On Noise-Induced Adaptive Network Control in Ad Hoc Networks Based on Biological Models

Submitted to
Graduate School of Information Science and Technology
Osaka University

July 2013

Narun ASVARUJANON

List of Publications

Journal Papers

1. Narun Asvarujanon, Kenji Leibnitz, Naoki Wakamiya and Masayuki Murata, "Resilient Mobile Ad Hoc Routing with Attractor Selection for Dense and Heavy Traffic Scenarios," to appear in *Special Issue of International Journal on Autonomous and Adaptive Communications Systems (IJAACS) on Self-* Systems*, 2013.

2. Narun Asvarujanon, Kenji Leibnitz, Naoki Wakamiya and Masayuki Murata, "Noise-Assisted Concurrent Multipath Traffic Distribution in Ad Hoc Networks," conditionally accepted for publication in *IEICE Transactions on Communications (Special Section on Progress in Information Network Science)*, November 2013.

Refereed Conference Papers

1. Narun Asvarujanon, Kenji Leibnitz, Naoki Wakamiya, and Masayuki Murata, "Robust and Adaptive Mobile Ad Hoc Routing with Attractor Selection," in *Proceedings of the 4th International workshop on Adaptive and DependAble Mobile Ubiquitous Systems (ADAMUS)*, (Berlin, Germany), July 2010.

Refereed Student Poster

1. Narun Asvarujanon, Kenji Leibnitz, Naoki Wakamiya, and Masayuki Murata, "Concurrent Multipath Traffic Distribution in Ad Hoc Networks based on Attractor Perturbation," presented at *32nd IEEE International Conference on Computer Communications (INFOCOM 2013)*, (Turin, Italy), April 2013.

Non-refereed Conference Papers

1. Narun Asvarujanon, Kenji Leibnitz, Naoki Wakamiya and Masayuki Murata, "Noise-Assisted Traffic Distribution over Multi-Path Ad Hoc Routing," in *Proceedings of 4th International Symposium on Applied Sciences in Biomedical and Communication Technologies (ISABEL)*, (Barcelona), October 2011. [Invited Paper]

2. Narun Asvarujanon, Kenji Leibnitz, Naoki Wakamiya, and Masayuki Murata, "An Attractor Perturbation-Based Traffic Distribution Method and Its Practical Experiments," to appear in *Proceedings of the 2013 International Symposium on Nonlinear Theory and its Applications (NOLTA)*, (Santa Fe, USA), September 2013.

Non-refereed Technical Papers

1. Narun Asvarujanon, Kenji Leibnitz, Naoki Wakamiya, and Masayuki Murata, "Extension and Evaluation of Biologically-Inspired Routing Protocol for MANETs," *Technical Report of IEICE (NS2009-52)*, pp. 55-60, July 2009.

2. Narun Asvarujanon, Kenji Leibnitz, Naoki Wakamiya, and Masayuki Murata, "Evaluation of Robustness and Adaptability of a Biologically-Inspired MANET Routing Protocol," *Technical Report of IEICE (NS2009-143)*, pp. 7-12, January 2010. [Encouragement Talk]

Preface

Computer networks have become highly complicated and less flexible to handle emerging problems which often occur nowadays. In order to cope with unpredictable problems, the concept of biologically inspired networks has been introduced to provide a high degree of robustness and adaptability to computer networks. In this thesis, we focus specifically on studying the role of randomness or fluctuation in biological systems. In conventional engineering systems, the randomness—usually referred to as *noise*—is normally seen as an undesirable factor for control mechanisms to the extent that there are efforts on removing and filtering noise to achieve higher signal to noise ratio (SNR) for strict control. On the contrary to artificial systems, biological systems adopt the concept of noise as a part of their mechanisms instead of eliminating it. By utilizing noise internally, biological systems are able to achieve high robustness and adaptability against external noise. Inspired by the concept of utilizing noise, we propose two adaptive noise-induced network control methods for wireless ad hoc networks in this thesis.

The first network control method is a routing protocol for mobile ad hoc networks (MANETs). Routing in MANETs is not a trivial task since it is greatly affected by external influences such as mobility/failure of nodes, unreliability/instability of wireless communication, arbitrarily initiated/terminated sessions, or uncontrollable joining/leaving of nodes. Such adverse and changing environment conditions can also be often observed in biology, where biological systems show a remarkable ability to survive and adapt to these changes. In this thesis, we further improve our previously proposed *MANET routing protocol, called MARAS*, which is based on *attractor selection*, a biological adaptation mechanism that is applied to the next hop selection process. In attractor selection, noise plays an essential role in coping with the uncertainties and variations of the system, which is network dynamics in our case. We will show that by utilizing noise to a certain extent, MARAS is capable of being more adaptive than other well-known protocols. Especially in the presence of large traffic volume and high node densities, generally considered as the worst case scenarios for MANET routing, MARAS retains its superiority as it is capable of still delivering a certain portion of traffic when the other mechanisms fail.

Through the study of the first network control method, we found that the limited bandwidth is one of the biggest challenges in MANETs and ad hoc networks in general. Therefore, we try to improve the available bandwidth in ad hoc networks by using multiple paths concurrently. However, since the quality of each path frequently changes in ad hoc networks due to its dynamic nature, a new challenge of appropriate traffic distribution over multiple paths arises. Unfortunately, traditional traffic distribution methods often rely heavily on the detailed knowledge of each network component and the preconfigured, i.e. fine-tuned parameters. Such detailed knowledge is difficult to obtain with the limited bandwidth, and preconfigured values are usually useful for considered situations but may not be suitable in case of unforeseeable changes. Therefore, we introduce a new concept, called *attractor perturbation* (AP), which enables an adaptive network performance control using only end-to-end statistical information. Based on AP, we propose the second network controlling method, a *concurrent multipath traffic distribution method*, which aims at lowering the average end-to-end delay by only adjusting a sending rate on each path. We demonstrate through simulations that by utilizing the noise-based attractor perturbation relationship, the proposed method achieves a lower average end-to-end delay compared to other methods which do not take fluctuations into account.

Finally, we summarize our observation of noise-based model behavior through research and implementations. Based on our experience, we present advantages and constraints of both models. Furthermore, we also provide guidelines of using our models in other applications with examples of existing implementations. Finally, we proposed a novel concept which combines both models to achieve a multi-objective application.

Acknowledgement

This thesis would never be accomplished without the support of many people. First, I would like to express my gratitude to my supervisor, Professor Masayuki Murata, who has always given me invaluable comments and encouraged me through the hardship of the research.

I would like to also thank the members of my thesis committee, Professor Teruo Higashino, Professor Toru Hasegawa, and Professor Takashi Watanabe of the Graduate School of Information Science and Technology, Osaka University, and Professor Morito Matsuoka of the Cybermedia Center, Osaka University, for their valuable comments, and efforts in reviewing this thesis.

Moreover, I would like to show my deepest appreciation to Professor Naoki Wakamiya who always devotes himself to guiding my research and gives me priceless advice. Without his understanding and support, I would never be able to complete the Ph.D. course.

I would like to particularly thank Invited Professor Kenji Leibnitz for his precious time discussing various topics with me. He does not only inspire me with many ideas but also continually shows his care and understanding. Furthermore, I am greatly indebted to his effort in proofreading my papers, including this thesis.

Furthermore, I would like to show my gratitude to Associate Professor Go Hasegawa, Associate Professor Shin'ichi Arakawa, and Assistant Professor Yuichi Ohsita for their suggestions on my research. I would like to also acknowledge Professor Koso Murakami, Professor Makoto Imase, and Professor Hirotaka Nakano for their comments on this study since the beginning of Ph.D. course until their retirement from Osaka University.

I am also grateful to the Ministry of Education, Culture, Sports, Science and Technology

(MEXT) for the financial support during the course.

Last but not least, I would like to seize this opportunity to express my personal appreciation to my family. No words of gratitude would be adequate to their priceless unconditional support and love. I would never have been able to complete this thesis without all of you. Thank you for everything.

Contents

1 Introduction **1**

1.1 Challenges in Communications Networks 2

1.1.1 Routing in Ad Hoc Networks 2

1.1.2 Multipath Communications 3

1.2 Bio-inspired Concepts 4

1.2.1 From Noise to Biological Systems 4

1.2.2 Biological Systems and Computer Networking 5

1.2.3 Attractor Selection Mechanism 6

1.2.4 Attractor Perturbation 7

1.3 Thesis Outline 8

2 Resilient Mobile Ad Hoc Routing with Attractor Selection **11**

2.1 Introduction 11

2.2 Related Work 13

2.3 Mathematical model 14

2.3.1 Extended Attractor Selection-based Model 14

2.3.2 Design of the Parameters in the Mathematical Model 16

2.3.3 Attractor States 16

2.3.4 Designing the Synthesis and Degradation Functions 17

2.3.5 Mathematical Model for Next Hop Selection 18

2.4 Our Routing Protocol 20

2.4.1 Route Establishment 20
2.4.2 Routing Information 22
2.4.3 Data Packet Forwarding 23
2.4.4 Route Maintenance 23
2.5 Evaluation 27
2.5.1 Common Simulation Settings 27
2.5.2 Evaluation against Failures 29
2.5.3 Evaluation against Node Density 30
2.5.4 Evaluation against Number of Traffic Sessions 31
2.5.5 Evaluation under Mobility Scenarios 33
2.6 Summary 35

3 Traffic Distribution over Multiple Paths with Attractor Perturbation 37
3.1 Introduction 37
3.2 Motivation and Problem Formulation 39
3.2.1 System Model 41
3.2.2 Problem Definition: 2 Paths 41
3.2.3 Lagrangian Optimization 43
3.2.4 Optimal Solution 44
3.2.5 Extended Model for n Paths 44
3.3 Preliminary Investigation 46
3.3.1 Evaluation of Linearity between Fluctuation and Response 46
3.3.2 Simulation of Network Traffic 47
3.3.3 Discussion 48
3.4 Concurrent Multipath Traffic Distribution 49
3.5 Evaluation 50
3.5.1 Comparison Target 50
3.5.2 Bandwidth Improvement 52
3.5.3 End-to-End Delay Improvement 55

3.5.4 Discussion on Bio-inspired Adaptability 58
3.6 Summary 58

4 Design Considerations for Future Applications of Noise-based Models 61
4.1 Introduction 61
4.2 Advantages and Constraints of Noise-based Models 63
4.2.1 Non-rule-based Functionality 63
4.2.2 Robustness against Uncertainties 64
4.2.3 Simplified Design 65
4.3 Application Design Guidelines 66
4.3.1 Using Attractor Selection 66
4.3.2 Using Attractor Perturbation 67
4.3.3 Multi-objective Applications 69
4.4 Example of Multi-objective Applications 70
4.4.1 Layered Attractor Selection 70
4.4.2 Attractor Perturbation over Attractor Selection 72
4.5 Summary 75

5 Conclusion 77
5.1 Contributions of This Thesis 77
5.2 Future Research Direction 78

Bibliography 81

List of Figures

1.1 Overview of attractor selection 7

1.2 Dynamics of attractor perturbation 8

2.1 Behavior of attractor selection system 15

2.2 Dynamics of M values from attractor selection model 19

2.3 Overview of MARAS 21

2.4 Evaluation results against number of failure occurrences 30

2.5 Evaluation results against node density 31

2.6 Evaluation results against number of traffic sessions 32

2.7 Evaluation results under mobility scenario 34

2.8 Evaluation results against both increasing node density and traffic 35

3.1 Overall system model 42

3.2 Numerically simulated fluctuation and response of dynamic system 46

3.3 Histogram of packet delays when doubling the traffic rate at 500 s 47

3.4 Simulation Scenario from [12] 52

3.5 Bandwidth comparison between AP-based proposals and CMT unlimited *rBuf* from [12] 53

3.6 Performance comparison under mobility scenario 56

3.7 Average delay comparison under mobility scenario (y-axis is capped for visibility) 57

3.8 Results from different values of b 58

4.1 Overview of cooperative routing and clustering controls [53] 71

4.2 Layered routing based on attractor selection model [69] 72

List of Tables

2.1 Simulation parameters of MARAS 28

Chapter 1

Introduction

Traditional network control mechanisms often rely on a certain set of predefined rules and fine-tuned parameters for known situations. However, computer network architectures and their protocols have become increasingly sophisticated over time through addition of many features to support new applications, where different applications may require different settings of protocol parameters. Since the total number of possible situations occurring in the real world is too large to be handled by preprogrammed sets of definitions, it is necessary that new networking mechanisms are designed in a flexible and adaptive manner to cater for any changes in the environment. One of the possible solutions investigated and proposed in this thesis is the concept of bio-inspired networking which has recently been introduced to to tackle unpredictable and unstable situations in computer networks.

One particular type of computer networks which has gained our interest is the ad hoc network. Ad hoc networks, or wireless ad hoc networks, are communications networks formed by wireless nodes in an ad hoc manner. In this type of networks, there is no infrastructure nor centralized control body and every node acts as both client and control node, i.e. router, at the same time. With these features, the deployment of ad hoc networks is easier and more flexible when compared to traditional wired networks. Despite this benefit, there are quite a few challenges in ad hoc networks, e.g, a dynamic topology due to arbitrary node participation, unstable wireless connection, and also mobility if nodes are mobile. Second is a limited amount of resources, i.e., bandwidth and energy, which are crucial for communications and controls. In contradiction to the limited resources, ad hoc network protocols need to be scalable since the number of participating nodes are theoretically unlimited. Therefore, the ad hoc network can be considered one of the networks with unpredictable problems that need a new adaptive mechanism to overcome the addressed challenges. Hence, we have chosen to focus our bio-inspired network control research on ad hoc networks.

In this chapter, we first explain challenges in communication networks in Section 1.1. Then, we introduce bio-inspired concepts for communication networks in Section 1.2. Finally, an outline of this thesis is explained in Section 1.3.

1.1 Challenges in Communications Networks

1.1.1 Routing in Ad Hoc Networks

One of the most fundamental problems of ad hoc networks is routing, which must be done in a multi-hop fashion due to a lack of infrastructure. Routing in ad hoc networks is not a trivial task since it is greatly affected by external influences such as failure of nodes, unreliability/instability of wireless communication, arbitrarily initiated/terminated sessions, or uncontrollable joining/leaving of nodes [18, 52], especially in a special type of ad hoc networks where nodes are mobile, which is called mobile ad hoc networks (MANETs), the network dynamics is even more immense. Widely adopted and well-known ones are ad hoc distance vector (AODV), dynamic source routing (DSR), and optimized link state routing (OLSR). More details on existing protocols can be found in [1, 60].

Even though there are many existing ad hoc network routing protocols, MANET is still questioned for scalability. Due to the limited bandwidth and the sensitivity to interference on the wireless channel, the effectiveness of communication in MANETs could be drastically decreased as the number of nodes and the amount of traffic increases. On the other hand, MANETs are expected to show a better survivability since they operate in a completely decentralized manner, which is generally more robust against single points of failure. Due to these issues and expectations, achieving scalability and survivability are major points in designing routing protocols for MANETs.

Another widely discussed feature of MANETs is their adaptability. Due to the lack of centralized control, a routing protocol has to be able to learn about the condition of the environment and adapt itself to any changes. Hence, we need to find a mechanism which provides this kind of self-organizing and environment-aware abilities. Such features are often exhibited in biological systems and their mechanisms usually consist of simple rules [25] among distributed entities, which is also very suitable for routing in MANETs.

