure 4.12	Expected total cost versus cell crossing rate under various cell residence time distributions and paging delay constraints: (a) maximum paging delay = 1, (b) maximum paging delay = 2, (c) no paging delay constraint.	104
Figure 4.13	Representation of a cellular network topology by a graph model. ...	105
Figure 4.14	Graph model with an average node degree of 4. The nodes represent the location of the base stations. An edge between two nodes represents those two base stations are neighbors to each other.	105
Figure 4.15	Optimal distance threshold distribution.	106
Figure 4.16	Cost ratio versus call arrival rate and location update cost under different cell crossing rate.	106
Figure 4.17	(a) Cost ratio versus percentage change in average time between call arrivals under different call arrival rate; (b) Cost ratio versus percentage change in average cell residence time under different cell crossing rate. .	107
Figure A.1	Timing diagram	122

List of Tables

Table 2.1	Summary of parameters used in simulation and analytical models.....	34
-----------	---	----

Acknowledgements

First of all, I would like to thank my supervisor, Professor Victor Leung, for his guidance, advice, and support throughout my graduate studies. I am also grateful to him for allowing me to attend and present our work at various conferences.

During different stages on my research, I have the privilege to work with two other people. The first one is Dr. Henry Chan who is now teaching at the Hong Kong Polytechnic University. The second one is Dr. Mark Lewis who is now teaching at the University of Michigan, Ann Arbor. I thank them sincerely for their comments on my work on path optimization in wireless ATM networks. Mark's guidance on Markov decision processes also gave me another perspective on location update problems.

A major part of this thesis is based on the results in Markov decision processes. I learned this branch of mathematics through a graduate course taught by Professor Martin Puterman. And I thank him for offering such an interesting course.

I would like to thank Professor Cyril Leung and Professor Son Vuong for serving on my dissertation committee.

The area of wireless/mobile ATM was first introduced to me by my former supervisor, Professor Jon Mark, at the University of Waterloo. He also wrote the recommendation letters for me so that I could have the opportunity to pursue my Ph.D. studies. For that, I thank him wholeheartedly.

I would also like to thank all the former and present members in the Communications group for their help and support, especially Hansen Wang and Peter Chong.

I wish to express my sincere thanks to my parents, Denise F. C. Chan and Stanley C. M. Wong, for their constant love and support throughout all these years.

Many thanks to my wife, Joyce Lee, for her love, understanding, and encouragement which helped me go through this difficult yet rewarding process.

This work was supported in part by the (1) Natural Sciences and Engineering

Research Council of Canada (NSERC) under a Postgraduate Scholarship and Grant No. OGP0044286; (2) Communications Research Centre, Industry Canada, under the Fessenden Postgraduate Scholarship; and (3) the University of British Columbia under the University Graduate Fellowship.

Chapter 1 Introduction

In recent years, there has been tremendous growth in the use of wireless cellular telephony for communications. Mobile subscribers are increasing at an exponential rate and will continue to increase in the near future. There are various wireless networks currently in use or proposed in the literature. Examples include Personal Communication Services (PCS), paging [78], wireless Asynchronous Transfer Mode (ATM) [28][29], mobile Internet Protocol (IP) [44], and mobile satellite. The next generation wireless broadband networks, such as the International Mobile Telecommunications 2000 (IMT-2000) [30] defined by the International Telecommunication Union (ITU) and the Universal Mobile Telecommunications System (UMTS) defined by the European Telecommunications Standards Institute (ETSI), aim to unify many of these diverse systems existing today into a seamless radio infrastructure capable of offering a wide range of services.

In order to utilize the radio spectrum efficiently, a cellular architecture is used in wireless networks. The geographical coverage area is partitioned into *cells*, each served by a base station. Mobile users and their terminals are connected to the network via the base stations. Cells can have different sizes: picocells are commonly used in indoor environments; microcells are used within cities; macrocells are used in highway and rural areas. Smaller cells use less power for transmissions and allow a greater frequency re-use.

One of the issues in mobility management is to *track the location of the users and their terminals between call arrivals*. Since mobile users are free to move within the coverage area, the network only maintains the approximate location of each user. When a connection needs to be established to a particular user, the network has to determine the user's exact location. The operation of informing the network about the current location of the mobile user is known as *location update*, and the operation of determining the exact location of the mobile user is called *terminal paging* or *searching*.

It is well-known that there is a trade-off between location update and paging. If the mobile terminal updates its location whenever it crosses a cell boundary, then the network can maintain its precise location. However, if the call arrival rate is low, the network wastes its resources by processing frequent update information and the mobile terminal wastes its power by transmitting the update signal. On the other hand, if the mobile terminal does not perform location update frequently, a large coverage area has to be paged when a call arrives, which wastes the radio bandwidth. Thus, *the central problem of location management is to devise algorithms which minimize the overall cost for location update and paging.*

Another issue in mobility management is to *maintain the connections when the mobile user moves from one access point to another.* The operation of transferring a mobile connection from one access point to another is known as *handoff* or *handover*. Handoffs for multimedia traffic differ from conventional voice handoffs in that a mobile user may have several active connections with different bandwidth requirements and quality-of-service (QoS) constraints. The handoff function should ensure that all ongoing connections are rerouted to another access point in a seamless manner. In other words, the design goal is to minimize any service disruption and degradation during and after the handoff process.

Since handoff may involve operations such as rerouting, resource allocation, and packet forwarding, the handoff protocols are network dependent. For example, the handoff operations for Mobile IP are different than those for Wireless ATM. On the other hand, the same location update and paging mechanism can be deployed in different networks with minor modifications. The only difference is how the update information is stored, disseminated, and retrieved within the network.

In our work, we focus on (1) *handoff management in wireless ATM networks*, and

(2) *location update in wireless cellular networks*. Among different kinds of wireless network technologies, wireless ATM is chosen because of its potential to support different kind of services with QoS guarantee. A number of prototypes reported in the literature such as the MAGIC WAND from Advanced Communications Technologies & Services (ACTS) [39] and the WATMnet from NEC [51] have demonstrated the feasibility of using ATM to support high speed services in the wireless domain.

There are two types of handoffs in wireless ATM networks, namely *intra-switch handoffs* and *inter-switch handoffs*. An intra-switch handoff occurs when the mobile terminal moves from one base station to another, and both base stations are connected to the same switch. An inter-switch handoff occurs when the mobile terminal moves to a new base station that is connected to another switch. In both cases, channel allocation is performed at the new base station. However, *connection rerouting* is also required for inter-switch handoffs. Several local connection rerouting protocols have been proposed in the literature. Although most of these protocols incur a low handoff delay, the end-to-end path after rerouting may become “suboptimal”, which implies an inefficient use of network resources. Part of our work focuses on optimizing the path after an inter-switch handoff.

The rest of this chapter is organized as follows: Sections 1.1-1.3 describe the related work on inter-switch handoff, location update, and terminal paging, respectively. Section 1.4 discusses the motivations and objectives of our work. Section 1.5 presents an overview of our contributions. Section 1.6 describes the scope of this thesis.

1.1 Related Work on Inter-Switch Handoff

Several connection rerouting protocols to facilitate inter-switch handoffs for wireless ATM networks have been proposed in the literature. In this section, we summarize these protocols and point out the strengths and weaknesses of each method.

1.1.1 Virtual Connection Tree

Call and handoff control for wireless ATM networks based on hierarchical grouping of backbone and wireless network resources is proposed in [1]. A *virtual connection tree* is a collection of base stations and ATM switching nodes and links. The root of the tree is an ATM switching node and the leaves of the tree are the base stations. The advantage of the proposed scheme is that as long as the mobile terminals stay within the virtual connection tree, they can move to any base stations without involving the network call processor. However, the proposed solution minimized the handoff processing delay at the expense of allocating more resources that may not be used. Over allocation of network resources is contrary to the original idea of flexibility and efficiency inherent in ATM networks. In addition, packet loss may occur and out-of-sequence packets may be received at the mobile terminal or the end station.

1.1.2 Multicast-Based Rerouting

In this method [25], the controlling switch establishes a multicast connection to the current serving base station and its neighboring base stations. When ATM packets arrive for the mobile terminal, the controlling switch will multicast those ATM packets to this multicast connection group. Thus when the mobile terminal moves to one of the neighboring base stations, ATM packets are already available. After each handoff, the controlling switch has to add the new neighboring base stations to the multicast connection group as well as delete the previous neighboring base stations from the group. The major drawback of this scheme is that extra buffer space has to be allocated to the neighboring base stations. In addition, with a large number of mobile terminals, frequent handoff increases the processing load of the controlling switch.

1.1.3 Path Extension

The rationale behind path extension is to extend the original connection to the switch to

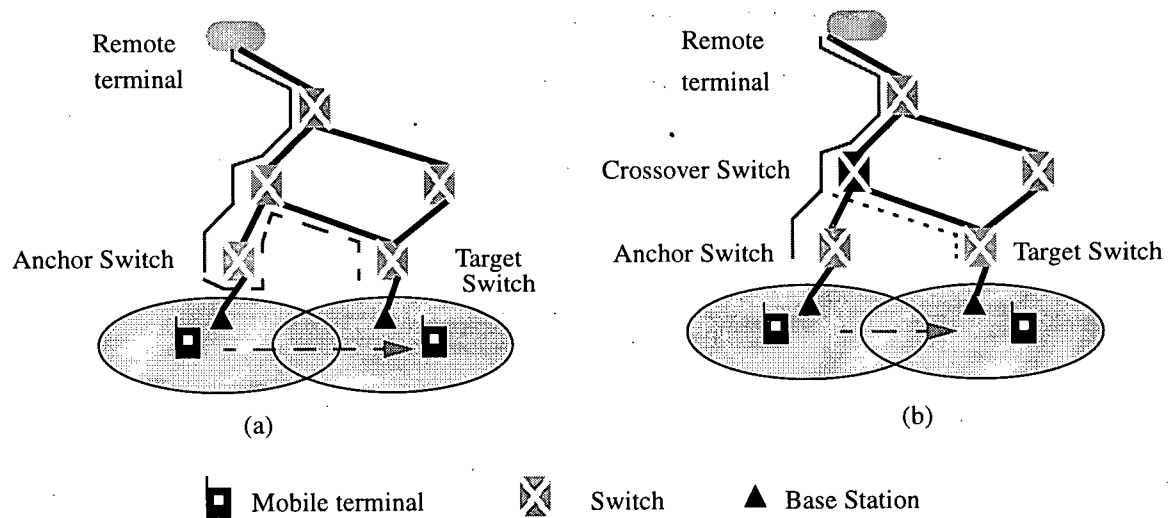


Figure 1.1 (a) Path extension scheme, (b) Path rerouting scheme.

which the new base station is connected. As shown in Figure 1.1 (a), the switches to which the original and new base stations are connected are usually referred, respectively, as the *anchor switch* and the *target switch* [6]. The path extension method extends the connection from the anchor switch to the target switch during handoff. The minimum hop path between these two switches is usually chosen as the extended path. The path extension scheme is fast and simple to implement. QoS degradations such as packet loss, duplicate packets, and mis-sequence packets do not occur. However, since the extended path is longer than the original one, certain QoS requirements, such as packet transfer delay and packet delay variation, may not be guaranteed after a handoff. In addition, data looping may occur when the mobile terminal moves back to the previous anchor switch later. This leads to an inefficient utilization of network resources.

1.1.4 Path Rerouting

Path rerouting can be considered as a generalization of the path extension scheme. In path

extension, the anchor switch extends the original connection to the target switch, while in path rerouting, any switch along the original connection can be selected to set up a branch connection to the target switch. As shown in Figure 1.1 (b), the switch chosen to perform this function is usually referred to as the *crossover switch* [61]. Depending on the performance criteria of the crossover switch discovery algorithms [61][14], the end-to-end path after rerouting may not be *optimal*. The optimal path is defined as the best path among a set of feasible paths that can satisfy the prescribed end-to-end QoS constraints.

The performance comparisons between various connection schemes including the virtual connection tree, path extension, path rerouting, and multicast-based rerouting are reported in [11]. Results indicate that the virtual connection tree incurs the lowest handoff delay at the expense of the highest bandwidth requirements. The multicast-based rerouting has the shortest disruption time but a relatively high buffer requirements. The performance of path extension and path rerouting lie in between the virtual connection tree and the multicast-based rerouting schemes. However, these analytical results are only true for a hierarchical symmetric tree network configuration. Performance comparisons between these rerouting schemes in another network topology may be different.

1.1.5 Two-Phase Handoff Scheme

The two-phase handoff protocol proposed in [65] combines the advantages of path extension and path rerouting. The two-phase handoff protocol consists of two stages: path extension and possible path optimization. Referring to Figure 1.2, path extension is performed for each inter-switch handoff, and path optimization is performed whenever it is necessary. During path optimization, the network determines the optimal path between the source and the destination (i.e., the path between the remote terminal and the current target switch in Figure 1.2) and transfers the user information from the old path to the new path. The major steps during path optimization execution generally involve [20]:

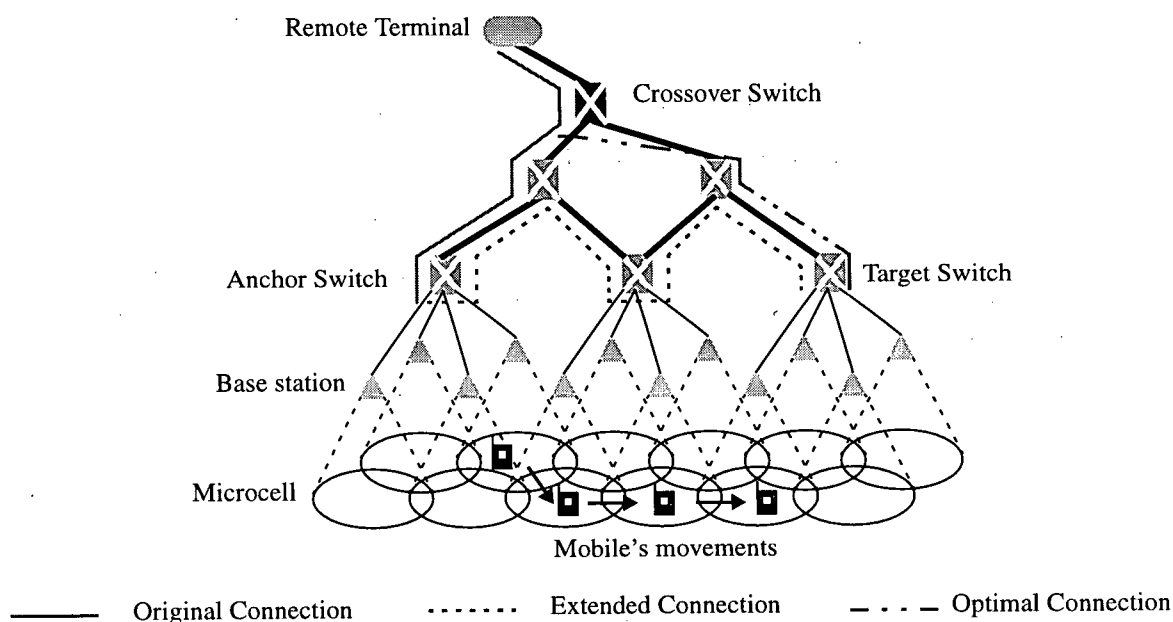


Figure 1.2 Two-phase handoff protocol.

1. Determining the location of the crossover switch;
2. Setting up a new branch connection;
3. Transferring the user information from the old branch connection to the new one;
4. Terminating the old branch connection.

Since the mobile terminal is still communicating over the extended path via the current base station while path optimization takes place, this gives enough time for the network to perform the necessary functions while minimizing any service disruptions. Notice that the path optimization process described above is not restricted to the two-phase handoff protocol. It can also be applied to other connection rerouting protocols where the location of the crossover switch is not the optimal one. In addition, when the mobile terminal moves to another switch during the execution of path optimization, path extension can still be used to extend the connection to the target switch.

In [47], an experimental testbed is used to compare the performance between

different rerouting schemes. Results indicate that for the two-phase handoff scheme, path extension from the old base station to the new base station in the first phase incurs a small handoff delay (6.5 ms). For CD-quality audio (128 kb/s), there is no packet loss during path extension and that only 1% of the subsequent path optimizations suffer a loss of a single packet.

To ensure a seamless path optimization, some important issues need to be addressed:

1. How to determine the location of the crossover switch?
2. How can the service disruptions be minimized during path optimization?
3. When and how often should path optimization be performed?

For the first issue, a crossover switch determination algorithm based on PNNI (Private Network-to-Network Interface) standard [10] was proposed in [20]. Five different crossover switch algorithms for wireless ATM local area networks are proposed in [61]. For the second issue, packet loss and packet mis-sequencing can be prevented by using appropriate signaling and buffering at the anchor and crossover switches during path optimization, see [20][70][71]. However, little work has been reported on the third issue.

1.2 Related Work on Location Update

In order to reduce its location uncertainty, each mobile terminal has to report its location from time to time. The location update procedure begins with an update message sent by the mobile over the uplink control channel, which is followed by some signaling procedures which update the database. Location update algorithms can be divided into two main groups: *static* and *dynamic*. In a static algorithm, location update is triggered based on the topology of the network. In a dynamic algorithm, location update is based on the user's call and mobility patterns. In this section, we summarize various location update schemes currently in use or proposed in the literature.

1.2.1 Location Area (LA)-based

In this update algorithm, the coverage area is partitioned into a number of location areas (LAs). Each LA contains a group of cells. All base stations within the same LA broadcast the identifier (ID) of its LA periodically. Each mobile terminal compares its registered LA ID with the current broadcast LA ID. Location update is triggered if the two IDs are different. Upon a call arrival for a particular mobile terminal, all the cells within its current LA are polled simultaneously, ensuring a success within a single step. The LA-based update scheme is widely adopted by the current cellular systems, including the EIA/TIA IS-41 [22] and the GSM (Global System for Mobile Communication) networks [41]. The main drawback of this scheme is that for an LA with a large number of cells, significant amount of radio bandwidth is consumed in paging for each call arrival. In addition, mobile terminals located close to an LA boundary may perform excessive location updates as they move back and forth between two LAs.

1.2.2 Selective LA Update

The rationale behind the *selective LA update scheme* [58] is that a daily commuter may cross a number of LAs on his way to and from work. However, he may only stay in some LAs for a very short period of time. Rather than performing location update whenever he crosses a new LA, the update process at certain LAs can be skipped. In [58], an analytical model is introduced in which the interconnections of the LAs are characterized by a graph model. The movement model is Markovian. The residence time in each LA follows a geometric distribution. A genetic algorithm is used to obtain the near-optimal solutions. For low residing probability in certain LAs and high update cost, results show that this scheme incurs a lower location management cost than the conventional LA-based scheme.

For implementation, information regarding the transition probabilities and residence time is required. To estimate the transition probabilities between LAs for a

particular user, his movements throughout the day can be observed over long periods of time. Since the LA-based update scheme is used in the current PCS networks, information about the frequency of his transition from one LA to another can be retrieved from the database.

1.2.3 Profile-based

The goal of the *profile-based location update scheme* [45] (also known as the *alternative location strategy* [60]) is to reduce the update cost by taking advantage of the user's mobility pattern. The network maintains a profile for each user, which includes a sequential list of the most likely LAs that the user is located at different time periods. This list is sorted from the most to the least likely LA where a user can be found. When a call arrives, the LA is being paged sequentially. As long as the mobile terminal moves between LAs within the list, no update is necessary. Location update is performed only when the mobile terminal moves to a new LA which is not on the list. The list may be derived from the user's movement history.

1.2.4 Movement-based

In the *movement-based update scheme* [4][12], each mobile terminal counts the number of boundary crossings between cells incurred by its movements. Location update is performed when this number exceeds a predefined movement threshold M (e.g., $M = 6$). This scheme allows the dynamic selection of the movement threshold on a per-user basis.

For implementation, the mobile terminal only needs a counter to count the number of cell boundary crossings. The counter is reset whenever it reaches the movement threshold. The *cell identification code* proposed in [43] can also be used to identify the boundary crossing.

In [4], an analytical model is introduced to determine the optimal movement threshold. The model is applicable for mesh and hexagonal cell configurations under the

assumptions of a general cell residence time distribution and symmetric random walk movement pattern. The maximum paging delay constraint is considered and a shortest-distance-first order paging scheme is used.

1.2.5 Timer-based

In the *timer-based update scheme* [12][53], each mobile terminal updates its location every T time units (e.g., $T = 1$ hour). This scheme does not require the mobile terminal to record or process location information during the time between updates. For implementation, the timer threshold can be programmed into the mobile terminal by a hardware or software timer. An analytical model is introduced in [53] to study the timer-based scheme. Assuming the Gaussian distribution on user location probability and Poisson call arrivals, the update period which minimizes the cost of location update and paging is derived. Results show that the timer-based scheme performs substantially better than the LA-based scheme.

A variation of the timer-based scheme, called the *adaptive threshold scheme*, is proposed in [42]. The mobile terminal transmits an update message every T' time units, where the parameter T' (referred as the *location registration threshold*) is not a constant, but varies with the current signaling load on the uplink control channel of the base station. Numerical results, under the assumptions of a linear cell configuration and random walk movement pattern, show that the adaptive threshold scheme has a better performance than the static timer-based scheme.

1.2.6 Distance-based

In the *distance-based update scheme* [12][27][38], each mobile terminal tracks the distance it has moved (in number of cells) since the last update, and transmits an update signal whenever the distance exceeds a certain threshold. For implementation, the mobile terminal requires some knowledge about the cell topology. In order to identify the cells

within the distance threshold or the cells along the distance threshold boundary, the mobile terminal needs to download a set of these cell IDs after each location update.

The distance-based scheme has been studied extensively. In [12], the authors compared the movement, timer, and distance-based schemes, under the assumptions of random walk mobility movements and a ring topology of cells. Analytical results show that the distance-based scheme gives the lowest location management cost. In [38], the distance-based update scheme is formulated as an optimization problem. The goal is to minimize the expected total cost for update and paging between call arrivals. Under a linear cell configuration and symmetric random walk movement pattern, the optimal distance threshold is determined by dynamic programming. In [27], an iterative approach is used to compute the optimal distance threshold in a hexagonal cell configuration under the assumption of symmetric random walk mobility pattern.

1.2.7 Predictive Distance-based

In the *predictive distance-based update scheme* [35], the mobile terminal reports both its location and velocity during the update process. Based on the above information, the network determines the probability density function of the mobile's location, which is used to predict the mobile's location in future time. This prediction information is made available to both the network and the mobile terminal. The mobile terminal checks its position periodically and performs location update whenever its distance exceeds the threshold distance measured from the predicted location. Upon a call arrival, the network pages the mobile terminal starting from the predicted location (which may be the one that performed the last update) and outward, in a shortest-distance-first order, until the mobile terminal is found.

For performance analysis, a Gauss-Markovian process is used to model the user's mobility pattern. The Gauss-Markov model captures the correlation of the mobile's

velocity in time, and can represent different user mobility patterns, including the random walk and the constant velocity fluid-flow model. Numerical results, under the assumptions of an infinite one-dimensional linear model and Poisson call arrivals, show that the predictive distance-based scheme has a better performance than the non-predictive one.

1.2.8 State-based

In the *state-based update scheme* [55], the mobile terminal determines whether to perform location update based on its current *state*. The state information can include the time elapsed or the number of cell crossings since the last update, the cell-distance between the current and last registered locations, or some other criteria. Thus, maintaining different state information corresponds to different location update schemes. In [55], a state-based update scheme is analyzed where the system state includes the current location and the time elapsed since the last update. A time-varying Gaussian process [54] is used to model the user's movement. The sub-optimal solution for the average cost of location update and paging under no paging delay constraint is obtained by a greedy method. Results show that the state-based update scheme achieves a 10% improvement in average cost compared to the timer-based scheme.

1.2.9 LeZi Update

The idea of the *LeZi update* (pronounced as "lazy update") algorithm [16] is based on a compression algorithm proposed by Ziv and Lempel. The LeZi update algorithm can be considered to be a *path-based* update scheme in which the *movement history* rather than the current location is sent in an update message. The movement history consists of a list of *zone* (i.e., LA or cell) IDs where the mobile terminal has crossed after the last update. The network database maintains the movement history in a compact form by a *trie* or *digital search tree*. This trie can be considered to be a part of the user's profile. Upon a call arrival, selective paging based on the trie information is used to locate the mobile terminal.

1.3 Related Work on Terminal Paging

Terminal paging is the process in which the network determines the exact location of a particular mobile terminal. In each *polling cycle* [27] or *search iteration*, polling signals are sent over the downlink control channel to all cells where the mobile terminal is likely to be present. All the mobile terminals listen to the page message and only the target mobile terminal sends a response message back over the uplink control channel. In each polling cycle, there is a *timeout* period. If the target mobile terminal replies before the timeout, then the paging process is terminated. Otherwise, another group of cells are chosen in the next polling cycle. To avoid call dropping, the mobile terminal must be located within an allowable *time constraint*. The *maximum paging delay* corresponds to the maximum number of polling cycles allowed to locate the mobile terminal. For example, if the maximum paging delay is equal to 1, the mobile terminal has to be located within a single search iteration.

Since radio bandwidth is consumed during the paging process, the paging cost is proportional to the number of polling cycles, as well as the number of cells being polled in each cycle. The paging area depends on the information provided by the location update function. The paging cost can be reduced by predicting the current location of the mobile terminal. In this section, we summarize different paging strategies proposed in the literature. A classification of various paging strategies is shown in Figure 1.3.

1.3.1 Blanket Polling

In *blanket polling*, all the cells within the LA in which the mobile terminal is located are being polled simultaneously when a call arrives. Since the mobile terminal is located within the LA, its location can be determined within a single polling cycle. This paging strategy is currently deployed on top of the LA-based update scheme in the existing PCS networks. The major drawback of blanket polling is that since the number of cells within a

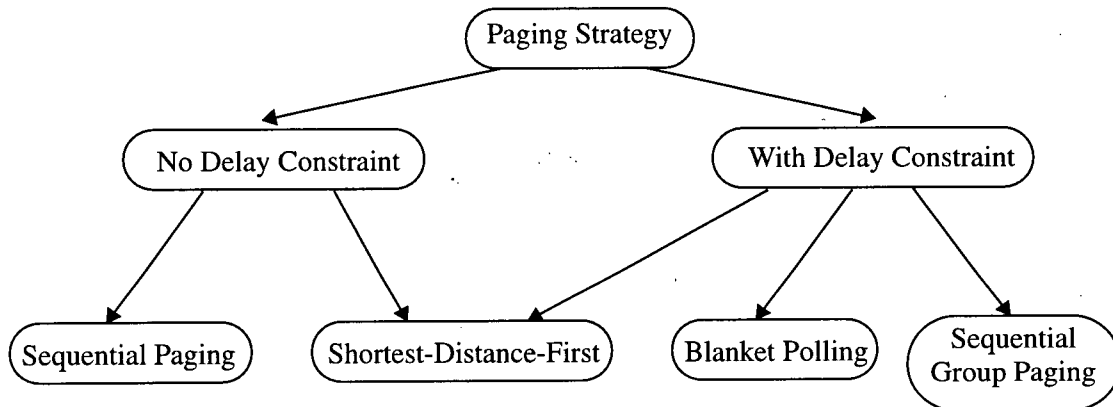


Figure 1.3 Classification of different paging schemes.

typical LA is large, the paging cost is very high.

1.3.2 Shortest-Distance-First

In this paging strategy, the network pages the mobile terminal starting from the cell where the mobile terminal last updated its location and moving outward, in a shortest-distance-first order. The distance is measured in terms of the number of cells away from the last update location. If a threshold-based update scheme (e.g., distance or movement) is used, the paging area or the residing area of the mobile terminal is bounded [27]. The mobile terminal can be located within a fixed number of polling cycles. The paging delay constraint can be incorporated by grouping cells with different distances for each polling cycle. Various location update algorithms such as the distance and movement-based schemes have used this paging strategy to determine the location management cost for performance comparisons (e.g., [27][35][38][43]).

To illustrate the mechanism of this paging strategy, consider Figure 1.4 in which the cell topology is hexagonal. Suppose the cell labeled 0 is the location where the mobile terminal performed the last update. Assume the distance-based update scheme is used with a distance threshold equal to 4. That is, location update is performed whenever the mobile

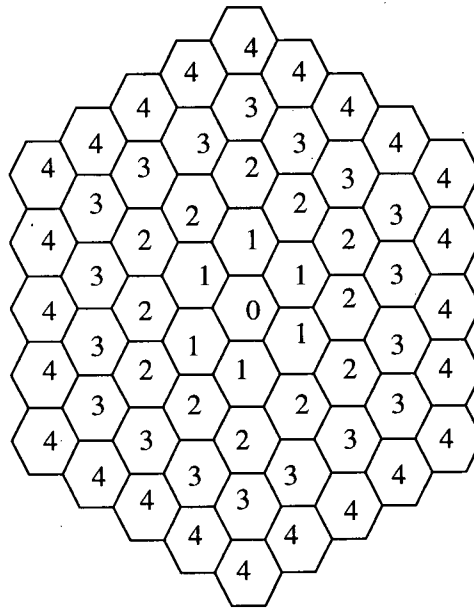


Figure 1.4 Hexagonal cell configuration.

terminal moves to any cell labeled 4. With no paging delay constraint, the polling sequence is $\{0, 1, 2, 3\}$. That is, cell 0 is polled first. If no response is received from the mobile terminal after a timeout, all the cells labeled 1 are polled in the next polling cycle. This continues until either the mobile terminal sends a response message to the base station or all the cells within the sequence list have been polled. With delay constraints, cells with different labels or distances may be polled as a group in a polling cycle. For example, if the maximum paging delay is equal to 3, the paging sequence list can be $\{\{0, 1\}, \{2\}, \{3\}\}$. That is, cells with label 0 or 1 are polled in the first polling cycle. Cells labeled 2 and 3 are polled in the second and third polling cycles, respectively.

1.3.3 Sequential Paging based on User's Location Probability

In this paging strategy, the current location of the mobile terminal is predicted based on its location probability distribution. Polling signals are sent only to those cells where the user is likely to be present. An intuitive result derived in [52] states that: "*Given the probability*

distribution on user location, under no paging delay constraint, the paging cost is minimized by sequentially polling the cells in decreasing order of probability". Clearly, the uniform location distribution gives the highest paging cost and delay.

When there is a maximum paging delay constraint, a group of cells can be polled together in each polling cycle. Dynamic programming [46] can be used to determine the optimal group size which minimizes the paging cost. In [52], the authors obtained the optimal paging sequence that results in the minimum paging cost with average paging delay constraint. The sequential paging strategy has been used in [53][55] for performance analysis of the timer and state-based update schemes.

1.3.4 Velocity Paging

The *velocity paging scheme* [68] aims to reduce the paging cost by decreasing the size of the paging area. The goal is achieved by grouping the users into different *velocity classes*, based on their velocity at the location update instant. When a call arrives, the paging area is dynamically generated based on the user's last registration time and the velocity class index. The velocity paging scheme can be deployed on top of other location update algorithms. To implement this paging strategy, information such as the mobile's last known location, velocity class index, and the last registration time is required at the user's database profile. When the velocity paging scheme is combined with the movement-based update algorithm, numerical results [68] indicate that this combined scheme may not always result in a reduction in cost as compared to the LA-based update scheme with blanket polling. The authors determine the range of cell radius under which this combined scheme should be used based on system parameters.

1.4 Motivations and Objectives

To facilitate inter-switch handoff in wireless ATM networks, several connection rerouting

protocols have been reported in the literature. Although most of these protocols incur a low handoff delay, the path after rerouting may become suboptimal, which implies an inefficient use of network resources. Path optimization is required to reroute the connection to a more efficient route. However, the question of when path optimization should be performed has not been addressed in the literature, and is investigated in the first part of the thesis.

Our work is motivated by the fact that path optimization does not have to be performed after each inter-switch handoff. Although path optimization can increase the network utilization by rerouting the connection to a more efficient route, transient QoS degradations such as packet loss and an increase in packet delay variation may occur. In addition, if there are a large number of mobile users with high movement patterns, performing path optimization after each path extension will increase the processing load of certain switches and the signaling load of the network. We believe the decision of performing path optimization should be based on several factors including: (1) the amount of network resources (e.g., bandwidth) utilized by the connection, (2) the QoS requirements of the connection, (3) the remaining time of the connection, or (4) signaling load of the network.

Another part of our work focuses on the determination of the optimal update boundary for the distance-based location update algorithm. Although it has been shown that the distance-based update algorithm has a better performance than the LA, movement, and timer based update schemes, the models used to determine the optimal distance threshold are often under certain unrealistic assumptions. First of all, structured cell configurations are commonly used. For example, mesh or hexagonal cell configurations are used in two-dimensional models (e.g., [27]), and a linear model is used in the one-dimensional case (e.g., [12][38]). Although these cell topologies simplify the analytical

computation, they do not give an accurate representation of a realistic cellular network topology, where the size of each cell depends on the transmit power, receiver sensitivity, antenna radiation pattern, and propagation environment, and the number of neighboring cells varies from cell to cell. We believe a graph model such as the one proposed in [16][58] is more appropriate to characterize the topology of a cellular network.