1.1.2 Multipath Communications

Since one of major problems of ad hoc networks is the limited bandwidth, there are research attempts to solve it by reducing bandwidth loss due to radio interference from other adjacent communications, e.g. by using directional antenna [73]. However, a more common approach is to use multiple paths—not only the shortest path—in order to increase overall network bandwidth. In addition to bandwidth improvement, multipath communications can also increase fault tolerance, improve end-to-end delay, provide better load-balancing, and improve energy consumption in ad hoc networks [49]. Examples of multipath approaches in ad hoc networks are split multipath routing (SMR) [37], extensions of AODV to find link-disjoint paths (AOMDV) [47], a multipath extension of DSR (MP-DSR) [43], and a multipath extension of OLSR (MP-OLSR) [72].

Nevertheless, little has been addressed regarding how to distribute traffic over multiple paths. Existing protocols either use additional paths as backup paths upon link failure [43] or use a round-robin scheme to distribute traffic equally on each path [72]. Moreover, according to the current trend of having multiple radio access technologies (RATs) per device, the concept of multihoming, i.e. using multiple paths over different interfaces or networks, has gained more interest. Using all available paths blindly—within or across networks—could degrade overall performance and we realized that there is a need for a new concurrent multipath traffic distribution algorithm. There have been a few existing work on this topic, such as, concurrent multipath transmission (CMT) [12, 31] and multipath realtime transport protocol (MP-RTP) [57].

1.2 Bio-inspired Concepts

In this section, we explain the features of biological systems, biologically inspired mechanisms and their applications, and the bio-inspired concepts that are used in this thesis.

1.2.1 From Noise to Biological Systems

Noise or fluctuation is an undesirable factor for conventional engineering system controls. Therefore, it is common for those systems to try to remove noise using filters. Contradictively, randomness has been introduced into improving optimality search algorithm [55], which random walk is one of the simplest examples. Hence, it is learned that a negative feedback process to suppress noise is not preferable at all times.

The fact that noise is found everywhere in biological systems, for examples, marine predator search behavior [56] can be modeled as a Levy walk—a special type of random walk following Levy distribution, and a lot of research found that gene expressions that regulate cellular functions are subject to noise or stochastic fluctuation [27, 33, 46], shows that instead of removing noise, living systems unavoidably take positive feedback process and function in a presence of large fluctuations [32].

There is an explanation on how biological systems can remain stable with large fluctuation in [32]. From a macroscopic view of a cellular system, the system is considered dissipative. In such dissipative system, a macroscopic description is robust against microscopic change, where many microscopic states fall into the same macroscopic state. Therefore, even if a state is perturbed by fluctuations and it is deviated from the original deterministic rate equation, there is a region where the state tends to return to, which is called attractor. Due to this attraction mechanism, a macroscopic state in cellular system is stable against molecular fluctuations.

1.2.2 Biological Systems and Computer Networking

The complexity of computer networking has increased after many years of development. It has become very difficult to solve emerging problems by traditional approaches due to both computational and technological limitations. Among the promising approaches to tackle complex problems is to make networks self-organized, where no central unit dominates the whole network system and interactions among simple entities realize the globally controlled behavior. Designing self-organizing mechanisms for solving emerging problems is not a simple task. Therefore, researchers started looking for existing self-organizing mechanisms and found them in biological systems. Since they are the evolutionary product over many generations of the individuals, often driven by random natural selection, they are known to be fault tolerant, robust, adaptive, survivable, and scalable. Hence, bio-inspired mechanisms are widely adopted for handling pervasive scenarios [25].

Bio-inspired self-organizing concepts are usually heuristic, or metaheuristic [70], algorithms designed for finding optimal or sub-optimal solution to optimization problems in a reasonably practical time. In recent years, many new bio-inspired optimization algorithms have been proposed, such as, harmony search [36] and firefly synchronization [61]. However, among those algorithms, the most well known and widely used for computer networking would be *swarm intelligence*.

The concept of swarm intelligence [13, 14] originates from the social behavior of insect colonies, such as ant colonies. The main algorithm of ant-based routing protocol is the Ant Colony Optimization (ACO) [23]. ACO is derived from the foraging process of ants which is a random walk when searching for food. Once the food is found, the ant returns to the nest via its own trail. While returning, the ant deposits pheromones on the way as chemical markers for other ants to follow its trail to the food. The indirect communication, which is based on the pheromone trail mediated by the environment, is called *stigmergy*. Using this approach, the shortest path between the source and the destination can be found. Example applications of swarm intelligence are routing protocols, such as, AntNet [21], ARA [30], AntHocNet [22], BeeAdHoc [66], and HOPNET [65].

In the following subsections, we explain the bio-inspired concepts used in this thesis, attractor selection and attractor perturbation mechanisms.

1.2.3 Attractor Selection Mechanism

The attractor selection mechanism is modeled after the behavior of *E. coli* cells, which are capable of adapting to dynamically changing nutrient conditions in the environment without any predefined adaptation rules [33]. A mutant *E. coli* cell has a gene regulatory network consisting of two mutually inhibitory sequences of chemical reactions which synthesize two corresponding nutrients. When one of the nutrients becomes scarce, the protein concentration activating a sequence for the missing nutrient increases to return the cell to a stable gene expression. However, there is no explicit rule-based mechanism to switch between the sequences of chemical reactions. In [33], a mathematical model describing this bistable behavior of protein concentrations m_1 and m_2 is proposed as

$$\begin{aligned} \frac{dm_1}{dt} &= \frac{s(\alpha)}{1+m_2^2} - d(\alpha)\, m_1 + \eta_1 \\ \frac{dm_2}{dt} &= \frac{s(\alpha)}{1+m_1^2} - d(\alpha)\, m_2 + \eta_2 \end{aligned} \tag{1.1}$$

where $s(\alpha)$ and $d(\alpha)$ are the rate coefficients of protein synthesis and decomposition, respectively. Both of them depend on α which represents the cell activity or cell volume growth. The terms η_i are independent white noise that exists in gene expression.

The essential point in Eqn. (1.1) is the interaction between activity α and noise terms η_i, as shown in Figure 1.1. If the ratio between activity and noise is sufficiently large, the system's behavior remains rather unaffected by noise. On the other hand, if activity approaches zero, the dynamics of the system states m_1 and m_2 become entirely determined by noise, i.e., they perform a random walk. When the state randomly approaches a new attractor, activity α will increase which results in the state being locked at the new attractor.

Example applications of attractor selection are self-adaptive multi-path routing in overlay networks [40] and layered attractor selection for clustering and data gathering in wireless

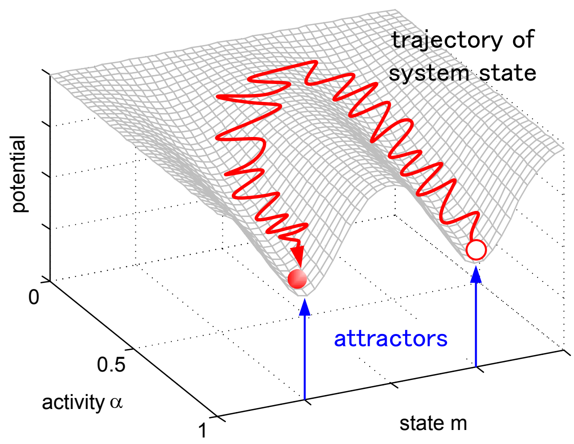


Figure 1.1: Overview of attractor selection

sensor networks [53].

1.2.4 Attractor Perturbation

The attractor perturbation model is derived from observations of fluctuation and response in biological systems, in particular, an experiment on the evolution of functional proteins in a clone bacteria cell. In [54], it was found that the fluctuation, which is expressed by the variance of the fluorescence of a bacterial protein, and its response, which is the average change in this fluorescence, have a linear relationship modeled as follows when a force is introduced:

$$\bar{x}_{a+\Delta a} - \bar{x}_a = b\,\Delta a\,\sigma_a^2 \tag{1.2}$$

where b is a scalar constant, x is a time dependent measurable variable in the system with mean $\bar{x}$ and variance σ_a^2, and a is a controllable parameter.

There are two major assumptions underlying the model formulation of AP. First, the variable x must have a Gaussian-like distribution which is often observed in biology. Second, the variable x and the parameter a are closely associated, in other words, a change in the parameter a would strongly affect the distribution of the variable x.

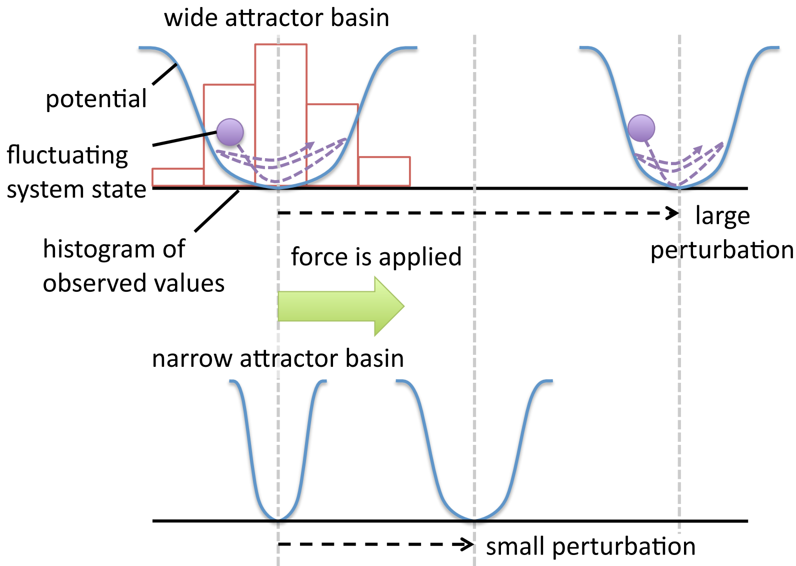


Figure 1.2: Dynamics of attractor perturbation

Equation (1.2) reveals that the difference in the average of the variable x before and after applying a change to the parameter a is linearly proportional to the amount of change in a and the variance of the variable x prior to the change. Since the amount of change in a, called *force* can be seen as controllable, it is possible to adjust the difference in average of x, called *perturbation*, by taking the current variance of x into consideration. Obviously, using the same amount of force Δa to perturb the average of x when the variance σ_a^2 is large will also lead to a larger perturbation, as shown in Fig. 1.2. This figure also shows the attractor basins corresponding to each empirical distribution of x.

1.3 Thesis Outline

Based on the challenges explained in Section 1.1, we realized that the current communications networks need new adaptive and robust mechanisms to overcome those challenges. In particular, we considered using noise-assisted bio-inspired mechanisms [63] to achieve adaptability and robustness in our proposals due to their inherent robust and adaptive features explained in Section 1.2. In this thesis, we propose two network controls for ad hoc networks which are routing protocol and traffic distribution method. Moreover, we also present design considerations on future applications of noise based models which are results of our study throughout this thesis.

The work in this thesis is organized as follows:

Chapter 2

Resilient Mobile Ad Hoc Routing with Attractor Selection [5–7, 10]

This chapter explains our noise-based mobile ad hoc routing protocol. To address challenges described in Section 1.1.1, we aim to achieve scalability, survivability, and adaptability in MANET routing by applying a bio-inspired model, called attractor selection. We first show how we extend the attractor selection model and design its parameters. Then, we describe how we apply the extended model to a next hop selection process, along with algorithmic details of the routing protocol. Finally, evaluation results of our proposal over various scenarios in comparison to state-of-the-art routing protocols are shown.

Chapter 3

Traffic Distribution over Multiple Paths with Attractor Perturbation [8, 9]

This chapter explains our noise-based traffic distribution method. Our method aims to address challenges raised in Section 1.1.2, which is to distribute traffic concurrently over multiple paths without using round-robin or blind scheduling. We first explain our motivation and problem formulation based on attractor perturbation model. Then, we perform a preliminary investigation of the model applicability. After that is confirmed, we propose an algorithm for concurrent multipath traffic distribution. Finally, simulation results in comparison to other concurrent multipath traffic control protocols are shown.

Chapter 4

Design Considerations for Future Applications of Noise-based Models

This chapter is dedicated for noise-based model discussions for audience who is interested in grasping a better understanding of our the models. We first introduce the background of noise-based models. Then, we provide a list of advantages and constraints of our models along with crucial points that require application designers considerations based on actual experience in designing and implementing them in network control applications. We further discuss the important design points for applying each model on applications. Finally, we provide application examples, both existing work and new ones.

Chapter 2

Resilient Mobile Ad Hoc Routing with Attractor Selection

Preface

Portions of this chapter were previously published as ‘Resilient Mobile Ad Hoc Routing with Attractor Selection for Dense and Heavy Traffic Scenarios,” in Special Issue of International Journal on Autonomous and Adaptive Communications Systems (IJAACS) on Self-* Systems, and have been reproduced with permission. Copyright is held by Inderscience Enterprises Ltd.

2.1 Introduction

In Mobile Ad Hoc Networks (MANETs) there is no predetermined infrastructure and each node can move around, arbitrarily join, or leave the network. With these features, the deployment of MANETs is easier and more flexible when compared to traditional wired networks. Despite this benefit, however, MANETs have been questioned due to doubts regarding their scalability. Due to the limited bandwidth and the sensitivity to interference on the wireless channel, the effectiveness of communication in MANETs could be drastically decreased as the number of nodes and the amount of traffic increases. On the other hand, MANETs are expected to show a better survivability since they operate in a completely decentralized manner, which is generally more robust against single points of failure. In the context of routing, survivability is about maintaining connectivity to each node [45]. Due to these issues and expectations, achieving scalability and survivability are major points in designing routing protocols for MANETs.

Another widely discussed feature of MANETs is their adaptability. Due to the lack of centralized control, a routing protocol has to be able to learn about the condition of the environment and adapt itself to any changes. Hence, we need to find a mechanism which provides this kind of self-organizing and environment-aware abilities. Such features are often exhibited in biological systems and their mechanisms usually consist of simple rules [25] among distributed entities, which is also very suitable for routing in MANETs.

Among various biologically inspired mechanisms described in the literature, we apply the *attractor selection mechanism* [33] in our routing protocol. This mechanism was shown in [41,42] to be useful as a basic control mechanism for robust and adaptive routing. Based on the concept in [41,42], we extended and implemented the MARAS routing protocol (Mobile Ad hoc Routing with Attractor Selection) in a commercial QualNet simulator to guarantee that interactions on the underlying MAC and PHY layers of the IEEE 802.11 protocol stack are considered in our protocol, hence, it can be utilized by real world applications. Our first evaluation in [7] already showed that MARAS is robust and adaptive to failures or mobility of nodes, but it yielded a lower packet delivery performance when no failures of nodes were assumed than in the presence of few failures. Such phenomenon implies that the performance of MANET routing depends on the number of nodes in the network. Therefore, we decided to further evaluate our protocol to study its limitations under heavy traffic and high node densities.

As an extended study of [7], we focus in this chapter on investigating the scalability and survivability of our protocol. A similar study on scalability of routing protocols was made in [4], but the number of traffic sessions there was low and the inter-packet interval was large so that extreme conditions were not reached. As we aim at investigating the limitations of our protocol, we evaluate here our protocol against other well-studied protocols under high node density and heavy traffic conditions. Under these conditions, traditional protocols may need to be fine-tuned in order to operate well or else the performance could severely deteriorate and could even completely stop functioning at all. Combining results from [7] and this chapter, we would like to show that MARAS is able to operate under a wider range of conditions than AODV [51] and AntHocNet [22] because it has better adaptability to a changing environment and is more resilient under severe conditions. We demonstrate the main benefit of MARAS through simulations, which lies in its survivability under changing conditions without any additional external management effort.

The rest of this chapter is organized as follows. Based on the the biologically inspired concept in Section 1.2, we introduce our proposed attractor selection mechanism in Section 2.3.1. Next, we describe our protocol in greater detail in Section 2.4. Then, the evaluation results from simulation are presented and discussed in Section 2.5. Finally, we conclude this chapter and list future work.

2.2 Related Work

Regarding biologically inspired networking technologies, swarm intelligence-based MANET routing protocols [22, 65] or firefly-based synchronization mechanisms have been proposed. Further examples are the Bio-Networking middleware architecture in [59] or the Perplexus project [15], in which a wireless network system with reconfigurable hardware has been developed to achieve biological features of "growth, evolution, and learning".

Among these protocols, we have chosen AntHocNet [22] as a comparison protocol in our evaluation section. AntHocNet is an Ant Colony Optimization (ACO) inspired routing algorithm for MANETs. It combines both reactive and proactive routing strategies where it reactively establishes and maintains routes only when they are needed and proactively maintain and improve the on-going routes by gathering more routing information. AntHocNet uses two processes to gather routing information. One is an ACO based path sampling process using artificial ants. The other, called *pheromone diffusion*, is used to spread out the pheromone information which is placed by ants to the neighbor nodes, aiming to guide the next ant to the destination in a similar manner to the Bellman-Ford algorithm. Combining both processes, AntHocNet can obtain the routing information in a time efficient, adaptive, and robust manner.

2.3 Mathematical model

2.3.1 Extended Attractor Selection-based Model

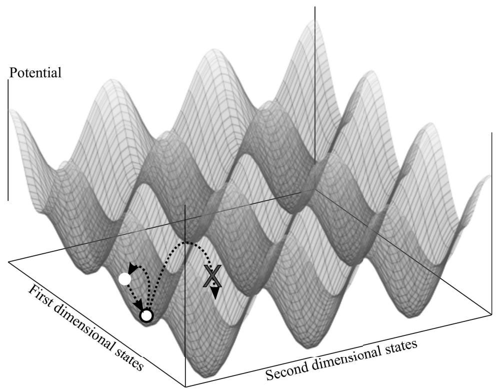
We generalized the model based on Eqn. (1.1) from 2 to M dimensions. Let m_i be the value representing whether the i-th choice should be selected. Moreover, let us define the M-dimensional state vector $\vec{m} = (m_1, \ldots, m_M)$. The attractor selection among M alternatives shall have the general form as

$$\frac{d\vec{m}}{dt} = \vec{f}(\vec{m}) \times \alpha + (1 - \alpha) \times \vec{\eta}, \tag{2.1}$$

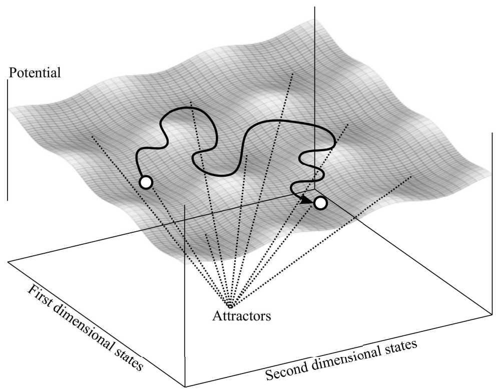
where α expresses the goodness of the current condition and $\vec{\eta} = (\eta_1, \ldots, \eta_M)$ is the vector of the noise affecting the selection.

The activity $\alpha \in [0, 1]$ is the main feedback variable, which corresponds to the current performance of the system and it is used to control the influence of randomness on attractor selection. When the current condition of the system becomes undesirable, the activity decreases. As a result, the value of term $\vec{f}(\vec{m}) \times \alpha$ decreases and a larger effect from noise $\vec{\eta}$ takes place to shift the system to another attractor by a random walk. Once the system approaches a suitable attractor, the activity increases and the effect of noise is suppressed, which then allows the system to become stable again. Moreover, to suppress the effect of noise even further when the activity becomes very high, we added the coefficient $(1 - \alpha)$ to the original equation.

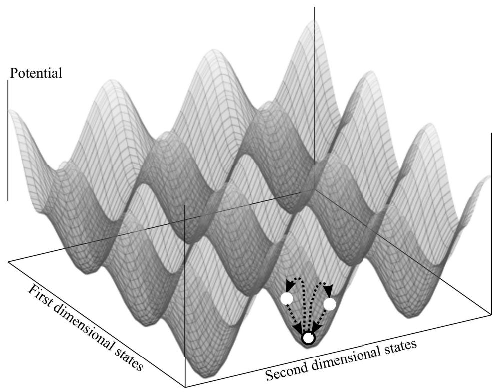
In Figure 2.1, we show the general principle of the attractor selection concept. The x-axis shows the first dimensional state m_1 and the y-axis shows the second dimensional state m_2 where the attractors are shown as valleys. The z-axis indicates the potential at



(a) With high activity, the system state cannot move away from the current attractor by only the small noise effect



(b) When the activity becomes lower, the potential landscape becomes smoother which allows random walk by noise



(c) Once the system state reaches another suitable attractor, the activity will increase and the current state will remain at the current attractor

Figure 2.1: Behavior of attractor selection system

each state. The current system state is illustrated as a circle that is constantly in motion due to the effect of the noise. It can be observed in Figure 2.1(a) that when the activity is high, moving the system's state away from the current attractor by the effect of noise is difficult because of the steepness of the potential landscape. However, due to external influences, for example, the activity decreases in Figure 2.1(b) leading to a flatter potential landscape and the system's state can be changed by the small effect of noise. After a better state is found, the activity increases again as can be seen in Figure 2.1(c). As a result, the potential landscape becomes steep and the current state is once again stable at the new attractor.

At first, the concept of having noise in the system may look undesirable. However, adding noise into the system makes it in general more robust to external fluctuations. In sensor networks, noise and random walk can provide load-balancing and scalable properties as shown in [11]. Moreover, getting stuck in local minima can be avoided using noise and random walk as explained in [55].

2.3.2 Design of the Parameters in the Mathematical Model

We now briefly discuss how the parameters have been extended in our mathematical model and we further show how to design appropriate values. This derivation is not to be considered as a formal mathematical proof, but an informal discussion of the underlying model.

2.3.3 Attractor States

The attractors can be obtained from the steady state solutions of the system in Eqn. (2.8). Without loss of generality, we can assume that the maximum index value of m_i is at some index $k = \arg\max_i m_i$. For the system to converge at an attractor, we assume $\alpha = 1$ and since the mean of η_i is 0, we can ignore it in the following. The equilibrium state is then obtained from the solution of the following equation system

$$\frac{dm_i}{dt} = 0 \qquad \Rightarrow \qquad m_i = \frac{\varphi}{1 + m_k^2 - m_i^2} \qquad i = 1, \dots M \tag{2.2}$$

where we define $\varphi = s(\alpha)/d(\alpha)$. This equation system basically can be separated to two cases that either m_i is maximal or not.

For $i = k$, Eqn. (2.2) simply reduces to $m_k = \varphi$. On the other hand, for $i \neq k$, it means

m_i is not maximal and we have the following equation

$$m_i = \frac{\varphi}{1 + m_k^2 - m_i^2} \tag{2.3}$$

which translates into the following polynomial after replacing $m_i = x$ and $m_k = \varphi$ from above.

$$(x - \varphi)\left(x^2 + \varphi x - 1\right) = 0 \tag{2.4}$$

The roots of this polynomial in Eqn. (2.4) are then x_1, x_2, x_3.

$$x_{1/2} = \frac{1}{2}\left(-\varphi \pm \sqrt{\varphi^2 + 4}\right) \qquad\qquad x_3 = \varphi. \tag{2.5}$$

Since $m_i \geq 0$, we can eliminate x_2 having the minus in Eqn. (2.5) and we can also eliminate x_3, since $i \neq k$. Thus, the equilibrium values of Eqn. (2.8) consist of M-dimensional vectors having one high value $H = \varphi$ and all other $M - 1$ are low values $L = \frac{1}{2}\left(\sqrt{\varphi^2 + 4} - \varphi\right)$. Note that this high value may be any element, so in total we have such M attractor states.

2.3.4 Designing the Synthesis and Degradation Functions

We still need to formulate the functions $s(\alpha)$ and $d(\alpha)$ in Eqn. (2.8). For reducing the number of unknown values, we first set $d(\alpha) = \alpha$ and our task is now to define the function $s(\alpha)$ appropriately.

We should bear in mind that our requirements were such that we have attractors where the elements have a high value H and $M - 1$ low values $L \leq H$. Thus, it must also hold that H is larger than L.

$$\frac{1}{2}\left(\sqrt{\varphi^2 + 4} - \varphi\right) \leq \varphi \qquad \Rightarrow \qquad \frac{1}{\sqrt{2}} \leq \varphi. \tag{2.6}$$

We define this lower bound as $\varphi^* = 1/\sqrt{2}$.

Now we can define any monotonous and differentiable increasing function $s(\alpha)$, which has the boundary condition of $\varphi(0) = 1/\sqrt{2}$. Since we restrict the domain of α to the interval $[0, 1]$, the target value at $s(1) = \beta$ would have to be defined in such a way that it is sufficiently "far" from the low values that any fluctuations would not spontaneously perturb the system state out of this attractor. The function $s(\alpha)$ which we use in this paper is then simply defined as

$$s(\alpha) = \alpha \left[\beta\alpha^{\gamma} + \varphi^{*}\right] \tag{2.7}$$

where γ is a tunable parameter for the increase of φ with respect to α.

2.3.5 Mathematical Model for Next Hop Selection

Attractor selection is adopted in our protocol for next hop selection among neighbor nodes. Hence, we map the vector of neighbors to $\vec{m}$, which contains state value m_i, indicating whether the i-th neighbor should be selected among M neighbors as a next hop node for a certain destination. We further map activity α to the information reflecting the goodness of the current routing condition. Since the next hop selection shall provide a single next hop neighbor as the solution, we design the controlling function of attractor selection as shown in Eqn. (2.8).

For neighbor node $1 \leq i \leq M$:

$$\frac{dm_i}{dt} = \frac{s(\alpha)}{1 + m_{max}^2 - m_i^2} - d(\alpha)\, m_i + (1 - \alpha)\, \eta_i, \tag{2.8}$$

where $m_{max} = \max_{j=1,\ldots,M}\{m_j\}$, $s(\alpha) = \alpha[\beta\alpha^{\gamma} + \varphi^{*}]$, $d(\alpha) = \alpha$, $\varphi^{*} = 1/\sqrt{2}$, and η_i is Gaussian white noise with mean 0 and variance of 1. Parameters β and γ control the influence of activity over state values and we use empirically determined values $\beta = 10$ and $\gamma = 3$ throughout this study. A detailed discussion on how to design these parameters can be found in Appendix 2.3.2.

In case of high activity α, Eqn. (2.8) yields the $\vec{m}$, which has a single high m_i value and $M - 1$ low values m_j, $j \neq i$. This follows from the deterministic part of Eqn. (2.8)

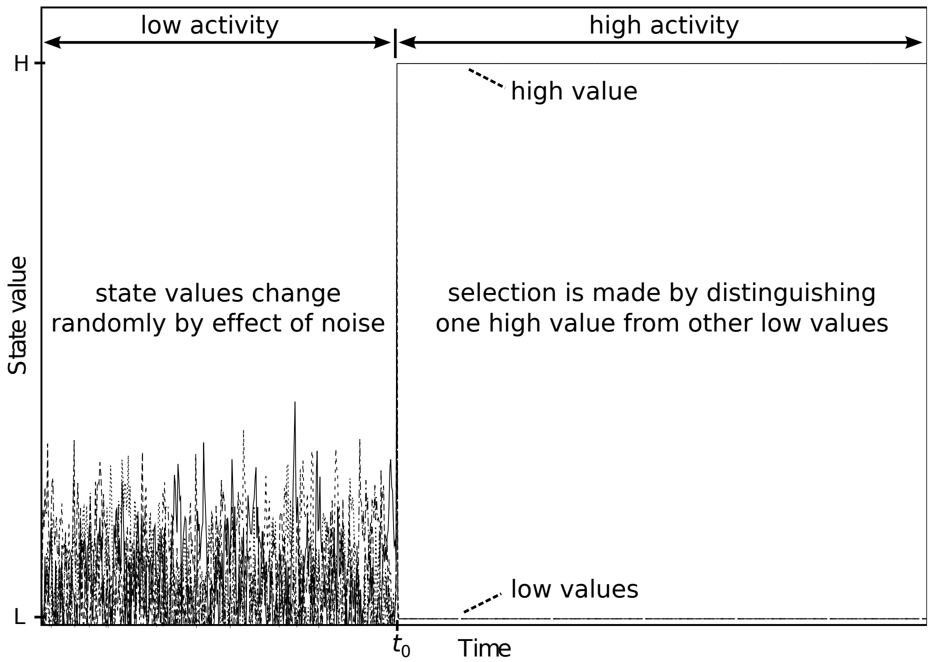


Figure 2.2: Dynamics of M values from attractor selection model

and the solution can be obtained by selecting the maximum m_i value. On the other hand, in case of low activity α, Eqn. (2.8) gives a random $\vec{m}$ where each element m_i has roughly the same value. This permits that a new attractor, which is now more suitable for the current conditions, is easily switched only through the small effect of noise. According to this approach, the appropriate selection can be adaptively made.

The dynamics of state values from Eqn. (2.8) is illustrated in Figure 2.2 where each curve represents the state value for each neighbor. During the time that the activity α is low (prior to t_0), each state value m_i receives more effect from noise and its value changes randomly. Then, at t_0 a solution is found and the activity α increases to the maximum, which is used for selection of the next hop neighbor. As a result, the difference between selected value and not selected values increases and one high value and a set of $M-1$ low values are distinguished from each other, which indicates that the system reaches a suitable attractor.

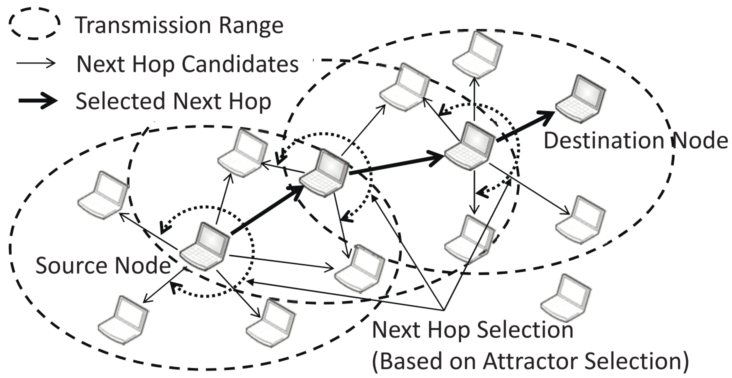
The attractor selection mechanism is feedback-controlled, so if any certain link becomes congested, the evaluated activity of the link will decrease and that neighbor will be less likely to be selected. Therefore, any bottleneck problem will be automatically solved without any external control.