Another commonly used assumption is related to the cell residence time distribution. The cell residence time denotes the amount of time that the mobile terminal stays in a particular cell before moving to another one. Most of the work assumed the cell residence time follows a geometric (or exponential) distribution. And the distribution is assumed to be independent and identically distributed (i.i.d.) for all cells. The major limitation of the i.i.d. geometric residence time assumption is that it does not capture an accurate representation of individual user mobility patterns, where a user may stay at certain locations (e.g., his home or office) for a relatively long period of time.

The last assumption is related to the movement models. The symmetric random walk is commonly used to characterize individual movement behavior. When the mobile user leaves a cell, there is an equal probability that the user will move to any neighboring cells. Although the random walk model simplifies the analysis, its main drawback is that the direction of the mobile user is not taken into account. In general, a mobile user usually travels with a destination in mind. Thus, the mobile's location in the future is likely to be correlated with its movement history.

Another goal of our work is to develop an analytical model which can eliminate some of the unrealistic assumptions commonly used and will allow us to determine the optimal distance threshold (or update boundary) for the distance-based location update algorithm.

1.5 Contributions of the Thesis

The main contributions of this dissertation are as follows:

- **Exponential, periodic, and Bernoulli path optimization schemes:** To address the issue of when and how often path optimization should be performed, we propose three heuristics for the initiation of path optimization. These heuristics may not be optimal but they are simple to implement. Given the cost and mobility parameters, closed-form solutions of the expected cost per call and the optimal operating point are derived for each scheme.
- **Optimal policy to initiate path optimization:** We further investigate the path optimization problem and propose a stochastic model to determine the optimal time to perform path optimization. Link cost and signaling cost functions are introduced to capture the trade-off between the network resources utilized by a connection and the signaling and processing load incurred on the network. Different link cost functions can be assigned to different service classes with different bandwidth requirements. Different signaling cost functions can be used based on the complexity of the path optimization procedures and the signaling load of the network. The time between inter-switch handoffs can follow any arbitrary general distributions. A stationary deterministic optimal policy is obtained. Results indicate that the optimal policy derived from our model has a better performance compared to some other heuristics.
- **Distance-based location update algorithm:** We propose a novel stochastic model to analyze the distance-based location update algorithm. The model is formulated as a semi-Markov decision process. There is a cost function associated with location update and another cost function associated with terminal paging. Unlike other models, our model is applicable to arbitrary cell topologies and the cell

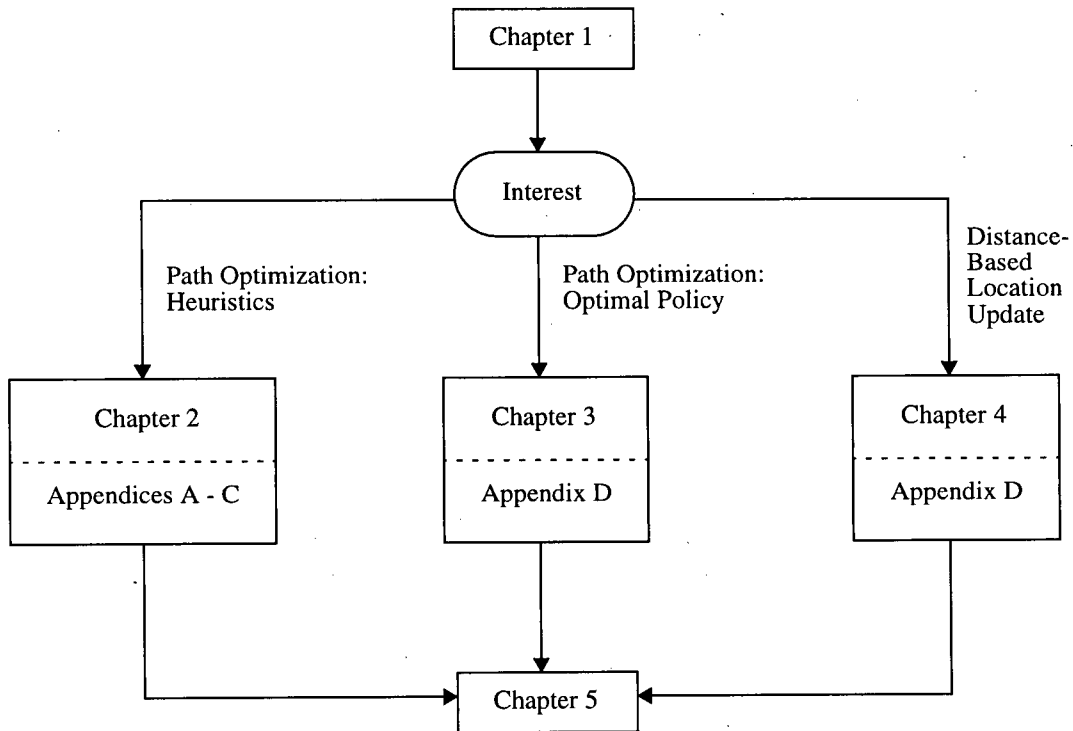


Figure 1.5 Structure of the thesis.

residence time can follow general distributions. We consider Markovian movement patterns in which the probability that the mobile user moves to a particular neighboring cell can depend on the location of the current cell or a list of cells recently visited. Results indicate that the distance thresholds determined from our model have a better performance than those derived from a hexagonal cell configuration with random walk movement pattern.

1.6 Organization of the Thesis

Depending on the specific interest the reader may choose one of the following *chapter routes* (see Figure 1.5). The thesis is organized as follows: In Chapter 2, we propose and analyze three path optimization schemes to facilitate the two-phase handoff protocol. A

discrete time analytical model and a discrete event simulation model are proposed to evaluate the performance of different path optimization schemes. We present performance comparisons between these schemes. In Chapter 3, we propose a stochastic model to determine the optimal time to perform path optimization. We describe the optimality equations, the value iteration algorithm, and the structure of the optimal policy. We present numerical results and compare the optimal policy with four heuristics. In Chapter 4, we propose a stochastic model to determine the optimal update boundary for the distance-based location update algorithm. We present numerical results for the hexagonal cell configuration and random graph topology. Finally, Chapter 5 concludes the thesis with a summary of the presented work and some suggestions for future work.

*"Take a method and try it. If it fails, admit it frankly
and try another. But above all, try something."*

Franklin Delano Roosevelt, 1881-1945.

Chapter 2 Path Optimization for Inter-Switch Hand-off: Heuristics

2.1 Introduction

In this chapter, we address the issue of when to perform path optimization for the two-phase handoff protocol by proposing and analyzing three path optimization schemes, namely: exponential, periodic, and Bernoulli [72]-[75]. These schemes may not be optimal but they are simple to implement. An analytical model and a simulation model are used to analyze the performance of these path optimization schemes. For the discrete time analytical model, closed-form solution of the expected cost per call and the optimal operating point are derived for each path optimization scheme. The analytical results are corroborated by simulations based on non-hierarchical random graphs.

The rest of this chapter is organized as follows. The proposed path optimization schemes are described in Section 2.2. The discrete time analytical model and the discrete event simulation model proposed for evaluating the performance of different path optimization schemes are explained in Sections 2.3 and 2.4, respectively. Section 2.5 discusses the analytical and simulation results. A summary is given in Section 2.6. The proofs of the expected cost per call for the proposed path optimization schemes are shown in the Appendices A-C.

2.2 Path Optimization Schemes

As mentioned in Chapter 1, the end-to-end path after connection rerouting may not be optimal. For a large number of mobile connections within the fixed network, these sub-optimal paths can lead to an inefficient use of network resources. In this section, we discuss different methods to initiate path optimization. Path optimization schemes are grouped into four types, namely, *QoS-based*, *network-based*, *timer-based*, and *handoff-based*. These path optimization initiation schemes are described below.

2.2.1 QoS-based

As the name implies, QoS-based path optimization schemes trigger path optimization of each mobile connection based on its current QoS measures. For example, path optimization can be initiated if the number of hops of the path is greater than a certain number, or if the average end-to-end packet transfer delay bound is violated. To implement those QoS-based path optimization schemes, information about the quality of the current path in terms of the defined QoS measures (e.g., hop count, current average delay, delay variation) must be maintained by the network.

2.2.2 Network-based

Network-based path optimization schemes trigger path optimizations for a group of connections based on the existing traffic load of a switch or the utilization of the network. For example, a network switch can initiate path optimization for a group of mobile connections whenever the new call dropping probability of a certain traffic class exceeds a particular threshold. During path optimization, a number of connections will be rerouted to some other switches, thereby reducing the traffic load of that switch.

2.2.3 Timer-based

For the timer-based schemes, path optimizations are triggered at time instants which are independent of the current QoS of the connection or the utilization of the network. The time instants can be deterministic or random. For example, the time between path optimization can be based on some random processes. In addition, it can also be a function of the velocity of the mobile terminal, the dwell time, and the residual service time of the mobile connection. In this chapter, we analyze two of these timer-based schemes which are simple to implement.

1. *Periodic path optimization scheme*: path optimization is performed every T time units.
2. *Exponential path optimization scheme*: The time between path optimizations is modeled as an exponentially distributed random variable with mean $1/\nu$.

The periodic scheme has been proposed within the ATM Forum wireless ATM working group [8] to facilitate inter-switch handoff. Notice that both periodic and the exponential schemes trigger path optimization regardless of whether or not there is an inter-switch handoff. Thus, unnecessary path optimizations may be performed for stationary mobile connections.

2.2.4 Handoff-based

The handoff-based path optimization schemes trigger path optimization for each mobile connection based on some criteria after each inter-switch handoff. For example, it can be a function of the number of previous handoffs, the velocity of the mobile terminal, or the residual service time of the connection. In this chapter, we also analyze the following scheme:

1. *Bernoulli path optimization scheme*: After each path extension, there is a probability p , $0 \leq p \leq 1$, such that path optimization is performed.

For the Bernoulli scheme, path optimization may only occur on condition that

there is an inter-switch handoff. Thus, path optimization is never triggered after each handoff if $p = 0$. On the other hand, path optimization is always performed after each path extension if $p = 1$.

Path optimization can also be invoked based on a combination of the above schemes. In the remainder of this chapter, we will compare the performance of the periodic, exponential, and the Bernoulli path optimization schemes, and determine how to assign the optimal values for v , T and p given a set of call and mobility parameters.

2.3 Analytical Model

In this section, we present a discrete time analytical model proposed to evaluate the performance of different path optimization schemes [73]. Our proposed model is a generalization of the model in [48]. Events occur at each fixed time interval. The time interval can be minutes or seconds. When the time interval is sufficiently small (e.g., fraction of a second), this model is approximately the same as a continuous time model. For each mobile connection, the following events can occur at each time interval: (1) path extension due to inter-switch handoff; (2) path optimization; and (3) call termination. At each time interval, we let:

- λ denote the probability that there is an inter-switch handoff (and thus a path extension);
- μ denote the probability that the call terminates.

Thus, the time between inter-switch handoff and the call duration follow geometric distribution with mean $1/\lambda$ and $1/\mu$, respectively. For each mobile connection, we let:

- L denote the number of links between the source and the destination during call setup;

- H denote the number of links between the anchor switch and the target switch during path extension.

The random variables L and H are assumed to be independent, each having a general distribution with mean \bar{L} and \bar{H} , respectively. During path optimization, the path between the current anchor switch and the source is recomputed. We assume the number of links of this optimal path is a random variable also with mean \bar{L} .

The performance metric is the *expected cost per call*, which captures the amount of network resources used and the processing and signaling load of the network. The total cost per call is the sum of the link cost and the signaling cost due to inter-switch handoff. More precisely, we let:

- C_{link} denote the link cost per unit time interval per link;
- C_{PE} denote the signaling cost for each path extension event;
- C_{PO} denote the signaling cost for each path optimization event.

The term C_{link} captures the amount of resources (e.g., bandwidth) used by the connection. Different traffic classes can be assigned different values for C_{link} . The total link cost of the call is a function of time and the number of links. The term C_{PE} captures the cost for setting up the extended path between the anchor switch and the target switch. The term C_{PO} includes the cost for (1) locating the crossover switch; (2) setting up the new branch connection; (3) terminating the old branch connection; and (4) updating the connection server about the status of the existing route. Since path optimization is more complex than path extension, the cost of performing path optimization is higher than the cost of performing path extension. Thus, it is reasonable to assume that $C_{PO} \gg C_{PE}$. As we are only interested in the relative performance between different path optimization schemes, other signaling costs including the costs for call setup and termination as well as

the costs for location update and intra-switch handoff are not considered.

In the following subsections, we describe the closed-form expressions of the expected cost for the exponential, periodic, and Bernoulli path optimization schemes. The derivation of these expressions are presented in the Appendices A-C.

2.3.1 Exponential Path Optimization Scheme

In the exponential scheme, the time between path optimization is modeled as an exponentially distributed random variable with mean $1/v$. In the discrete time model, the time between path optimization is modeled as a geometrically distributed random variable. At each time interval, we let v denote the probability that path optimization is invoked. The expected cost per call, denoted as $E[C_{\text{exponential}}]$, is:

$$E[C_{\text{exponential}}] = \frac{\bar{L}}{\mu} C_{\text{link}} + \frac{\lambda(1-\mu)(1-v)\bar{H}}{\mu[1-(1-\mu)(1-v)]} C_{\text{link}} + \frac{(1-\mu)}{\mu} (\lambda C_{PE} + v C_{PO}). \quad (2.1)$$

The proof of equation (2.1) is shown in Appendix A. In (2.1), the first term is the link cost of the call if the mobile terminal remains connected to the same switch during its connection lifetime. The second term is the increase in link cost due to the path extension associated with each inter-switch handoff. The third term is the signaling cost due to path extension. And the last term is the signaling cost due to path optimization. When $v = 0$,

$$E[C_{\text{exponential}}]_{v=0} = \frac{\bar{L}}{\mu} C_{\text{link}} + \frac{\lambda(1-\mu)\bar{H}}{\mu[1-(1-\mu)]} C_{\text{link}} + \frac{\lambda(1-\mu)}{\mu} C_{PE}. \quad (2.2)$$

The exponential path optimization scheme reduces to pure path extension. That is, only path extension is performed for each inter-switch handoff and there is no path optimization. On the other hand, if $v = 1$,

$$E[C_{\text{exponential}}]_{|v=1} = \frac{\bar{L}}{\mu} C_{\text{link}} + \frac{\lambda(1-\mu)}{\mu} C_{PE} + \frac{(1-\mu)}{\mu} C_{PO}. \quad (2.3)$$

In this case, path optimization is invoked at each time interval.

By taking the derivative of (2.1) with respect to v and setting the expression to zero, we obtain the optimal v , denoted as v_{opt} , such that the expected cost is minimized.

The expression is:

$$v_{\text{opt}} = \frac{1}{1-\mu} \left[\sqrt{\lambda \bar{H} \frac{C_{\text{link}}}{C_{PO}}} - \mu \right]. \quad (2.4)$$

Since v is a probability, it has to be within the range $0 \leq v \leq 1$. Thus, based on (2.4), the sub-optimal value of v is:

$$v_{\text{sub-opt}} = \begin{cases} 0, & \text{if } v_{\text{opt}} \leq 0 \\ v_{\text{opt}} & \text{if } 0 \leq v_{\text{opt}} \leq 1 \\ 1, & \text{if } v_{\text{opt}} \geq 1. \end{cases} \quad (2.5)$$

2.3.2 Periodic Path Optimization Scheme

The periodic scheme invokes path optimization periodically. In the discrete time model, we let positive integer k denote the time period, measured in number of discrete time intervals, between successive path optimizations. The expected cost per call, denoted as

$E[C_{\text{periodic}}]$, is:

$$\begin{aligned} E[C_{\text{periodic}}] = & \frac{\bar{L}}{\mu} C_{\text{link}} + \frac{\lambda(1-\mu)\bar{H}}{\mu^2[1-(1-\mu)^k]} [1 - k(1-\mu)^{k-1} + (k-1)(1-\mu)^k] C_{\text{link}} \\ & + \frac{\lambda(1-\mu)}{\mu} C_{PE} + \frac{(1-\mu)^k}{1-(1-\mu)^k} C_{PO}. \end{aligned} \quad (2.6)$$

The proof for equation (2.6) is shown in Appendix B. Similar to (2.1), the first term in (2.6) is the link cost of the call if no inter-switch handoff is generated. The second

term is the increase in link cost due to path extension at each inter-switch handoff. The last two terms are the signaling cost due to path extension and path optimization, respectively.

When $k = 1$ in (2.6),

$$E[C_{periodic}]|_{k=1} = \frac{\bar{L}}{\mu} C_{link} + \frac{\lambda(1-\mu)}{\mu} C_{PE} + \frac{(1-\mu)}{\mu} C_{PO}. \quad (2.7)$$

Equations (2.3) and (2.7) are the same. That is, path optimization is performed at each time interval.

By taking the derivative of (2.6) with respect to k and setting the expression to zero, the optimal k , denoted as k_{opt} , such that the expected cost is minimized, is given by the solution of:

$$\lambda \bar{H} C_{link} [1 + (1-\mu)^k - k \ln(1-\mu)] = \mu C_{PO} \ln(1-\mu). \quad (2.8)$$

Although the value of k that satisfies (2.8) is always positive, it may not be an integer. Thus this value needs to be rounded to the closest integer.

2.3.3 Bernoulli Path Optimization Scheme

For the Bernoulli scheme, path optimization is performed with probability p after each path extension. As derived in Appendix C, the expected cost of the call, denoted as $E[C_{Bernoulli}]$, is:

$$\begin{aligned} E[C_{Bernoulli}] = & \frac{\bar{L}}{\mu} C_{link} + \frac{\lambda(1-p)(1-\mu)\bar{H}}{\mu[1-(1-\lambda p)(1-\mu)]} C_{link} \\ & + \frac{\lambda(1-\mu)}{\mu} (C_{PE} + p C_{PO}). \end{aligned} \quad (2.9)$$

In (2.9), the first term is the link cost of the call if the mobile terminal moves under the coverage of the same edge switch during its connection lifetime. The second term is the additional link cost due to path extension at each inter-switch handoff. The last term is

the signaling cost due to path extension and path optimization. When $p = 0$, equation (2.9) reduces to the same expression as (2.2), as follows:

$$E[C_{Bernoulli}]|_{p=0} = \frac{\bar{L}}{\mu} C_{link} + \frac{\lambda(1-\mu)\bar{H}}{\mu[1-(1-\mu)]} C_{link} + \frac{\lambda(1-\mu)}{\mu} C_{PE}. \quad (2.10)$$

The Bernoulli scheme reduces to pure path extension. On the other hand, when $p = 1$,

$$E[C_{Bernoulli}]|_{p=1} = \frac{\bar{L}}{\mu} C_{link} + \frac{\lambda(1-\mu)}{\mu} (C_{PE} + C_{PO}). \quad (2.11)$$

Path optimization is performed after every path extension.

By taking the derivative of (2.9) with respect to p and setting the expression to zero, we obtain the optimal p , denoted as p_{opt} , such that the expected cost is minimized, as follows:

$$p_{opt} = \frac{1}{\lambda(1-\mu)} \left\{ \sqrt{[\mu + \lambda(1-\mu)] \bar{H} \frac{C_{link}}{C_{PO}}} - \mu \right\}. \quad (2.12)$$

Since p is a probability, it has to be within the range $0 \leq p \leq 1$. Thus, based on (2.12), the sub-optimal value of p is:

$$p_{sub-opt} = \begin{cases} 0, & \text{if } p_{opt} < 0 \\ p_{opt} & \text{if } 0 \leq p_{opt} \leq 1 \\ 1, & \text{if } p_{opt} > 1. \end{cases} \quad (2.13)$$

2.4 Simulation Model

In the simulation model, a wireless ATM network is modeled as a non-hierarchical random graph. Random graphs have been used to model ATM networks [24]. Different variations of random graph models have also been proposed to model the topology of the Internet [19][80]. The generation of a non-hierarchical random graph consists of the following steps [19]:

1. N nodes are randomly distributed over a rectangular coordinate grid. Each node is placed at a location with integer coordinates. A minimum distance is specified so that a node is rejected if it is too close to another node. The Euclidean metric is then used to calculate the distance $\alpha(i, j)$ between each pair of nodes (i, j) .
2. A fully connected graph is constructed with the link weight equal to the Euclidean distance.
3. Based on the fully connected graph, a minimum weight spanning tree is constructed.
4. To achieve a specified average node degree¹ of the graph, edges are added one at a time with increasing distance.

If node i and j are connected, then the link weight, denoted as ω_{ij} , is assumed to be equal to:

$$\omega_{ij} = \alpha(i, j) + \beta \quad (2.14)$$

where β is a uniformly distributed random variable in the range $0 \leq \beta \leq \beta_{max}$. In (2.14), the first term can be interpreted as the propagation delay of the link, and the second term approximately models the queueing delay of the link.

A 20-node random graph generated from the above model is shown in Figure 2.1. The size of the rectangular coordinate grid is 100×100 . The minimum distance between any two nodes is 15. The average node degree of the graph is 3. For (2.14), β_{max} is set to be equal to 100. Each node represents an ATM switch and each edge represents a physical link connecting the two switches. Since we are only concerned about inter-switch handoff, base stations are not included in the model.

For our simulations, ten different 20-node random graphs, similar to the example shown above, were generated. Each random graph is employed in 10,000 simulation runs.

¹ The average node degree is defined as the average number of links connected to a node.

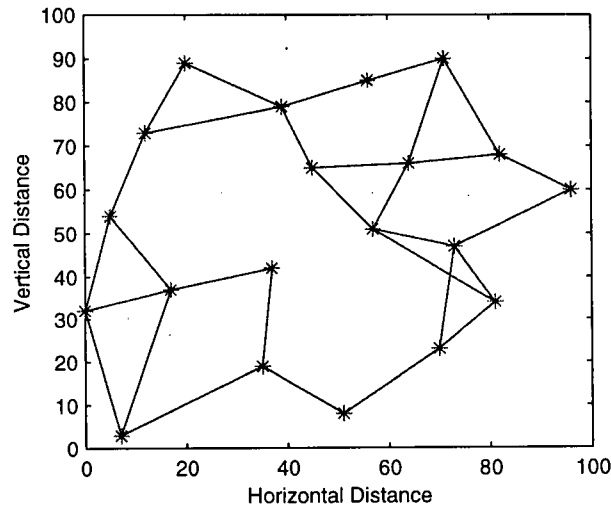


Figure 2.1 A 20-node random graph with node degree of 3; minimum distance between any two nodes is 15.

For each simulation run, a call is generated with two nodes chosen randomly as the source and destination nodes. Dijkstra's algorithm [13] is used to compute the shortest delay path between these two nodes. The source node is assumed to be stationary. The destination node becomes the anchor switch of the mobile connection. The call duration and the time between inter-switch handoff are modeled as exponentially distributed random variables with mean $1/\mu$ and $1/\lambda$, respectively.

During each inter-switch handoff, the target switch is restricted to be one of the neighboring switches of the current anchor switch. Path extension is used to extend the connection from the anchor switch to the target switch. Subsequent path optimizations may be triggered based on different initiation schemes as described in Section 2.2.

Similar to the performance metric used in our analytical model, the performance metric in our simulation model is the *average cost per call*, which is defined as the sum of the link cost and the signaling cost due to inter-switch handoff, and is given by:

$$Cost = n_{PE}C_{PE} + n_{PO}C_{PO} + \sum_{i=1}^m \tau_i l_i C_{link} \quad (2.15)$$

Table 2.1 Summary of parameters used in simulation and analytical models

item	symbol	value
number of nodes in the network	-	20
size of the rectangular coordinate grid	-	100×100
average node degree	-	3
maximum queueing delay (ms)	β_{max}	100
time between inter-switch handoff (minutes)	$1/\lambda$	5
average call duration (minutes)	$1/\mu$	20
signaling cost per path extension	C_{PE}	1
signaling cost per path optimization	C_{PO}	5
link cost per link per minute	C_{link}	variable
time interval in analytical model (minutes)	-	1×10^{-3}
average number of links during call setup (from simulation)	\bar{L}	3.44
average increase in number of links during path extension	\bar{H}	1

where n_{PE} and n_{PO} denote the number of path extension and path optimization, respectively, during a call; C_{link} , C_{PE} and C_{PO} are as defined in the previous section; m is the number of events per call; τ_i denotes the time elapsed between event $i - 1$ and i ; and l_i denotes the number of links between two end nodes during τ_i . The average cost per call is calculated by averaging the total cost per call over all (100,000) simulation runs.

The simulation model described above can be extended to handle other network topologies (e.g., star, ring, and hierarchical), as well as other statistical distributions for the call duration and the time between inter-switch handoff.

2.5 Results and Discussions

The baseline parameters for the analytical and simulation models are summarized in Table 1. Since the analytical model is a discrete time model and the simulation model is a continuous time model, for a fair comparison, the time interval in the analytical model has

to be sufficiently small in order to approximate it as a continuous time model. We have chosen a time interval of 1×10^{-3} minutes. Thus, if $C_{link} = 0.5/\text{link}/\text{minute}$ in the simulation model, the corresponding value is $C_{link} = 0.5 \times 10^{-3}/\text{link}/\text{time interval}$ in the analytical model. The inter-switch handoff rate λ , the call termination rate μ , the path optimization rate v for the exponential scheme, and the path optimization period k for the periodic scheme also have to be scaled accordingly (e.g., $\lambda = 0.2/\text{minute}$ in the simulation model is expressed as $0.2 \times 10^{-3}/\text{time interval}$ in the analytical model).

We first present the analytical results for the three proposed path optimization schemes. The expressions of the expected cost per call as shown in equations (2.1), (2.6), and (2.9) are normalized with respect to the signaling cost per path optimization, C_{PO} . Figures 2.2 - 2.4 show the analytical results for the exponential, periodic, and Bernoulli path optimization schemes, respectively. Given the link cost to signaling cost ratio, C_{link}/C_{PO} , and the mobility parameters λ and μ , there exists an optimal operating point for each scheme such that the expected cost per call is minimized. This optimal value can be determined directly from equations (2.4), (2.8), and (2.12).

For the exponential path optimization scheme, as shown in Figure 2.2, the optimal path optimization probability v_{opt} increases as C_{link}/C_{PO} increases. From equations (2.4) and (2.5), $v_{sub-opt}$ is always equal to 1 when

$$\frac{C_{link}}{C_{PO}} \geq \frac{1}{\lambda \bar{H}}. \quad (2.16)$$

For the periodic path optimization scheme, as shown in Figure 2.3, the optimal path optimization period k_{opt} decreases as C_{link}/C_{PO} increases. When C_{link}/C_{PO} exceeds a certain threshold, k_{opt} is always equal to 1. This threshold can be determined from equation (2.8).

For the Bernoulli path optimization scheme, as shown in Figure 2.4, the optimal

path optimization probability p_{opt} increases as C_{link}/C_{PO} increases. From equations (2.12) and (2.13), when

$$\frac{C_{link}}{C_{PO}} \geq \frac{\lambda(1-\mu) + \mu}{\bar{H}} \quad (2.17)$$

$p_{sub-opt}$ is always equal to 1.

Figures 2.5 - 2.7 compare the analytical and simulation results for the periodic, exponential, and the Bernoulli path optimization schemes, respectively. As shown in the figures, the results of the expected cost per call from the analytical model and the average cost per call from the simulation model are closely in agreement. The simulation results corroborate the closed-form solutions derived from the analytical model.

Although the analytical and simulation models give the same results, there are subtle differences between these two models. In the simulation model, a fixed network topology is required; while in the analytical model, only the parameters \bar{L} and \bar{H} are fixed. Thus, even though the average cost per call from the simulation model and the expected cost per call from the analytical model are the same, their second moments (i.e., cost variance) are different.

Figure 2.8 shows the performance comparisons between different path optimization schemes. For each path optimization scheme, its optimal value (i.e., respectively, k_{opt} , v_{opt} , and p_{opt}) is first calculated and then used to determine the minimum expected cost per call. On the whole, the exponential scheme gives the highest expected cost per connection. The performance of the periodic scheme lies between the exponential and the Bernoulli schemes. The Bernoulli scheme outperforms the other two schemes by giving the lowest expected cost per call. This is due to the fact that in the Bernoulli scheme, path optimization may be invoked only after each inter-switch handoff, while in the exponential and periodic cases, multiple path optimizations may occur unnecessarily between succes-

sive inter-switch handoff. Other results for different values of λ and μ also indicate that the Bernoulli scheme has a better performance over the other two schemes. Another observation from Figure 2.8 is that the minimum expected cost per call increases linearly in C_{link}/C_{PO} . This is due to the fact that as C_{link}/C_{PO} increases, the first term in equations (2.1), (2.6), (2.9), (i.e., $\bar{L}C_{link}/\mu$) becomes the dominant term of the respective expression.

Note that in our simulation, 10 different 20-node random graphs are used. If the number of nodes of the random graph remains unchanged, the average total cost of using either 10, 25, or 50 random graphs is equal. However, if the number of nodes of the random graph increases, the average number of links between two end-points will increase. Thus, the expected total costs, averaged over all source and destination pairs, will also increase. In spite of that, results show that the Bernoulli scheme still has a better performance than the other two heuristics.

In order to determine the minimum expected cost for each scheme, its optimal operating point needs to be calculated. The operating point for each scheme is a function of λ , μ , \bar{H} , and C_{link}/C_{PO} . Although the parameters \bar{H} and C_{link}/C_{PO} can be determined by the network, the values of λ and μ may not always be estimated correctly by the mobile terminal during call setup. If that is the case, the path optimization operating point may not indeed be the optimal one. We are interested in determining the percentage change of the expected cost per call to the variation of the average call duration $1/\mu$ and the average time between inter-switch handoffs $1/\lambda$.

First of all, we assume the actual call termination rate $\mu = 0.03$ per minute. Thus, the average call duration of the mobile connection is approximately 33 minutes. The optimal expected total cost, denoted as $E[\text{Cost (optimal)}]$, is determined. The average call duration is then varied within the range of $(-90, 100)\%$, (i.e., between 3 and 66 minutes).

Let $\hat{\mu}$ denote the estimated call termination rate. The values of μ and $\hat{\mu}$ are related by the following equation:

$$\hat{\mu}^{-1} = (1 + \Delta_{\mu})\mu^{-1} \quad (2.18)$$

where Δ_{μ} , $(-90\% \leq \Delta_{\mu} \leq 100\%)$, is the percentage change of the average call duration. Based on the estimated call duration rate $\hat{\mu}$ and other network parameters (i.e., \bar{L} , \bar{H} , C_{link}/C_{PO} , etc), the sub-optimal operating point for each path optimization scheme is determined. From this sub-optimal operating point and other cost and mobility parameters (i.e., λ , μ , etc), the sub-optimal expected total cost, denoted as $E[\text{Cost (sub-optimal)}]$, is computed. The change in the expected total cost is determined by the following equation:

$$\frac{E[\text{Cost (sub-optimal)}]}{E[\text{Cost (optimal)}]} \quad (2.19)$$

For the mobility and cost parameters, we assume that $\lambda = 0.1$, $C_{link} = 0.5$, $C_{PE} = 1$, $C_{PO} = 5$, $\bar{L} = 3.5$, and $\bar{H} = 1$. The result is shown in Figure 2.9. When the average call duration is over-estimated by more than -40% , the cost ratio of all three schemes is equal to one, which implies that the operating point is insensitive to the change of average call duration. However, within the $(-90, -40)$ percentage range, there is an increase in the cost ratio for both exponential and Bernoulli schemes. The periodic scheme has the smallest change in expected cost (less than 1%) within the variation range. These results imply that if there is uncertainty in estimating the average call duration, it may be better to over-estimate the value.

For the variation of the time between inter-switch handoff, we assume that the actual inter-switch handoff rate $\lambda = 0.1$ per minute. Thus, the average time between inter-switch handoffs is 10 minutes. The time between inter-switch handoffs is then varied within the range of $(-90, 100)\%$, (i.e., between 1 and 20 minutes). Let $\hat{\lambda}$ denote the

estimate inter-switch handoff rate. The values of λ and $\hat{\lambda}$ are related by the following equation:

$$\hat{\lambda}^{-1} = (1 + \Delta_{\lambda})\lambda^{-1} \quad (2.20)$$

where Δ_{λ} , $(-90\% \leq \Delta_{\lambda} \leq 100\%)$, is the percentage change of the time between inter-switch handoffs. Based on the estimated inter-switch handoff rate $\hat{\lambda}$ and other network parameters, the operating point for each path optimization scheme is calculated. The percentage change in expected cost is determined from eqn. (2.19). The result is shown in Figure 2.10. When the average time between inter-switch handoffs is over-estimated by more than -60% , the change in expected total cost is less than 5% for all three schemes. Among these three schemes, the Bernoulli scheme has the smallest change in expected cost. These results imply that if there is uncertainty in estimating the time between inter-switch handoffs, it may be better to over-estimate the value.

Although both the analytical and simulation models provide us with some useful insights on the network resources used and the processing and signaling load per connection, these models are not without drawbacks. First of all, path extension and path optimization are assumed to be point processes, i.e., the time to perform these operations has not been taken into account. In addition, the signaling overhead for path optimization is assumed to be constant. In practice, this signaling overhead may also depend on the location of the crossover switch. To simplify the analysis, we assume the average number of links of the optimal path is the same as the average number of links of the path during call setup. In real networks, that assumption may not be true.