2.4 Our Routing Protocol

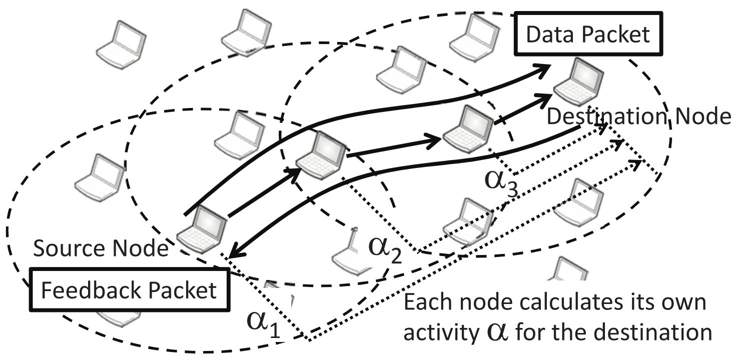
In this section, we explain our protocol MARAS in detail. The overview of the general behavior of MARAS is shown in Figure 2.3. MARAS reactively establishes a route to the destination. Using the information in the route entry and attractor selection model as shown in Figure 2.3(a), the appropriate next hop is selected and the path to the destination can be found. MARAS uses the feedback packets to evaluate the path that the data packets have taken, based on which each node along the path calculates the activity as shown in Figure 2.3(b). When there is a link error as illustrated in Figure 2.3(c), the activity decreases and it triggers a random walk of data packets to search for an alternative path to the destination. Once a good path to the destination is found, the next hop selection is no longer random and the routing will be deterministic and stable again. Note that because of the use of feedback packets, we assume a bidirectional connectivity between each neighbors as in most other MANET routing protocols. However, MARAS can also operate in a network containing unidirectional links as we will explain later in Section 2.4.4.

2.4.1 Route Establishment

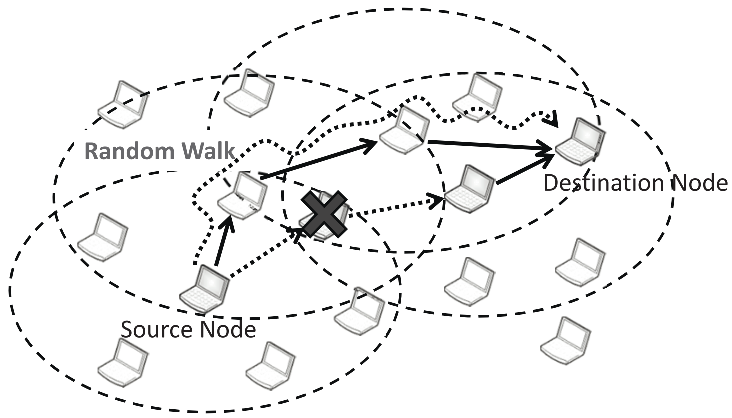
We adopt the broadcast route discovery mechanism from AODV [51] and make a few modifications. In our protocol, when a node has data to send but no route is available, a *route-request packet* (RREQ) is broadcast from the source node and re-broadcast by other nodes until it reaches the destination. When the RREQ packet arrives at the destination or the number of traveled hops exceeds the specified *Time-To-Live* (TTL), a *route-reply packet* (RREP) is generated and forwarded in unicast manner via the memorized reverse path to the source. When a node on the path receives the RREP packet, it sets up the route entry for the destination of the data packet, which favors the selection of the previous



(a) Next hop selection



(b) Activity calculation



(c) Route recovery

Figure 2.3: Overview of MARAS

hop of the RREP packet. Moreover, the route entry is marked with a maximum activity $\alpha = 1$ because of the availability of a route to the destination. Once the RREP arrives at the source node, it starts sending data packets.

Due to the random selection of the next hop in the low activity case, a node which is not on the path still occasionally receives a data packet. If the current node has no route entry for that destination, then it will set up a new route entry by a routing vector which contains random state values for every neighbor node. Consequently, based on the new route entry, the node randomly selects the next hop by the effect of noise. When forwarded data packets occasionally reach the destination, a feedback packet is sent back to the source by retracing the path used by data packet. Along the path, when each node receives a feedback packet, the attractor selection mechanism is applied to the routing information of that node, see Figure 2.3(c).

2.4.2 Routing Information

The routing information stored at each node in the route entry are as follows.

- *Destination address* is used for looking up the corresponding route entry when a data packet is received.
- *Neighbour vector* $\vec{n} = (n_1, n_2, \ldots, n_M)$ is a list of neighbor addresses, maintained by HELLO packets like in AODV where M is the number of neighbors.
- *Attractor selection vector*, called *routing vector* $\vec{m} = (m_1, m_2, \ldots, m_M)$ has the same dimension as the neighbor vector and contains the *state values*, where each state value is mapped to a neighbor in the neighbor vector. These state values are used to determine the next hop of each data packet.
- *Activity* α reflects the current goodness condition of the path to the destination. The routing vector is updated according to this value, allowing the next hop selection to adapt to the current condition.

- *Precursor list* contains pairs of the address of the source node and the address of the most recent neighbor that forwarded the data packet originating at that source node to the destination via the current node.

- *Feedback window* is a sliding window where each frame contains the traveled hop count of the feedback packet, originating at the destination and sent via the current node. Each frame is added to the feedback window on the reception of a feedback packet, kept for a *window interval* of $T = 1.0\,\mathrm{s}$, and then discarded to avoid using outdated information.

2.4.3 Data Packet Forwarding

Next hop selection in data packet forwarding is controlled by the attractor selection mechanism. Using attractor selection, MARAS selects the neighbor as the next hop, which has the *maximum state value* m_{max} in the routing vector, because the maximum value shows the highest chance of that neighbor on delivering the data packet to the destination. The data packet is forwarded to this next hop and the process repeats itself until it reaches the destination.

The concept of attractor selection along with the maximum state value favors the next hop selection in a way that MARAS will keep selecting the same next hop as long as the activity is high. When the activity drastically decreases, which reflects an undesirable condition, e.g., connectivity loss or congested channel, state values will approach each other. Then, the effect of noise changes the state values and allows a different neighbor to be chosen. In such conditions, other routing protocols, e.g., AODV, DSR, issue additional control messages to find a new route. However, MARAS is able to recover from such conditions without using additional explicit control messages.

2.4.4 Route Maintenance

MARAS maintains the routes as long as they are being used and removes unused routes after a certain period of time to save the resources required to maintain them. In order to keep the routing information updated, MARAS uses feedback packets to learn the current condition of the network. Every time when the activity changes at reception of a feedback packet or when it decays, Eqn. (2.8) is evaluated and the routing vector is updated. The route maintenance mechanisms are explained in this section.

Feedback Packets

Each time a data packet arrives at the destination, a new feedback packet is generated and sent back to the source. The feedback packet contains only the source and the destination addresses and headers of underlying protocols. It utilizes the memorized previous hop in the precursor list at each intermediate node to take the most recent route back to the source and to avoid getting lost. During its journey, its traveled hop count information is kept in the *feedback window* of each intermediate node for the purpose of activity calculation. The hop count information in the feedback window is deleted after the *window interval* T to avoid using outdated information.

In our previous study, the feedback packets were sent back to the source in a unicast manner, which caused a high overhead. Since feedback packets can be used to update the path to the destination, it would be more beneficial if all the neighbor nodes, not only the target node of the feedback packet, receive them along the path. Moreover, we also embed the activity of the current routing vector in the feedback packet to allow the neighbor nodes to utilize this activity as a probabilistic value to decide whether to send out the HELLO packets or not. When the activity is high, candidates other than the currently selected next hop are not necessary. Hence, the neighbor nodes can send out less HELLO packets to reduce its bandwidth consumption. Moreover, the smaller number of neighbor nodes in case of high activity is favorable because there will be less unnecessary random walk effects. As a result, MARAS with broadcast feedback packets should have a higher performance than the original one and this is shown through simulations in Section 2.5.5.

Activity Calculation

The activity of each routing vector is calculated upon the arrival of a feedback packet at time t_0 based on the traveled hop count of the most recent feedback packet and the minimum traveled hop count in the feedback window:

$$\alpha(t_0) = \frac{\min_{t_0 - T < t \leq t_0} \{w(t)\}}{w(t_0)}, \tag{2.9}$$

where $w(t)$ is the traveled hop count of the feedback packet arriving at time t. Moreover, the activity remains the same until the next activity recalculation, which occurs by either the next arrival of feedback packet or the activity decay mechanism (see the next section).

The activity changes according to the hop count to the destination in the range between 0 and 1. If the hop count to the destination becomes larger, then it means that the current path to the destination is no longer appropriate and an attempt to find a better path should be made. Therefore, the activity will decrease and the effect from noise will induce a random walk. On the other hand, once a shorter path is found, $\alpha(t)$ will instantly become 1. MARAS will continue using this new path until another change occurs in the network.

Activity Decay and Routing Vector Update

When a route is broken, data packets cannot arrive at the destination and there will be no feedback packet returning to the source. In such condition, the activity must decay to let the system escape from a stalling condition. In case of unidirectional links, the absence of feedback is treated similarly to the case of a broken link. This implies that paths consisting of bidirectional links are preferably chosen.

In our protocol, the decay process is periodically performed over interval $\tau = 1.0\,\mathrm{s}$, i.e., at $\tau, 2\tau, \ldots, k\tau$. Given the current time t, where $k\tau \leq t < (k+1)\tau$, and the most recent feedback packet arrival at t_0, we use the simple activity decay equation of the stored activity:

$$\alpha(t) = \begin{cases} \alpha(t_0) & \text{if } t - \tau \leq k\tau < t_0 < t \\ \alpha(t_0) - \delta & \text{if } t - \tau \leq t_0 \leq k\tau < t \\ \alpha(t - \tau) - \delta & \text{otherwise,} \end{cases} \tag{2.10}$$

where the decay constant $\delta = 0.1$ is used for the current implementation. The activity decay mechanism is performed regardless of the feedback packet arrival. Therefore, when there is no incoming feedback packet, the activity will continuously decay and the routing vector will be updated using the decayed activity.

Attractor Selection-based Route Recovery

In MARAS, data packets occasionally take a random walk looking for a new path to the destination due to the noise term and the random routing vector as explained in Section 2.4.1. This behavior of data packets inherently contributes to route recovery. As such, MARAS does not require any specific mechanism designated for route recovery.

However, using the data packet as a route recovery packet has the drawback of possibly lower delivery ratio due to loss of data packets. From the viewpoint of delivery ratio and overhead, the number of hops should be limited to as small number as possible. On the other hand, from the viewpoint of route recovery, the number of hops should be as large as possible to increase a chance of finding a new route. This trade-off is considered in our implementation and we introduce the *random walk range* ρ and *random walk threshold* θ for this purpose. Whenever the activity is lower than the random walk threshold θ, the TTL of data packets is limited to the random walk range ρ instead of the default TTL to avoid the negative effect of too long paths and infinite loops as MARAS has no explicit loop-avoiding capability. We use empirical values $\rho = 10$ and $\theta = 0.6$ throughout this study.

2.5 Evaluation

We evaluate MARAS with the network simulator QualNet and compare its performance with that of AntHocNet [22] and AODV [51]. We use the code of AntHocNet from the developers available at [26] and the implementation of AODV in QualNet, which uses HELLO packets, local route repair, and intermediate node reply features. We choose AODV as it is a well-known reactive routing protocol and our protocol is also reactive. Moreover, AODV shows a very good performance in the evaluation of [4]. The other protocol, AntHocNet, is an ant colony optimization-based ad hoc routing protocol and is a hybrid routing protocol that uses both reactive route establishment and recovery together with a proactive route maintenance. We choose AntHocNet as it is also a biologically inspired routing protocol and the routing decision is made stochastically.

The evaluation section is separated into four parts. First, we evaluate MARAS and compared protocols in a failure scenario, in which we aim to show its robustness against various levels of failures. Then, we move on to the evaluation of scalability in terms of node density. Afterwards, the adaptability of each protocol over the amount of traffic load is evaluated. Lastly, we evaluate the protocols under a mobility scenario. In each part, we use the parameters varying from the normal values to the extreme ones because our objective is to study the survivability of our protocol in extreme cases, i.e., unexpected events. In case of unexpected events, fine-tuning of parameters is not possible for any protocol but only its adaptability and, in the worst case, survivability features can be relied on in such conditions. Therefore, we kept the default values of parameters of each protocol, including ours, throughout the evaluation to observe such features of each protocol.

2.5.1 Common Simulation Settings

In this section, we describe the common simulation settings and parameters that we use in the non-mobility cases (Sections 2.5.2–2.5.4).

The area of the evaluation scenario is $1500 \times 1500\,\mathrm{m}^2$. We use the uniform node placement in QualNet, which divides the simulation area into grids and uniformly places a node

Table 2.1: Simulation parameters of MARAS

Category	**Parameter Name**	**Value**
Attractor selection	High value β	10
	Activity exponent γ	3
Activity calculation	Window interval T	1.0 s
	Decay constant δ	0.1
	Decay interval τ	1.0 s
Routing	Random walk threshold θ	0.6
	Random walk range ρ	10

within each tile. Each node in the simulation uses the IEEE 802.11b wireless module with data rate of 2 Mbps which is the common configuration in many other protocol evaluations [22, 65]. The free-space radio propagation model is used here to avoid additional complexity of interpreting our results and the approximate radio range is 510 m. Additionally, we assume an infinite wireless interface buffer at each node and CBR traffic is used with UDP as a transport layer protocol to avoid observing effects from the congestion control mechanisms of TCP. We use CBR of 8 kbps which sends out 10 packets per second. The simulation time is 1000 s where the traffic generation starts and ends with the simulation. Please note that all results shown here are average values from 100 simulation runs.

We consider two performance metrics in this evaluation: delivery ratio and transmission overhead. The *delivery ratio* is the ratio of successfully delivered data packets at the destination over the number of all data packets sent from the source. The *transmission overhead* is the ratio of the sum of all unicast and broadcast transmissions in the network for the whole simulation to the number of the successfully delivered data packets. This metric reflects the amount of network load inflicted by the delivery of each data packet.

The parameters of MARAS are summarized in Table 2.1 with their default values. The parameters of AntHocNet are set according to the configuration file provided with the code in [26]. The other parameters of AODV and MARAS, which are not given here, are default values according to their implementations in QualNet.

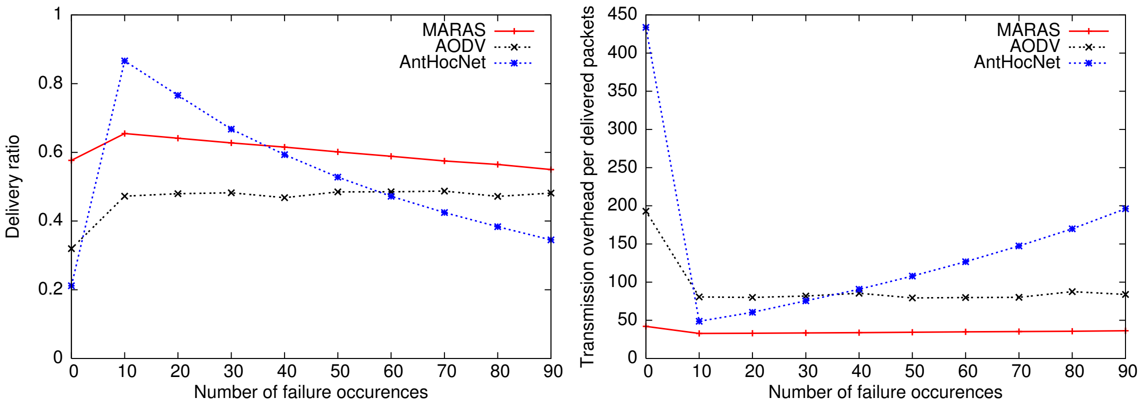
2.5.2 Evaluation against Failures

In this section, we briefly revisit the evaluation results against failures of MARAS from [7]. We use a failure model to simulate topology changes caused by joining and leaving nodes. We force 25% among all 256 nodes to fail at the same time by switching their wireless interfaces off using the API available in QualNet. Consequently, link failures occur and the route recovery performance can be evaluated. Failing nodes are randomly selected among all nodes excluding the source and the destination.