2.6 Summary

Path optimization may be necessary if the end-to-end path after connection rerouting is not optimal. In this chapter, we proposed and analyzed three different path optimization

schemes which are simple to implement. For the *exponential path optimization scheme*, the time between path optimizations is modeled as an exponentially distributed random variable. The *periodic path optimization scheme* invokes path optimization at periodic time intervals. For the *Bernoulli path optimization scheme*, path optimization is performed with a fixed probability after each path extension. A discrete time analytical model and a discrete event simulation model were proposed to compare the performance of these schemes by evaluating the expected cost during a call. The analytical model enables a closed-form expression and optimal operating point to be obtained for each path optimization scheme. The analytical and simulation results agree with each other, corroborating the two models. The Bernoulli scheme outperforms the other two schemes by providing a lower expected cost per call and a lower percentage change of expected cost relative to the variation of the average time between inter-switch handoffs.

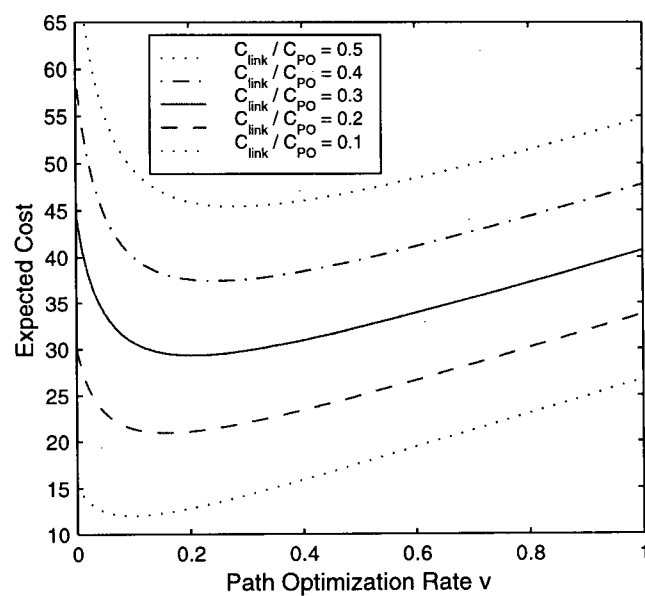


Figure 2.2 Exponential path optimization scheme.

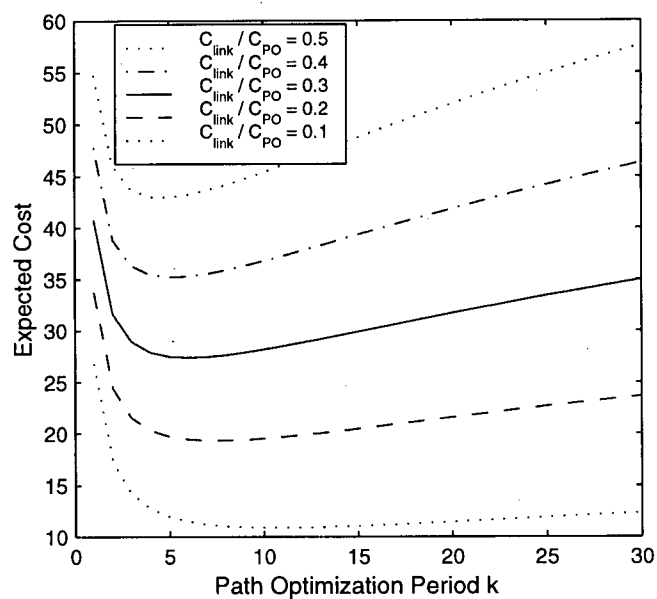


Figure 2.3 Periodic path optimization scheme.

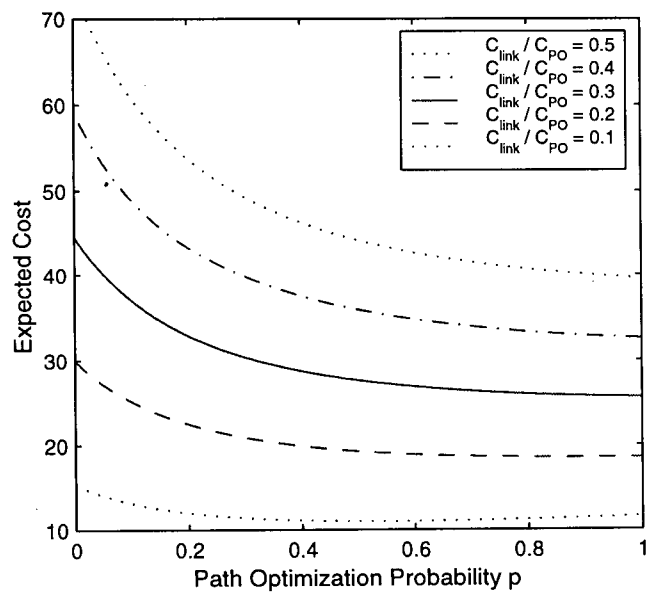


Figure 2.4 Bernoulli path optimization scheme.

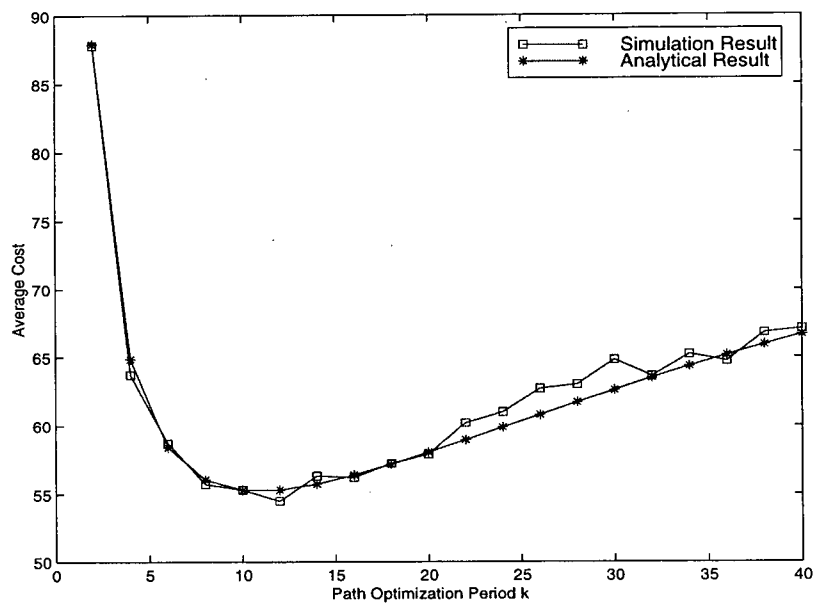


Figure 2.5 Analytical and simulation results of the periodic path optimization scheme.

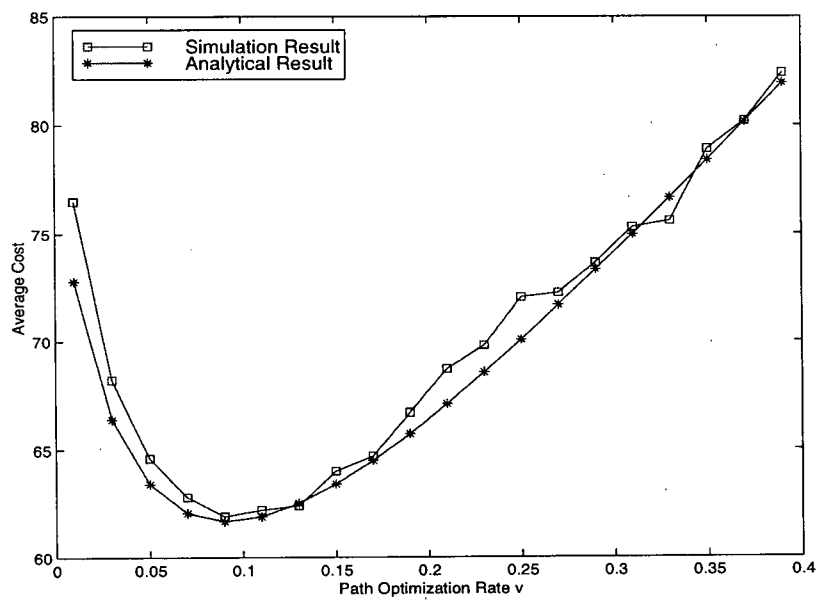


Figure 2.6 Analytical and simulation results of the exponential path optimization scheme.

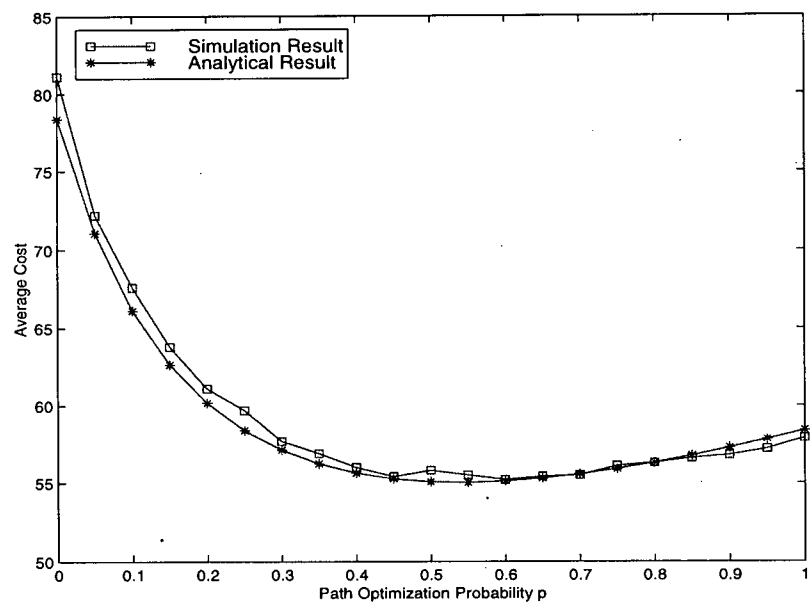


Figure 2.7 Analytical and simulation results of the Bernoulli path optimization scheme.

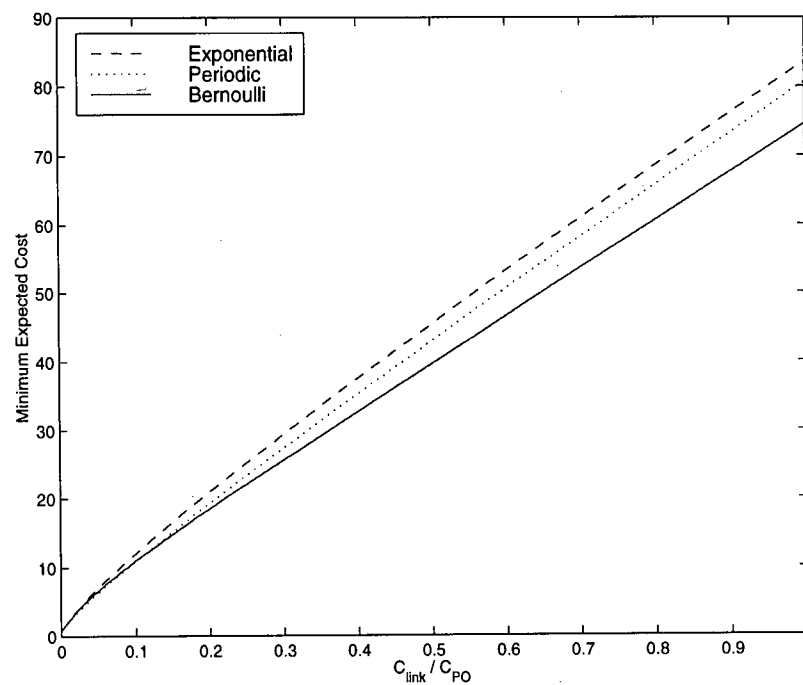


Figure 2.8 Comparisons between different path optimization schemes.

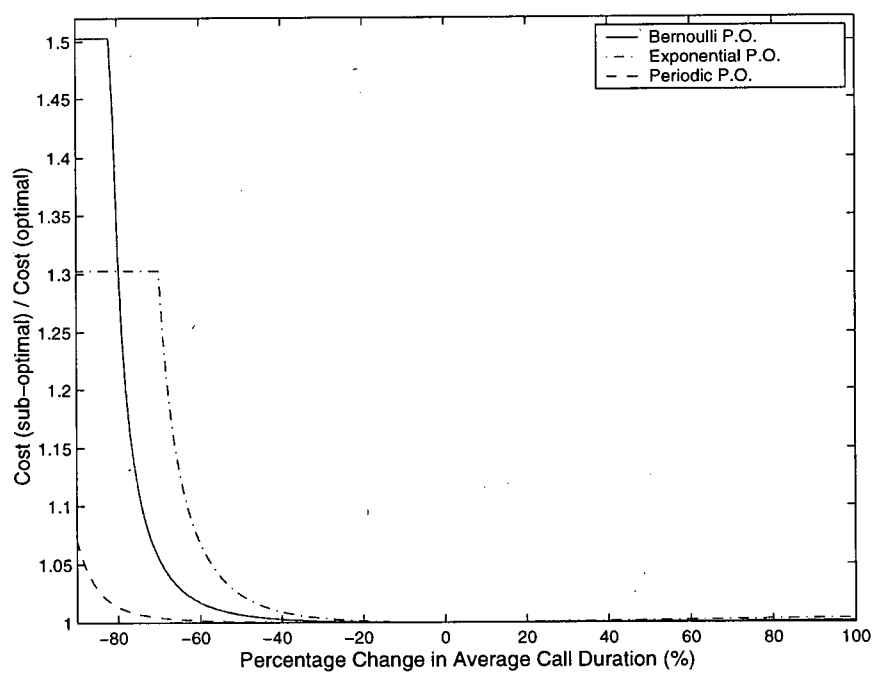


Figure 2.9 Variation of the average call termination time.

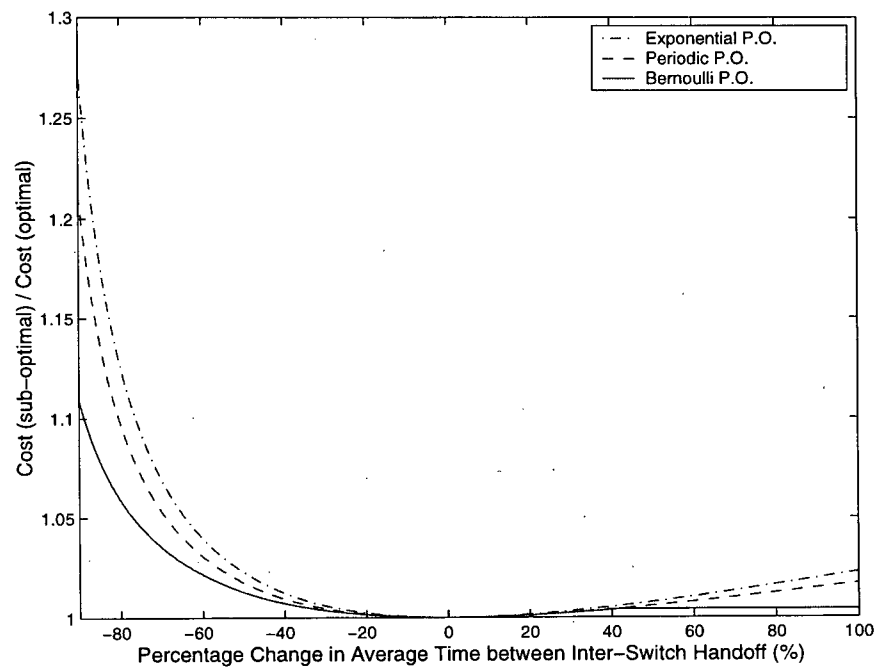


Figure 2.10 Variation of the time between inter-switch handoff.

"Two roads diverged in a yellow woods...

I took the one less traveled by.

And that has made all the difference."

Robert Frost, The Road Not Taken.

Chapter 3 Path Optimization for Inter-Switch Hand-off: Optimal Policy

3.1 Introduction

In this chapter, we propose a stochastic model to determine the optimal time to perform path optimization [76]. The path optimization problem is formulated as a semi-Markov decision process. *Link cost* and *signaling cost* functions are introduced to capture the trade-off between the network resources utilized by a connection and the signaling and processing load incurred on the network. The objective is to determine the optimal policy which minimizes the expected total cost per call. The major contribution lies in the formulation of a general model that is applicable to a wide range of conditions. Distinct features of our model include:

1. Different link cost functions can be assigned to different service classes with different bandwidth requirements.
2. Different signaling cost functions can be used based on the complexity of the path optimization procedures and the signaling load of the network.
3. The time between inter-switch handoffs can follow an arbitrary general distribution.

The rest of this chapter is organized as follows. The model formulation of the path optimization problem is described in Section 3.2. In Section 3.3, we describe the optimality equations, the value iteration algorithm, and the structure of the optimal policy. The

implementation issues are described in Section 3.4. Extension of the above model to mobile-to-mobile connection and other QoS constraints are described in Section 3.5. In Section 3.6, we present numerical results and compare the optimal policy with four other heuristics. A summary is given in Section 3.7. The proofs of the propositions stated in this chapter are shown in the Appendix D. For general background on Markov decision processes, please refer to [46] or [56].

3.2 Model Formulation

Each mobile connection may experience a number of inter-switch handoffs during its connection lifetime. During each inter-switch handoff, path extension can be used to extend the connection from the current anchor switch to the target switch. Although path extension is simple to implement, the connection utilizes more network resources than necessary. Occasional path optimization is required to reroute the connection to an optimal path. Path optimization is a complex process. It increases the processing and signaling load of the network. Thus, there is a trade-off between the network resources utilized by the call and the processing and signaling load incurred on the network. We formulate the above problem as a semi-Markov decision process. After each path extension, the network must decide whether to perform subsequent path optimization. The decision is based on the current number of links of the path and the locations of the anchor and target switches. The model is described below.

3.2.1 Semi-Markov Decision Process Model

When an inter-switch handoff occurs, a path extension is performed. After that, a decision must be made whether to perform subsequent path optimization. Those time instants are called *decision epochs*. Referring to Figure 3.1, the sequence $\sigma_0, \sigma_1, \dots$ represents the time of successive decision epochs. Since inter-switch handoff only occurs during the call

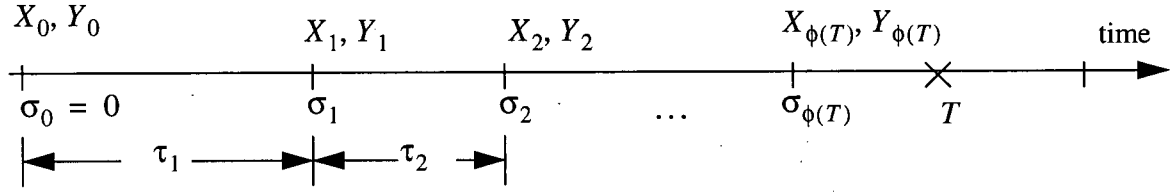


Figure 3.1 Timing diagram.

lifetime, the time interval requiring mobility monitoring is between a call arrival and its termination. The term $\sigma_0 = 0$ represents the arrival time of a new call and the random variable T denotes the call termination time. The random variable $\phi(T)$ denotes the total number of inter-switch handoffs that has occurred before the termination time T .

At each decision epoch, the network must decide whether to perform subsequent path optimization. Let $A = \{NPO, PO\}$ denote the *action* set, where PO corresponds to performing path optimization after path extension, and NPO corresponds to performing path extension only. The random variable Y_n is used to denote the action chosen at decision epoch n .

The action chosen is based on the current *state* of the connection. The state space is denoted by S . For each state $s \in S$, the state information includes the locations of the target and anchor switches, and the number of links of the current path. Although the decision is only based on the state at the current decision epoch, state change may occur between decision epochs. Consequently, we distinguish the *natural process* and the semi-Markov decision process. The natural process models the state evolution of the system as if it were observed continually throughout time, while the semi-Markov decision process represents the evolution of the system state at decision epochs only. These two processes coincide at decision epochs. The random variable X_n is used to denote the state at decision epoch n , and W_τ is used to denote the state of the natural process at time τ .

Two cost functions are introduced to account for the network resources utilized

and the signaling load incurred due to inter-switch handoff. The link cost function reflects the amount of network resources used during the connection lifetime, while the signaling cost function captures the processing and signaling load incurred on the network due to path extension and path optimization. The signaling costs are incurred only at the decision epochs, while the link cost is accrued continuously during the call lifetime.

The function $f(\tilde{s}, s, a)$ denotes the link cost rate as long as the natural process occupies state \tilde{s} and action a was chosen in state s at the preceding decision epoch. If the natural process occupies state \tilde{s} during the time interval $(\tau, \tau + d\tau)$, then the link cost accrued during this period is $f(\tilde{s}, s, a) \cdot d\tau$.

The function $b(s, a)$ denotes the signaling cost incurred when the decision maker chooses action a in state s . Thus, $b(s, NPO)$ represents the signaling cost of performing path extension, and $b(s, PO)$ represents the signaling cost of performing path extension and subsequent path optimization. All cost functions are assumed to be bounded and nondecreasing with respect to the number of links of the current path.

A decision rule prescribes a procedure for action selection in each state at a specified decision epoch. Deterministic Markovian decision rules are functions $\delta_t: S \rightarrow A_s$, which specify the action choice when the system occupies state s at decision epoch $t < T$; i.e., for each $s \in S$, $\delta_t(s) \in A_s$. This decision rule is said to be *Markovian* (memoryless) because it depends on previous system states and actions only through the current state of the system, and *deterministic* because it chooses an action with certainty. A policy π specifies the decision rule to be used at all decision epochs. That is, a policy is a sequence of decision rules, $\pi = (\delta_1, \delta_2, \dots)$. The set of all policies is denoted by Π .

Let $v^\pi(s)$ denote the *expected total cost per call* given policy π is used with initial state s . Thus,

$$\begin{aligned}
v^\pi(s) = E_s^\pi \left\{ \sum_{n=0}^{\phi(T)} b(X_n, Y_n) + \sum_{n=0}^{\phi(T)-1} \left[\int_{\sigma_n}^{\sigma_{n+1}} f(W_\tau, X_n, Y_n) d\tau \right] \right. \\
\left. + \int_{\sigma_{\phi(T)}}^T f(W_\tau, X_{\phi(T)}, Y_{\phi(T)}) d\tau \right\}
\end{aligned} \tag{3.1}$$

where E_s^π denotes the expectation with respect to policy π and initial state s . In the above expression, the first summation corresponds to the lump sum portion of the signaling cost. In the second summation, each term corresponds to the continuous portion of the link cost that is incurred at rate $f(W_\tau, X_n, Y_n)$ between decision epochs n and $n+1$. The last term corresponds to the link cost that is incurred at rate $f(W_\tau, X_{\phi(T)}, Y_{\phi(T)})$ between decision epoch $\phi(T)$ and termination time T . We assume the termination time is exponentially distributed with rate μ . In that case, equation (3.1) can be written as:

$$v^\pi(s) = E_s^\pi \left\{ \sum_{n=0}^{\infty} e^{-\mu \sigma_n} c(X_n, Y_n) \right\} \tag{3.2}$$

where

$$c(s, a) = b(s, a) + E_s^a \left\{ \int_0^{\tau_1} e^{-\mu t} f(W_t, s, a) dt \right\}. \tag{3.3}$$

For a proof of this fact, see Proposition D.1 in Appendix D. The expression in (3.2) is the expected total cost of an infinite-horizon semi-Markov decision process with discount rate μ . The function $c(s, a)$ in (3.3) is the expected total cost between two decision epochs, given the system occupies state s and the decision maker chooses action a in state s . This cost function is further discussed in Section 3.2.3.

Since the optimization problem that we consider is to minimize the expected total cost, we define a policy π^* is optimal in Π if

$$v^{\pi^*}(s) \leq v^{\pi}(s) \quad (3.4)$$

for all $\pi \in \Pi$.

Let $G(t|X_n, Y_n)$ denote the cumulative distribution function of the time between decision epochs n and $n + 1$, given that current state is X_n and action Y_n is chosen. The time between decision epochs corresponds to the time between inter-switch handoffs. In this formulation, the time between inter-switch handoffs follows a general distribution and can depend on the location of a particular anchor switch that the mobile terminal is connected to. We use $G(dt|X_n, Y_n)$ to represent the time-differential. That is,

$$G(dt|X_n, Y_n) = dG(t|X_n, Y_n).$$

A policy is said to be *stationary* if $\delta_t = \delta$ for all t . A stationary policy has the form $\pi = (\delta, \delta, \dots)$; for convenience we denote it by δ . For a stationary policy δ , equation (3.2) can be written as:

$$v^{\delta}(s) = c[s, \delta(s)] + \sum_{s' \in S} \left\{ \int_0^{\infty} e^{-\mu t} v^{\delta}(s') P[s'|s, \delta(s)] G[dt|s, \delta(s)] \right\} \quad (3.5)$$

where $P[s'|s, \delta(s)]$ denotes the transition probability that the next state is s' , given the current state is s and action $\delta(s)$ is chosen. For a proof of this fact, see Proposition D.1 in Appendix D. Our objective is to determine an optimal stationary deterministic policy δ^* which minimizes (3.5).

To simplify the analysis, two assumptions are made. We assume the time between inter-switch handoffs is independent of the state and action chosen, i.e., the cumulative distribution function $G(t|X_n, Y_n) = G(t)$. We assume the mobile terminal is communicating with a remote terminal which is stationary; i.e., we consider a mobile-to-fixed connection. The model formulation for mobile-to-mobile connection is described in Section 3.5.1.

3.2.2 State Transition Probability Function

A state change occurs when there is an inter-switch handoff. The state space S is three dimensional. For each state $(i, j, k) \in S$, i denotes the location of the target switch; j denotes the location of the current anchor switch; and k denotes the number of links of the current path. Thus,

$$S = \{1, 2, \dots, N\} \times \{1, 2, \dots, N\} \times \{1, 2, \dots, L\} \quad (3.6)$$

where N denotes the total number of nodes in the network and L represents the maximum number of links allowed in a path. The number of links of any path is always finite. We assume the number of links increased during path extension is bounded by M which is much smaller than L (i.e., $M \ll L$).

Since the end-to-end delay is proportional to the number of links of the path, a sub-optimal path with a high number of links not only increases the delay but also increases the call dropping probability and the congestion level of the network. We impose the condition that whenever the number of links is greater than or equal to $L - M$ and there is an inter-switch handoff, path extension is performed followed by path optimization with certainty. For convenience, we let $K = L - M$. Later we show that path optimization is always performed when the number of links exceeds a certain threshold, and this threshold is much smaller than K .

Given the current state $s = (i, j, k)$, the available action set is:

$$A_{(i, j, k)} = \begin{cases} \{NPO, PO\}, & 1 \leq k < K \\ \{PO\}, & K \leq k \leq L. \end{cases} \quad (3.7)$$

Thus, after each path extension, path optimization may be performed if the number of links is less than K , while path optimization is performed with certainty whenever the number of links is greater than or equal to K .

Two probability distribution functions are introduced to govern the state changes.

Let

- $p(m|i, j)$ denote the probability that the number of links of the optimal path is m , given the locations of the two end-points are i and j , respectively.
- $q(l|i)$ denote the probability that the location of the target switch in the next decision epoch is l , given the location of the target switch at the current decision epoch is i .

In ATM networks, source routing is used for all connection setup requests. That is, the source switch selects a path based on topology, loading, and reachability information in its database. As networks grow in size and complexity, full knowledge of network parameters is typically unavailable. Each single entity in the network cannot be expected to have detailed and instantaneous access to all nodes and links. Routing must rely on partial or approximate information, and still meet the QoS demands [26][37]. The ATM Forum PNNI standard [10] introduces a hierarchical process that aggregates information as the network gets more and more remote. However, the aggregation process inherently reduces the accuracy of the information and introduces uncertainty. Thus, in large networks it is more appropriate to model the number of links of a path between two endpoints in a probabilistic manner.

On the other hand, for small networks with periodic routing information update, the number of links of a path between the two endpoints can be modeled in a deterministic manner. Let $Y(i, j)$ denote the number of links of the optimal path between the two endpoints i and j . The functions $p(m|i, j)$ and $Y(i, j)$ are related by

$$p(m|i, j) = \begin{cases} 1, & \text{if } Y(i, j) = m \\ 0, & \text{if } Y(i, j) \neq m. \end{cases} \quad (3.8)$$

Let D denote the location of the destination (i.e., the remote terminal), which is assumed to be fixed. The transition probability that the next state $s' = (i', j', k')$ given the current state $s = (i, j, k)$ and action a chosen, is given by:

$$P(i', j', k' | i, j, k, a) = \begin{cases} q(i'|i) \cdot p(m|i, j), & i' \neq i, j' = i, k' = k + m, a = NPO \\ q(i'|i) \cdot p(n|i, D), & i' \neq i, j' = i, k' = n, a = PO \\ 0, & \text{otherwise.} \end{cases} \quad (3.9)$$

Equation (3.9) states that if action NPO is chosen, the number of links is increased by m with probability $p(m|i, j)$ after path extension. On the other hand, if action PO is chosen, the number of links is equal to n with probability $p(n|i, D)$ after path optimization. In both cases, the location of the target switch at the next decision epoch is equal to i' with probability $q(i'|i)$.

3.2.3 Cost Functions

For each path extension event, the network incurs a fixed signaling cost $C_{PE} > 0$ and a variable signaling cost $h_{PE}(m)$ where m represents the number of links increased during path extension. The terms C_{PE} and $h_{PE}(m)$ capture the cost of setting up the extended path between the anchor and target switches.

For each path optimization performed, the network incurs a fixed signaling cost $C_{PO} > 0$ and a variable signaling cost $h_{PO}(l)$ where l represents the number of links reduced during path optimization. These two terms capture the cost of (1) locating the crossover switch; (2) setting up the new branch connection; (3) terminating the old branch connection; and (4) updating the connection server about the status of the existing route.

We assume the link cost rate only depends on the number of links of the current path. That is, $f(\tilde{s}, s, a) = f(k)$ for all $\tilde{s}, s \in S$. Recall from equation (3.2) that $c(i, j, k, a)$ denotes the expected total cost between two decision epochs, given the system

occupies state (i, j, k) and action a is chosen. Since the first inter-switch handoff occurs at time σ_1 , the locations of the anchor and target switches are the same at the call setup time σ_0 . Thus, during the time interval $(\sigma_0, \sigma_1]$, we have $i = j$ and the cost function

$$c(j, j, k, NPO) = I_1 f(k) \quad (3.10)$$

where $I_1 = \int_0^\infty \int_0^t e^{-\mu\tau} d\tau G(dt)$. The function $I_1 f(k)$ is the expected discounted link cost between two decision epochs, given the current number of links is k .

For other decision epochs not equal to σ_0 , the locations of the anchor and target switches are always different (i.e., $i \neq j$ if $\sigma \neq \sigma_0$). In that case, if action NPO is chosen, then the cost function

$$c(i, j, k, NPO) = C_{PE} + \sum_{m=1}^M \{h_{PE}(m) + I_1 f(k+m)\} p(m|i, j). \quad (3.11)$$

The function $C_{PE} + \sum h_{PE}(m)p(m|i, j)$ is the expected signaling cost for path extension, given the locations of the anchor and target switches are i and j , respectively.

For path optimization, we assume the number of links of the optimal path is always less than or equal to the number of links of the current path, and less than K . For decision epoch σ not equal to σ_0 , if action PO is chosen, then the cost function

$$\begin{aligned} c(i, j, k, PO) = & C_{PE} + \sum_{m=1}^M h_{PE}(m)p(m|i, j) + C_{PO} \\ & + \sum_{m=1}^M \sum_{n=1}^{(k+m) \wedge (K-1)} \{h_{PO}(k+m-n) + I_1 f(n)\} p(n|i, D)p(m|i, j) \end{aligned} \quad (3.12)$$

where $x \wedge y = \min(x, y)$. The expression $C_{PO} + \sum h_{PO}(k-n)p(n|i, D)$ is the expected signaling cost for path optimization, given the current number of links is k , and the locations of the source and destination are i and D , respectively.

3.3 Optimality Equations

In this section, we introduce the optimality equations (sometimes referred to as the Bellman equations or functional equations) and investigate their properties. We show that solutions of these equations correspond to optimal value functions and that they also provide a basis for determining optimal policies. Let $v(s)$ denote the minimum expected total cost per call given state s . That is,

$$v(s) = \min_{\pi \in \Pi} v^{\pi}(s). \quad (3.13)$$

The optimality equations are given by

$$v(s) = \min_{a \in A} \left\{ c(s, a) + \sum_{s' \in S} \left[\int_0^{\infty} e^{-\mu t} v(s') P(s'|s, a) G(dt) \right] \right\}. \quad (3.14)$$

Let $I_2 = \int_0^{\infty} e^{-\mu t} G(dt)$. Equation (3.14) can be expanded as follows:

For $i = j$ and $1 \leq k < K$,

$$v(j, j, k) = c(j, j, k, NPO) + \sum_{l=1}^N I_2 v(l, j, k) q(l|j). \quad (3.15)$$

For $i \neq j$ and $1 \leq k < K$,

$$v(i, j, k) = \min \left\{ c(i, j, k, NPO) + \sum_{l=1}^N \sum_{m=1}^M I_2 v(l, i, k+m) p(m|i, j) q(l|i), \right. \\ \left. c(i, j, k, PO) + \sum_{l=1}^N \sum_{m=1}^M \sum_{n=1}^{(k+m) \wedge (K-1)} I_2 v(l, i, n) p(n|i, D) p(m|i, j) q(l|i) \right\} \quad (3.16)$$

For $i \neq j$ and $K \leq k \leq L$,

$$v(i, j, k) = c(i, j, k, PO) + \sum_{l=1}^N \sum_{n=1}^{K-1} I_2 v(l, i, n) p(n|i, D) q(l|i). \quad (3.17)$$

At call setup time σ_0 , the locations of the anchor and target switches are the same. Thus in (3.15), no path extension or path optimization is performed. For other decision epochs not equal to σ_0 , the locations of the anchor and target switches are different. If the number of links of the path is less than K , then after each path extension, the network will decide whether to perform subsequent path optimization. This fact is stated in (3.16). Since path optimization is always performed if the number of links is greater than K , in (3.17), the action PO is chosen when there is an inter-switch handoff.