The failure interval is calculated by dividing the total simulation time by the number of failure occurrences, which ranges between 100 s in case of 10 failures to 11.11 s in case of 90 failures. We decrease the failure interval when the number of failures increases to maintain the same number of active nodes and to evaluate the effect of higher failure frequency. The first group of nodes is forced to fail at 0 s. After a failure interval, the previously failing nodes recover and a new group of randomly failing nodes is iteratively selected. Note that the value 0 means no failure occurrences.

In this scenario we use two source/destination pairs. The two source nodes are those closest to the lower left corner of the simulation area and the two destination nodes are those closest to the upper right corner. The purpose of the selection of source and destination pairs is to observe the interference caused between both sessions.

In Figure 2.4(a), MARAS achieves higher delivery ratio and lower transmission overhead per successfully delivered packet than AODV for all cases. Although the delivery ratio of AntHocNet is higher under less dynamic conditions with low failure occurrences, it becomes much lower and drops faster over an increasing number of failure occurrences. The reason of the decreased delivery ratio can be explained by Figure 2.4(b) where the overhead of AntHocNet increases much faster than AODV and MARAS. The increased overhead here is the effect of proactive broadcast forward ants and the slow adaptation to changes of AntHocNet because it relies too much on the old pheromone information. Therefore, it is sufficient to say that MARAS is more robust and adaptive against large number of failures and changes in topology than AODV and AntHocNet.



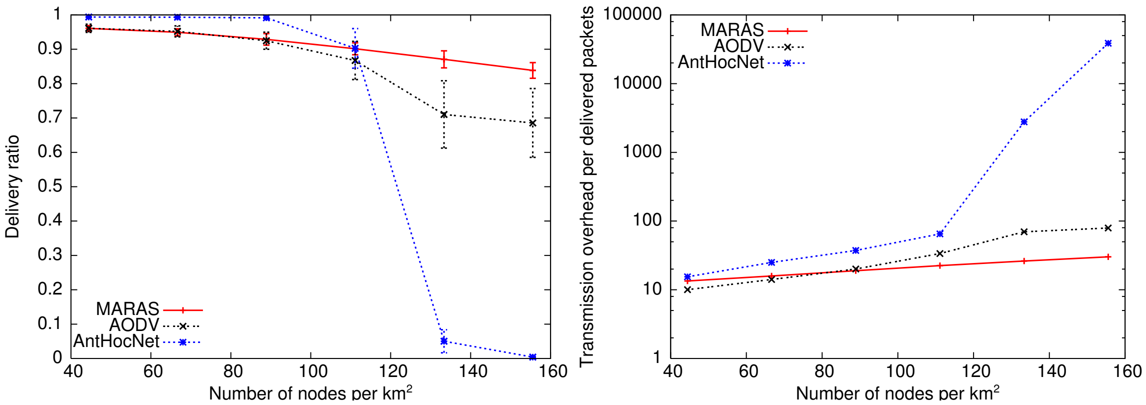
(a) Delivery ratio (b) Transmission overhead

Figure 2.4: Evaluation results against number of failure occurrences

2.5.3 Evaluation against Node Density

In the previous simulation scenario, we have seen that MARAS is more robust to failure than AODV and AntHocNet. However, the significant change of performance between 0 and 10 failures in Figure 2.4 raises the question whether the number of nodes has a direct relation to the performance of the protocols. Therefore, we perform further evaluations against node density over the same terrain size. Since node failures can be interpreted as thinning the density, all nodes operate failure-free in the following evaluation. Moreover, we randomly select two source nodes from 25% of the nodes in the leftmost area and connect them with two random destinations among 25% on the rightmost area. This is done to ensure that source and destination node are not located in direct transmission range of each other.

In this scenario, the number of nodes is varied from 100 to 350 nodes in the $1500 \times 1500\,\mathrm{m}^2$ area which corresponds to each point on the x-axis of Figure 2.5. Note that the error bars in Figure 2.5(a) are the confidence intervals for 99.95%. From Figure 2.5(a), it can be clearly seen that MARAS has a lower delivery ratio in the low node density cases. However, MARAS can nearly maintain this performance when the node density increases because MARAS creates less interference than AODV and AntHocNet as it does not use broadcast



(a) Delivery ratio

(b) Transmission overhead

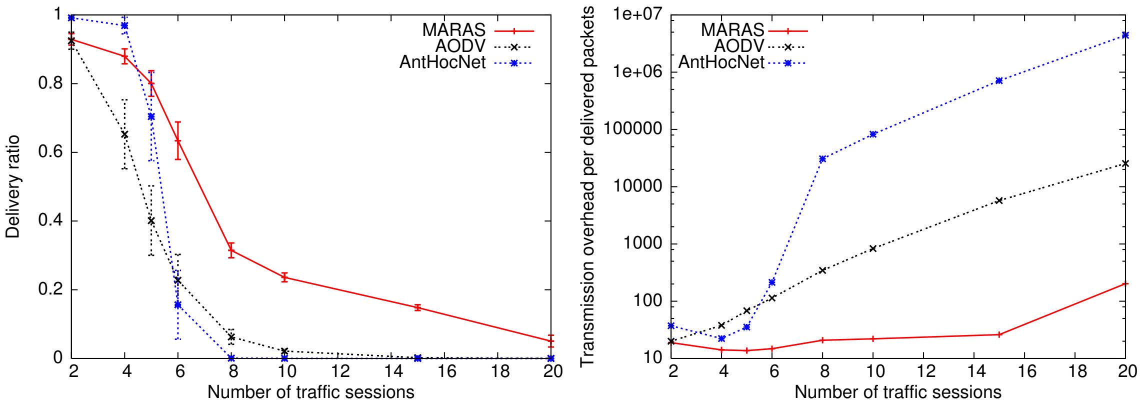
Figure 2.5: Evaluation results against node density

control packets. Moreover, the overhead of MARAS does not increase much as the node density increases, as shown in Figure 2.5(b). On the other hand, AntHocNet, which can achieve higher delivery ratio in low node density cases, cannot maintain its performance due to its much higher overhead requirement when the node density increases. The performance of AODV lies between MARAS and AntHocNet in both delivery ratio and overhead. From these results, we can say that MARAS is more resilient to increasing node density than AODV and AntHocNet.

2.5.4 Evaluation against Number of Traffic Sessions

In MANETs, it is not possible to directly control the traffic or network admission. In the previous section, we investigated how each protocol behaves when there are too many nodes joining the network. In this section, we investigate how each protocol operates under heavy traffic conditions. The source and the destination pairs are randomly selected in the same way as described in Section 2.5.3. The results from varying the number of traffic sessions from 2 to 20 sessions with 256 nodes are shown in Figure 2.6.

In Figure 2.6(a), the delivery ratio of AntHocNet is very high when the number of traffic sessions is low and the delivery ratio of AODV and MARAS is also sufficiently high in such cases. However, as the number of traffic sessions increases, it can be observed that



(a) Delivery ratio (b) Transmission overhead

Figure 2.6: Evaluation results against number of traffic sessions

the delivery ratio of MARAS degrades slower than that of AODV and AntHocNet, which means that MARAS can tolerate more traffic load. In heavy traffic conditions, AODV and AntHocNet keep trying to use the shortest path to the destination because they can not detect that it has already been congested. Therefore, AODV and AntHocNet suffer from transmission errors, lower delivery ratio, and consequently the high overhead as shown in Figure 2.6(b). On the other hand, MARAS can avoid such problems because the activity decreases when data does not arrive at the destination, which allows MARAS to be able to find another path to the destination. As a result, MARAS can still keep delivering data packets in a small portion while AODV and AntHocNet are totally unable to provide routing service under such intense conditions. Considering this situation as an emergency condition where every node tries to send out data at the same time, MARAS can survive in such a condition to deliver a small amount of emergency messages or controlled messages and does not completely break down. Hence, MARAS can be regarded as more resilient to high load conditions than AODV and AntHocNet.

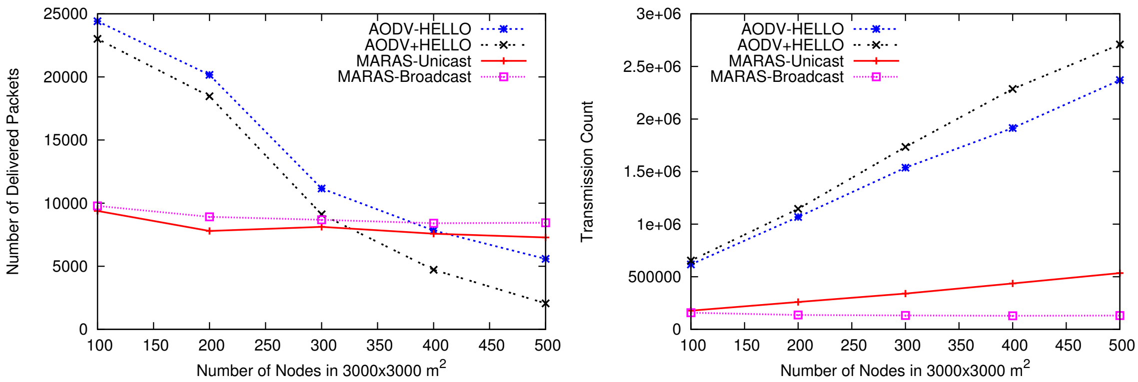
2.5.5 Evaluation under Mobility Scenarios

Mobility is a crucial characteristic of MANETs, therefore, we perform an additional evaluation of MARAS against AODV under mobility scenario in this section. Under this mobility scenario, we evaluate not only the previous version of MARAS (denoted by MARAS-Unicast), but also the newly improved MARAS with broadcast feedback packet (denoted by MARAS-Broadcast), as mentioned in Section 2.4.4. For AODV, we use two different settings, one using HELLO packet (denoted by AODV+HELLO), and the other that does not use HELLO packet (denoted by AODV-HELLO).

The random waypoint mobility model is used in this scenario, where the maximum speed varies from 5, 10, to 20 m/s, which correspond to a walking speed, a bicycling or slow vehicles speed, and vehicular speed, respectively. The minimum speed of 1 m/s has been used with 0-second pause time. The simulation area is $3000 \times 3000\, m^2$ with the number of nodes varying from 100 to 500 nodes, initially distributed uniformly in the whole simulation area. The simulation duration is 1000 seconds. There are 5 sessions of CBR traffic with 40 Kbps data rate, in which the source and the destination pairs are randomly selected with a constraint that both are not initially in each other's transmission range.

The metrics that we consider here are the number of delivered packets, which reflects the network capacity, and the network-wide transmission count, which reflects the amount of overhead induced by the routing protocol.

The obtained results of each case of maximum speed are very similar. Therefore, only the results from the case of 20 m/s maximum speed are shown in Figure 2.7. In this figure, each point in the figures is the average result of 50 simulation runs. When the number of nodes increases with the fixed amount of traffic, it can be seen that the number of delivered packet of AODV adaptations decreases drastically, see Figure 2.7(a). This result shows that AODV cannot cope with the increased amount of network nodes, which is likely to occur in everyday situations, for instance, in downtown areas, in crowded halls, or while commuting. On the other hand, MARAS, regardless its inferior number of delivered packets, can maintain its performance through the changes in the number of nodes. The



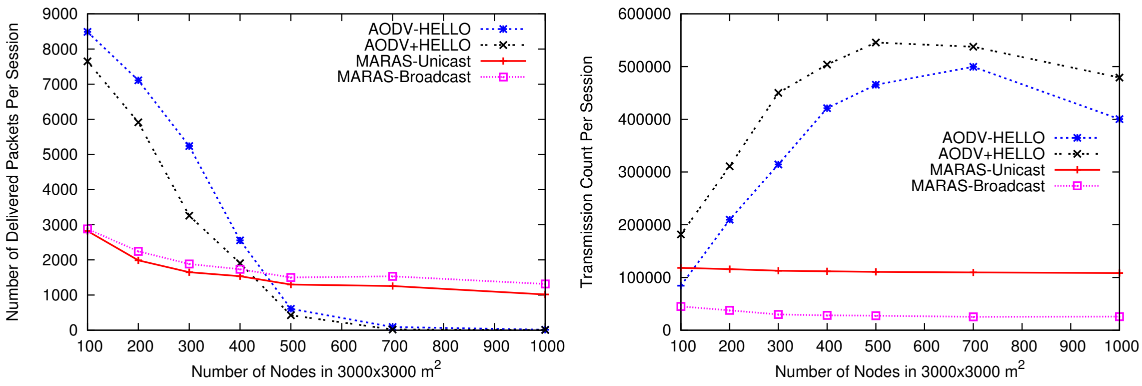
(a) Number of Delivered Packets (b) Transmission overhead

Figure 2.7: Evaluation results under mobility scenario

reason behind this is shown in Figure 2.7(b), where it can be seen that MARAS, which uses less control message flooding, has much lower transmission overhead than adaptations of AODV. Please also note that the newly improved adaptation of MARAS noticeably outperforms the original one in both terms of the higher network capacity and the lower overhead.

Additionally, we evaluate MARAS against AODV in an even more realistic scenario, where the amount of traffic also increases when the number of nodes increases. This scenario can be found in networks managed by telecommunication providers where the amount of traffic is estimated by a certain percentage of the number of participating nodes. Due to the constraints of MANETs which have limited bandwidth, we estimate that there is only 1 pair of active users per 100 users (1 percent). Other than the traffic parameter, the settings and the metrics used are the same as the above scenario. The results of this scenario are shown in Figure 2.8 where each point is an average of 10 simulation runs and the value on the y-axis is normalized per session for ease of interpreting the results.

In Figure 2.8(a), we can see a similar tendency as in Figure 2.7(a). However, the decreasing slope of the number of delivered packets of AODV is steeper than in Figure 2.7(a), which shows that the parameter that affects this metric is not only the number of nodes but also the number of traffic sessions. We also extended the x-axis further than



(a) Number of Delivered Packets

(b) Transmission overhead

Figure 2.8: Evaluation results against both increasing node density and traffic

in Figure 2.7 to 1000 nodes, which confirms that AODV cannot survive in this high dense and heavy traffic. To the contrary, MARAS can maintain its performance and especially MARAS-Broadcast has a very low transmission overhead compared to the other comparison protocols, see Figure 2.8(b).

2.6 Summary

In this chapter, we described MARAS, a resilient routing protocol for MANETs inspired by the biological attractor selection mechanism. This mechanism is formulated by non-linear stochastic differential equations with a control factor, called activity, which influences the degree of randomness in the selection process. Feedback packets are used to evaluate the route that each data packet takes and to update the activity at each node in the route by using the hop count information, allowing the route to react to changes in the network without creating additional control overhead on changes. As a result, MARAS is adaptive and resilient to failures and can operate under high node density or heavy load conditions.

Our focus of this study is on the survivability of the routing protocol under extreme conditions, in other words, unexpected events. We strongly believe that it is necessary for the routing protocol to operate even though at lower performance level instead of ceasing its operation in such condition. Therefore, we evaluated MARAS against AODV and AntHocNet in extreme conditions, i.e., frequent failures, high node densities, heavy traffic, mobility, and the combination of them. The evaluation results show that MARAS can achieve a higher delivery ratio and a lower overhead than the other well-studied routing protocols, AODV and AntHocNet, in such conditions.

MARAS should be used in scenarios that are generally appropriate for reactive routing protocols, i.e., bandwidth-scarce and high dynamics scenarios. Reactive routing protocols, like MARAS, generally cannot handle many concurrent traffic sessions due to multiplicative overhead required to maintain each session. However, based on evaluation results, it can be seen that MARAS can handle more traffic sessions than other reactive routing protocols, i.e., AODV and AntHocNet. Moreover, MARAS can operate using a range of parameters without the needs of fine-tuning. This characteristic is useful in comparison to zone-based or cluster-based routing, which usually needs such fine-tuning effort. In conclusion, our proposal is superior than existing protocols like AODV and AntHocNet under extreme condition without sacrificing much performance under normal conditions.

In the future, we would like to study the behavior of MARAS under more realistic fading models, e.g., two-ray model, Rayleigh model, in which we expect similar results as long as key parameters, such as, number of nodes, node density, relative mobility between nodes, the distance of radio communication, and the distance of radio interference, are kept the same.

Chapter 3

Traffic Distribution over Multiple Paths with Attractor Perturbation

Preface

Portions of this chapter were previously published as 'Noise-assisted traffic distribution over multi-path ad hoc routing,' in Proceedings of 4th International Symposium on Applied Sciences in Biomedical and Communication Technologies (ISABEL) where copyright is held by ACM, and are under reviewing process (conditionally accepted) as 'Noise-assisted Concurrent Multipath Traffic Distribution in Ad Hoc Networks,' for publication in IEICE Transactions on Communications (Special Section on Progress in Information Network Science) where copyright is held by IEICE, and have been reproduced with permission.

3.1 Introduction

Computer network architectures and their protocols have become increasingly sophisticated over time through addition of many features to support new applications, such as multimedia streaming, voice-over-IP (VoIP), or online gaming. These new protocols often require a careful fine-tuning of parameter values to operate at their best performance. However, different traffic conditions may require different settings of protocol parameters that need to be manually readjusted. Since the total number of possible situations occurring in the real world is too large to be handled by preprogrammed sets of definitions, it is necessary that new networking mechanisms are designed in a flexible and adaptive manner to cater for any changes in the environment. Reliability of the communication channel is particularly important for wireless networks due to the limited available wireless spectrum and fluctuating channel characteristics. Additionally, in mobile ad hoc networks (MANETs), a specific type of infrastructure-less wireless network, the nodes can be mobile which leads to sudden changes in connectivity and network topology.

Beside conventional approaches that have been proposed to improve adaptability in ad hoc networks, also concepts based on biological mechanisms have been proposed [24, 48] for self-organized control since they are able to provide greater robustness and adaptability to external influences. The underlying idea is to derive a protocol that is based on the model of a natural phenomenon. For example, swarm intelligence is a concept where individual agents mimic the behavior of insect swarms, e.g. ants or bees, during foraging and it has been successfully applied to routing problems [13]. Firefly groups perform a distributed synchronization of their flashing behavior and this was applied to synchronization in sensor networks [61]. Reaction-diffusion describes the chemical dynamics of morphogens in the development of stripes or spots on animal furs. Based on the reaction-diffusion dynamics the coding rate for camera sensor networks can be controlled [68].

Since biological systems are often described as dynamic systems, they rely on a mathematical formulation given as differential equations. In dynamic systems, attractors describe the states to which the system evolves over time. In the past, we studied the concept of attractor selection, which is based on the dynamics found in gene expression [33] and has been previously also applied to tackling problems in communication networks [7, 40]. In this chapter, we apply a similar biological mechanism called attractor perturbation (AP), which is derived from the fluctuation-response relationship observed in an experiment on the evolution of functional proteins in a cell [54]. A previous application of AP to wireless networks can be found in [39, 64].

In this chapter, we focus on bandwidth improvement and end-to-end delay minimization in ad hoc networks. In terms of bandwidth improvement, one of the most common approaches is using multiple paths in the same or across different media (multihoming). To enable the ability to utilize multiple paths concurrently, there are a few existing work in both wired, e.g., Opportunistic Multipath Scheduling (OMS) [17], and wireless networks, e.g., Concurrent Multipath Transfer (CMT) [12, 31] and Adaptive Load Balancing Algorithm (ALBAM) [74]. However, most existing control methods require a full knowledge of the current network status, e.g., queue length on each node, which is difficult to obtain or requires frequent probing which causes bandwidth degradation. Therefore, we consider applying AP to concurrent multipath traffic distribution to improve the available bandwidth while utilizing the AP relationship to predict the outcome of traffic adjustment and attempt to also minimize the end-to-end delay at the same time.

The rest of this chapter is organized as follows. Based on the biologically-inspired mechanisms explained in Section 1.2.4, we describe our problem scenario of traffic distribution in multi-path routing and investigate the applicability of AP on eht problem in Section 3.2. The proposed concurrent traffic distribution algorithm is explained in Section 3.4. Then, the evaluation results from simulation are presented and discussed in Section 3.5. Finally, we conclude this chapter and describe future work.

3.2 Motivation and Problem Formulation

The advantage of using multiple paths is that if one path breaks due to failures at intermediate links or nodes, the other paths can still be maintained. Furthermore, using multiple paths permits a better balancing of loads by distributing traffic more evenly in the network. Especially, if nodes in an ad hoc scenario are operated by batteries, this may lead to reduced energy consumption of intermediate nodes. Finally, using multiple paths concurrently can improve the total available bandwidth in the network.

In today's wireless networks, it becomes common that participating devices can connect to more than one radio access technologies (RAT) and even within the same RAT, there are multiple possible separated channels to use. Therefore, the concept of multipath can now be extended to multi-channel and multi-homing in heterogeneous wireless networks. Even though our current work is focused on ad hoc networks, our concept of path is applicable to traffic allocation over multi-channel and multi-homing scenarios. The allocation granularity, which describes the unit of information allocated to each path, is also of great importance [49]. Coarse granularities, such as per-connection or per-flow, tend to reduce the management overhead, but are not as flexible as small granularities, e.g., per-packet, since these permit a better distribution of traffic. However, per-packet granularity may require reordering at the destination, if the latencies differ too much among paths.

Eqn. (1.2) reveals that the difference in the average of the variable x before and after applying a change to the force a is linearly proportional to the amount of change in a and the variance of the variable x prior to the change. Therefore, one can predict the response to the applied force from the fluctuation of the targeted system. Since the amount of change in a can be seen as controllable, it is possible to adjust the difference in average of x, called *perturbation*, by taking the current variance of x into consideration. Obviously, using the same amount of force Δa to perturb the average of x when the variance σ_a^2 is large will also lead to a larger perturbation, as shown in Fig. 1.2. Based on this relationship, we plan to use this model to estimate the required amount of force required to achieve the desired amount of perturbation.

The requirement of applying the AP model is that the variable x has a Gaussian-like distribution as assumed in model derivation [54]. Theoretically speaking, end-to-end delays in ad hoc networks should follow Gaussian distribution. However, it might not be the case in real world scenarios. Therefore, to confirm the applicability of AP model for our proposal, we have performed simulations to confirm the delay distribution in ad hoc networks and discovered that it is similar to the Gaussian distribution. Moreover, there exists an AP application for traffic rate control to achieve target delay on wired networks [64]. Even though the work was designed for wired networks, it should also be applicable ad hoc networks since AP allows simplifying the system as a black box by observing only the end-to-end variables and overlooking the underlying details. Combining the above two reasons, we decided to use AP for concurrent multipath traffic distribution which could be achieved by performing AP-based traffic rate control on each path, aiming to obtain overall higher bandwidth and lower average end-to-end delay. The minimization problem for the application is formulated in the following subsections.

3.2.1 System Model

In this study, we consider a situation where a source node is connected to the destination node via multiple paths and each path i does not cause interference with one another, as illustrated in Fig. 3.1. This network model covers both ad hoc (or mesh) networks with multiple radio channels and also multihoming system. For the sake of simplicity, we consider only $n = 2$ in this study but the proposed method can be extended to $n > 2$ cases as shown in 3.2.5.

The notations of variables on each path i are as follows:

- Observed end-to-end delay (measurable variable): x_i
- Current traffic rate (controllable variable or force): a_i
- Amount of traffic rate adjustment: Δa_i
- Average end-to-end delay prior to applying Δa_i: $\bar{x}_i$
- Average end-to-end delay after applying Δa_i: $\bar{x}'_i$
- Delivered packet count: n_i

3.2.2 Problem Definition: 2 Paths

Our proposal aims at minimizing the average end-to-end delay of all packets. Using AP, we attempt to minimize the *total delay sum*, which directly corresponds to the average delay of all packets on both paths. The delay sum can be estimated through the product of the expected delay and the adjusted traffic rate on each path.

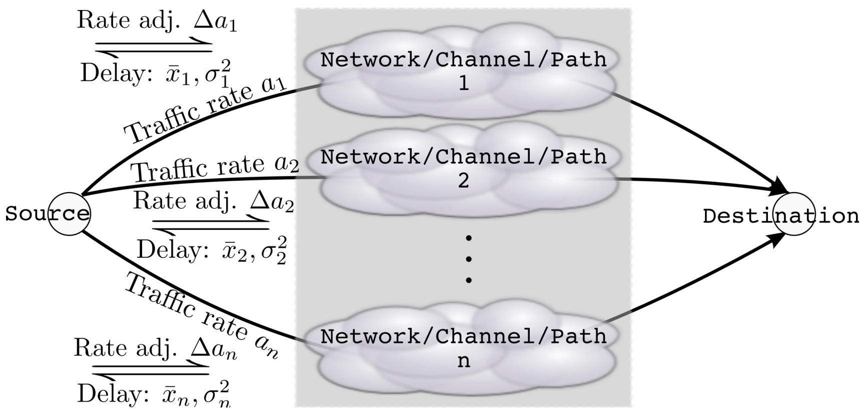


Figure 3.1: Overall system model

According to the AP concept, in case of two paths, we can calculate the expected average delay $\bar{x}'_i$ as follows:

$$\bar{x}'_1 = \bar{x}_1 + b_1\, \Delta a_1\, \sigma_1^2 \tag{3.1}$$

$$\bar{x}'_2 = \bar{x}_2 + b_2\, \Delta a_2\, \sigma_2^2 \tag{3.2}$$

Therefore, we can define a function $f(\Delta a_1, \Delta a_2)$ as an estimation of the average delay after applying traffic rate adjustment Δa_i as follows.

$$\begin{aligned} & f(\Delta a_1, \Delta a_2) \\ & = (a_1 + \Delta a_1)\, \bar{x}'_1 + (a_2 + \Delta a_2)\, \bar{x}'_2 \\ & = (a_1\bar{x}_1 + a_2\bar{x}_2) + \left(\bar{x}_1 + a_1 b_1 \sigma_1^2\right) \Delta a_1 \\ & \quad + \left(\bar{x}_2 + a_2 b_2 \sigma_2^2\right) \Delta a_2 + b_1\sigma_1^2 \Delta a_1^2 + b_2 \sigma_2^2 \Delta a_2^2 \end{aligned} \tag{3.3}$$

Given that $c' = (a_1\bar{x}_1 + a_2\bar{x}_2)$, $c_1 = \left(\bar{x}_1 + a_1 b_1 \sigma_1^2\right)$, $c_2 = \left(\bar{x}_2 + a_2 b_2 \sigma_2^2\right)$, $k_1 = b_1\sigma_1^2$, and $k_2 = b_2\sigma_2^2$, Eqn. (3.3) can be formulated as a constrained optimization (minimization) problem as follows:

$$\begin{aligned}
&\textbf{Minimize} \\
&f(\Delta a_1, \Delta a_2) = c' + c_1\Delta a_1 + c_2\Delta a_2 + k_1\Delta a_1^2 + k_2\Delta a_2^2 \\
&\quad \textit{subject to} \qquad \Delta a_1 + \Delta a_2 = 0
\end{aligned} \tag{3.4}$$

The solution of the minimization problem in Eqn. (3.4) is the amount of the adjustment in traffic rate to be applied to each path in order to achieve minimal average end-to-end delay of all packets. The *subject to* condition is required since the total amount of traffic prior and after adjustment has to be the same.

3.2.3 Lagrangian Optimization

The minimization problem which has the form as in Eqn. (3.4) can be solved using *Lagrangian Optimization*.

The Lagrangian has the general form of

$$L(x^*, \lambda^*) = f(x) - \Sigma_i[\lambda_i(g_i(x) - b_i)]$$

where x^* is the optimal solution of x and λ^* is the penalizing Lagrangian multiplier.

The associated Lagrangian of Eqn. (3.4) is:

$$\begin{aligned}
L(\Delta a_1^*, \Delta a_2^*, \lambda^*) = c' + c_1\Delta a_1^* + c_2\Delta a_2^* + k_1\Delta a_1^{*2} \\
+ k_2\Delta a_2^{*2} - \lambda^* \left(\Delta a_1^* + \Delta a_2^*\right)
\end{aligned} \tag{3.5}$$

$$\frac{\partial L}{\partial \Delta a_1^*} = c_1 + 2k_1\Delta a_1^* - \lambda = 0 \tag{3.6}$$

$$\frac{\partial L}{\partial \Delta a_2^*} = c_2 + 2k_2\Delta a_2^* - \lambda = 0 \tag{3.7}$$

$$\frac{\partial L}{\partial \lambda^*} = -\left(\Delta a_1^* + \Delta a_2^*\right) = 0 \tag{3.8}$$

In the three Eqns. (3.5)–(3.7), there are three unknown variables Δa_1^*, Δa_2^*, and λ^*. Therefore, this optimization problem can be solved and we obtain the optimal amount of traffic rate adjustment Δa_i for each path i to minimize the sum of average delays.