If the signaling cost function for path optimization is zero (i.e., $C_{PO} = 0$ and $h_{PO}(l) = 0$), the problem of finding an optimal policy is trivial. It is optimal to perform path optimization after each inter-switch handoff. This is because the link cost function is nondecreasing with respect to the number of links of the current path. After each path optimization, there is a reduction in the number of links. However, if the signaling cost function for path optimization is nonzero, it is not obvious as to what constitutes the optimal policy. Note that if $\mu > 0$, the state space is finite, and the cost functions are bounded, then the solutions for equations (3.15)-(3.17) exist. By solving these equations, a stationary deterministic optimal policy can be obtained.

3.3.1 Value Iteration Algorithm

There are a number of iteration algorithms available to solve the above optimality equations. Examples include the value iteration, policy iteration, and linear programming algorithms [46]. Value iteration is the most widely used and best understood algorithm for solving discounted Markov decision problems. Value iteration is also known by other names including successive approximations, over-relaxation, and pre-Jacobi iteration. The following value iteration algorithm finds a stationary deterministic optimal policy and the corresponding expected total cost.

Algorithm:

1. Set $v^0(s) = 0$ for each state $s \in S$. Specify $\epsilon > 0$ and set $n = 0$.
2. For each $s \in S$, compute $v^{n+1}(s)$ by:

$$v^{n+1}(s) = \min_{a \in A} \left\{ c(s, a) + \sum_{s' \in S} \left[\int_0^\infty e^{-\mu t} v^n(s') P(s'|s, a) G(dt) \right] \right\}.$$

3. If $\|v^{n+1} - v^n\| < \epsilon$, go to step 4. Otherwise increment n by 1 and return to Step 2.
4. For each $s \in S$, the stationary optimal policy

$$\delta^*(s) = \arg \min_{a \in A} \left\{ c(s, a) + \sum_{s' \in S} \left[\int_0^\infty e^{-\mu t} v^n(s') P(s'|s, a) G(dt) \right] \right\}$$

and stop.

There are a number of definitions for the function norm $\|\cdot\|$. In this thesis, the function norm is defined as:

$$\|v\| = \max_{s \in S} v(s).$$

Convergence of the value iteration algorithm is ensured since the operation in Step 2 corresponds to a contraction mapping. Thus, the function $v^n(s)$ converges in norm to $v(s)$.

In small networks, if each node maintains perfect information of all nodes and links, then the function $v[i, i, Y(i, D)]$ is the minimum expected total cost per call given source i and destination D . On the other hand, in large networks, the number of links of a path determined by the source is modeled in a probabilistic manner. In that case, the expression

$$\sum_k v(i, i, k) p(k|i, D) \tag{3.18}$$

is the minimum expected total cost per call given source i and destination D , averaged

over the number of links of the optimal path.

3.3.2 Structure of the Optimal Policy

We now provide a condition under which the optimal policy has a *control limit* (or *threshold*) structure. The control limit structure states that path optimization is performed with certainty whenever the number of links of the current path exceeds a certain threshold. For convenience, we let $\Delta\gamma(k) = \gamma(k+1) - \gamma(k)$ for some function γ .

Proposition 3.1: Given state $(i, j, k) \in S$, the optimal policy δ^* has a control limit structure:

$$\delta^*(i, j, k) = \begin{cases} NPO, & 1 \leq k < k^*, \\ PO, & k^* \leq k \leq L \end{cases} \quad (3.19)$$

when $I_1 \Delta f(k+m) - \sum_n \Delta h_{PO}(k+m-n)p(n|i, D) \geq 0$ for all m such that $p(m|i, j) \neq 0$ and $k^* \leq k < K$.

The proof of the above proposition is shown in Proposition D.2 in Appendix D. The value k^* is the control limit or threshold. Consider the special case where the cost functions are linear. That is, $f(k) = C_{link} \cdot k$ and $h_{PO}(l) = w_{PO} \cdot l$ where C_{link} and w_{PO} are positive constant. In this case, if $I_1 C_{link} - w_{PO} \geq 0$, then path optimization is always performed when the number of links is greater than or equal to k^* . An optimal policy with threshold structure facilitates its implementation. For each mobile connection, the network only has to maintain the information of the minimum number of links to initiate path optimization for all anchor and target switch pairs. The decision of performing path optimization can be made by a table lookup.

3.4 Implementation Aspects

Having identified the different parameters involved in the model, we are now in a position

to explain the steps that need to be taken in order to implement the model. For each mobile connection, during its connection setup phase, the network controller assigns the cost functions based on the service class and the signaling load of the network. Different service classes (e.g., CBR, VBR, ABR) with different bandwidth requirements are assigned different link cost functions to reflect the network resources consumed. The assigned signaling cost function reflects the complexity of the path optimization procedures and the current signaling load of the network. By keeping the mobility profile of each user (i.e., the movement history and call history), the average time between inter-switch handoffs as well as the average duration of the connection can be estimated [81].

Given the input parameters (i.e., cost functions and various distributions), the value iteration algorithm can be used to determine the optimal policy. The optimal policy is then stored in a table format. Each entry of the table specifies the minimum number of links to initiate path optimization for a specific pair of anchor and target switches. Whenever there is an inter-switch handoff, the network performs a table lookup at the corresponding anchor and target switch entry. Path optimization is performed if the number of links is greater than the threshold.

The optimal policy table needs to be updated when there are changes in network topology or signaling load of the network. The update can be performed off-line; i.e., whenever spare processing capacity is available at the network controller.

3.5 Model Extensions

In the previous sections, we have considered the connection between a mobile terminal and a fixed endpoint. In this section we extend the above model to the connection between two mobile terminals and to take into the consideration of other QoS constraints.

3.5.1 Extension to Mobile-to-Mobile Connection

The problem formulation for mobile-to-mobile connection is similar to that of mobile-to-fixed connection. Consider mobile terminals 1 and 2 communicating with each other via a wireless ATM network. Each mobile terminal has its own movement pattern. Path extension is performed when there is an inter-switch handoff, (initiated from either side), followed by path optimization if necessary. In this formulation, the state space needs to include the locations of the two endpoints, as well as the information of which mobile terminal initiates the path extension. For each state $(i, j, k, l, m) \in S$, i and j denote the locations of the anchor switch connected to mobile terminals 1 and 2, respectively; k denotes the number of links of the current path; l denotes the identifier of the mobile terminal which initiates the inter-switch handoff; and m denotes the location of the target switch.

Since the movement pattern of each mobile user is different, the time between inter-switch handoffs for each mobile user is also different. Suppose the time between inter-switch handoffs for mobile terminal r , ($r \in \{1, 2\}$), is exponentially distributed with rate λ_r , then the time between decision epochs is also exponentially distributed with rate $\lambda_1 + \lambda_2$.

Since the state space has changed, the cost functions and the state transition probability function have to be modified accordingly. As the modification is conceptually similar to the functions derived in Section 3.3, the details are omitted. The optimality equations are

$$v(s) = \min_{a \in A} \left\{ c(s, a) + \left(\frac{\lambda_1 + \lambda_2}{\lambda_1 + \lambda_2 + \mu} \right) \cdot \sum_{s' \in S} [v(s') P(s'|s, a)] \right\}. \quad (3.20)$$

The value iteration algorithm described in Section 3.3.1 can be used to evaluate the expected total cost and the optimal policy. The conditions for the optimal policy with a

threshold structure can also be derived.

3.5.2 Extension to QoS Constraints

In Sections 3.2 and 3.3, path optimization is triggered based on the number of links of the current path. In general, a mobile connection can have multiple QoS constraints such as bandwidth, delay, delay jitter, etc. Suppose the connection has to maintain a delay constraint. In this case path optimization is performed with certainty if the end-to-end delay after path extension exceeds the delay constraint, while path optimization may be performed if the delay after path extension is still below the constraint.

To incorporate the delay constraint into the model, the state space needs to be extended to include the end-to-end delay of the current path. We assume the end-to-end delay of a path is the sum of the delay on each link of the path. The delay information on each link can be obtained from the network by measurement. Let ζ denote the end-to-end delay of the current path and Ψ be the delay constraint. Let $\Phi(i, j)$ denote the delay of the path between the two endpoints i and j . The optimality equations described in Section 3.3 have to include the following constraint

$$\zeta + \Phi(i, j) \leq \Psi \quad (3.21)$$

where i and j denote the locations of target and anchor switches respectively. Note that the value iteration algorithm described in Section 3.3.1 cannot be used to solve the optimality equations with constraints. However, the optimality equations can be transformed into primal or dual linear programs, which can then be solved by the simplex algorithm. Interested readers are referred to [46] for details of the transformation.

In summary, multiple QoS constraints can be incorporated into the model by extending the state space and including the constraint equations into the set of optimality equations. The expected total cost and the optimal policy can be obtained by transforming

the model into a linear programming model.

3.6 Numerical Results and Discussions

In this section, we compare the performance of the optimal policy with four heuristics. For the first heuristic, path optimization is performed after each path extension. We denote this policy as “*always perform path optimization*” or δ^{PO} . For the second heuristic, no path optimization is performed during the connection lifetime. We denote this policy as “*never perform PO*” or δ^{NPO} . For the third heuristic, periodic path optimization is considered. The use of periodic path optimization has been proposed within the ATM Forum [8]. For periodic path optimization, after every Γ time units, the network determines if the connection requires path optimization. Suppose the current time is t , path optimization is performed if an inter-switch handoff has occurred during the time interval $(t - \Gamma, t)$. In this section, we assume that Γ is equal to the average time between inter-switch handoffs. For the last heuristic, we consider the Bernoulli path optimization scheme which is proposed and analyzed in the previous chapter.

The performance metrics are the *expected total cost per call* and the *expected number of path optimizations per call*. The expected total cost per call is defined in Section 3.1. The expected number of path optimizations per call given policy π with initial state s is:

$$\gamma^\pi(s) = E_s^\pi \left\{ \sum_{n=0}^{\infty} e^{-\mu \sigma_n} \cdot \mathbf{1}[Y_n = 1] \right\} \quad (3.22)$$

where $\mathbf{1}[\cdot]$ denotes the indicator function. That is, $\mathbf{1}[a = 1]$ is equal to one if $a = 1$ and zero otherwise.

3.6.1 Simulation Model

In the simulation model, a wireless ATM network is modeled as a non-hierarchical random graph, similar to the one proposed in Chapter 2. Based on network model, we obtain the adjacency matrix of the network as well as the number of links of the shortest path between any two nodes. We assume the number of links of the shortest path estimated by the source is deterministic. The call duration is assumed to be exponential. The time between inter-switch handoffs follows a Gamma distribution.

For each source and destination pair, the value iteration algorithm described in Section 3.3.1 is used to determine the minimum expected total cost and the optimal policy. From the optimal policy, the value iteration algorithm is used again to calculate the expected number of path optimizations. The minimum expected total cost and the expected number of path optimizations are then averaged over all possible source and destination pairs. We repeat this for 100 random graphs and determine the averages.

For the two heuristics δ^{PO} and δ^{NPO} , the expected total cost and the expected number of path optimizations for each source and destination pair are also determined by the value iteration algorithm. These values are then averaged over all possible source and destination pairs. Again, we repeat this for 100 random graphs and determine the averages.

For the periodic and Bernoulli path optimization policies, simulation must be used. Given the network topology, a call is generated with two nodes chosen as the source and destination. Dijkstra's algorithm is used to compute the shortest path between these two nodes. The destination node is assumed to be stationary. The source node becomes the anchor switch of the mobile connection. During each inter-switch handoff, the target switch is restricted to be one of the neighboring switches of the current anchor switch. Path extension is used to extend the connection from the anchor switch to the target

switch. Path optimization is performed periodically for the periodic scheme. For the Bernoulli scheme, its operating point, denoted as p_{opt} , is determined from equations (2.12) and (2.13). After each path extension, path optimization is performed with probability p_{opt} . For each source and destination pair, 1000 simulation runs are performed. The average total cost and the average number of path optimizations per call are determined. We repeat this for 100 random graphs and determine the averages.

All the cost functions are assumed to be linear. The link cost function $f(k) = C_{link} \cdot k$ where $C_{link} > 0$. The term C_{link} captures the amount of resources (e.g., bandwidth) used by the connection. Different traffic classes can be assigned different values for C_{link} . The variable cost function for path extension $h_{PE}(m) = w_{PE} \cdot m$ where $w_{PE} > 0$ and m denotes the number of links increased during path extension. The variable cost function for path optimization $h_{PO}(l) = w_{PO} \cdot l$ where $w_{PO} > 0$ and l denotes the number of links reduced during path optimization. For the cost and mobility parameters, we assume that $C_{link} = 0.1$, $C_{PE} = 1$, $C_{PO} = 5$, $w_{PE} = 0.5$, $w_{PO} = 0.5$, $\lambda = 0.1$, and $\mu = 0.03$.

3.6.2 Results

Figure 3.2 shows the expected total cost versus the link cost rate C_{link} . The optimal policy gives the lowest expected total cost compared to the other four heuristics. When the link cost rate is small, there is no incentive to perform path optimization. The operating point, p_{opt} , for the Bernoulli policy is close to zero. The optimal policy is to perform path extension only. Thus, results of the Bernoulli, δ^{NPO} , and optimal policies are the same. When the link cost rate increases, the optimal policy for some source and destination pairs is to perform path optimization. Results of the optimal, Bernoulli, and δ^{NPO} policies diverge, while the results of the Bernoulli and δ^{PO} policies begin to converge.

Figure 3.3 shows the expected number of path optimizations versus the link cost

rate C_{link} . Since no path optimization is performed for the δ^{NPO} policy, the expected number of path optimizations is always equal to zero. Note that since both the call termination rate and the inter-switch handoff rate are constant, in this case the expected number of inter-switch handoffs is also a constant. Thus, results for the periodic and δ^{PO} policies are independent of the link cost rate. For the Bernoulli and optimal policies, when the link cost rate is small, there is no incentive to perform path optimization. The expected number of path optimizations is small. As the link cost rate increases, some source and destination pairs perform path optimization after inter-switch handoff. Thus, there is an increase in the number of path optimizations performed.

Figure 3.4 shows the expected total cost versus the inter-switch handoff rate λ . The expected total cost increases as the inter-switch handoff rate increases. When λ is small (i.e., the average time between inter-switch handoffs is larger than the average call duration), an inter-switch handoff is unlikely to occur during the connection lifetime. Thus, the results between the five policies are close. As the inter-switch handoff rate increases, these five curves begin to diverge. The δ^{PO} policy gives the highest expected total cost, which is followed by the periodic, δ^{NPO} , and Bernoulli policies. Results of the δ^{NPO} and Bernoulli policies are very close. Although we can conclude that the expected total cost increases in λ and the optimal policy always gives the minimum expected total cost, the performance comparisons between the other four heuristics differ when another set of parameters are chosen. That is, the δ^{PO} policy can sometimes have a better performance than the periodic and δ^{NPO} policies.

Figure 3.5 shows the expected number of path optimizations versus the inter-switch handoff rate λ . The expected number of path optimizations increases as λ increases. Results of the Bernoulli and optimal policies are quite close. Due to the threshold structure of the optimal policy, path optimization is performed only after a certain

number of inter-switch handoffs. Thus, the expected number of path optimizations for the optimal policy is smaller than the periodic and δ^{PO} policies.

Figure 3.6 shows the expected total cost versus the call termination rate μ . The expected total cost decreases as the call termination rate increases, which is intuitive since the link cost is accrued continuously during the call lifetime. When μ is large (i.e., the call duration is short), all the connections experience a small number of inter-switch handoffs. Thus, the results of all these policies are close. When the call duration increases, the results begin to diverge. We can see a significant cost difference between the optimal policy and the other heuristics when the call duration is long (i.e., μ is small).

Figure 3.7 shows the expected number of path optimizations versus the call termination rate μ . The expected number of path optimizations decreases as μ increases. Due to the threshold structure of the optimal policy, path optimization is performed only after a certain number of path extensions. Thus, the expected number of path optimizations performed for the optimal policy is much smaller than the periodic and δ^{PO} policies.

In the previous results, we assume the time between inter-switch handoffs follows a Gamma distribution. We also consider exponential and hyper-exponential distributions for the time between inter-switch handoff. For a fair comparison, the average time between inter-switch handoffs is the same for various distributions. Figure 3.8 and Figure 3.9 show the minimum expected total cost of the optimal policy versus λ and C_{link} , respectively. These results indicate that the expected total cost is relatively insensitive to the distributions of the time between inter-switch handoffs.

We also perform the sensitivity analysis for the optimal policy with respect to the variation of the average call duration and the average time between inter-switch handoffs. The procedures for the sensitivity analysis are similar to those described in Section 2.5 in Chapter 2. The results for different μ are shown in Figure 3.10. When the average call

duration is over-estimated by more than -40% , the cost ratio is almost equal to one, which implies that the optimal policy is insensitive to the change of the average call duration. However, within the $(-90, -50)$ percentage range, there is an increase in the cost ratio. The cost ratio can be as high as 1.53 for $\mu = 0.02$. These results imply that if there is uncertainty in estimating the average call duration, it may be better to over-estimate the value in order to reduce the cost ratio difference.

Figure 3.11 shows the cost ratio versus the percentage change in average time between inter-switch handoffs for different λ . Within the percentage range of interest, the cost ratio is always less than 1.07 (i.e., 7%). Within the $(-50, 100)$ percentage range, the cost ratio is less than 1.01 (i.e., 1%). These results imply that the optimal policy is relatively insensitive to the change of the average time between inter-switch handoffs.

In this chapter, the wireless ATM network is modeled as a non-hierarchical random graph. One question that arises is whether the results will differ if some other network topologies are used. The answer is affirmative. The relative performance between the four heuristics will change if another network topology is used. This is essentially the same as changing the values in the functions $p(m|i, j)$ or $Y(i, j)$. However, the optimal policy will always give the lowest expected total cost compared to the other heuristics.

One issue may arise as to the mapping of the network resources utilized by a connection into appropriate cost functions. That issue is beyond the scope of our work. Interested readers can refer to [7] or [17] for details.

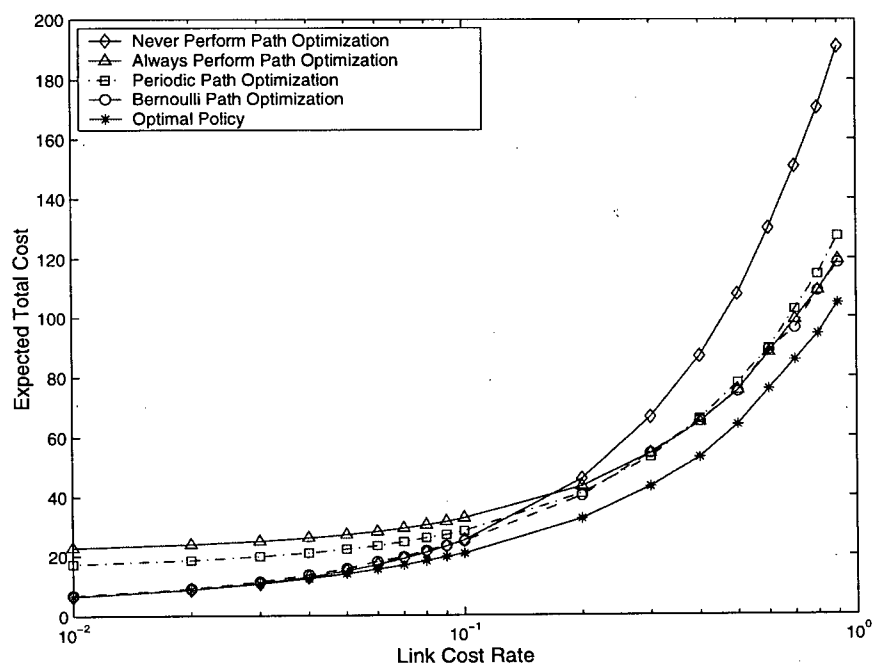
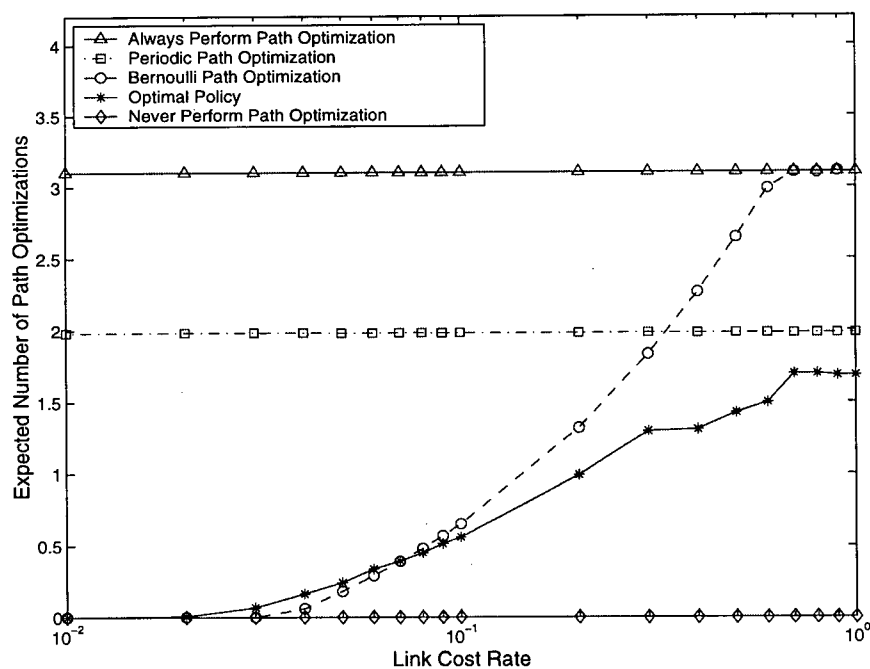
3.7 Summary

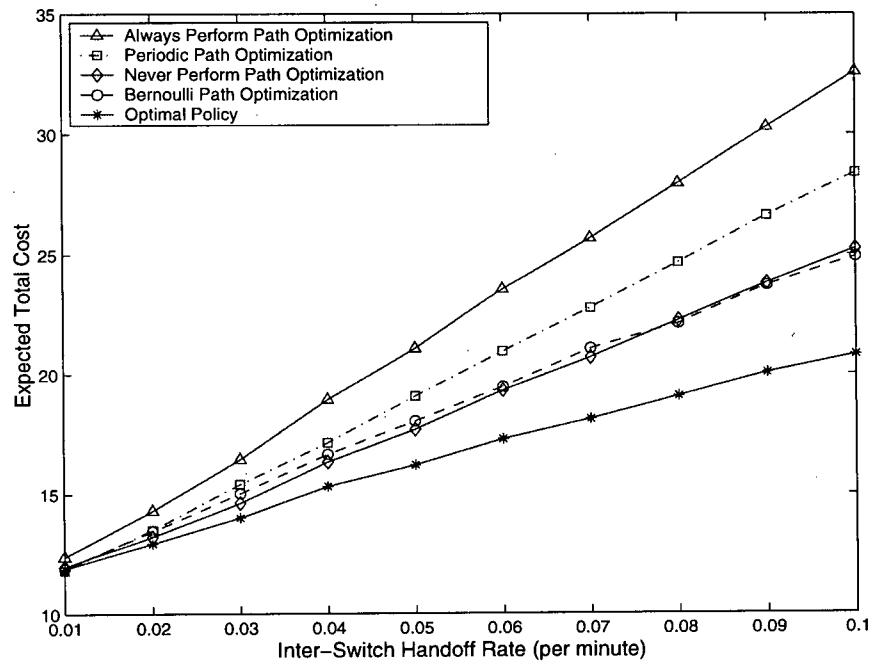
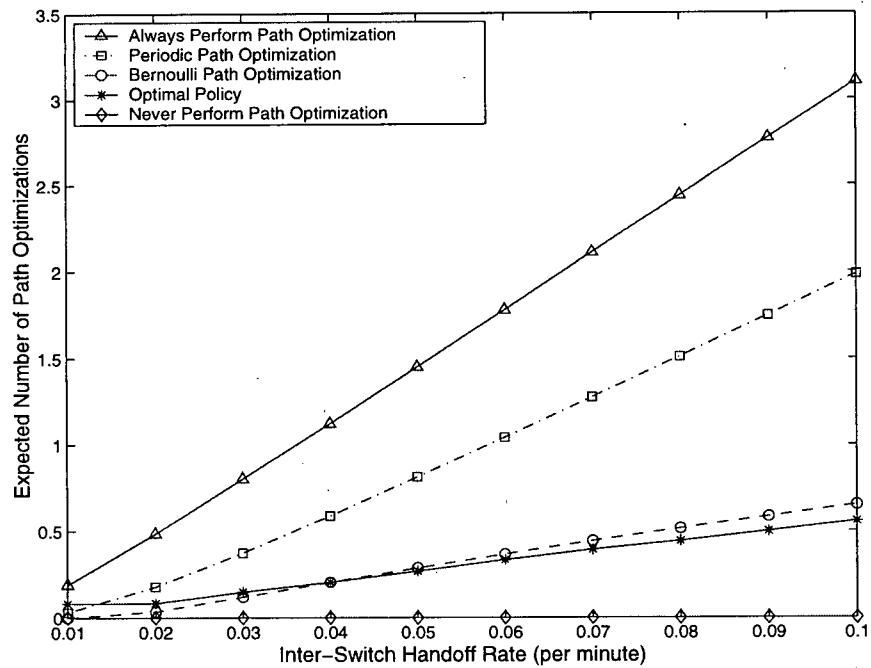
In this chapter, we addressed the issue of when to initiate path optimization for the two-phase handoff protocol. The path optimization problem is formulated as a semi-Markov decision process. Based on current state information, the network decides whether to perform path optimization after path extension. The time between inter-switch handoff fol-

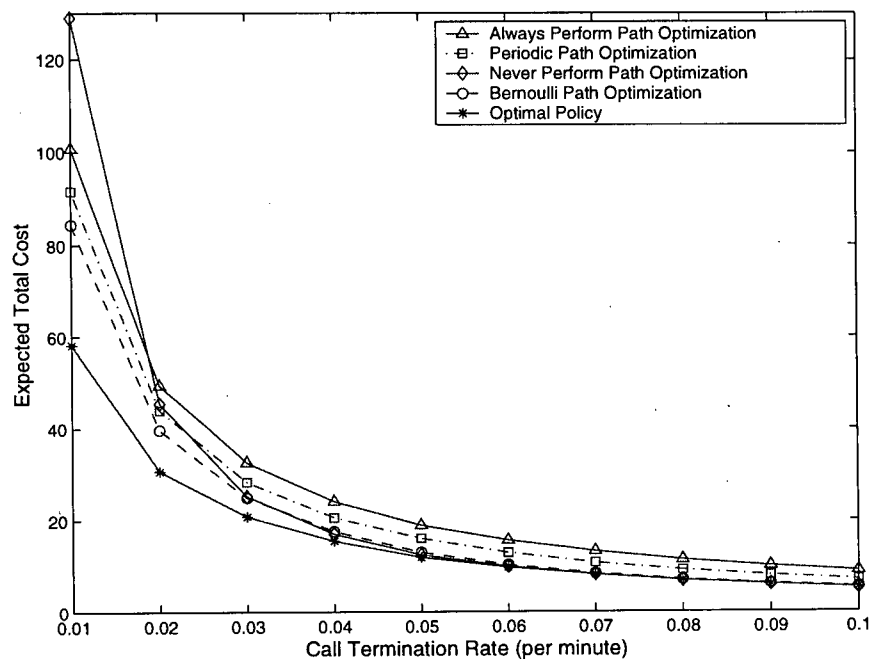
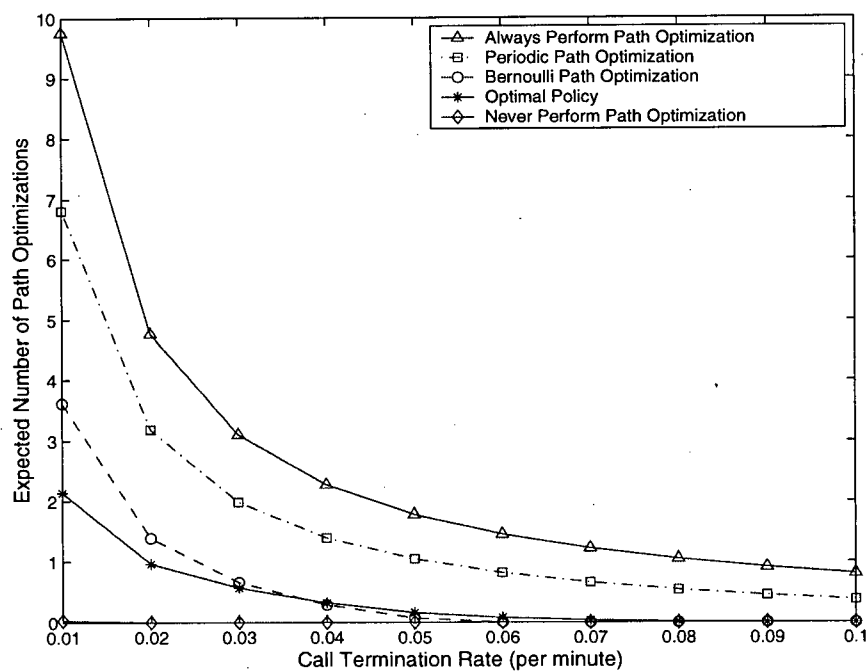
lows a general distribution and can depend on the location of the anchor switch. Although in this chapter, the network architecture is assumed to be ATM-based, the proposed model is also applicable for any connection-oriented mobile network architectures. In addition in the future IP network infrastructure which provides QoS, our model may also be used to ensure a continuous flow of the multimedia applications in the wireless domain.

We presented the value iteration algorithm which solves the optimality equations. A stationary deterministic optimal policy is obtained. Under certain conditions, the optimal policy has a threshold structure. That is, path optimization is always performed when the number of links of the path is greater than a certain threshold. The threshold structure of the optimal policy facilitates the implementation. When inter-switch handoff occurs, the decision of performing path optimization can be made by a table lookup.

The performance of the optimal policy is compared with four heuristics, namely, “always perform path optimization”, “never perform path optimization”, “periodic” and “Bernoulli” path optimizations. Simulation results indicate that the optimal policy gives a lower expected cost per call than the other four heuristics. These results imply that by using the optimal policy, the mobile connection maintains a good balance between the network resources utilized and the signaling load incurred on the network during its connection lifetime. We studied the effect of various distributions for the time between inter-switch handoffs on the expected total cost. Results indicate that the relative change in the expected total cost is not significant. We also performed the sensitivity analysis for the optimal policy with respect to the variation of the average call duration and the average time between inter-switch handoffs. Results indicate that the optimal policy is relatively insensitive to the change of the average time between inter-switch handoffs. If there is uncertainty in estimating the average call duration, it may be better to over-estimate the value in order to reduce the cost ratio difference.

Figure 3.2 Expected total cost versus link cost rate C_{link} .Figure 3.3 Expected number of path optimizations versus link cost rate C_{link} .

Figure 3.4 Expected total cost versus inter-switch handoff rate λ .Figure 3.5 Expected number of path optimizations versus inter-switch handoff rate λ .

Figure 3.6 Expected total cost versus call termination rate μ .Figure 3.7 Expected number of path optimizations versus call termination rate μ .

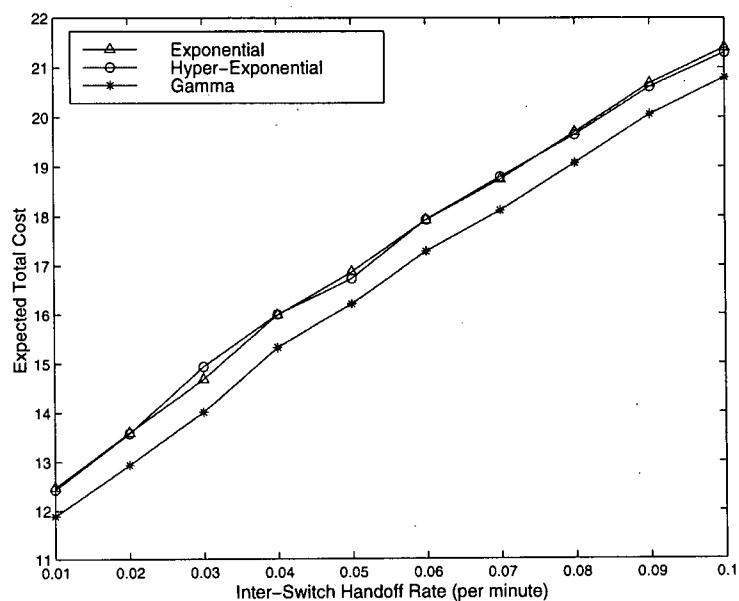


Figure 3.8 Minimum expected total cost of the optimal policy versus inter-switch handoff rate λ for various distributions.