3.2.4 Optimal Solution

According to steps taken in Section 3.2.3, the optimal solution in case of two paths is as follows.

$$\begin{aligned}\Delta a_1^* &= \frac{c_2 - c_1}{2(k_1 + k_2)} \\ &= \frac{(\bar{x}_2 + a_2 b_2 \sigma_2^2) - (\bar{x}_1 + a_1 b_1 \sigma_1^2)}{2(b_1 \sigma_1^2 + b_2 \sigma_2^2)}\end{aligned} \tag{3.9}$$

$$\Delta a_2^* = -\Delta a_1^* \tag{3.10}$$

3.2.5 Extended Model for n Paths

According to the AP concept, in case of n paths, we have:

$$\begin{aligned}\bar{x}_1' &= \bar{x}_1 + b_1\,\Delta a_1\,\sigma_1^2 \\ \bar{x}_2' &= \bar{x}_2 + b_2\,\Delta a_2\,\sigma_2^2 \\ &\vdots \\ \bar{x}_n' &= \bar{x}_n + b_n\,\Delta a_n\,\sigma_n^2\end{aligned}$$

Total delay summation of n-path case can be calculated as follows.

$$\begin{aligned}&f(\Delta a_1, \Delta a_2, \ldots, \Delta a_n) \\ &= (a_1 + \Delta a_1)\,\bar{x}_1' + \cdots + (a_n + \Delta a_n)\,\bar{x}_n' \\ &= (a_1 + \Delta a_1)\,(\bar{x}_1 + b_1\,\Delta a_1\,\sigma_1^2) + \ldots \\ &\quad + (a_n + \Delta a_n)\,(\bar{x}_n + b_n\,\Delta a_n\,\sigma_n^2) \\ &= (a_1\bar{x}_1 + \cdots + a_n\bar{x}_n) + \left(\bar{x}_1 + a_1 b_1 \sigma_1^2\right)\Delta a_1 + \ldots \\ &\quad + \left(\bar{x}_n + a_n b_n \sigma_n^2\right)\Delta a_n + b_1\sigma_1^2\Delta a_1^2 + \cdots + b_n\sigma_n^2\Delta a_n^2 \\ &= \Sigma_i^n \left(a_i\bar{x}_i + (\bar{x}_i + a_i b_i \sigma_i^2)\Delta a_i + (b_i\sigma_i^2)\Delta a_i^2\right)\end{aligned} \tag{3.11}$$

Minimization problem can be formulated similarly to the 2-path case:

$$\begin{aligned}
&\textbf{Minimize}\\
&f(\Delta a_1, \Delta a_2, \ldots, \Delta a_n)\\
&= \Sigma_i^n \left(a_i \bar{x}_i + (\bar{x}_i + a_i b_i \sigma_i^2)\Delta a_i + (b_i \sigma_i^2)\Delta a_i^2\right)\\
&\textit{subject to} \quad \Sigma_i^n \Delta a_i = 0
\end{aligned} \tag{3.12}$$

The associated Lagrangian of Eqn. (3.12) is:

$$\begin{aligned}
&L(\Delta a_1^*, \ldots, \Delta a_n^*, \lambda^*)\\
&= \Sigma_i^n \left(a_i \bar{x}_i + (\bar{x}_i + a_i b_i \sigma_i^2)\Delta a_i^* + (b_i \sigma_i^2)\Delta a_i^{*2}\right) - \lambda^* \Sigma_i^n \Delta a_i^*
\end{aligned} \tag{3.13}$$

$$\frac{\partial L}{\partial \Delta a_1^*} = a_1 b_1 \sigma_1^2 + 2 b_1 \sigma_1^2 \Delta a_1^* - \lambda = 0 \tag{3.14}$$

$$\vdots$$

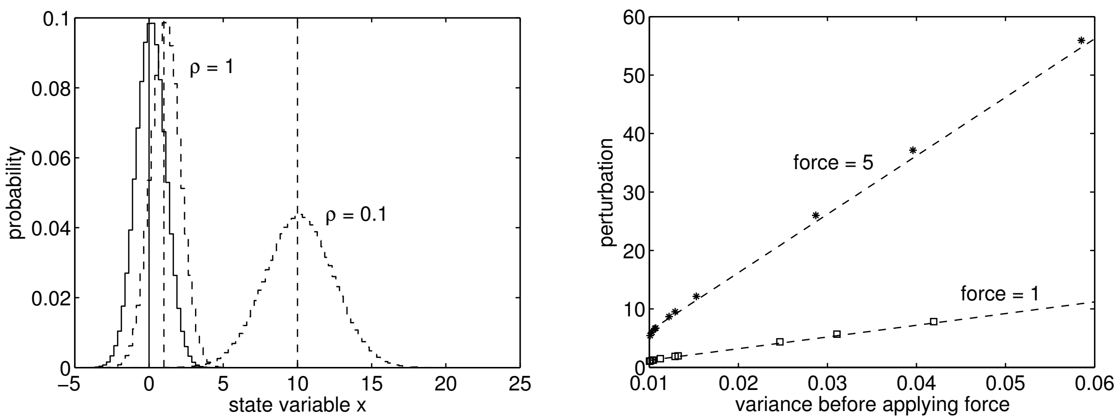
$$\frac{\partial L}{\partial \Delta a_n^*} = a_n b_n \sigma_n^2 + 2 b_n \sigma_n^2 \Delta a_n^* - \lambda = 0 \tag{3.15}$$

$$\frac{\partial L}{\partial \lambda^*} = -\Sigma_i^n \Delta a_i^* = 0 \tag{3.16}$$

From Eqn. (3.13)–(3.16), we can form an augmented matrix as follows:

$$\left[\begin{array}{cccccc|c}
2b_1\sigma_1^2 & 0 & 0 & \ldots & 0 & -1 & -2b_1\sigma_1^2\\
0 & 2b_2\sigma_2^2 & 0 & \ldots & 0 & -1 & -2b_2\sigma_2^2\\
\vdots & \vdots & \ddots & \ldots & \vdots & \vdots & \vdots\\
0 & 0 & \ldots & 2b_{n-1}\sigma_{n-1}^2 & 0 & -1 & -2b_{n-1}\sigma_{n-1}^2\\
0 & 0 & \ldots & 0 & 2b_n\sigma_n^2 & -1 & -2b_n\sigma_n^2\\
1 & 1 & \ldots & 1 & 1 & 0 & 0
\end{array}\right].$$

The above augmented matrix can be solved using row elimination technique.



(a) Normalized histograms

(b) Linear relationship in perturbation

Figure 3.2: Numerically simulated fluctuation and response of dynamic system

3.3 Preliminary Investigation

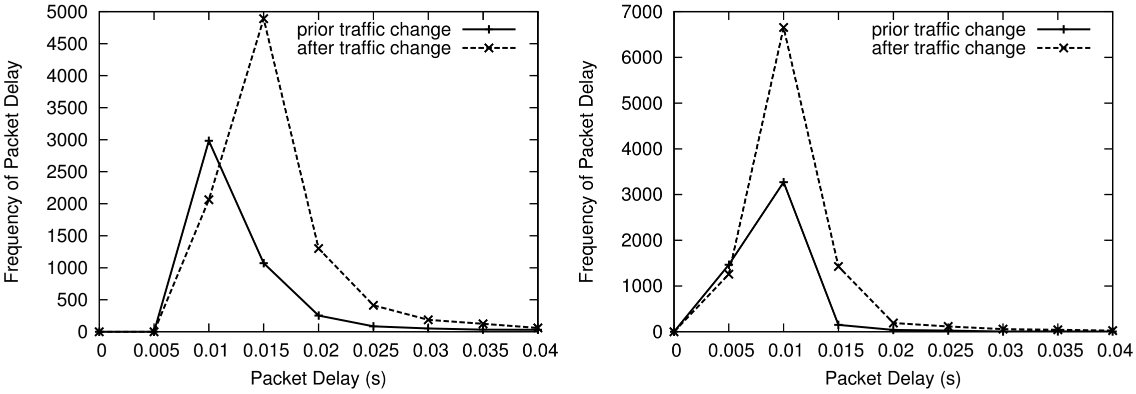
Let us now show the applicability of attractor perturbation as adaptive method for traffic distribution. In this section, we first perform a numerical verification of the AP model by evaluation of stochastic differential equations. Next, we study the behavior of AP based proposal in ad hoc network simulations.

3.3.1 Evaluation of Linearity between Fluctuation and Response

For our numerical evaluation we first show that this attractor perturbation principle actually holds in theory. To do this, we define a simple theoretical attractor model like the one in [38]:

$$\frac{dx}{dt} = -\rho\,(x - x_0) + \eta \tag{3.17}$$

where x is the state variable, ρ is the speed of adaptation, x_0 is the attractor, and η is noise. Fig. 3.2a shows how the initial black histogram at $x_0 = 0$ gets perturbed by the same force, but to different offsets for $\rho = 0.1$ and $\rho = 1.0$. The term ρ controls the softness of adaptation in the dynamic system and represent the internal fluctuations. A smaller ρ leads to slower adaptation of x in Eqn. (3.17) and therefore to a larger variance. Repeating



(a) High variance

(b) Low variance

Figure 3.3: Histogram of packet delays when doubling the traffic rate at 500 s

this experiment for different force values (1 and 5) and random ρ shows us the expected linear behavior, when we plot variance of x on the x-axis and average perturbation on the y-axis of Fig. 3.2b.

We can see from Fig. 3.2 that the linear relationship exists and we can exploit this for our traffic distribution method in multi-path networks. Note that the slope of the lines in Fig. 3.2b corresponds to the constant b term in Eqn. (1.2).

3.3.2 Simulation of Network Traffic

To demonstrate the validity of AP based traffic distribution, we performed simulations of a mobile ad hoc network using the QualNet network simulator. The scenario consists of an area of $1000 \times 1000\,\mathrm{m}^2$, where 25 nodes are uniformly distributed. The simulation duration is 1000 s for each run. There are 5 traffic sessions starting at 1 s: 1 CBR session with packet size of 250 Bytes and 100 ms sending interval, and 4 random traffic sessions with the same packet sizes and exponentially distributed sending intervals with average of 1000 ms serving as background traffic. The underlying routing protocol is MARAS [7].

In order to clearly observe the effect of AP, we change the sending interval of CBR packets from 100 ms to 50 ms at half of the simulation time (500 s) and the results are shown in Fig. 3.3. For several randomly seeded trials, we observe that the variance σ^2 of end-to-end delays during the first 500 s varied depending on the random initial configuration and we could categorize two cases, one with high variance and one with low variance. In Fig. 3.3a the high variance case is shown and the initial average and variance of end-to-end delay before the traffic rate change were $1.653 \cdot 10^{-2}$ s and $1.176 \cdot 10^{-4}$ s^2, respectively. After applying the force to the system through the traffic rate change, the new average delay became $2.009 \cdot 10^{-2}$ s. On the other hand, in the case of Fig. 3.3b which has much lower variance of $1.581 \cdot 10^{-5}$ s^2 than in the high variance case, the average delay changes from $1.108 \cdot 10^{-2}$ s to $1.391 \cdot 10^{-2}$ s. In summary, it can be seen that (i) the average delay can be influenced by the change in traffic rate, and (ii) the perturbation is larger in the case of larger variance.

3.3.3 Discussion

In this section, we introduced *attractor perturbation* (AP), a novel biologically inspired approach which can perform a simplified control of an underlying system. With AP, it is possible to regard the whole underlying system as a black box and perform control based on observed average and variance of the time series of the considered performance metric. According to our evaluation, it can be seen that the concept of AP is feasible for network control in ad hoc networks. Simulation results showed for a single path as well as numerical evaluations of the theoretical differential equation reveal that the fluctuation-response relationship is visible. As a result, this relationship can be used to estimate the optimal amount of traffic change to achieve minimal average end-to-end delays for all packets in order to distribute traffic over multi-path routing as proposed in this chapter.

This section reported on the first steps of our research on traffic distribution in a multi-path ad hoc network. Even though our simplified network simulations were made over only a single-path routing protocol, we can expect similar results in case of multi-path routing if disjoint paths are used. As a result we have further study the multiple paths case in the following section.

3.4 Concurrent Multipath Traffic Distribution

The optimal solution Δa_i^* from Eqn. (3.9)–(3.10) is used in the following Alg. 1, executed at the source every interval ρ (= 5 s in our simulation experiments). Currently, to study a pure behavior of our proposal, we assume that the end-to-end information is known to the source node without an actual measurement. However, a feedback mechanism can be easily implemented to deliver these information to the source node. Since the statistical information is needed only once every execution interval, the overhead can be considered negligible and the actual results should be similar to the simulation results shown in this chapter.

Algorithm 1 AP-based Traffic Distribution

procedure ADJTRAFFIC($\bar{x}_1, \sigma_1^2, a_1, n_1, \bar{x}_2, \sigma_2^2, a_2, n_2$) ▷ Only AP+Com uses n_1, n_2
    for all i **do**
        $\bar{x}_i \leftarrow (\rho(\rho a_i - n_i) + \bar{x}_i n_i) / \rho a_i$ ▷ Delay compensation (AP+Com only)
    end for
    $(\Delta a_1^*, \Delta a_2^*) \leftarrow \text{SolveMinimization}(\bar{x}_1, \sigma_1^2, \bar{x}_2, \sigma_2^2)$
    if $|\Delta a_1^*| > \alpha_{\max} \times (a_1 + a_2)$ **then**
        $\Delta a_1^* \leftarrow \alpha_{\max} \times (a_1 + a_2) \times \frac{\Delta a_1^*}{|\Delta a_1^*|}$
        $\Delta a_2^* \leftarrow -\Delta a_1^*$ ▷ Rate adjustment maximum ratio α_{max}
    end if
    $a_1 \leftarrow a_1 + \Delta a_1^*$
    $a_2 \leftarrow a_2 + \Delta a_2^*$
end procedure

In every iteration of the algorithm, the AP-based protocol (AP-Com) uses the measured average $\bar{x}_i$, the variance σ_i^2, and the current traffic rate a_i to solve minimization problem. The optimal solution is applied to the current traffic rate gradually which is controlled by the rate adjustment maximum ratio α_{max}.

An AP-based protocol with delay compensation (AP+Com) is also proposed here. The delay compensation process serves to maintain throughput in our mechanism with the price of using more information of the delivered packet count n_i on each path i for calculating the number of lost packets. Without delay compensation, AP-Com behaves in favor of lower delay regardless of the delivery performance on each path. In AP+Com, we compensate packet loss assuming that each lost packet has the end-to-end delay equals to ρ. We will show results of AP-based control with and without delay compensation in the evaluation section.

The coefficient b_i is required to solve the minimization problem. This value is crucial for estimating the average delay after applying the traffic rate adjustment and is determined in every iteration based on the current iteration's average delay, the previous iteration's average delay and variance, and the amount of rate adjustment applied in the previous iteration using Eqn. (1.2). In case that there is no traffic rate adjustment on the previous iteration, the default b given in the configuration is used in that iteration and the accurate b_i can be calculated on the subsequent iteration. Note that another research on AP in wired networks [64] has found that the AP-based control does not require a fine-tuning of coefficient b. We will show later in Section 3.5.4 that the same concept also holds in our wireless case.

3.5 Evaluation

To demonstrate the validity of the AP-based traffic distribution, we performed simulations using the QualNet network simulator. We divided the evaluation into two parts, bandwidth improvement and end-to-end delay improvement evaluation.

3.5.1 Comparison Target

In bandwidth improvement evaluation, we compare the performance of the AP-based proposal to two existing multipath transport layer controls: Concurrent Multipath Transfer (CMT) [31] and Multipath Real-time Transport Protocol (MPRTP) [57]. CMT utilizes SCTP [58] protocol to send data concurrently to the destination which has multiple networking interfaces, while MPRTP is an extended version of RTP protocol, which utilizes UDP, to allow scheduling RTP traffic over multiple paths concurrently.

Our proposal is closely related to the recently proposed MPRTP protocol because both are implemented over UDP. The scheduling algorithm of MPRTP uses loss rate, packet size sum, bytes sent, and interval information from RTP's receiver reports (RRs) to estimate the bandwidth of the current path. The original paper explained the bandwidth calculation on path j upon the arrival of i^{th} RR as:

$$RR[j] = \frac{(\sum_{k=HSN_{i-1}}^{HSN_i} sizeof(X_k)) \times (1 - L_i)}{t_i - t_{i-1}} \tag{3.18}$$

where HSN_i, HSN_{i-1} are the Highest Sequence Received reported by the receiver in two consecutive RRs, t_i, t_{i-1} are the reception timestamps in two consecutive RRs, L_i is the reported loss rate in the latest RR, and $sizeof(X_k)$ is the size of each packet k and the sum gives the total bytes sent during the latest report interval.