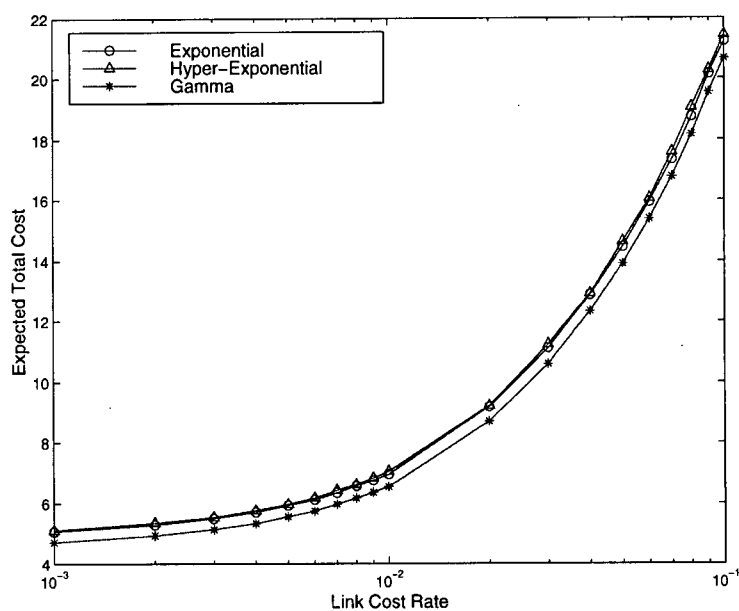


Figure 3.9 Minimum expected total cost versus link cost rate C_{link} for various inter-switch handoff rate distributions.

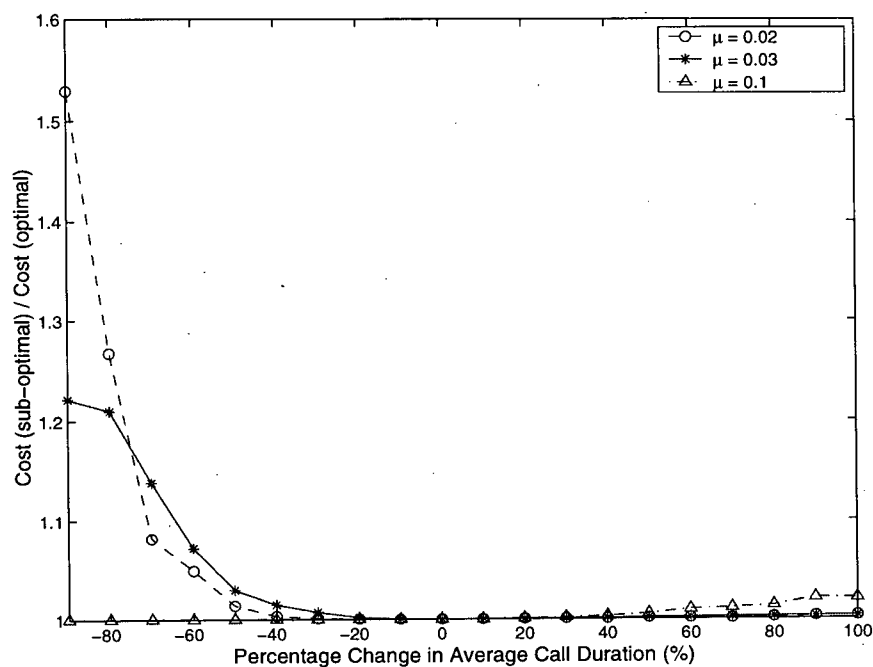


Figure 3.10 Variation of the average call termination time.

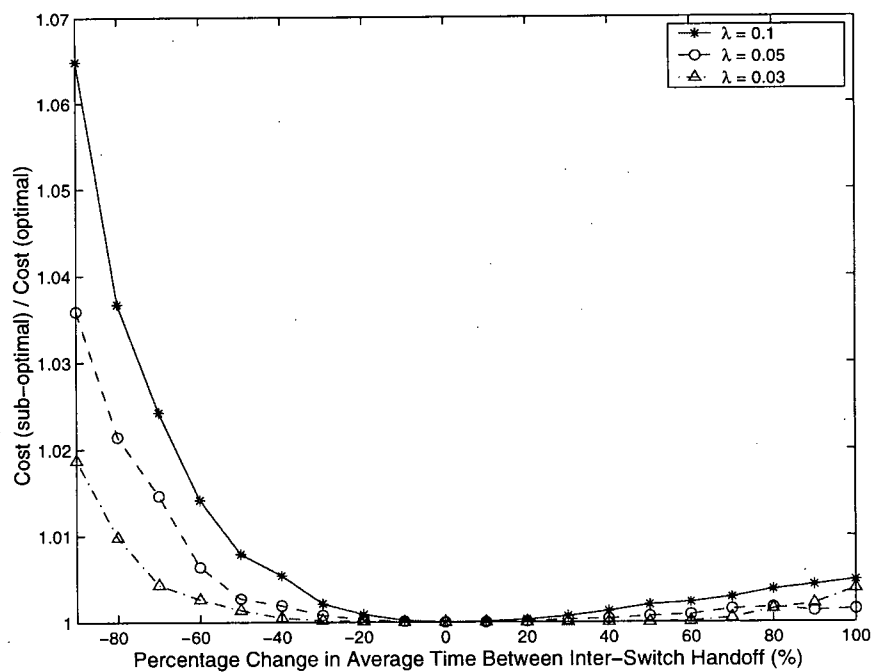


Figure 3.11 Variation of the average time between inter-switch handoffs.

"There is always another way to look at the same problem."

Richard P. Feynman, 1918-1988.

Chapter 4 Distance-based Location Update

4.1 Introduction

In this chapter, we propose a stochastic model to analyze the distance-based location update algorithm [77]. The location tracking problem is formulated as a semi-Markov decision process [46]. There is a cost function associated with location update and another cost function associated with terminal paging. The objective is to determine the optimal policy so as to minimize the expected total cost between call arrivals. Unlike other models proposed in the literature, our model eliminates some of the unrealistic assumptions commonly used. Distinct features of our model include:

1. **Cell residence time can follow general distributions:** This captures the fact that the mobile user may spend more time at certain cell locations (e.g., home or office) than some other locations. In addition, various distributions can be used to model different cell sizes (e.g., macrocell, microcell, or picocell). The average residence time in each cell can be different. The i.i.d. exponential or geometric cell residence time assumption can be relaxed.
2. **Applicable to arbitrary cell topologies:** This feature captures the fact that the number of neighboring base stations at different locations may vary in real life. Some base stations may only have two neighboring base stations while others can have as many as six. Thus, our model is not restricted to structured cell configuration such as mesh or hexagonal.
3. **Markovian movement patterns:** The probability that the mobile user moves to a par-

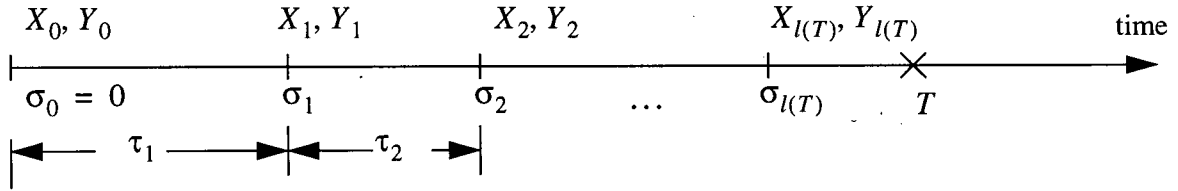


Figure 4.1 Timing diagram.

ticular neighboring cell can depend on the location of the current cell or a list of cells recently visited. The assumption of symmetric random walk movement pattern can be relaxed.

The structure of this chapter is as follows: In Section 4.2, we describe the model formulation. In Section 4.3, we derive some analytical results for the distance-based algorithm under a given cell residence time distribution. Numerical results are presented in Section 4.4. A summary is given in Section 4.5. The derivation of the optimality equations is shown in Appendix D. In the remainder of this chapter, we use the terms “mobile user” and “mobile terminal” interchangeably.

4.2 Model Formulation

The mobile terminal has to make a decision whenever it crosses a cell boundary or a certain time has elapsed. Those time instants are called *decision epochs*. Referring to Figure 4.1, the sequence $\sigma_0, \sigma_1, \dots$ represents the times of successive decision epochs. Since the network must track the user’s location perfectly during a call, the user’s location is known to the network when a call terminates. Thus, the time interval requiring mobility tracking is between the termination of the last call and the arrival of the next one. In Figure 4.1, $\sigma_0 = 0$ denotes the last call termination time and the random variable T represents the arrival time of the next call.

At each decision epoch, the mobile terminal has to decide whether to update its

location or not. The action set $A = \{0, 1\}$, where “1” represents the action of performing location update and “0” represents the null action of no intervention. The random variable Y_n is used to denote the action chosen at decision epoch n .

The mobile terminal chooses an action based on its current state information. The state information can include: the number of cells crossings since the last update (movement-based), the cell-distance between the current location and where the previous update was performed (distance-based), the velocity of the mobile terminal, or some other criteria. The random variable X_n is used to denote the state at decision epoch n .

Two cost functions are introduced to account for the network resources used for location update and terminal paging. The location update cost reflects the consumption of radio bandwidth and battery power, as well as the update processing incurred on the database. The paging cost reflects the number of cells being paged and the number of search iterations performed.

The function $f(X_n, Y_n)$ denotes the update cost at decision epoch n , given current state X_n and action Y_n chosen. Thus, $f(X_n, 1)$ represents the cost incurred, given location update is performed at decision epoch n with state X_n . On the other hand, we assume the cost is zero for no updating, i.e., $f(X_n, 0) = 0$.

For the paging cost function, referring to Figure 4.1, let $l(T)$ denote the last decision epoch before the next call arrival. The cost function $h(X_{l(T)})$ represents the cost incurred on terminal paging. We assume that the paging strategy follows the shortest-distance-first order. That is, when a paging event occurs, the search is conducted first at the user’s last reported cell. If the mobile is not found there, then the search is conducted in increasing distance order from the last reported cell, until the user is located. The maximum paging delay corresponds to the maximum number of search iterations allowed. At each search iteration, a set of cells is paged simultaneously. In this model, different

paging delay constraints are modeled by using different paging cost functions.

A decision rule prescribes a procedure for action selection in each state at a specified decision epoch. Deterministic Markovian decision rules are functions $\delta_t: S \rightarrow A$, which specify the action choice when the system occupies state s at decision epoch $t < T$. A policy $\pi = (\delta_1, \delta_2, \dots)$ is a sequence of decision rule to be used at all decision epochs.

Let $v^\pi(s)$ denote the expected total cost between call arrivals given policy π is used with initial state s . It is the sum of the location update and paging cost. Based on the above notations,

$$v^\pi(s) = E_s^\pi \left\{ \sum_{n=0}^{l(T)} f(X_n, Y_n) + h(X_{l(T)}) \right\} \quad (4.1)$$

where E_s^π denotes the expectation with respect to policy π and initial state s . In the above expression, the first term corresponds to the lump sum portion of the location update cost. The second term corresponds to the paging cost incurred upon a call arrival.

Let $G(t|X_n, Y_n)$ denote the cumulative distribution function of the time between decision epochs n and $n+1$, given current state X_n and action Y_n chosen. The time between decision epochs corresponds to the cell residence time (also known as the cell dwell time). In our formulation, the cell residence time follows a general distribution that can depend on the location of the cell. This captures the fact that a mobile user may spend more time at certain cell locations than the others. In addition, different cell sizes (e.g., macrocell, microcell, picocell) can have different cell residence time distributions. Thus, the usual i.i.d. exponential cell residence time assumption can be relaxed.

If the time between call arrivals at each mobile terminal is exponentially distributed with mean $1/\lambda$, then as shown in Appendix D, equation (4.1) can be written as:

$$v^\pi(s) = E_s^\pi \left\{ \sum_{n=0}^{\infty} e^{-\lambda \sigma_n} c(X_n, Y_n) \right\} \quad (4.2)$$

where

$$c(X_n, Y_n) = f(X_n, Y_n) + h(X_n) \int_0^{\infty} (1 - e^{-\lambda \tau}) G(d\tau | X_n, Y_n). \quad (4.3)$$

The cost function $c(X_n, Y_n)$ can be interpreted as the *effective cost* incurred at decision epoch n , given the current state X_n and that action Y_n is chosen.

A policy is said to be *stationary* if $\delta_t = \delta$ for all t . A stationary policy has the form $\pi = (\delta, \delta, \dots)$; for convenience we denote it by δ . For a stationary policy δ , equation (4.2) can be written as

$$v^\delta(s) = c[s, \delta(s)] + \sum_j \left\{ \int_0^{\infty} e^{-\lambda t} v^\delta(j) P[j|s, \delta(s)] G[dt|s, \delta(s)] \right\} \quad (4.4)$$

where $P[j|s, \delta(s)]$ denotes the transition probability that the next state is j , given the current state is s and action $\delta(s)$ is chosen. For a proof of this fact, see Proposition D.1 in Appendix D. Our objective is to determine an optimal stationary deterministic policy δ^* which minimizes (4.4). Note that for Markovian movement pattern, correlations between the directions of successive moves of the mobile user can also be incorporated if the state includes a history of the cells visited.

Since different location update algorithms correspond to maintaining different states, in the remainder of this chapter, we focus on the analysis of the distance-based location update algorithm subject to various paging delay constraints, cell residence time distributions, and network topologies.

...	-2	-1	0	1	2	...
-----	----	----	---	---	---	-----

Figure 4.2 One-dimensional linear model.

4.3 Distance-based Location Update Algorithm

In this section, we begin by analyzing the distance-based algorithm with a given cell residence time distribution $G(t)$ for all cells and a simple linear cell configuration. The value iteration algorithm is presented to evaluate the expected total cost and the optimal policy. The conditions under which the optimal policy has a threshold structure are given. These results are then extended to the hexagonal cell configuration. After that, we analyze the distance-based algorithm under arbitrary cell topology and Markovian movement patterns, and describe the implementation based on a simple table lookup.

4.3.1 Linear Cell Configuration

We assume that the mobile user moves according to a symmetric one-dimensional random walk (see Figure 4.2). That is, when the mobile user moves to a neighboring cell, the probability that the user moves to the left (or right) is equal to 0.5. The state $X(t) \in \{0, \pm 1, \pm 2, \dots\}$ of the mobile terminal at time t refers to the coordinates of the current position of the user relative to the position of the last update. For $i > 0$, $X(t) = i$ denotes the user being i units to the right at time t , and vice versa. The state $X(t) = 0$ denotes the user staying within the cell in which the mobile terminal performed the last update.

Whenever the mobile terminal crosses a cell boundary, it makes a decision whether or not to perform location update. Let $\delta(X(t))$ denote the action chosen at state $X(t)$. If $\delta(X(t)) = 0$, then $X(t^+) = X(t)$. If $\delta(X(t)) = 1$, then $X(t^+) = 0$, where $X(t^+)$ denotes the user's location immediately after the decision.

The cost for location update is assumed to be constant. That is,

$$f(X(t), a) = \begin{cases} 0, & a = 0 \\ C_{LU}, & a = 1. \end{cases} \quad (4.5)$$

The update cost represents the expenditure of radio bandwidth and processing at the database.

If a call arrives at time t , the paging cost incurred is given by $h(X(t))$. The form of the paging cost function is based on the assumption that the network starts its search for the user from the cell in which the mobile terminal last updated its location (i.e., shortest-distance-first).

Since the network must track the user's location precisely during a call, we assume the user's location is known to the network when a call ends. Starting from state $X(0) = 0$, we devise a distance-based location update policy which minimizes the expected sum of the location update costs and the paging cost of the next call.

We now introduce the optimality equations and investigate their properties. For simplicity, we denote the position $X(t)$ by i . Let $v(i)$ denote the minimum expected total cost between call arrivals given state i . The optimality equations are given by:

$$v(i) = \min \left\{ (1-K)h(i) + \frac{K}{2}[v(i+1) + v(i-1)], \right. \\ \left. C_{LU} + (1-K)h(0) + \frac{K}{2}[v(1) + v(-1)] \right\} \quad (4.6)$$

where $i = 0, \pm 1, \pm 2, \dots$, and $K = \int_0^\infty e^{-\lambda\tau} G(d\tau)$.

The first term in (4.6) denotes the expected total cost if no update is performed at state i , while the second term denotes the expected total cost if location update is performed at state i . For state $i = 0$, the minimum is achieved by the first term so that

$\delta^*(0) = 0$ and

$$v(0) = (1 - K)h(0) + \frac{K}{2}[v(1) + v(-1)]. \quad (4.7)$$

For $i \neq 0$, the optimal policy $\delta^*(i)$ is given by:

$$\delta^*(i) = \begin{cases} 0, & v(i) < C_{LU} + v(0) \\ 1, & v(i) = C_{LU} + v(0). \end{cases} \quad (4.8)$$

There are a number of iteration algorithms available to solve the optimality equations. The following value iteration algorithm finds a stationary deterministic optimal policy and the corresponding minimum expected total cost.

Value Iteration Algorithm:

1. Select $v^0(i)$, specify $\varepsilon > 0$ and set $n = 0$.
2. For each i , compute $v^{n+1}(i)$ by

$$v^{n+1}(0) = (1 - K)h(0) + \frac{K}{2}[v^n(1) + v^n(-1)] \quad (4.9)$$

$$v^{n+1}(i) = \min \left\{ (1 - K)h(i) + \frac{K}{2}[v^n(i+1) + v^n(i-1)], \right. \\ \left. C_{LU} + v^n(0) \right\} \quad (4.10)$$

where $i = \pm 1, \pm 2, \dots$

3. If $\|v^{n+1} - v^n\| < \varepsilon$, go to step 4. Otherwise increase n by 1 and return to Step 2.
4. For each i , choose $\delta^*(i) = 0$ if

$$(1 - K)h(i) + \frac{K}{2}[v^n(i+1) + v^n(i-1)] < C_{LU} + v^n(0). \quad (4.11)$$

Otherwise, choose $\delta^*(i) = 1$.

5. Stop.

We now provide a condition under which the optimal policy has a *control limit* (or *threshold*) structure. The control limit structure simply states that location update is performed with certainty whenever the distance exceeds a certain threshold.

Proposition 4.1: If the paging cost function $h(i)$ is nondecreasing in i for $i \geq 0$ and in $-i$ for $i \leq 0$, then the optimal policy has the control limit structure

$$d(i) = \begin{cases} 0, & I_L < i < I_U \\ 1, & \text{otherwise} \end{cases} \quad (4.12)$$

where $I_L < 0 < I_U$. We allow the values $I_L = -\infty$ and $I_U = \infty$, which correspond to the policy that location update is never performed. If $h(i)$ is *symmetric*, that is $h(i) = h(-i)$ for all $i > 0$, then $I_L = -I_U$.

Proof: Let $v^0(i) = 0$ for all i in the value iteration algorithm. By using induction in n , it is clear that for each n , v^n is nondecreasing in i for $i \geq 0$ and nonincreasing in i for $i < 0$. Since $v^n \rightarrow v$, the optimal expected total cost v is also nondecreasing in i for $i \geq 0$ and nonincreasing in i for $i < 0$. From (4.8), the optimal policy must have a control limit structure. \square

Let C_P denote the paging cost per cell. We now consider the scenario where $h(i)$ is symmetric and unbounded, such as

$$h(i) = C_P \cdot [2|i| + 1]. \quad (4.13)$$

In this case, the paging cost is proportional to the distance from the user's last known location. Since the paging cost function and the random walk model are both symmetric, we can deduce that v must also be symmetric in i . Thus, the value iteration algorithm only needs to consider non-negative values of i .

It is necessary to limit the range of i in order to obtain a practical value iteration algorithm. We assume the existence of an upper bound $\hat{v}(0)$ for $v(0)$, and let I_{max} be the minimal value of i such that

$$(1 - K)h(i) > C_{LU} + \hat{v}(0) > C_{LU} + v(0)$$

for all $|i| \geq I_{max}$. The existence of a finite I_{max} is guaranteed for h such as (4.13), since $(1 - K)h(i) \rightarrow \infty$ as $i \rightarrow \infty$. From (4.6)-(4.8), we obtain $v(i) = C_{LU} + v(0)$ and $\delta(i) = 1$ for $|i| \geq I_{max}$. Therefore in the value iteration algorithm, we only consider $v^n(i)$ for $|i| < I_{max}$ and set

$$v^{n+1}(-I_{max}) = v^{n+1}(I_{max}) = C_{LU} + v^n(0)$$

for all n .

To obtain an upper bound $\hat{v}(0)$, we consider the following policy:

$$\delta(i) = \begin{cases} 0, & i = 0 \\ 1, & |i| \geq 1 \end{cases}$$

in which the mobile terminal updates its position whenever it moves to another cell. We take the expected cost at state 0 of this policy as our upper bound of $v(0)$. Thus,

$$\begin{aligned} \hat{v}(0) &= (1 - K)h(0) + \frac{K}{2}[2C_{LU} + 2\hat{v}(0)] \\ &= h(0) + \frac{K}{1 - K}C_{LU}. \end{aligned}$$

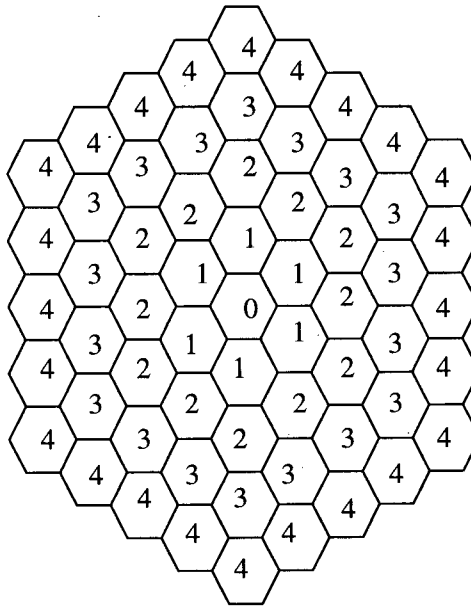


Figure 4.3 Hexagonal cell configuration.

In summary, given a general cell residence time distribution, the minimum expected total cost of updating and paging as well as the optimal policy can be determined by the value iteration algorithm. If the paging cost function is nondecreasing, the optimal policy has a threshold structure. Results in this section can be considered as an extension of [38] in which the cell residence time is restricted to be geometric, while our model allows a general cell residence time distribution.

4.3.2 Hexagonal Cell Configuration

We now extend the results in the previous section to the symmetric two-dimensional random walk model. An infinite two-dimensional hexagonal cell configuration is considered (see Figure 4.3). The probability that a user moves to each neighboring cell is equal to $1/6$. The cell residence time is assumed to be i.i.d. with a given cumulative distribution $G(t)$.

The state $X(t) \in \{0, 1, 2, \dots\}$ of the mobile terminal at time t refers to the

coordinates of the current position of the user relative to the most recent update position. The state $X(t) = 0$ denotes the user staying within the cell where the last update was performed. The state $X(t) = i$ denotes the user being located in one of those cells which is i units away from the last update location.

The cost for each location update is C_{LU} . If a call arrives at time t , the paging cost incurred is given by $h(X(t))$. For simplicity, we denote the two-dimensional position $X(t)$ by i . The optimality equations for the minimum expected total cost between call arrivals are given by:

$$v(0) = (1 - K)h(0) + Kv(1). \quad (4.14)$$

$$v(i) = \min \left\{ (1 - K)h(i) + K \left[\left(\frac{1}{3} - \frac{1}{6i} \right) v(i-1) + \frac{1}{3} v(i) + \left(\frac{1}{3} + \frac{1}{6i} \right) v(i+1) \right], \right. \\ \left. C_{LU} + (1 - K)h(0) + Kv(1) \right\} \quad (4.15)$$

where $i = 1, 2, 3, \dots$ and $K = \int_0^\infty e^{-\lambda\tau} G(d\tau)$.

The first term in (4.15) denotes the expected total cost if no update is performed at state i , while the second term denotes the expected total cost if location update is performed at state i .

If state $i = 0$, the optimal policy $\delta^*(0) = 0$. For state $i \geq 1$, the optimal policy $\delta^*(i)$ is given by

$$\delta^*(i) = \begin{cases} 0, & v(i) < C_{LU} + v(0) \\ 1, & v(i) = C_{LU} + v(0). \end{cases} \quad (4.16)$$

The value iteration algorithm in the previous section can be used to determine the optimal policy and the minimum expected total cost. We only need to replace equations (4.9) and (4.10) by

$$v^{n+1}(0) = (1-K)h(0) + Kv^n(1). \quad (4.17)$$

$$v^{n+1}(i) = \min \left\{ (1-K)h(i) + K \left[\left(\frac{1}{3} - \frac{1}{6i} \right) v^n(i-1) + \frac{1}{3} v^n(i) + \left(\frac{1}{3} + \frac{1}{6i} \right) v^n(i+1) \right], C_{LU} + v^n(0) \right\} \quad (4.18)$$

where $i = 1, 2, 3, \dots$. And replace equation (4.11) by:

$$(1-K)h(i) + K \left[\left(\frac{1}{3} - \frac{1}{6i} \right) v^n(i-1) + \frac{1}{3} v^n(i) + \left(\frac{1}{3} + \frac{1}{6i} \right) v^n(i+1) \right] < C_{LU} + v^n(0) \quad (4.19)$$

Similar to the linear cell configuration, the optimal policy in a hexagonal cell configuration also has a threshold structure if $h(i)$ is a non-decreasing function. In that case,

$$\delta^*(i) = \begin{cases} 0, & i < I \\ 1, & i \geq I. \end{cases} \quad (4.20)$$

where I denotes the distance threshold. Interested readers can refer to [59] for a proof of this fact.

In summary, for a hexagonal cell configuration, under the assumption of a given general cell residence time distribution and random walk movement pattern, the minimum expected total cost of updating and paging as well as the optimal policy can be determined by the value iteration algorithm. Results in this section can be considered as an extension of [27]. The key difference between our results and those in [27] is that we consider mobility tracking between calls and do not make the assumption that the user's location evolves to a steady-state by the time the next call arrives. In addition, our model can

analyze any given cell residence time distribution where the model in [27] is restricted to the exponential distribution.

4.3.3 Arbitrary Cell Topology

Although the model formulations of the distance-based algorithm in the previous sections give us some insight on the structure of the optimal policy and allow us to extend the work reported in [38] and [27], they are not practical for implementation simply because (1) it is difficult to keep track of the distance between two cells, and (2) the analysis are still under certain unrealistic assumptions (e.g., random walk movement pattern). To implement the distance-based algorithm efficiently, we propose that: *“the mobile terminal maintains the information of the identifiers of the current cell and the cell that performed the last update. Whenever the mobile terminal crosses a cell boundary, the decision of location update is made by a simple table lookup.”* This model is formally described below.

Let (i, j) denote the state, where i represents the identifier of the current cell that the mobile terminal is residing, and j represents the identifier of the cell in which the mobile terminal performed its last update.

For a particular mobile user, different cell residence time distributions can be assigned to different cell locations. The average residence time in each cell can be different. This is achieved by letting $G(t|i)$ denote the cumulative distribution function of the cell residence time, given the identifier of the current cell is i .

For the Markovian movement pattern in an arbitrary cell topology, we let $P(k|i)$ denote the probability that the mobile user moves to neighboring cell k in the next decision epoch, given the current cell identifier is i . This captures the correlations of the user movement between two neighboring cells. If the state information includes the identifiers of the recently visited cells, then a history-based mobility model for a particular user can be obtained.

The location update cost is C_{LU} and the paging cost function is given by $h(i, j)$. Let $v(i, j)$ denote the minimum expected total cost between call arrivals given state (i, j) . The optimality equations are given by:

$$v(i, j) = \min \left\{ \int_0^\infty (1 - e^{-\lambda t}) h(i, j) G(dt|i) + \sum_k \left[\int_0^\infty e^{-\lambda t} v(k, j) P(k|i) G(dt|i) \right], \right. \\ \left. C_{LU} + \int_0^\infty (1 - e^{-\lambda t}) h(i, i) G(dt|i) + \sum_k \left[\int_0^\infty e^{-\lambda t} v(k, i) P(k|i) G(dt|i) \right] \right\} \quad (4.21)$$

where $1 \leq i, j \leq N$ and N denotes the number of base stations within the coverage area. The first term in (4.21) denotes the expected total cost if no update is performed at state (i, j) , while the second term denotes the expected total cost if location update is performed at state (i, j) . Equation (4.21) can be solved by the value iteration algorithm to obtain the minimum expected total cost and the optimal policy $\delta^*(i, j)$. Since the mobile terminals usually have limited processing power, the computations can be performed either at the base stations or the switches.

Note that there are some differences between the distance thresholds computed in an arbitrary cell topology to those derived from a structured cell topology. In a structured cell topology (e.g., hexagonal), the number of neighboring cells is the same for all cell locations. Since the derivation of the optimal distance threshold in a structured cell configuration is under the assumptions of symmetric random walk and i.i.d. cell residence time distribution, *the optimal distance threshold is the same for all cell locations*. As an example, refer to Figure 4.3. Suppose cell "0" is the last update location and the distance threshold is equal to 3. Whenever the mobile terminal moves to a cell location with distance threshold greater than or equal to 3, location update is performed.

One the other hand, in an arbitrary cell topology, the number of neighboring cells

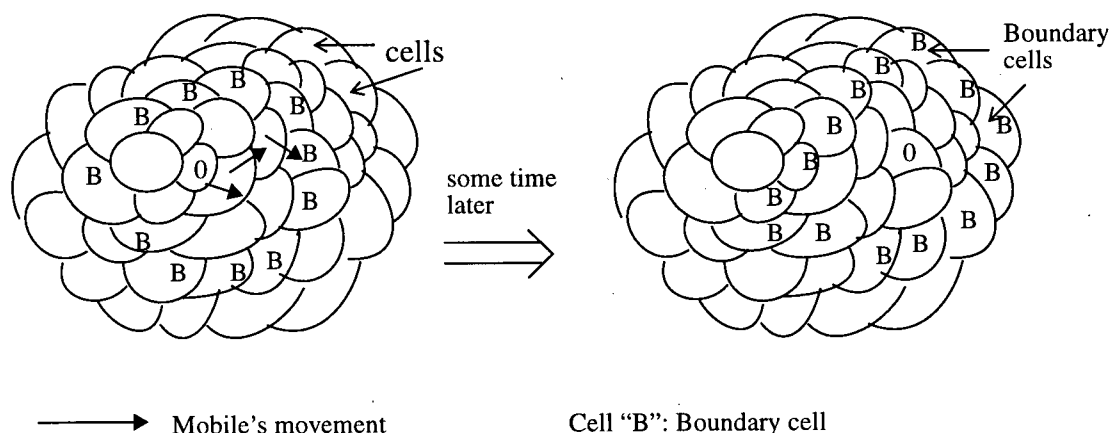


Figure 4.4 Update boundary derived from an arbitrary cell configuration. Location update is performed whenever the mobile terminal moves to any cells labeled with "B".

is different at different cell locations. Since our proposed model allows Markovian movement pattern and general cell residence time distributions at different cell locations:

1. For any cells located on the *update boundary*, their cell-distance measured from the last updated cell location may be different.
2. Different cell locations may have different update boundary.

As an example, refer to Figure 4.4. The cell labeled with "0" is the cell location where update was last performed. The update boundary consists of the set of cells labeled with "B". Whenever the mobile terminal moves to those boundary cells, location update is performed. Note that some boundary cells have cell-distance equals to 2 while some have cell-distance equal to 3. After an update is performed in a boundary cell, that cell will be labeled as "0" and a new set of boundary cells will be chosen.

For implementation, after each location update or a call termination, the mobile terminal needs to download the list of the update boundary cell identifiers. Whenever the mobile terminal moves to another cell, it compares the new cell identifier with the list of the update boundary cell identifiers. Location update is performed if the new cell is one of

those update boundary cells.

4.4 Numerical Results and Discussions

We now present some numerical results for the distance-based location update algorithm. The performance metrics are the expected total cost and the optimal distance threshold. We are interested in understanding how the (1) paging delay constraints, (2) cell residence time distributions, (3) cell topologies, and (4) movement patterns affect the expected total cost and the optimal distance threshold.

4.4.1 Paging Delay Constraints

We consider a hexagonal cell configuration with random walk movement pattern. The cell residence time is assumed to be Gamma distributed with the following cumulative distribution function:

$$G(t) = 1 - \sum_{n=0}^{\kappa-1} \frac{(\mu t)^n e^{-\mu t}}{n!} \quad \kappa = 1, 2, 3, \dots \quad (4.22)$$

If $1/\bar{\mu}$ represents the mean cell residence time, then $\bar{\mu} = \mu/\kappa$. The factor κ controls the variance of the distribution. A higher value of κ gives a lower variance. In this section, we assume $\kappa = 2$.

The maximum paging delay corresponds to the maximum number of search iterations allowed. When there is no paging delay constraint, we assume the paging strategy follows the shortest-distance-first order. Let state i denote the user being located in one of those cells which is i units away from the last update location. The paging cost function is:

$$h(i) = C_p[3i(i+1) + 1]. \quad (4.23)$$

When the maximum paging delay is equal to one, all cells within the distance

threshold are paged simultaneously. Let I denote the distance threshold. The paging cost function is:

$$h(i) = \begin{cases} C_P[3I(I-1)+1], & 0 \leq i < I \\ \infty, & i \geq I. \end{cases} \quad (4.24)$$

When the maximum paging delay is equal to two, we assume all the cells within half the distance of the threshold are paged first. If the mobile terminal is not located, then the remaining cells within the distance threshold are paged simultaneously. For the paging cost function with distance threshold $I = 1$,

$$h(i) = \begin{cases} C_P, & i = 0 \\ \infty, & i > 0. \end{cases} \quad (4.25)$$

For the paging cost function with distance threshold $I > 1$,

$$h(i) = \begin{cases} C_P \left\{ 3 \left(\left\lfloor \frac{I}{2} \right\rfloor - 1 \right) \left\lfloor \frac{I}{2} \right\rfloor + 1 \right\}, & 0 \leq i \leq \left\lfloor \frac{I}{2} \right\rfloor - 1 \\ C_P[3(I-1)I+1], & \left\lfloor \frac{I}{2} \right\rfloor \leq i < I \\ \infty, & |i| \geq I. \end{cases} \quad (4.26)$$

We assume that the call arrival rate λ is 0.01 per minute, the cell crossing rate $\bar{\mu}$ is 0.1 per minute, the location update cost C_{LU} is 10, and the paging cost per cell C_P is 1.

Figure 4.5 shows the expected total cost versus the call arrival rate λ of each user under various delay constraints. When the call arrival rate is high, the average inter-arrival time is much smaller than the average cell residence time, in which case the network tracks the user's location perfectly. The expected total cost decreases when the call arrival rate increases. The cases of "paging delay = 1" and "no paging delay constraint" provide the upper and lower bounds for the expected total cost, respectively. As the maximum

paging delay increases, there is a reduction in the expected total cost.

Figure 4.6 shows the optimal distance threshold versus the call arrival rate λ under various delay constraints. The optimal distance threshold decreases as the call arrival rate increases. The cases of “no delay constraint” and “paging delay = 1” provide the upper and lower bounds for the optimal distance threshold, respectively. As the maximum paging delay increases, there is an increase in the optimal distance threshold.

Figure 4.7 shows the expected total cost versus cell crossing rate $\bar{\mu}$ under various delay constraints. The expected total cost increases when the outgoing rate increases. When the cell crossing rate is high, the average cell residence time is small. In that case, the mobile user moves frequently which increases the need to perform location update.

Figure 4.8 shows the optimal distance threshold versus cell crossing rate $\bar{\mu}$ under various delay constraints. The optimal distance threshold increases when the cell crossing rate increases. When the cell residence time is large (i.e., cell crossing rate is small), the mobile terminal has a high chance to be within the last update location. A small distance threshold reduces the paging cost. On the other hand, when the cell residence time is small (i.e., cell crossing rate is large), the mobile terminal may have crossed a large number of cells before a call arrives. A high distance threshold reduces the update cost.

Figure 4.9 shows the expected total cost versus the location update cost C_{LU} under various delay constraints. The behavior for low and high values of location update costs is in accordance with intuition. When the update cost is low, more updates are being performed in order to reduce the paging cost. When the update cost increases, there is no incentive to perform location update. Figure 4.10 confirms the above reasoning. The optimal distance threshold increases as the update cost increases.

4.4.2 Cell Residence Time Distributions

In this section, we study the effect of the cell residence time distributions on the expected

total cost and the distance threshold. We assume a hexagonal cell configuration with random walk movement pattern. Three cell residence time distributions are chosen: exponential, hyper-exponential, and Gamma. The Gamma distribution is stated in equation (4.22). For the hyper-exponential distribution, it has the following cumulative distribution function:

$$G(t) = 1 - [pe^{-\mu_1 t} + (1-p)e^{-\mu_2 t}]. \quad (4.27)$$

We assume $\mu_1 = 2\bar{\mu}$, $\mu_2 = (2/3)\bar{\mu}$, and $p = 0.5$. We also assume that the call arrival rate λ is 0.01 per minute, the location update cost C_{LU} is 10, and the paging cost per cell C_P is 1.

Figure 4.11 shows the optimal distance threshold versus the cell crossing rate $\bar{\mu}$ under different cell residence time distributions and paging delay constraints. Results indicate that for a given cell crossing rate $\bar{\mu}$, the values of the optimal distance threshold are in general the same for these cell residence time distributions. Their values only differ for $0.007 \leq \bar{\mu} \leq 0.01$ with paging delay constraint equals to 1 and for $0.08 \leq \bar{\mu} \leq 0.09$ with no paging delay constraint.

Figure 4.12 shows the expected total cost versus the cell crossing rate $\bar{\mu}$ under three different cell residence time distributions and paging delay constraints. Results show that the expected total costs between various cell residence time distributions are quite close. One way to explain this is by observing equations (4.14)-(4.15). Different cell residence time distributions correspond to different values of K . For example, when $\lambda = 0.01$ and $\bar{\mu} = 0.1$: $K_{\text{exponential}} = 0.909$, $K_{\text{hyper-exponential}} = 0.911$, and $K_{\text{Gamma}} = 0.907$. For the range of parameters that we have chosen to investigate, the values of K under those three cell residence time distributions are quite close. Thus, the expected total costs between these distributions are also quite close. These results indicate

that for a hexagonal cell configuration with random walk movement pattern, the effect of changing the cell residence time distributions is not significant.

4.4.3 Cell Topologies

In the previous sections, we assume a hexagonal cell configuration for the analysis. Rather than using a structured cell topology, in this section, we use a graph model to model the cellular network. In general, the interconnection of the cells can be modeled as a connected graph $G = (N, E)$, where the *node set* N represents the set of cell or base station identifiers and the *edge set* E represents the connectivity between two neighboring cells. For example, referring to Figure 4.13, the node set $N = \{a, b, c, d, e, f, g\}$ and the edge set $E = \{(a, b), (a, c), (b, c), \dots, (f, g)\}$.

Since we are unable to obtain an actual cell topology from the public domain, we use a random graph model to represent the topology of a cellular network. The advantages of using a random graph model are: (1) the number of neighboring base stations for each base station can be different; and (2) only the nodes that are close together are connected. This models the connectivities of the neighboring base stations. The generation of a random graph is described in Section 2.4. In our model, we consider a coverage area that consists of 100 base stations with an average node degree of 4. An example of a random graph model is shown in Figure 4.14.

The cell residence time is assumed to be Gamma distributed. A symmetric random walk movement model is assumed. We also assume that the call arrival rate λ is 0.01 per minute, the cell crossing rate $\bar{\mu}$ is 0.1 per minute, the location update cost C_{LU} is 10, and the paging cost per cell C_p is 1.

Figure 4.15 shows the relative frequency distribution of the average optimal distance threshold. For each last updated location, its average optimal distance threshold is defined as the average of all the cell-distance of the update boundary cells relative to the

last updated cell location. For illustration purpose, the optimal distance threshold is rounded to the closest integer. Figure 4.15 shows that for a random graph topology, approximately 62% of the cells have a distance threshold of 4, 25% of the cells have a distance threshold of 5, 10% of the cells have a distance threshold of 3, and 3% of the cells have a distance threshold of 6. Note that, for the same set of cost and mobility parameters, the optimal distance threshold derived from a hexagonal cell configuration is equal to 4.

4.4.4 Performance Comparisons

One question arises as to the performance gain of using (1) a random graph model with Markovian movement pattern as compared to (2) a hexagonal cell configuration with symmetric random walk movement pattern. We now compare the performance between these two models.

Since the mobile user usually has a destination in mind, we model this behavior by choosing one particular node (or cell) in the random graph as the destination. Whenever the mobile user leaves the current cell, it moves to a neighboring cell which is closest to the destination. This captures the behavior of moving towards the destination. If the mobile user is staying within the destination cell, after a certain period of time it will move to one of the neighboring cells. This continues until the next call arrives.

We now describe the procedures of comparing the distance thresholds determined from our model to those derived from a hexagonal cell configuration.

1. Given the cost and mobility parameters, we first use a hexagonal cell configuration with symmetric random walk movement pattern to obtain the optimal distance threshold. This optimal distance threshold is then applied to the random graph model with Markovian movement pattern.
2. The expected total cost of location update and paging between call arrivals is then determined. This cost is denoted as "Cost (hexagonal)". The term "hexagonal" is used

to remind us that the optimal distance threshold is derived from the hexagonal cell configuration.

3. We also use the random graph model with the above movement pattern to determine the minimum expected total cost by solving the optimality equations. This cost is denoted as “Cost (optimal)”. The term “optimal” is used to remind us that the update boundary corresponds to the optimal policy.
4. The performance gain is the cost ratio which is defined as Cost (hexagonal) / Cost (optimal).

The parameters used include: call arrival rate $\lambda = 0.01$ per minute, the location update cost $C_{LU} = 10$, and the paging cost per cell $C_P = 1$. We assume that the cell residence time follows an i.i.d. Gamma distribution with rate μ .

Figure 4.16 shows the cost ratio versus the call arrival rate λ and the location update cost C_{LU} under different cell crossing rate μ . From these figures, we observe that the cost ratio increases when μ increases or λ decreases. And it approaches unity when λ or C_{LU} are large. In Figure 4.16 (a), when the average time between call arrivals is large (i.e., λ is small), the optimal update boundary obtained from our model gives a lower cost than the distance threshold derived from the hexagonal model. However, when the average time between call arrivals is small (i.e., λ is large), the mobile user does not travel much before a call arrives. Thus, the distance thresholds derived from both methods give the same performance. In Figure 4.16 (b), the variation of the cost ratio with respect to C_{LU} is due to the changes of the optimal update boundary or distance threshold for different update cost values. When C_{LU} is large, there is no incentive to perform location update. The expected total cost only consists of the paging cost. Thus, the cost ratio approaches unity.

These results imply that in real wireless cellular networks environments in which the cell topology is not structured and the user movement pattern is not random, our model

can provide a more accurate update boundary than that derived from a hexagonal cell configuration with random walk movement pattern. And by using a more accurate location update boundary, the network can maintain a better balance between the processing incurred due to location update and the radio bandwidth utilized for paging between call arrivals.

4.4.5 Sensitivity Analysis

We also perform sensitivity analysis for the optimal policy with respect to the variation of the average time between call arrivals and average cell residence time. The procedures for the sensitivity analysis are similar to those described in Section 2.5 in Chapter 2.

Figure 4.17 (a) shows the cost ratio versus Δ_λ under different call arrival rate λ and Figure 4.17 (b) shows the cost ratio versus Δ_μ under different cell crossing rate μ . From these figures, we observe that the cost ratio is more sensitive to the under-estimation of both λ and μ . If the target cost ratio has to be less than 1.05 (i.e., 5% difference between the optimal and sub-optimal cost), then Δ_λ has to be greater than -50% and Δ_μ has to be greater than -60%. These results imply that if there is uncertainty in estimating λ or μ , it may be better to over-estimate the values in order to reduce the cost ratio difference.

4.5 Summary

In this chapter, we proposed a stochastic model to analyze the distance-based location update algorithms. The location tracking problem is formulated as a semi-Markov decision process. Based on the current state information, the mobile terminal decides whether to update its location whenever it crosses a cell boundary.

The Markovian movement pattern allows the study of a variety of mobility models. For example, if the state information includes the identifiers of the recently visited cells,

then a history-based mobility model for a particular user can be obtained. The cell residence time follows a general distribution and can depend on the location of the cell. The usual i.i.d. assumption for the cell residence time distribution can be relaxed.

For the distance-based algorithm, we showed that the optimal policy has a threshold structure for nondecreasing paging cost functions. We proposed an efficient scheme to implement the distance-based algorithm in an arbitrary cell topology. After each location update or a call termination, the mobile terminal needs to download the list of the update boundary cell identifiers corresponds to its current location. Whenever the mobile terminal moves to another cell, it compares the new cell identifier with the list of the update boundary cell identifiers. Location update is performed if the new cell is one of those update boundary cells.

We presented numerical results for the distance-based update algorithm in both hexagonal and random graph models. Results indicate that the distance threshold (or update boundary) computed from our proposed model has a better performance (in terms of a lower cost) than the distance threshold derived from a hexagonal cell configuration with random walk movement pattern. These results imply that by using the optimal policy, the network maintains a good balance between the processing incurred on location update and the radio bandwidth utilized for paging between call arrivals.

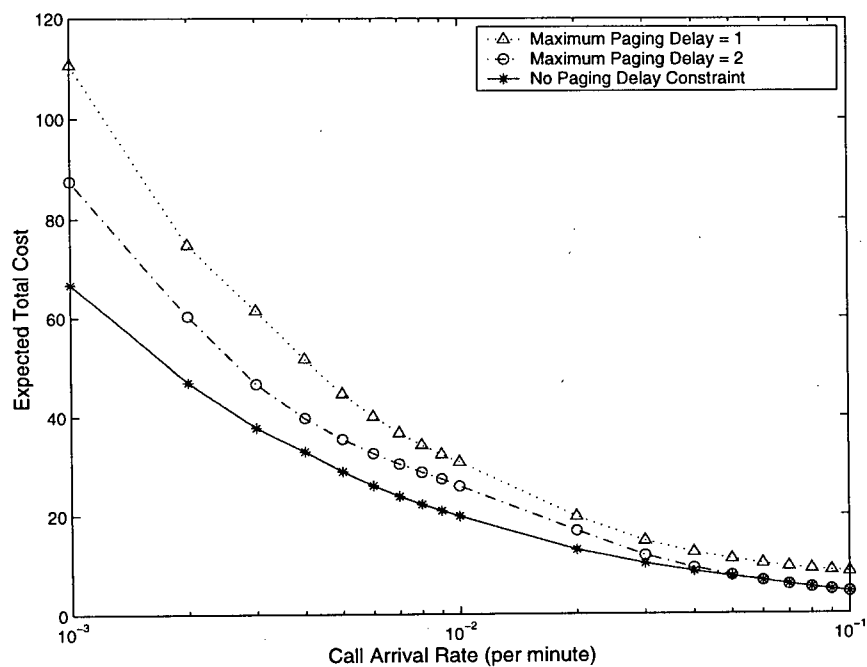


Figure 4.5 Expected total cost versus call arrival rate under various delay constraints.

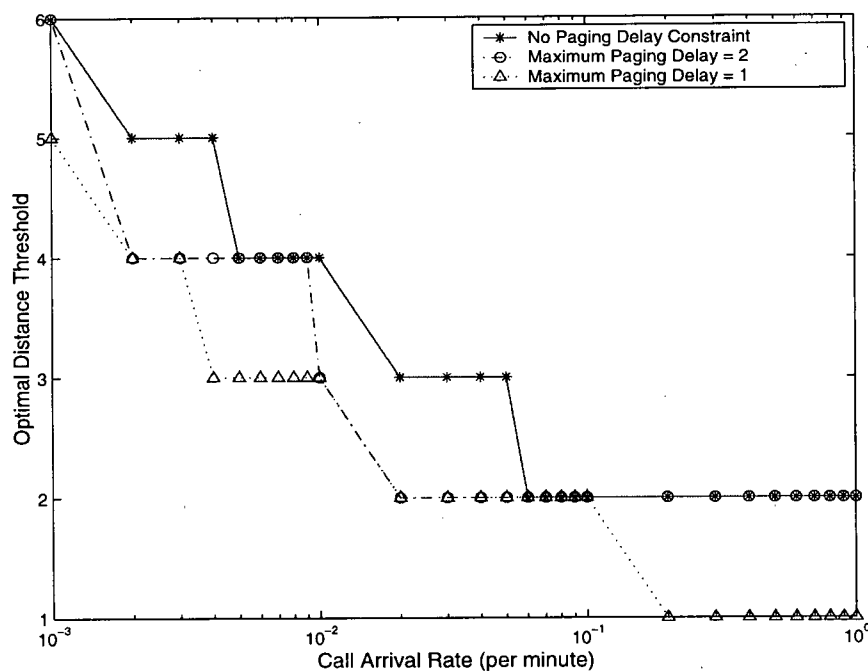


Figure 4.6 Optimal distance threshold versus call arrival rate under various delay constraints.

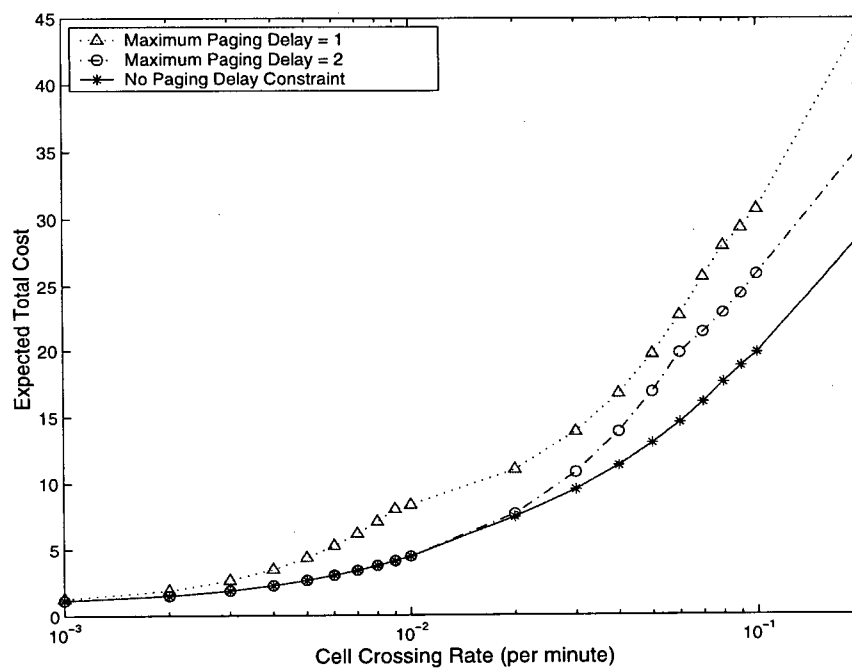


Figure 4.7 Expected total cost versus cell crossing rate under various delay constraints.

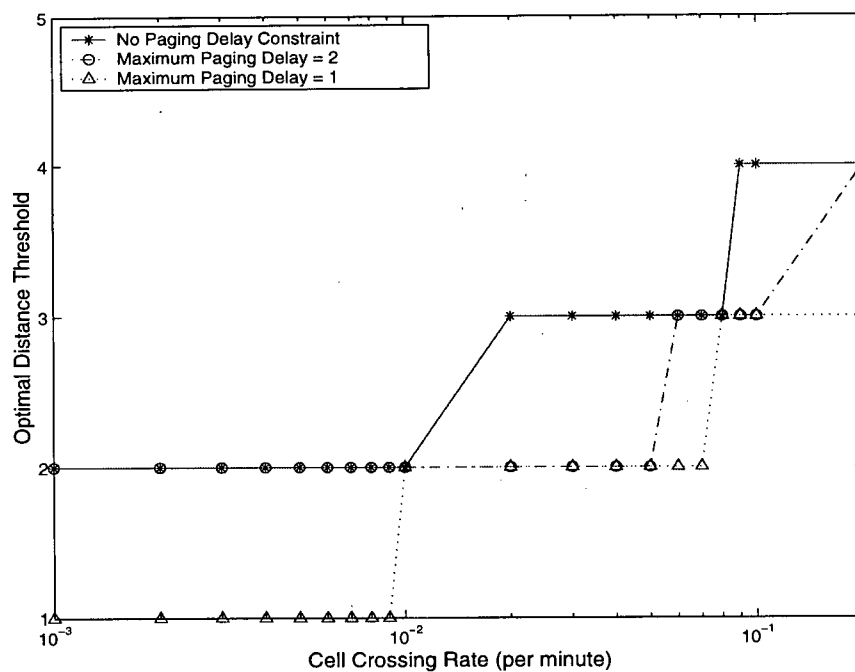


Figure 4.8 Optimal distance threshold versus cell crossing rate under various delay constraints.

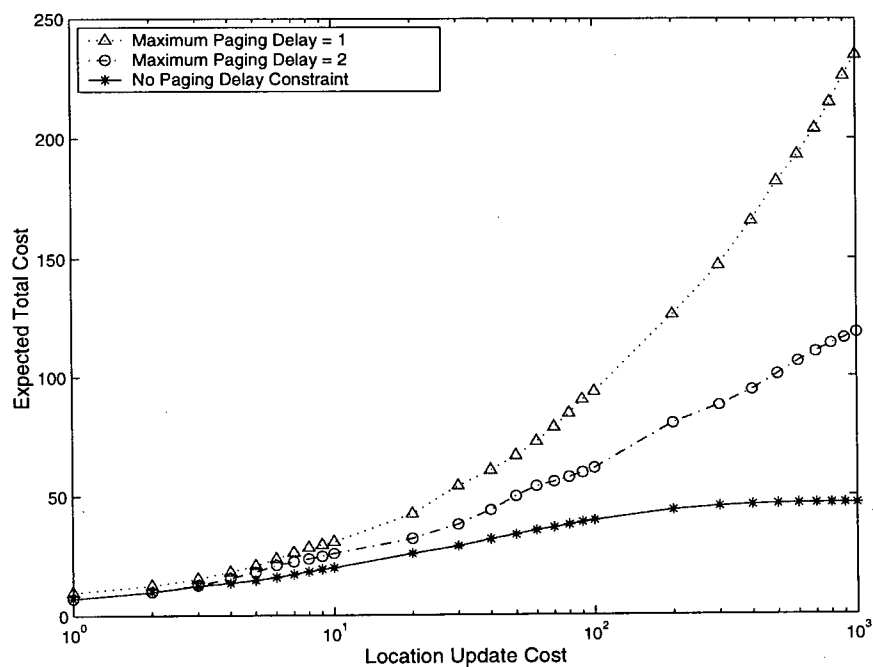


Figure 4.9 Expected total cost versus location update cost under various delay constraints.

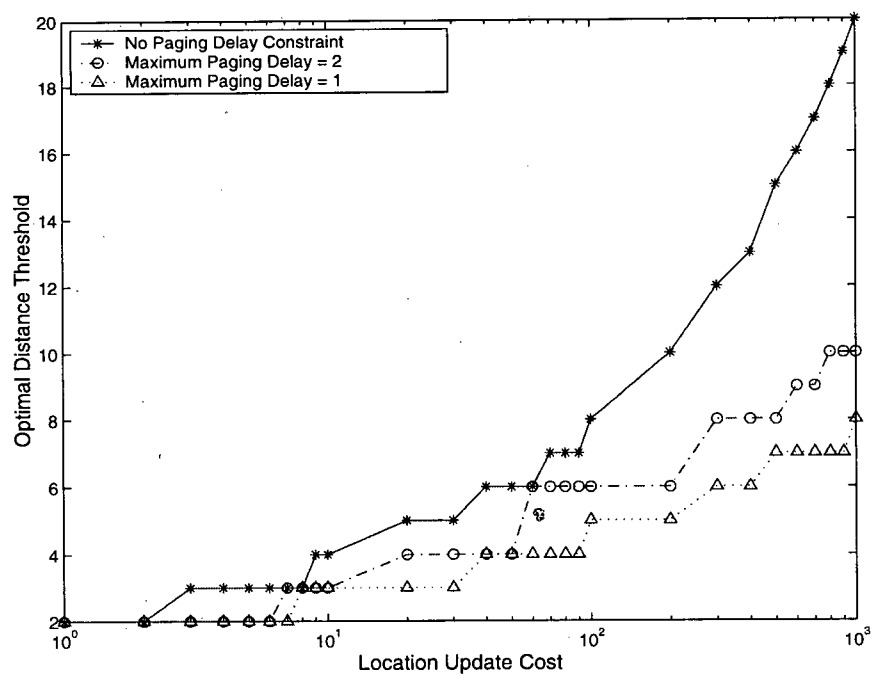


Figure 4.10 Optimal distance threshold versus location update cost under various delay constraints.

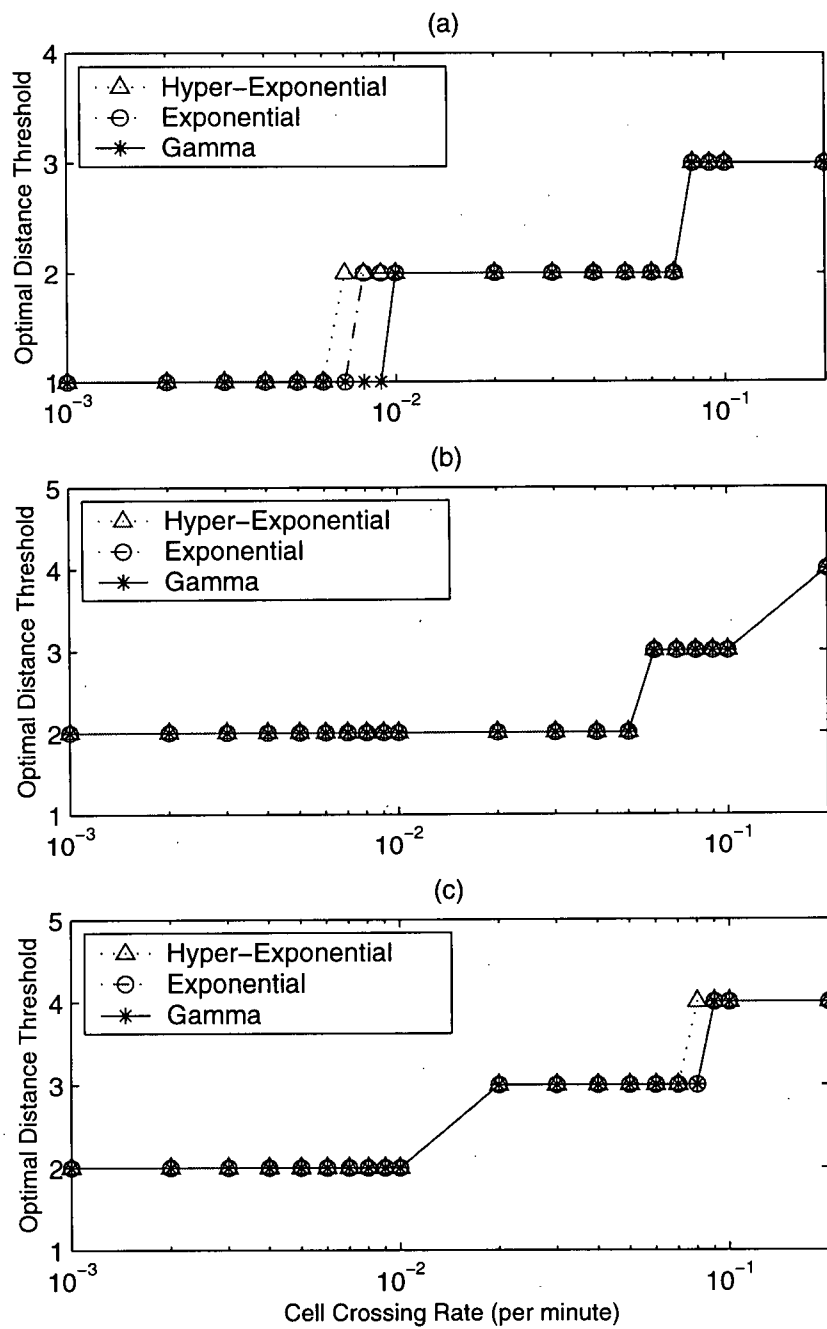


Figure 4.11 Optimal distance threshold versus cell crossing rate under various cell residence time distributions and delay constraints: (a) maximum paging delay constraint = 1, (b) maximum paging delay constraint = 2, (c) no paging delay constraint.

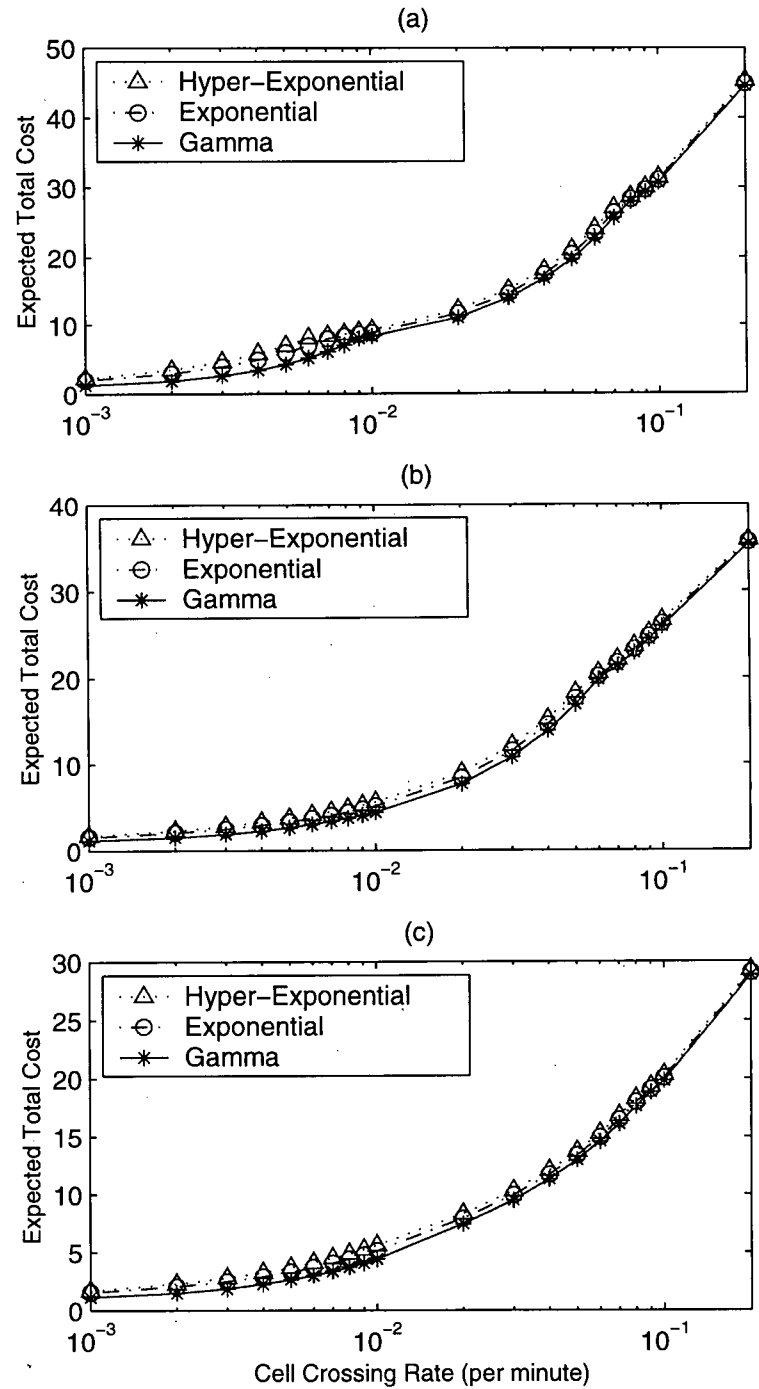
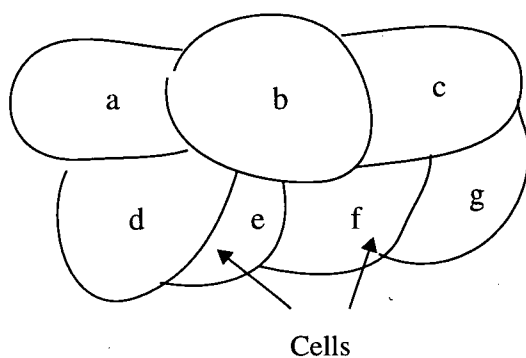
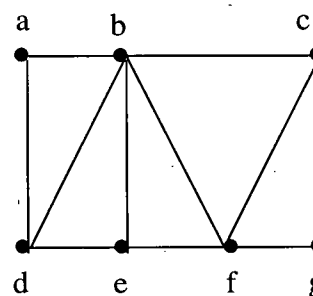


Figure 4.12 Expected total cost versus cell crossing rate under various cell residence time distributions and paging delay constraints: (a) maximum paging delay = 1, (b) maximum paging delay = 2, (c) no paging delay constraint.



(a) The cell topology.



(b) Graph model showing the interconnections of the cells.

Figure 4.13 Representation of a cellular network topology by a graph model.

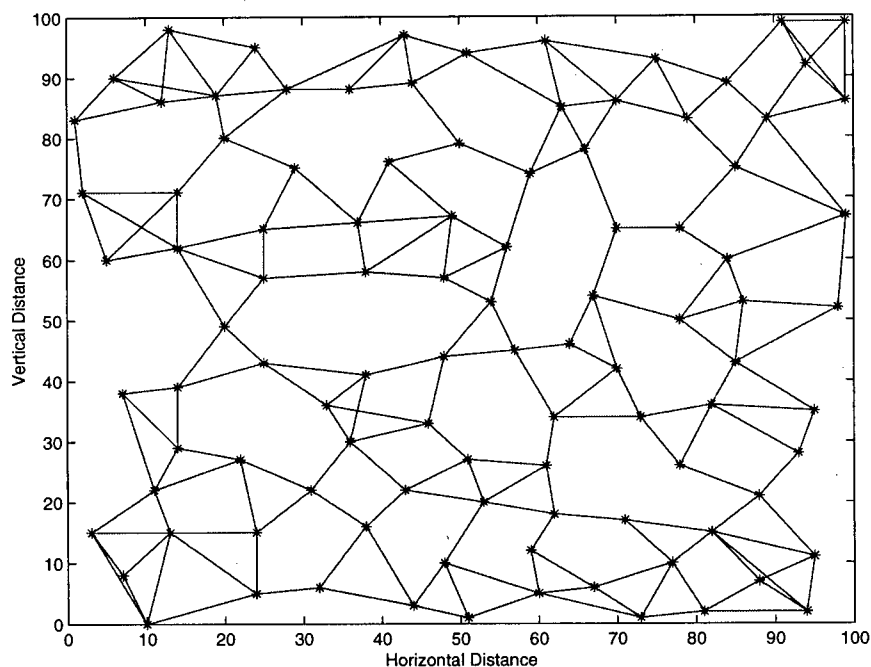


Figure 4.14 Graph model with an average node degree of 4. The nodes represent the location of the base stations. An edge between two nodes represents those two base stations are neighbors to each other.

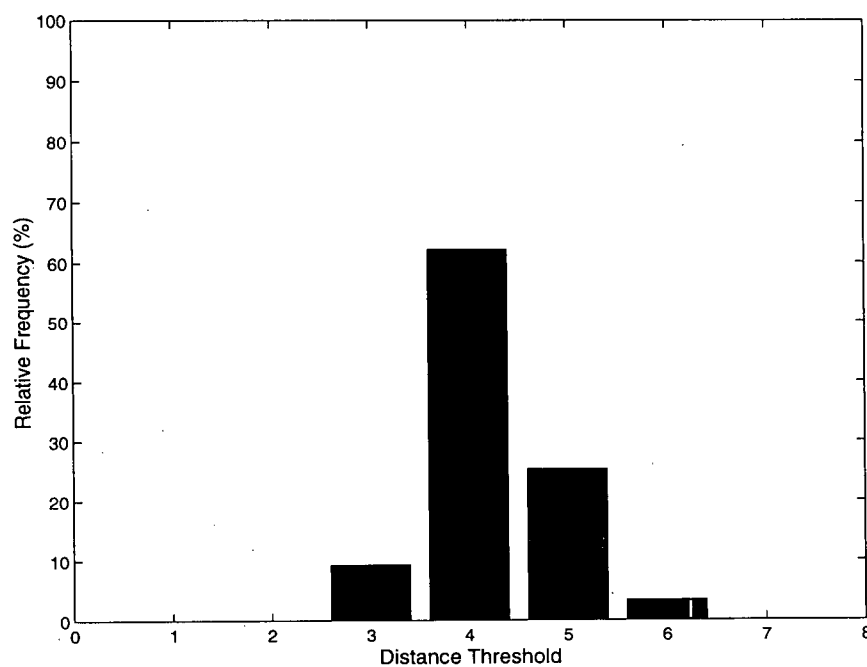


Figure 4.15 Optimal distance threshold distribution.

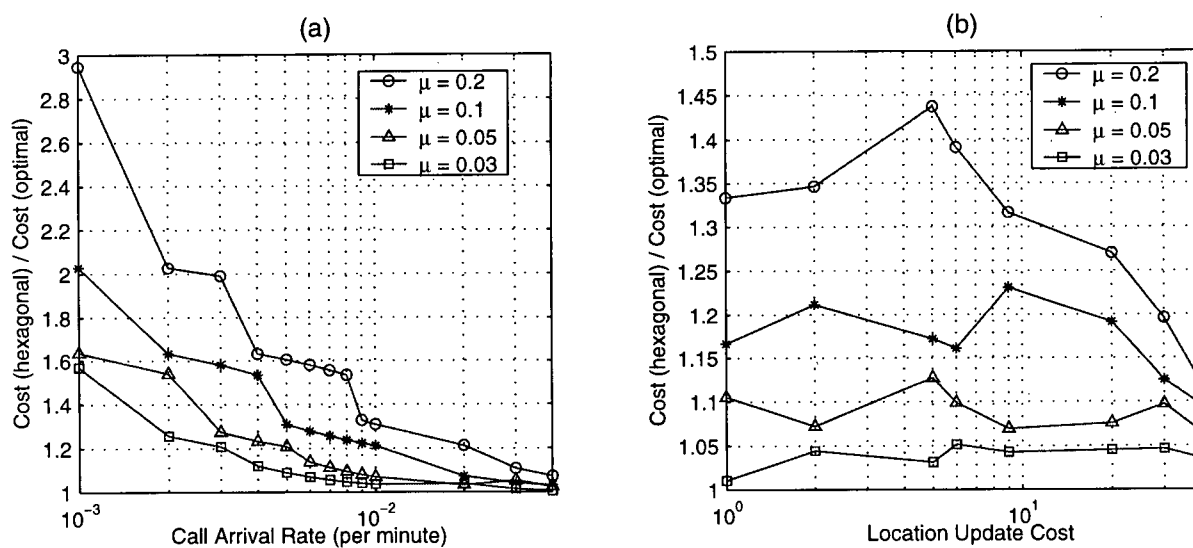


Figure 4.16 Cost ratio versus call arrival rate and location update cost under different cell crossing rate.

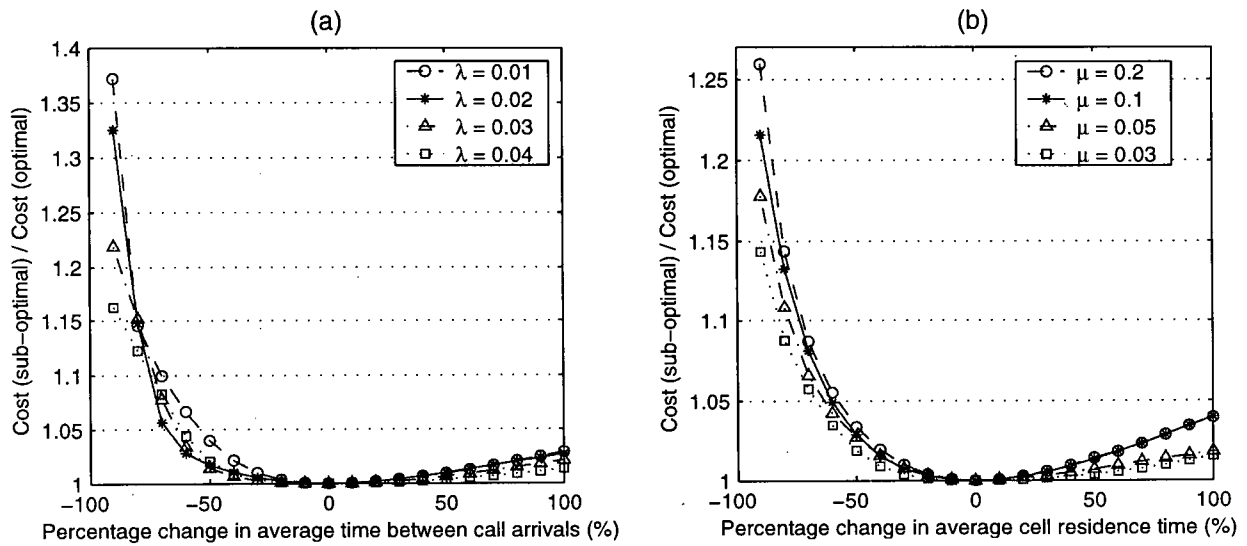


Figure 4.17 (a) Cost ratio versus percentage change in average time between call arrivals under different call arrival rate; (b) Cost ratio versus percentage change in average cell residence time under different cell crossing rate.

*"I fall, I stand still...
I trudge on, I gain a little...
I get more eager and climb higher and
begin to see the widening horizon.
Every struggle is a victory."*

Helen Keller, 1880-1968.

Chapter 5 Conclusions

We conclude this dissertation with a summary of our contributions and directions for future work.

5.1 Summary

This research begins with a study of inter-switch handoff in wireless ATM networks. Path optimization may be necessary if the end-to-end path after connection rerouting is not optimal.

- In Chapter 2, we proposed and analyzed three different path optimization schemes which are simple to implement. In the *exponential path optimization scheme*, the time between path optimizations is modeled as an exponentially distributed random variable. The *periodic path optimization scheme* invokes path optimization at periodic time intervals. In the *Bernoulli scheme*, path optimization is performed with a fixed probability after each path extension. A discrete time analytical model and a discrete event simulation model were proposed to compare the performance of these schemes by evaluating the expected total cost during a call. The analytical model enables a closed-form expression and optimal operating point to be obtained for each path optimization scheme. The analytical and simulation results agree with each other, corroborating the two models. Results indicate that the

Bernoulli scheme outperforms the other two schemes by providing a lower expected cost per call and a lower percentage change of expected cost relative to the variation of the average time between inter-switch handoffs.

- In Chapter 3, we addressed the issue of when to initiate path optimization by proposing a stochastic model. The path optimization problem is formulated as a semi-Markov decision process. Based on current state information, the network decides whether to perform path optimization after path extension. The time between inter-switch handoff follows a general distribution and can depend on the location of the anchor switch. We presented the value iteration algorithm which solves the optimality equations. A stationary deterministic optimal policy is obtained. Under certain conditions, the optimal policy has a threshold structure. That is, path optimization is always performed when the number of links of the path is greater than a certain threshold. The threshold structure of the optimal policy facilitates the implementation. When inter-switch handoff occurs, the decision of performing path optimization can be made by a simple table lookup. The performance of the optimal policy was compared with four heuristics, namely, “always perform path optimization”, “never perform path optimization”, “periodic”, and “Bernoulli” path optimization. Simulation results indicate the optimal policy gives a lower expected cost per call than the other four heuristics. These results imply that by using the optimal policy, the mobile connection maintains a good balance between the network resources utilized and the signaling load incurred on the network during its connection lifetime. We also performed the sensitivity analysis for the optimal policy with respect to the variation of the average call duration and the average time between inter-switch handoffs. Results indicate that the optimal policy is relatively insensitive to the change of the average

time between inter-switch handoffs. If there is uncertainty in estimating the average call duration, it may be better to over-estimate the value in order to reduce the cost ratio difference.

Another part of this thesis focused on location update in wireless cellular networks. Although it has been shown that the distance-based update algorithm has a better performance than the LA, movement, and timer based update schemes, the determination of the optimal distance threshold is often under certain unrealistic assumptions.

- In Chapter 4, we proposed a stochastic model to analyze the distance-based location update algorithm. The location tracking problem is formulated as a semi-Markov decision process. Based on the current state information, the mobile terminal decides whether to update its location whenever it crosses a cell boundary. The Markovian movement pattern allows the study of a variety of mobility models. For example, if the state information includes the identifiers of the recently visited cells, then a history-based mobility model for a particular user can be obtained. The cell residence time follows a general distribution and can depend on the location of the cell. The usual i.i.d. assumption for the cell residence time distribution can be relaxed. For the distance-based algorithm, we showed that the optimal policy has a threshold structure for nondecreasing paging cost functions. We proposed an efficient scheme to implement the distance-based algorithm in an arbitrary cell topology. Each mobile terminal only has to maintain the information of the identifiers of the current cell and the cell that performed the last update. Whenever the mobile terminal crosses a cell boundary, the decision of location update is made by a simple table lookup. We presented numerical results for the distance-based update algorithm in both hexagonal and random graph topologies. Results indicate that the distance threshold (or update boundary)

computed from our proposed model has a better performance (in terms of a lower cost) than the distance threshold determined from the hexagonal cell configuration with random walk movement pattern. These results imply that by using the optimal policy, the network can maintain a good balance between the processing incurred on location update and the radio bandwidth utilized for paging between call arrivals.

5.2 Further work

In the course of the investigations reported in this thesis, a number of interesting problems have been discovered which merit further research.

- **Multicast Handoff:** The support of multicast handoff is similar to the support of dynamic multicasting in which users can leave or join the multicast tree. If the target switch is part of the multicast tree, the handoff function can be invoked similar to the *dynamic joint function* in multicasting. If the target switch is not part of the multicast tree, path rerouting or path extension schemes can be still used. If path extension scheme is used [62], the crossover switch algorithm requires modification. Termination of the old subpath is required only if there is no group users connected to the old subpath within the multicast tree. Although path extension scheme seems to be easier to implement, issues like the increase of the path length and the formation of loops still need to be addressed. Path optimization within a multicast tree may not be a simple procedure. Further work is required in this area.
- **State-based models:** The proposed model in Chapter 4 can be used to analyze the movement-based and timer-based location update algorithms, and to study the effects of correlations between the cell residence times of adjacent cells. Our

proposed model can also be extended to study the location tracking algorithms in a wireless two-way messaging system [78], in which there is an additional holding cost to account for delaying the delivery of a message. Most of the models proposed in the literature (including the one we proposed in Chapter 4) assume the call arrival rate for a particular mobile user follows a Poisson distribution. Results in [31] show that it may not be the case. This points to the need for new analytical models for location tracking under general call arrival distributions.

- **User Profile:** Several location update and paging strategies improve the network performance by predicting the user's location based on his velocity, probability distribution, etc. This information has to be stored in the user profile. An efficient way to collect, store, update, and disseminate the user profile information is crucial.
- **Heterogeneous Networks:** As part of the IMT-2000 system, the mobile terminal will be able to communicate with several networks at the same time. Rather than performing location update with different networks, it is desirable if the mobile terminal only needs to report its location to a single network. This network will then disseminate the location information to other networks. An efficient way to disseminate and retrieve the information remains an open issue.

Bibliography

- [1] A. S. Acampora and M. Naghshineh, "An Architecture and Methodology for Mobile-Executed Hand-off in Cellular ATM Networks," *IEEE J. Select. Areas Commun.* vol. 12, no. 8, pp. 1365-1375, Oct. 1994.
- [2] A. Acharya, J. Li, and D. Raychaudhuri, "Primitives for Location Management and Handoff in Mobile ATM Networks," *ATM Forum Contribution 96-1121*, Aug. 1996.
- [3] A. Acharya, J. Li, B. Rajagopalan, and D. Raychaudhuri, "Mobility Management in Wireless ATM Networks," *IEEE Commun. Mag.*, vol. 35, no. 11, pp. 100-109, Nov. 1997.
- [4] I. F. Akyildiz, J. Ho, and Y.-B. Lin, "Movement-based Location Update and Selective Paging for PCS networks," *IEEE/ACM Trans. Networking*, vol. 4, no. 4, pp. 629-638, Aug. 1996.
- [5] I. F. Akyildiz, J. McNair, J. Ho, H. Uzunalioglu, and W. Wang, "Mobility Management in Next-Generation Wireless Systems," *Proc. of the IEEE*, vol. 87, no. 8, pp. 1347-1384, Aug. 1999.
- [6] B. A. Akyol and D. C. Cox, "Signaling Alternatives in a Wireless ATM Network," *IEEE J. Select. Areas in Commun.*, vol. 15, no. 1, pp. 35-49, Jan. 1997.
- [7] M. Asawa and W. E. Stark, "Optimal Scheduling of Handoffs in Cellular Networks," *IEEE/ACM Trans. on Networking*, vol. 4, no. 3, pp. 428-441, June 1996.
- [8] The ATM Forum Wireless ATM Working Group, "Wireless ATM Capability Set 1 Specification - Draft," *BTD-WATM-01.13*, Jan. 2000.

- [9] The ATM Forum Technical Committee, "ATM User-Networks Interface (UNI) Specification Version 3.1, Sept. 1994.
- [10] The ATM Forum Technical Committee, "Private Network-to-Network Interface Specification Version 1.0 (PNNI 1.0)," *af-pnni-0055.000*, March 1996.
- [11] B. A. J. Banh, G. J. Anido, and E. Dutkiewicz, "Handover Re-routing Schemes for Connection Oriented Services in Mobile ATM Networks," in *Proc. IEEE INFOCOM'98*, San Francisco, CA, pp. 1139-1146, March/April 1998.
- [12] A. Bar-Noy, I. Kessler, and M. Sidi, "Mobile Users: To Update or Not to Update?" *ACM/Baltzer J. Wireless Networks*, vol. 1, no. 2, pp. 175-195, July 1995.
- [13] D. Bertsekas and R. Gallager, *Data Networks*, 2nd edition, Prentice Hall, 1992.
- [14] R. R. Bhat and R. Gupta, "Framework for Dynamic COS Discovery in Wireless ATM," *ATM Forum Contribution 98-0005*, Feb. 1998.
- [15] R. R. Bhat, "Extension to Backward COS Discovery (BCD) Approach," *ATM Forum Contribution 98-0231*, April 1998.
- [16] A. Bhattacharya and S. K. Das, "LeZi-Update: An Information-Theoretic Approach to Track Mobile Users in PCS networks," in *Proc. of ACM/IEEE MobiCom'99*, Seattle, WA, pp. 1-12, Aug. 1999.
- [17] G. Bianchi, A. T. Campbell, and R. Liao, "On Utility-Fair Adaptive Services in Wireless Networks," in *Proc. IEEE/IFIP IWQoS'98*, Napa Valley, CA, May 1998.
- [18] Common ETSI - ATM Forum reference model for Wireless ATM Access Systems (WACS), BRAN-DTR-BRAN-040001 V0.3.2, July 1998.
- [19] M. B. Doar, "Better Model for Generating Test Networks," in *Proc. IEEE GLOBECOM'96*, London, UK, pp. 86-93, 1996.

- [20] G. Dommety, M. Veeraraghavan, and M. Singhal, "Route Optimization in Mobile ATM Networks," *ACM/Baltzer Mobile Networks and Applications*, vol. 3, no. 2, pp. 203-220, Aug. 1998.
- [21] G. Dommety, M. Veeraraghavan, and M. Singhal, "A Route Optimization Algorithm and Its Application to Mobile Location Management in ATM Networks," *IEEE J. Select. Areas Commun.*, vol. 16, no. 6, pp. 890-908, Aug. 1998.
- [22] EIA/TIA IS-41.3 (Revision B), *Cellular Radio-Telecommunications Intersystem Operations*, July 1991.
- [23] Y. Fang, I. Chlamtac, and Y.-B. Lin, "Modeling PCS Networks Under General Call Holding Time and Cell Residence Time Distributions," *IEEE/ACM Trans. Networking*, vol. 5, no. 6, pp. 893-906, Dec. 1997.
- [24] A. Farago and I. Chlamtac, "Virtual Path Network Topology Optimization Using Random Graphs," in *Proc. IEEE INFOCOM'99*, New York, N.Y., March 1999.
- [25] R. Ghai and S. Singh, "An Architecture and Communication Protocol for Picocellular Networks," *IEEE Personal Communications*, 3rd quarter, pp. 36-46, 1994.
- [26] R. Guerin and A. Orda, "QoS Routing in Networks with Inaccurate Information: Theory and Algorithms," *IEEE/ACM Trans. Networking*, vol. 7, no. 3, pp. 350-364, June 1999.
- [27] J. Ho and I. F. Akyildiz, "Mobile User Location Update and Paging under Delay Constraints," *ACM/Baltzer J. Wireless Networks*, vol. 1, no. 4, pp. 413-425, Dec. 1995.
- [28] *IEEE J. Select. Areas Commun.*, special issue on Wireless ATM, vol. 15, no. 1, Jan. 1997.
- [29] *IEEE Communications Magazine*, special issue on Introduction to Mobile and Wireless ATM, vol. 35, no. 11, Nov. 1997.

- [30] *IEEE Personal Communications*, special issue on IMT-2000: Standards Efforts of the ITU, vol. 4, no. 4, Aug. 1997.
- [31] J. Jannink, D. Lam, N. Shivakumar, J. Widom, And D. Cox, "Efficient and Flexible Location Management Techniques for Wireless Communication Systems," *ACM/Baltzer J. Wireless Networks*, vol. 3, pp. 361-374, Oct. 1997.
- [32] D. Lam, D. Cox, and J. Widom, "Teletraffic Modeling for Personal Communications Services," *IEEE Commun. Mag.*, vol. 35, no. 2, pp. 79-87, Feb. 1997.
- [33] J. Li and A. Acharya, "Handoff Control for Point to Multipoint Connections in Mobile ATM Networks," in *Proc. IEEE Globecom'98*, Sydney, Australia, November 1998.
- [34] J. Li, R. Yates, and D. Raychaudhuri, "Performance Analysis on Path Rerouting Algorithms for Handoff Control in Mobile ATM Networks," in *Proc. IEEE INFOCOM'99*, New York, NY, March 1999.
- [35] B. Liang and Z. Haas, "Predictive Distance-based Mobility Management for PCS Networks," in *Proc. IEEE INFOCOM'99*, New York, NY, March 1999.
- [36] T. Liu, P. Bahl, and I. Chlamtac, "Mobility Modeling, Location Tracking, and Trajectory Prediction in Wireless ATM Networks," *IEEE J. Select. Areas Commun.*, vol. 16, no. 6, pp. 922-936, August 1998.
- [37] D. H. Lorenz and A. Orda, "QoS Routing in Networks with Uncertain Parameters," *IEEE/ACM Trans. Networking*, vol. 6, no. 6, pp. 768-778, Dec. 1998.
- [38] U. Madhow, M. Honig, and K. Steiglitz, "Optimization of Wireless Resources for Personal Communications Mobility Tracking," *IEEE/ACM Trans. Networking*, vol. 3, no. 4, pp. 698-707, Dec. 1995.
- [39] The MAGIC WAND homepage: <http://www.tik.ee.ethz.ch/~wand>

- [40] H. Mitts, H. Hansen, J. Immonen, and S. Veikkolainen, "Lossless Handover for Wireless ATM," *ACM/Baltzer Mobile Networks and Applications*, vol. 1, no. 3, pp. 299-312, Dec. 1996.
- [41] M. Mouly and M.-B. Pautet, *The GSM System for Mobile Communications*, Palaiseau, France, 1992.
- [42] Z. Naor and H. Levy, "Minimizing the Wireless Cost of Tracking Mobile Users: An Adaptive Threshold Scheme," in *Proc. IEEE INFOCOM'98*, San Francisco, CA, pp. 720-727, March/April 1998.
- [43] Z. Naor and H. Levy, "Cell Identification Codes for Tracking Mobile Users," in *Proc. IEEE INFOCOM'99*, New York, NY, March 1999.
- [44] C. E. Perkins and D. B. Johnson, "Mobility Support in IPv6," *draft-ietf-mobileip-ipv6-09.txt*, work in progress, Oct. 1999.
- [45] G. P. Pollini and C.-L. I, "A Profile-Based Location Strategy and Its Performance," *IEEE J. Select. Areas Commun.*, vol. 15, no. 8, pp. 1415-1424, Oct. 1997.
- [46] M. L. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, Inc., 1994.
- [47] R. Ramjee, T. F. La Porta, J. Kurose, and D. Towsley, "Performance Evaluation of Connection Rerouting Schemes for ATM-Based Wireless Networks," *IEEE/ACM Trans. Networking*, vol. 6, no. 3, pp. 249-261, June 1998.
- [48] S. Rao, B. Gopinath, and D. Kurshan, "Optimizing Call Management of Mobile Units," in *Proc. IEEE PIMRC'92*, Boston, MA, pp. 225-229, 1992.
- [49] K. Rauhala, "Requirements for Mobile ATM Handover," *ATM Forum Contribution 96-989*, August 1996.

- [50] D. Raychaudhuri and N. D. Wilson, "ATM-Based Transport Architecture for Multiservices Wireless Personal Communication Networks," *IEEE J. Select. Areas Commun.*, vol. 12, no. 8, pp. 1401-1414, Oct. 1994.
- [51] D. Raychaudhuri, L. J. French, R. J. Siracusa, S. K. Biswas, R. Yuan, P. Narasimhan, and C. A. Johnson, "WATMnet: A Prototype Wireless ATM System for Multimedia Personal Communication," *IEEE J. Select. Areas Commun.*, vol. 15, no. 1, pp. 83-95, Jan. 1997.
- [52] C. Rose and R. Yates, "Minimizing the Average Cost of Paging under Delay Constraints," *ACM/Baltzer J. Wireless Networks*, vol. 1, no. 2, pp. 211-219, July 1995.
- [53] C. Rose, "Minimizing the Average Cost of Paging and Registration: A Timer-based Method," *ACM/Baltzer J. Wireless Networks*, vol. 2, no. 2, pp. 109-116, June 1996.
- [54] C. Rose and R. Yates, "Location Uncertainty in Mobile Networks: A Theoretical Framework," *IEEE Commun. Mag.*, vol. 35, no. 2, pp. 94-101, Feb. 1997.
- [55] C. Rose, "State-based Paging/Registration: A Greedy Technique," *IEEE Trans. Vehicular Technology*, vol. 48, no. 1, pp. 166-173, Jan. 1999.
- [56] S. M. Ross, *Introduction to Stochastic Dynamic Programming*. Academic Press, New York, 1983.
- [57] R. Sanchez, J. Evans, G. Minden, V. Frost, and K. Shanmugan, "RDRN: A Prototype for a Rapidly Deployable Radio Network," *ACM Computing and Commun. Review*, vol. 2, no. 2, pp. 15-22.
- [58] S. K. Sen, A. Bhattacharya, and S. K. Das, "A Selective Location Update Strategy for PCS Users," *ACM/Baltzer J. Wireless Networks*, vol. 5, no. 5, pp. 313-326, Sept. 1999.
- [59] S. S. Stidham, Jr., "Socially and Individually Optimal Control of Arrivals to a GI/M/1 Queue," *Man. Sci.* vol. 24, pp. 1598-1610, 1978.

- [60] S. Tabbane, "An Alternative Strategy for Location Tracking," *IEEE J. Select. Areas Commun.*, vol. 13, no. 5, pp. 880-892, June 1995.
- [61] C.-K. Toh, "Crossover Switch Discovery for Wireless ATM LANs," *ACM/Baltzer Mobile Networks and Applications*, vol. 1, no. 2 pp. 141-165, Oct. 1996.
- [62] C.-K. Toh, "A Unifying Methodology for Handover of Heterogeneous Connections in Wireless ATM Networks," *ACM SIGCOMM Computer Communications Review*, vol. 27, no. 1, pp. 12-30, Jan. 1997.
- [63] C.-K. Toh, *Wireless ATM and Ad-Hoc Networks*, Kluwer Academic Publishers, Boston 1997.
- [64] L. Van Hauwermeiren, L. Vercauteren, A. Saidi, and T. Van Landegem, "Requirements for Mobility Support in ATM," in *Proc. IEEE GLOBECOM'94*, San Francisco, CA, pp. 1691-1695, 1994.
- [65] M. Veeraraghavan, M. Karol, and K. Eng, "A Combined Handoff Scheme for Mobile ATM Networks," *ATM Forum Contribution 96-1700*, Dec. 1996.
- [66] M. Veeraraghavan, M. Karol, and K. Eng, "Mobility and Connection Management in a Wireless ATM LAN," *IEEE J. Select. Areas Commun.*, vol. 15, no. 1, pp. 50-68, Jan. 1997.
- [67] M. Veeraraghavan and G. Dommety, "Mobile Location Management in ATM Networks," *IEEE J. Select. Areas Commun.*, vol. 15, no. 8, pp. 1437-1454, Oct. 1997.
- [68] G. Wan and E. Lin, "A Dynamic Paging Scheme for Wireless Communication Systems," in *Proc. ACM/IEEE MobiCom'97*, Budapest, Hungary, pp. 195-203, 1997.
- [69] Z. Wang and J. Crowcroft, "Quality-of Service Routing for Supporting Multimedia Applications," *IEEE J. Select. Areas Commun.*, vol. 14, no. 7, pp. 1228-1234, Sept. 1996.

- [70] W. S. V. Wong and V. Leung, "A Path Optimization Signaling Protocol for Inter-Switch Handoff in Wireless ATM Networks," in *Proc. of the 1st International Workshop on Wireless Mobile ATM Implementation (wmATM)*, Hangzhou, China, pp. 92-98, April 1998.
- [71] W. S. V. Wong and V. C. M. Leung, "A Path Optimization Signaling Protocol for Inter-Switch Handoff in Wireless ATM Networks," *Computer Networks*, vol. 31, no. 9-10, pp. 975-984, May 1999.
- [72] W. S. V. Wong, H. Chan, and V. C. M. Leung, "Path Optimization for Inter-Switch Handoff in Wireless ATM Networks," in *Proc. IEEE International Conf. on Universal Personal Communications (ICUPC)*, Florence, Italy, pp. 615-619, Oct. 1998.
- [73] W. S. V. Wong, H. Chan, and V. C. M. Leung, "Performance Evaluations of Path Optimization Schemes for Inter-Switch Handoffs in Wireless ATM Networks," in *Proc. ACM/IEEE MobiCom'98*, Dallas, Texas, pp. 242-251, Oct. 1998.
- [74] W. S. V. Wong, H. Chan, and V. C. M. Leung, "A Framework for Analyzing Path Optimization Schemes for Inter-Switch Handoff in Wireless ATM Networks," in *Proc. IEEE ICC'99*, Vancouver, Canada, June 1999.
- [75] W. S. V. Wong, H. Chan, and V. C. M. Leung, "Performance Evaluations of Path Optimization Schemes for Inter-Switch Handoffs in Wireless ATM Networks," accepted for publication in *ACM/Baltzer Wireless Networks*.
- [76] W. S. V. Wong, M. E. Lewis, and V. C. M. Leung, "Stochastic Control of Path Optimization for Inter-Switch Handoffs in Wireless ATM Networks," submitted to *IEEE/ACM Trans. on Networking*.
- [77] W. S. V. Wong and V. C. M. Leung, "A Framework for Analyzing Location Update Algorithms in Wireless Cellular Networks," submitted.

- [78] T. Woo, T. La Porta, J. Golestani, and A. Agarwal, "Update and Search Algorithms for Wireless Two-Way Messaging: Design and Performance," in *Proc. IEEE INFOCOM*, San Francisco, CA, March/April 1998.
- [79] R. Yuan, S. K. Biswas, and D. Raychaudhuri, "A Signaling and Control Architecture for Mobility Support in Wireless ATM Networks," in *Proc. IEEE ICC'96*, Dallas, Texas, pp. 478-484, 1996.
- [80] E. W. Zegura, K. L. Calvert, and M. J. Donahoo, "A Quantitative Comparison of Graph-Based Models for Internet Topology," *IEEE/ACM Trans. Networking*, vol. 5, no. 6, pp. 770-783, Dec. 1997.
- [81] M. M. Zonoozi and P. Dassanayake, "User Mobility Modeling and Characterization of Mobility Patterns," *IEEE J. Select. Areas Commun.*, vol. 15, no. 7, pp. 1239-1252, Sept. 1997.

Appendix A. Expected Cost for the Exponential Path Optimization Scheme

In this appendix, we derive the expected cost per call for the exponential path optimization scheme. The notations used follow the analytical model described in Chapter 2. Referring to Figure A.1, the following notations are defined to facilitate the derivation:

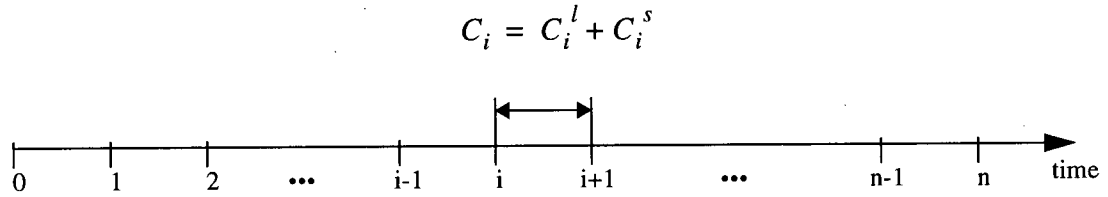


Figure A.1 Timing diagram

- n denote the call termination time.
- C_i denote the total cost during time interval $[i, i+1)^1$.
- C_i^l denote the link cost during time interval $[i, i+1)$.
- C_i^s denote the signaling cost at time i .

The cost functions C_i , C_i^l , and C_i^s are related by the following equation:

$$C_i = C_i^l + C_i^s. \quad (\text{A.1})$$

Two indicator functions are defined:

$$I_i = \begin{cases} 1, & \text{if path extension occurs at time } i, \\ 0, & \text{otherwise.} \end{cases} \quad (\text{A.2})$$

¹ $[a, b)$ denote the interval between a and b that includes a but not b .

$$\tilde{I}_i = \begin{cases} 1, & \text{if path optimization occurs at time } i, \\ 0, & \text{otherwise.} \end{cases} \quad (\text{A.3})$$

Since at each time interval, path extension and path optimization are modeled as Bernoulli processes with probabilities λ and ν , respectively, we have:

$$E[I_i] = \lambda \quad \text{and} \quad E[\tilde{I}_i] = \nu. \quad (\text{A.4})$$

For the initial conditions, i.e., at $i = 0$, we assume:

$$C_0^s = 0 \quad \text{and} \quad C_0^l = LC_{link}. \quad (\text{A.5})$$

Their expected values are:

$$E[C_0^s] = 0 \quad \text{and} \quad E[C_0^l] = LC_{link}. \quad (\text{A.6})$$

For the signaling cost due to mobility at time i :

$$C_i^s = I_i C_{PE} + \tilde{I}_i C_{PO} \quad i \geq 1. \quad (\text{A.7})$$

Taking the expectation:

$$E[C_i^s] = \lambda C_{PE} + \nu C_{PO} \quad i \geq 1. \quad (\text{A.8})$$

For the link cost between time interval $[i, i+1)$:

$$C_i^l = (C_{i-1}^l + I_i H C_{link})(1 - \tilde{I}_i) + LC_{link} \tilde{I}_i \quad i \geq 1. \quad (\text{A.9})$$

Taking the expectation, we have:

$$E[C_i^l] = (1 - v)E[C_{i-1}^l] + [(1 - v)\lambda\bar{H} + v\bar{L}]C_{link}. \quad (A.10)$$

For $i \geq 1$, it can be proved by mathematical induction that:

$$E[C_i^l] = (1 - v)^i \bar{L}C_{link} + [1 + (1 - v) + \dots + (1 - v)^{i-1}]M \quad (A.11)$$

where $M = (1 - v)\lambda\bar{H}C_{link} + v\bar{L}C_{link}$.

Equation (A.11) can also be written as:

$$E[C_i^l] = (1 - v)^i \bar{L}C_{link} + \frac{1 - (1 - v)^i}{v} M \quad i \geq 0. \quad (A.12)$$

The expected cost of the call when using the exponential path optimization scheme is:

$$\begin{aligned} E[C_{exponential}] &= E\left\{E\left[\sum_{i=0}^{n-1} C_i \middle| n\right]\right\} \\ &= \sum_{n=1}^{\infty} E\left[\sum_{i=0}^{n-1} C_i \middle| N = n\right] P(N = n) \\ &= \sum_{n=1}^{\infty} E\left[\sum_{i=0}^{n-1} (C_i^l + C_i^s) \middle| N = n\right] P(N = n) \end{aligned} \quad (A.13)$$

The expected signaling cost of the call:

$$\begin{aligned}
& \sum_{n=1}^{\infty} E \left[\sum_{i=0}^{n-1} (C_i^s | N=n) \right] P(N=n) \\
&= \sum_{n=2}^{\infty} \sum_{i=1}^{n-1} (\lambda C_{PE} + \nu C_{PO}) P(N=n) \\
&= (\lambda C_{PE} + \nu C_{PO}) \sum_{n=2}^{\infty} (n-1)(1-\mu)^{n-1} \mu \\
&= \frac{(1-\mu)}{\mu} (\lambda C_{PE} + \nu C_{PO}).
\end{aligned} \tag{A.14}$$

The expected link cost of the call:

$$\begin{aligned}
& \sum_{n=1}^{\infty} E \left[\sum_{i=0}^{n-1} (C_i^l | N=n) \right] P(N=n) \\
&= \sum_{n=1}^{\infty} \left\{ \sum_{i=0}^{n-1} \left[(1-\nu)^i \bar{L} C_{link} + \frac{1-(1-\nu)^i}{\nu} M \right] \right\} P(N=n) \\
&= \sum_{n=1}^{\infty} \left\{ \frac{1-(1-\nu)^n}{\nu} \bar{L} C_{link} + \left[\frac{n}{\nu} - \frac{1-(1-\nu)^n}{\nu^2} \right] M \right\} P(N=n) \\
&= \frac{\bar{L}}{\mu} C_{link} + \frac{\lambda(1-\mu)(1-\nu)\bar{H}}{\mu[1-(1-\mu)(1-\nu)]} C_{link}.
\end{aligned} \tag{A.15}$$

Substituting equations (A.14) and (A.15) into equation (A.13), we have:

$$\begin{aligned}
E[C_{exponential}] &= \frac{\bar{L}}{\mu} C_{link} + \frac{\lambda(1-\mu)(1-\nu)\bar{H}}{\mu[1-(1-\mu)(1-\nu)]} C_{link} \\
&\quad + \frac{(1-\mu)}{\mu} (\lambda C_{PE} + \nu C_{PO}).
\end{aligned} \tag{A.16}$$

Appendix B. Expected Cost for the Periodic Path Optimization Scheme

In this appendix, we derive the expected cost per call for the periodic path optimization scheme. The notations and definitions used follow those defined in Chapter 2 and Appendix A. In addition, we let k denote the period to invoke path optimization. The value of k is a positive integer and is measured in time interval as defined in Chapter 2. An indicator function, which models the time to perform periodic path optimization, is defined as:

$$\widehat{I}_i = \begin{cases} 1, & \text{if } i \bmod k \equiv 0, \\ 0, & \text{otherwise.} \end{cases} \quad (\text{B.1})$$

Thus, \widehat{I}_i is equal to 1 whenever time i is a multiple of k .

For the signaling cost due to mobility at time i :

$$C_i^s = I_i C_{PE} + \widehat{I}_i C_{PO} \quad i \geq 1. \quad (\text{B.2})$$

Taking the expectation:

$$E[C_i^s] = \lambda C_{PE} + \widehat{I}_i C_{PO} \quad i \geq 1. \quad (\text{B.3})$$

For the link cost between time interval $[i, i+1)$:

$$C_i^l = (C_{i-1}^l + I_i H C_{link})(1 - \widehat{I}_i) + \widehat{I}_i L C_{link}. \quad (\text{B.4})$$

Taking the expectation:

$$E[C_i^l] = (1 - \widehat{I}_i)E[C_{i-1}^l] + [(1 - \widehat{I}_i)\lambda \bar{H} + \widehat{I}_i \bar{L}]C_{link}. \quad (\text{B.5})$$

Since $E[C_0^l] = \bar{L}C_{link}$, equation (B.5) can also be written as:

$$E[C_i^l] = \bar{L}C_{link} + \lambda \bar{H}C_{link}(i \bmod k) \quad i \geq 0. \quad (B.6)$$

The expected cost of the call is:

$$E[C_{periodic}] = \sum_{n=1}^{\infty} E\left[\sum_{i=0}^{n-1} (C_i^l + C_i^s) \middle| N=n\right] P(N=n). \quad (B.7)$$

The expected signaling cost of the call:

$$\begin{aligned} & \sum_{n=1}^{\infty} E\left[\sum_{i=0}^{n-1} (C_i^s) \middle| N=n\right] P(N=n) \\ &= \sum_{n=2}^{\infty} \sum_{i=0}^{n-1} (\lambda C_{PE} + \widehat{I}_i C_{PO}) P(N=n) \\ &= \lambda C_{PE} \sum_{n=2}^{\infty} [nP(N=n)] + C_{PO} \sum_{n=2}^{\infty} \sum_{i=0}^{n-1} \widehat{I}_i P(N=n) \\ &= \lambda C_{PE} \sum_{n=2}^{\infty} [n(1-\mu)^{n-1}\mu] + C_{PO} \sum_{j=1}^{\infty} \sum_{m=1}^k jP(N=jk+m) \\ &= \frac{\lambda(1-\mu)}{\mu} C_{PE} + C_{PO} \sum_{j=1}^{\infty} \sum_{m=1}^k j(1-\mu)^{jk+m-1}\mu \\ &= \frac{\lambda(1-\mu)}{\mu} C_{PE} + \frac{(1-\mu)^k}{1-(1-\mu)^k} C_{PO}. \end{aligned} \quad (B.8)$$

The expected link cost of the call:

$$\begin{aligned}
& \sum_{n=1}^{\infty} E \left[\sum_{i=0}^{n-1} (C_i^l | N=n) \right] P(N=n) \\
&= \sum_{n=1}^{\infty} \sum_{i=0}^{n-1} [\bar{L}C_{link} + \lambda \bar{H}C_{link}(i \bmod k)] P(N=n) \\
&= \bar{L}C_{link} \sum_{n=1}^{\infty} [nP(N=n)] + \lambda \bar{H}C_{link} \sum_{m=1}^{k-1} \left[m \sum_{i=1}^{\infty} \sum_{j=ki-(k-1-m)}^{\infty} P(N=j) \right] \quad (B.9) \\
&= \bar{L}C_{link} \sum_{n=1}^{\infty} [n(1-\mu)^{n-1}\mu] + \lambda \bar{H}C_{link} \sum_{m=1}^{k-1} \left[m \sum_{i=1}^{\infty} \sum_{j=ki-(k-1-m)}^{\infty} (1-\mu)^{j-1}\mu \right] \\
&= \frac{\bar{L}}{\mu} C_{link} + \frac{\lambda \bar{H}C_{link}}{1-(1-\mu)^k} \sum_{m=1}^{k-1} m(1-\mu)^m \\
&= \frac{\bar{L}}{\mu} C_{link} + \frac{\lambda(1-\mu)\bar{H}C_{link}}{\mu^2[1-(1-\mu)^k]} [1-k(1-\mu)^{k-1} + (k-1)(1-\mu)^k].
\end{aligned}$$

Substituting equations (B.8) and (B.9) into equation (B.7), we have:

$$\begin{aligned}
E[C_{periodic}] &= \frac{\bar{L}}{\mu} C_{link} + \frac{\lambda(1-\mu)\bar{H}C_{link}}{\mu^2[1-(1-\mu)^k]} [1-k(1-\mu)^{k-1} + (k-1)(1-\mu)^k] \\
&\quad + \frac{\lambda(1-\mu)}{\mu} C_{PE} + \frac{(1-\mu)^k}{1-(1-\mu)^k} C_{PO}. \quad (B.10)
\end{aligned}$$

Appendix C. Expected Cost for the Bernoulli Path Optimization Scheme

In this appendix, we derive the expected cost per call for the Bernoulli path optimization scheme. Following the notations and definitions used in Chapter 2 and Appendix A, we also define an indicator random variable:

$$\hat{I}_i = \begin{cases} 1, & \text{perform path optimization after path extension at time } i, \\ 0, & \text{otherwise.} \end{cases} \quad (\text{C.1})$$

Since path optimization is invoked with probability p after each path extension, we have:

$$E[\hat{I}_i] = p. \quad (\text{C.2})$$

Note that in our analytical model, path extension and path optimization are modeled as point processes. Thus, the time required to perform these operations is not taken into account.

For the signaling cost due to mobility at time i :

$$C_i^s = I_i(C_{PE} + \hat{I}_i C_{PO}) \quad i \geq 1. \quad (\text{C.3})$$

Taking the expectation:

$$E[C_i^s] = \lambda(C_{PE} + pC_{PO}) \quad i \geq 1. \quad (\text{C.4})$$

For the link cost between time interval $[i, i+1)$:

$$C_i^l = C_{i-1}^l(1 - I_i) + (C_{i-1}^l + HC_{link})I_i(1 - \hat{I}_i) + LC_{link}I_i\hat{I}_i. \quad (\text{C.5})$$

Taking the expectation:

$$E[C_i^l] = (1 - \lambda p)E[C_{i-1}^l] + [\lambda(1 - p)\bar{H} + \lambda p\bar{L}]C_{link} \quad (\text{C.6})$$

Since $E[C_0^l] = \bar{L}C_{link}$, it can be proved by mathematical induction that for $i \geq 1$:

$$\begin{aligned} E[C_i^l] &= (1 - \lambda p)^i \bar{L}C_{link} + [1 + (1 - \lambda p) + \dots + (1 - \lambda p)^{i-1}]K \\ &= (1 - \lambda p)^i \bar{L}C_{link} + \frac{1 - (1 - \lambda p)^i}{\lambda p} K \end{aligned} \quad (C.7)$$

where $K = [\lambda(1 - p)\bar{H} + \lambda p\bar{L}]C_{link}$.

The expected cost of the call when using the Bernoulli path optimization scheme is:

$$E[C_{Bernoulli}] = \sum_{n=1}^{\infty} E\left[\sum_{i=0}^{n-1} (C_i^l + C_i^s) \middle| N = n\right] P(N = n). \quad (C.8)$$

The expected signaling cost of the call:

$$\begin{aligned} &\sum_{n=1}^{\infty} E\left[\sum_{i=0}^{n-1} (C_i^s) \middle| N = n\right] P(N = n) \\ &= \sum_{n=2}^{\infty} \sum_{i=1}^{n-1} \lambda(C_{PE} + pC_{PO}) P(N = n) \\ &= \lambda(C_{PE} + pC_{PO}) \sum_{n=2}^{\infty} (n-1)(1-\mu)^{n-1} \mu \\ &= \frac{\lambda(1-\mu)}{\mu} (C_{PE} + pC_{PO}). \end{aligned} \quad (C.9)$$

The expected link cost of the call:

$$\begin{aligned} &\sum_{n=1}^{\infty} E\left[\sum_{i=0}^{n-1} (C_i^l) \middle| N = n\right] P(N = n) \\ &= \sum_{n=1}^{\infty} \left\{ \sum_{i=0}^{n-1} \left[(1 - \lambda p)^i \bar{L}C_{link} + \frac{1 - (1 - \lambda p)^i}{\lambda p} K \right] \right\} P(N = n) \\ &= \frac{\bar{L}}{\mu} C_{link} + \frac{\lambda(1-p)(1-\mu)\bar{H}}{\mu[1 - (1 - \lambda p)(1 - \mu)]} C_{link}. \end{aligned} \quad (C.10)$$

Substituting equations (C.9) and (C.10) into equation (C.8), we have:

$$\begin{aligned}
 E[C_{Bernoulli}] = & \frac{\bar{L}}{\mu} C_{link} + \frac{\lambda(1-p)(1-\mu)\bar{H}C_{link}}{\mu[1-(1-\lambda p)(1-\mu)]} \\
 & + \frac{\lambda(1-\mu)}{\mu} (C_{PE} + pC_{PO}).
 \end{aligned} \tag{C.11}$$

"Another roof, another proof."

Paul Erdos, 1913-1996.

Appendix D. Semi-Markov Decision Processes

Proposition D.1: Assume the cost, transition probabilities, and sojourn times are time homogeneous. If the termination time of a finite-horizon semi-Markov decision process is exponentially distributed with mean $1/\mu$, then it is equivalent to an infinite-horizon semi-Markov decision process with discount rate μ . That is,

$$\begin{aligned} v^\pi(s) = E_s^\pi \left\{ \sum_{n=0}^{\phi(T)} b(X_n, Y_n) + \sum_{n=0}^{\phi(T)-1} \left[\int_{\sigma_n}^{\sigma_{n+1}} f(W_\tau, X_n, Y_n) d\tau \right] \right. \\ \left. + \int_{\sigma_{\phi(T)}}^T f(W_\tau, X_{l(T)}, Y_{l(T)}) d\tau + h(X_{\phi(T)}) \right\} \end{aligned} \quad (D.1)$$

is equivalent to

$$v^\pi(s) = E_s^\pi \left\{ \sum_{n=0}^{\infty} e^{-\mu \sigma_n} c(X_n, Y_n) \right\} \quad (D.2)$$

where

$$c(s, a) = b(s, a) + E_s^a \left\{ \int_0^{\tau_1} e^{-\mu t} f(W_t, s, a) dt + (1 - e^{-\mu \tau_1}) h(s) \right\}. \quad (D.3)$$

Proof: For clarity, we will analyze the four terms in (D.1) separately. Let

$$v^\pi(s) = u_1^\pi(s) + u_2^\pi(s) + u_3^\pi(s) + u_4^\pi(s) \quad (D.4)$$

where

$$u_1^\pi(s) = E_s^\pi \left\{ \sum_{n=0}^{\phi(T)} b(X_n, Y_n) \right\} \quad (D.5)$$

$$u_2^\pi(s) = E_s^\pi \left\{ \sum_{n=0}^{\phi(T)-1} \left[\int_{\sigma_n}^{\sigma_{n+1}} f(W_\tau, X_n, Y_n) d\tau \right] \right\} \quad (D.6)$$

$$u_3^\pi(s) = E_s^\pi \left\{ \int_{\sigma_{\phi(T)}}^T f(W_\tau, X_{l(T)}, Y_{l(T)}) d\tau \right\} \quad (D.7)$$

$$u_4^\pi(s) = E_s^\pi \{ h(X_{\phi(T)}) \}. \quad (D.8)$$

For (D.5),

$$\begin{aligned} u_1^\pi(s) &= E_s^\pi \left\{ \sum_{n=0}^{\phi(T)} b(X_n, Y_n) \right\} \\ &= E_s^\pi \left\{ \int_0^\infty \left[\sum_{n=0}^{\phi(t)} b(X_n, Y_n) \right] \mu e^{-\mu t} dt \right\}. \end{aligned}$$

If m represents the last decision epoch before termination, then

$$u_1^\pi(s) = E_s^\pi \left\{ \sum_{m=0}^{\infty} \int_{\sigma_m}^{\sigma_{m+1}} \left[\sum_{n=0}^m b(X_n, Y_n) \right] \mu e^{-\mu t} dt \right\}.$$

By interchanging the order of summation, we have

$$u_1^\pi(s) = E_s^\pi \left\{ \sum_{n=0}^{\infty} \sum_{m=n}^{\infty} \int_{\sigma_m}^{\sigma_{m+1}} b(X_n, Y_n) \mu e^{-\mu t} dt \right\}.$$

Since $\sum_{m=n}^{\infty} \int_{\sigma_m}^{\sigma_{m+1}} (\cdot) dt = \int_{\sigma_n}^{\infty} (\cdot) dt$,

$$\begin{aligned}
u_1^\pi(s) &= E_s^\pi \left\{ \sum_{n=0}^{\infty} \int_{\sigma_n}^{\infty} b(X_n, Y_n) \mu e^{-\mu t} dt \right\} \\
&= E_s^\pi \left\{ \sum_{n=0}^{\infty} e^{-\mu \sigma_n} b(X_n, Y_n) \right\}.
\end{aligned} \tag{D.9}$$

For (D.6), we have

$$\begin{aligned}
u_2^\pi(s) &= E_s^\pi \left\{ \sum_{n=0}^{\phi(T)-1} \left[\int_{\sigma_n}^{\sigma_{n+1}} f(W_\tau, X_n, Y_n) d\tau \right] \right\} \\
&= E_s^\pi \left\{ \sum_{m=1}^{\infty} \int_{\sigma_m}^{\sigma_{m+1}} \left[\sum_{n=0}^{m-1} \left(\int_{\sigma_n}^{\sigma_{n+1}} f(W_\tau, X_n, Y_n) d\tau \right) \right] \mu e^{-\mu t} dt \right\}.
\end{aligned}$$

By interchanging the order of summation, we obtain,

$$\begin{aligned}
u_2^\pi(s) &= E_s^\pi \left\{ \sum_{n=0}^{\infty} \sum_{m=n+1}^{\infty} \int_{\sigma_m}^{\sigma_{m+1}} \left[\int_{\sigma_n}^{\sigma_{n+1}} f(W_\tau, X_n, Y_n) d\tau \right] \mu e^{-\mu t} dt \right\} \\
&= E_s^\pi \left\{ \sum_{n=0}^{\infty} \int_{\sigma_{n+1}}^{\infty} \left[\int_{\sigma_n}^{\sigma_{n+1}} f(W_\tau, X_n, Y_n) d\tau \right] \mu e^{-\mu t} dt \right\} \\
&= E_s^\pi \left\{ \sum_{n=0}^{\infty} e^{-\mu \sigma_{n+1}} \left[\int_{\sigma_n}^{\sigma_{n+1}} f(W_\tau, X_n, Y_n) d\tau \right] \right\}.
\end{aligned} \tag{D.10}$$

For (D.7),

$$\begin{aligned}
u_3^\pi(s) &= E_s^\pi \left\{ \int_{\sigma_{\phi(T)}}^T f(W_\tau, X_{\phi(T)}, Y_{\phi(T)}) d\tau \right\} \\
&= E_s^\pi \left\{ \int_0^\infty \left[\int_{\sigma_{\phi(t)}}^t f(W_\tau, X_{\phi(t)}, Y_{\phi(t)}) d\tau \right] \mu e^{-\mu t} dt \right\}.
\end{aligned}$$

If n represents the last decision epoch before termination, then

$$u_3^\pi(s) = E_s^\pi \left\{ \sum_{n=0}^{\infty} \left[\int_{\sigma_n}^{\sigma_{n+1}} \left(\int_{\sigma_n}^t f(W_\tau, X_n, Y_n) d\tau \right) \mu e^{-\mu t} dt \right] \right\}.$$

By interchanging the order of integration, we have,

$$\begin{aligned} u_3^\pi(s) &= E_s^\pi \left\{ \sum_{n=0}^{\infty} \int_{\sigma_n}^{\sigma_{n+1}} \left[\int_{\tau}^{\sigma_{n+1}} f(W_\tau, X_n, Y_n) \mu e^{-\mu t} dt \right] d\tau \right\} \\ &= E_s^\pi \left\{ \sum_{n=0}^{\infty} \int_{\sigma_n}^{\sigma_{n+1}} [e^{-\mu \tau} - e^{-\mu \sigma_{n+1}}] f(W_\tau, X_n, Y_n) d\tau \right\}. \end{aligned} \quad (D.11)$$

For (D.8)

$$\begin{aligned} u_4^\pi(s) &= E_s^\pi \{ h(X_{\phi(T)}) \} \\ &= E_s^\pi \left\{ \int_0^{\infty} h(X_{\phi(t)}) \mu e^{-\mu t} dt \right\} \\ &= E_s^\pi \left\{ \sum_{n=0}^{\infty} \left[\int_{\sigma_n}^{\sigma_{n+1}} h(X_n) \mu e^{-\mu t} dt \right] \right\} \\ &= E_s^\pi \left\{ \sum_{n=0}^{\infty} [e^{-\mu \sigma_n} - e^{-\mu \sigma_{n+1}}] h(X_n) \right\}. \end{aligned} \quad (D.12)$$

Substitute (D.9)-(D.12) into (D.4), we obtain:

$$\begin{aligned} v^\pi(s) &= E_s^\pi \left\{ \sum_{n=0}^{\infty} \left[e^{-\mu \sigma_n} b(X_n, Y_n) + \int_{\sigma_n}^{\sigma_{n+1}} e^{-\mu \tau} f(W_\tau, X_n, Y_n) d\tau \right] \right\} \\ &= E_s^\pi \left\{ \sum_{n=0}^{\infty} e^{-\mu \sigma_n} \left[b(X_n, Y_n) + \int_{\sigma_n}^{\sigma_{n+1}} e^{-\mu(\tau - \sigma_n)} f(W_\tau, X_n, Y_n) d\tau \right] \right\} \end{aligned} \quad (D.13)$$

$$\begin{aligned}
v^\pi(s) &= E_s^\pi \left\{ \sum_{n=0}^{\infty} \left[e^{-\mu\sigma_n} b(X_n, Y_n) + \int_{\sigma_n}^{\sigma_{n+1}} e^{-\mu\tau} f(W_\tau, X_n, Y_n) d\tau \right. \right. \\
&\quad \left. \left. + (e^{-\mu\sigma_n} - e^{-\mu\sigma_{n+1}}) h(X_n) \right] \right\} \\
&= E_s^\pi \left\{ \sum_{n=0}^{\infty} e^{-\mu\sigma_n} \left[b(X_n, Y_n) + \int_{\sigma_n}^{\sigma_{n+1}} e^{-\mu(\tau - \sigma_n)} f(W_\tau, X_n, Y_n) d\tau \right. \right. \\
&\quad \left. \left. + (1 - e^{-\mu\tau_{n+1}}) h(X_n) \right] \right\}.
\end{aligned} \tag{D.14}$$

Recall $c(s, a)$ denotes the expected total cost between two decision epochs, given that the system occupies state s at the first decision epoch and that the decision maker choose action a in state s . Since the cost, transition probabilities, and sojourn times are assumed to be time homogeneous,

$$c(s, a) = b(s, a) + E_s^a \left\{ \int_0^{\tau_1} e^{-\mu t} f(W_t, s, a) dt + (1 - e^{-\mu\tau_1}) h(s) \right\}. \tag{D.15}$$

Substitute (D.15) into (D.13), we have

$$v^\pi(s) = E_s^\pi \left\{ \sum_{n=0}^{\infty} e^{-\mu\sigma_n} c(X_n, Y_n) \right\}.$$

The discrete-time version of this result can be found in Chapter 5 of [46].

For stationary deterministic policy δ :

$$\begin{aligned}
v^\delta(s) &= c[s, \delta(s)] + E_s^\delta \left\{ e^{-\mu\tau_1} v^\delta(X_1) \right\} \\
&= c[s, \delta(s)] + \sum_{s' \in S} \left\{ \int_0^{\infty} e^{-\mu t} v^\delta(s') P[s' | s, \delta(s)] G[dt | s, \delta(s)] \right\} \quad \square
\end{aligned}$$

Lemma D.1: For each state $(i, j, k) \in S$, the expected total cost $v(i, j, k)$ is a nondecreasing function in the number of links k .

Proof: The proof of this lemma is by induction. We must show

$$v(i, j, k) - v(i, j, k+1) \leq 0. \text{ Recall } I_2 = \int_0^\infty e^{-\mu t} G(dt). \text{ For } K \leq k < L:$$

$$v(i, j, k) = c(i, j, k, PO) + \sum_{l=1}^N \sum_{n=1}^{K-1} I_2 v(l, i, n) p(n|i, D) q(l|i).$$

From (3.10) and (3.11), it is clear that $c(i, j, k, a) \leq c(i, j, k+1, a)$ for all k . Hence,

$$\begin{aligned} v(i, j, k) &< c(i, j, k+1, PO) + \sum_{l=1}^N \sum_{n=1}^{K-1} I_2 v(l, i, n) p(n|i, D) q(l|i) \\ &= v(i, j, k+1). \end{aligned}$$

Thus, $v(i, j, k) \leq v(i, j, k+1)$ for $K \leq k < L$.

Since $(K-1+m) \wedge (K-1) = K-1$, for state (i, j, K) :

$$\begin{aligned} &v(i, j, K-1) \\ &= \min \left\{ c(i, j, K-1, NPO) + \sum_{l=1}^N \sum_{m=1}^M I_2 v(l, i, K-1+m) p(m|i, j) q(l|i), \right. \\ &\quad \left. c(i, j, K-1, PO) + \sum_{l=1}^N \sum_{m=1}^M \sum_{n=1}^{K-1} I_2 v(l, i, n) p(n|i, D) p(m|i, j) q(l|i) \right\} \\ &\leq c(i, j, K-1, PO) + \sum_{l=1}^N \sum_{n=1}^{K-1} I_2 v(l, i, n) p(n|i, D) q(l|i) \\ &\leq c(i, j, K, PO) + \sum_{l=1}^N \sum_{n=1}^{K-1} I_2 v(l, i, n) p(n|i, D) q(l|i) \\ &\leq v(i, j, K). \end{aligned}$$

For $i \neq j$ and $1 \leq k < K$, assume $v(i, j, k+1) \leq v(i, j, k+2) \leq \dots \leq v(i, j, L)$. We need

to show $v(i, j, k) - v(i, j, k + 1) \leq 0$. From (3.16),

$$v(i, j, k) = \min \left\{ c(i, j, k, NPO) + \sum_{l=1}^N \sum_{m=1}^M I_2 v(l, i, k+m) p(m|i, j) q(l|i), \right. \\ \left. c(i, j, k, PO) + \sum_{l=1}^N \sum_{m=1}^M \sum_{n=1}^{(k+m) \wedge (K-1)} I_2 v(l, i, n) p(n|i, D) p(m|i, j) q(l|i) \right\}.$$

Let a^* denote the optimal action of state $(i, j, k + 1)$. If $a^* = PO$,

$$\begin{aligned} v(i, j, k) &\leq c(i, j, k, PO) + \sum_{l=1}^N \sum_{m=1}^M \sum_{n=1}^{(k+m) \wedge (K-1)} I_2 v(l, i, n) p(n|i, D) p(m|i, j) q(l|i) \\ &\leq c(i, j, k+1, PO) + \sum_{l=1}^N \sum_{m=1}^M \sum_{n=1}^{(k+1+m) \wedge (K-1)} I_2 v(l, i, n) p(n|i, D) p(m|i, j) q(l|i) \\ &= v(i, j, k+1). \end{aligned}$$

On the other hand, if $a^* = NPO$,

$$\begin{aligned} v(i, j, k) &\leq c(i, j, k, NPO) + \sum_{l=1}^N \sum_{m=1}^M I_2 v(l, i, k+m) p(m|i, j) q(l|i) \\ &\leq c(i, j, k+1, NPO) + \sum_{l=1}^N \sum_{m=1}^M I_2 v(l, i, k+1+m) p(m|i, j) q(l|i) \\ &= v(i, j, k+1). \end{aligned}$$

To complete the proof, we need to show $v(j, j, k) - v(j, j, k + 1) \leq 0$ for $1 \leq k < K$. From (3.15),

$$\begin{aligned} v(j, j, k) &= c(j, j, k, NPO) + \sum_{l=1}^N I_2 v(l, j, k) q(l|j) \\ &\leq c(j, j, k+1, NPO) + \sum_{l=1}^N I_2 v(l, j, k+1) q(l|j) \\ &\leq v(j, j, k+1). \end{aligned}$$

Thus, by the principle of induction, for each state $(i, j, k) \in S$, the expected total cost $v(i, j, k)$ is a nondecreasing function in k . \square

The following proposition states the conditions under which the optimal policy has a threshold structure.

Proposition D.2: Given state $(i, j, k) \in S$, the optimal policy δ^* has a control limit (or threshold) structure:

$$\delta^*(i, j, k) = \begin{cases} NPO, & 1 \leq k < k^*, \\ PO, & k^* \leq k \leq L \end{cases} \quad (D.16)$$

when $I_1 \Delta f(k+m) - \sum_n \Delta h_{PO}(k+m-n)p(n|i, D) \geq 0$ for all m such that $p(m|i, j) \neq 0$

and $k^* \leq k < K$.

Proof: Let

$$\begin{aligned} r(i, j, k) = & \sum_{m=1}^M I_1 f(k+m)p(m|i, j) + \sum_{l=1}^N \sum_{m=1}^M I_2 v(l, i, k+m)p(m|i, j)q(l|i) \\ & - C_{PO} - \sum_{m=1}^M \sum_{n=1}^{(k+m) \wedge (K-1)} \{h_{PO}(k+m-n) + I_1 f(n)\} p(n|i, D)p(m|i, j) \\ & - \sum_{l=1}^N \sum_{m=1}^M \sum_{n=1}^{(k+m) \wedge (K-1)} I_2 v(l, i, n)p(n|i, D)p(m|i, j)q(l|i). \end{aligned}$$

Thus, the action PO is chosen if $r(i, j, k) \geq 0$ and the action NPO is chosen if

$r(i, j, k) < 0$. Let \hat{k} be the smallest k such $r(i, j, k) \geq 0$.

For convenience, let $\Delta r(i, j, \hat{k}) = r(i, j, \hat{k} + 1) - r(i, j, \hat{k})$.

$$\begin{aligned}
\Delta r(i, j, \hat{k}) = & \sum_{m=1}^M I_1 \Delta f(\hat{k} + m) p(m|i, j) + \sum_{l=1}^N \sum_{m=1}^M I_2 \Delta v(l, i, \hat{k} + m) p(m|i, j) q(l|i) \\
& - \sum_{m=1}^M \sum_{n=1}^{(\hat{k}+m) \wedge (K-1)} \Delta h_{PO}(\hat{k} + m - n) p(n|i, D) p(m|i, j).
\end{aligned} \tag{D.17}$$

Since $v(i, j, k)$ is a nondecreasing function in k , $\Delta v(l, i, \hat{k} + m) \geq 0$. Thus,

$\Delta r(i, j, \hat{k}) \geq 0$ when

$$I_1 \Delta f(\hat{k} + m) - \sum_{n=1}^{(\hat{k}+m) \wedge (K-1)} \Delta h_{PO}(\hat{k} + m - n) p(n|i, D) \geq 0$$

for all m such that $p(m|i, j) \neq 0$. Now for some k , assume $\Delta r(i, j, \hat{k}), \Delta r(i, j, \hat{k} + 1),$

$\dots, \Delta r(i, j, k - 1) \geq 0$. Then

$$\begin{aligned}
\Delta r(i, j, k) = & \sum_{l=1}^N \sum_{m=1}^M I_2 \Delta v(l, i, k + m) p(m|i, j) q(l|i) + \sum_{m=1}^M I_1 \Delta f(k + m) p(m|i, j) \\
& - \sum_{m=1}^M \sum_{n=1}^{(k+m) \wedge (K-1)} \Delta h_{PO}(k + m - n) p(n|i, D) p(m|i, j).
\end{aligned}$$

Since $v(i, j, k)$ is a nondecreasing function in k , $\Delta v(l, i, k + m) \geq 0$. Thus,

$\Delta r(i, j, k) \geq 0$ when

$$I_1 \Delta f(k + m) - \sum_{n=1}^{(k+m) \wedge (K-1)} \Delta h_{PO}(k + m - n) p(n|i, D) \geq 0$$

for all m such that $p(m|i, j) \neq 0$. Thus, by induction, the optimal policy has the control

limit structure when $I_1 \Delta f(k + m) - \sum_n \Delta h_{PO}(k + m - n) p(n|i, D) \geq 0$ for all m such that

$p(m|i, j) \neq 0$ and $k^* \leq k < K$. \square

Glossary of Acronyms

ABR	Available Bit Rate
ACTS	Advanced Communications Technologies & Services
ATM	Asynchronous Transfer Mode
CBR	Constant Bit Rate
GSM	Global System for Mobile Communication
ETSI	European Telecommunications Standards Institute
ID	Identifier
IMT-2000	International Mobile Telecommunications 2000
IP	Internet Protocol
ITU	International Telecommunication Union
LA	Location Area
PCS	Personal Communication Services
PNNI	Private Network-to-Network Interface
QoS	Quality of Service
UMTS	Universal Mobile Telecommunications System
VBR	Variable Bit Rate