Moreover, MPRTP uses packet loss information, to categorize paths into congested, mildly congested, and non-congested conditions. The scheduler then continuously assigns a part of traffic to each path, more if it is non-congested and less if it is congested, but keeping the total rate the same. We have implemented the *Algorithm 1* of [57] in QualNet, assuming a perfect knowledge of end-to-end information instead of using real RR packets, and compare its performance with our proposal.

As already mentioned above, CMT is implemented over SCTP which is supported by IETF alongside with TCP and UDP as a general purpose reliable transport protocol with connection-oriented, reliable data transfer, window-based, congestion control and flow control features, similar to TCP. One important feature of SCTP is its built-in multihoming where a connection can be established between a set of IP addresses. However, a standard SCTP uses only a pair of primary IP addresses at a time which does not allow concurrent transmissions. CMT is a modified version of SCTP that allows concurrent transmissions and includes few improvement on fast retransmission, congestion window update, and delayed acknowledgment algorithms. It was found in [12] that the receiving buffer, referred to as *rBuf* in the original paper, can be a performance bottleneck to CMT. Therefore, we only show the best CMT results without such constraint, called *CMT Unlimited*, as a reference in this chapter.

We did not implement our proposal over TCP or SCTP because in TCP scheme, due to

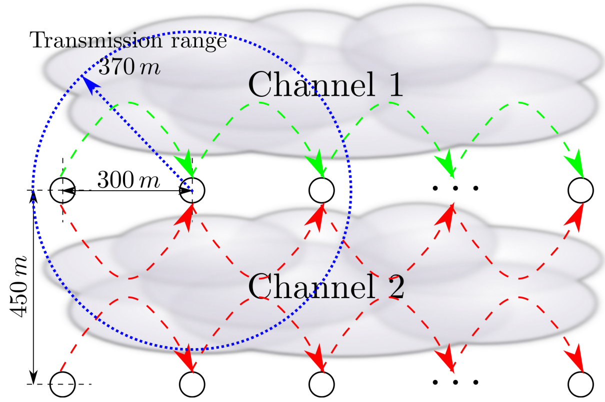


Figure 3.4: Simulation Scenario from [12]

various control mechanisms, e.g., rate control and congestion avoidance control, end-to-end delays do not generally follow Gaussian distribution. There are a few special cases when TCP traffic follows Gaussian distribution [34, 62], however, we leave the investigation on those cases as a future work.

3.5.2 Bandwidth Improvement

We have set up the simulation scenario exactly the same as described in [12], see Figure 3.4. There are two chains of nodes where the distance between nodes on the same chain is 300 m and the distance between chains is 450 m. The transmission range of each node is approximately 370 m where the carrier sensing range and the interference range span further under two-ray pathloss model without fading. The default transmission range of QualNet 5.2 is only 300 m and we matched the transmission range to [12] by slightly increasing the TX power.

In this scenario, one chain serves as the main concurrent multipath bandwidth evaluation and the other chain serves as interfering background traffic. On the main chain, each node is equipped with two IEEE 802.11b interfaces connected to two non-interfering channels. On the background traffic chain, each node is equipped with only one interface connected to the second channel which is used in the main chain. The data rate for IEEE 802.11b is 2 Mbps and the RTS/CTS mechanism is enabled. Static routes are used in this simulation

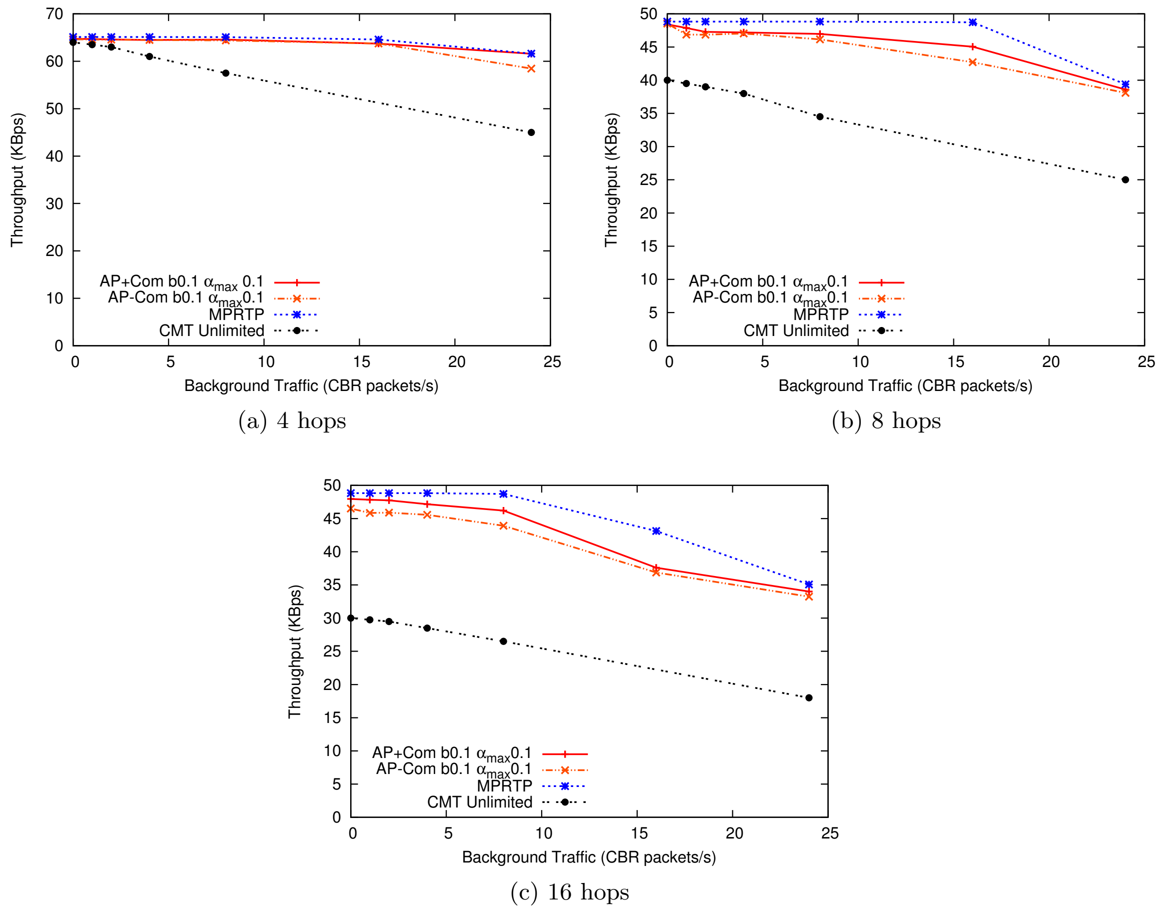


Figure 3.5: Bandwidth comparison between AP-based proposals and CMT unlimited *rBuf* from [12]

to eliminate complications due to the effect of the routing protocol.

The number of nodes varies from 10 to 34 (4, 8, and 16 hops on each chain). The traffic used in this evaluation is CBR with 1000 bytes per packet. We have performed the simulations using a few traffic rates on the main chain and have selected the one with highest obtained throughput and show the results in Figure 3.5. The main total traffic rates for 4, 8, and 16 hop cases are 65.1, 48.8, and 48.8 KBps respectively, which are decided based on the number of hops to the destination and the ratio of capacity explained in [44]. The main traffic is sent from the source during 60–360 seconds from a 420-second long simulation run. The amount of background traffic varies from 0 to 24 packets per second. The results of our protocol shown in Figure 3.5 are the average of 30 runs exactly the same as in [12].

From Fig. 3.5, it can be clearly seen that MPRTP and AP-based methods can achieve much higher throughput and is less susceptible to the interference from background traffic, in comparison to CMT. Even though now the implementations of both methods do not fully use feedback packets to gather the statistical information, a single feedback packet per decision interval ρ (= 5 s in this study) can hardly affect the higher bandwidth shown here. Therefore, we can claim here that the AP-based method and MPRTP are alternatives to CMT, which can provide better bandwidth improvement when an application can tolerate or handle packet loss.

Among UDP based proposals, MPRTP could achieve higher bandwidth due to its accurate rule-based bandwidth prediction in cases of low interference and background traffic load. However, when congestion occurs and more packet loss is observed, the bandwidth difference becomes smaller, most likely because of the less accuracy of rule-based bandwidth prediction of MPRTP. A similar behavior can be observed between AP+Com, which estimates delay compensation using packet loss, and AP-Com which does not use delay compensation. With the delay compensation process added in *AP+Com*, the performance of the AP-based method is slightly better than in *AP-Com* because the compensated delay reflects the actual network conditions better and enhances AP accuracy in estimating delay after adjusting traffic rate. However, the performance difference becomes smaller in the same manner to MPRTP when the load is high.

It is important to emphasize that, AP-Com which uses much less information, i.e., only delay information without delivered packet count nor lost packet count in comparison to AP+Com and MPRTP, can achieve comparable throughput to other protocols. This is an evidence showing the adaptability of attractor perturbation based method and more supportive results will be shown in the next subsection.

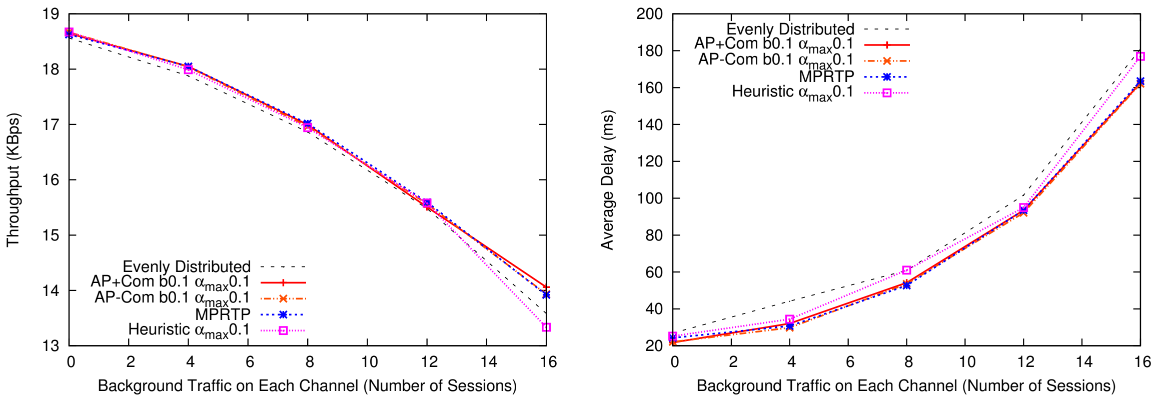
3.5.3 End-to-End Delay Improvement

In this section, we evaluate the average delay of the AP-based proposal in mobile scenarios. In mobile scenarios, an adaptive traffic distribution method is require since a traffic pattern on a certain path is affected by changes in other paths due to re-routing, topology changes, etc. To the authors' knowledge, most concurrent multipath traffic distribution methods do not support/consider mobile scenarios. Therefore, in addition to the baseline strategy where the traffic is split evenly on both paths (*evenly distributed*) and MPRTP approach, we developed another comparison method, called *heuristic* method, which operates based on the end-to-end average delay in a similar manner to our AP-based method. The main differences are that the heuristic method

- adjusts the traffic with the fixed ratio of the total traffic rate α_{max} (the AP-based method calculates the optimal solution in the range of $[-\alpha_{max}, \alpha_{max}]$),
- cannot estimate the delay after applying the traffic rate adjustment, and
- makes the decision to transfer the traffic purely from the path with higher average delay or the path with higher loss rate (in case of no delivered packet) to the path with lower one.

We expect that the evaluation against the *heuristic* method shall reveal the importance of taking the fluctuation into consideration when performing traffic distribution.

The scenario settings are as follows. 100 mobile nodes are distributed randomly in a $1500{\times}1500\,\mathrm{m}^2$ area. The random waypoint model is used with a minimum speed of 2 m/s, a maximum speed of 10 m/s, and a pause time of 30 s. Each node is equipped with two 802.11b interfaces with the data rate of 2 Mbps, connected to two non-interfering radio channels. There is one main multipath traffic session with total traffic rate of 20 packets/s and the packet size of 1000 bytes which is the same as the previous scenario. The number of background CBR traffic sessions varies from 0, 4, 8, 12 to 16 sessions per channel. Every background traffic session has the traffic rate of 1 packet/s. We chose a relatively low bit



(a) Thoughput

(b) Average Delay

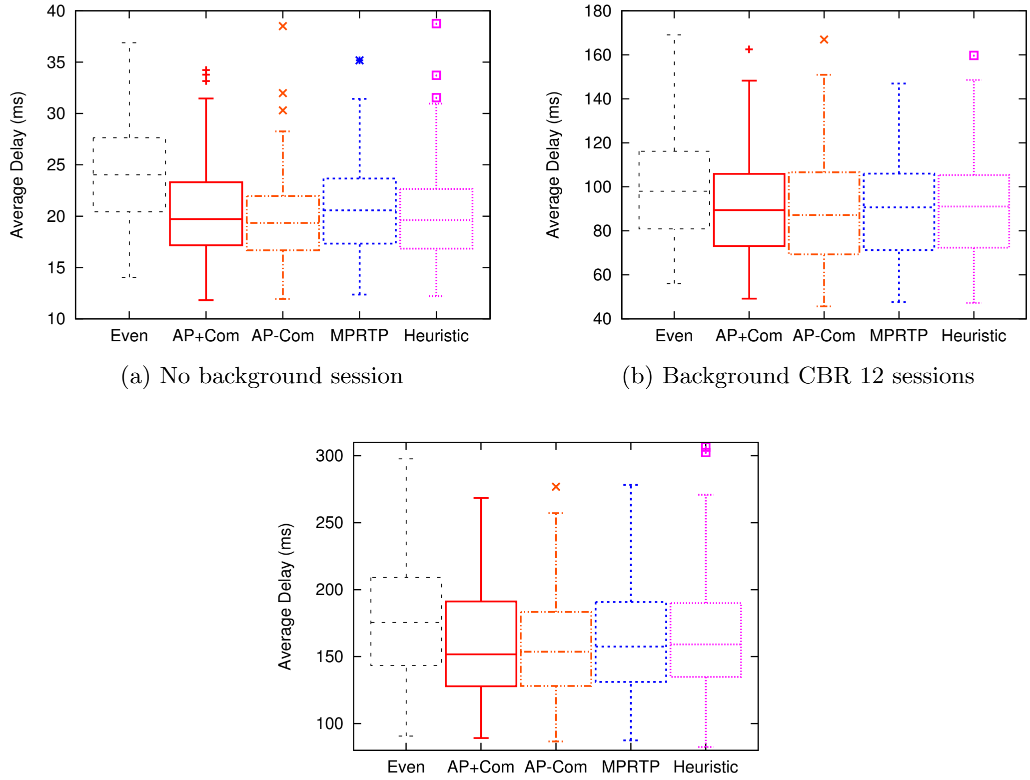
Figure 3.6: Performance comparison under mobility scenario

rate of background traffic to only increase interference while preserving bandwidth for the main session.

The average results from 100 runs are shown in Fig. 3.6–3.7. Fig. 3.6 shows the throughput and average delay againt the amount of background traffic. More details of average delay on each run is shown in Fig. 3.7 using box-and-whisker diagram where the box reflects lower quartile (Q1), median (Q2), and upper quartile (Q3). The bars show the range of ± 1.5 IQR and the dots show the data that are out of that range.

It can be observed from Fig. 3.6 that the throughput of each approach is quite similar. However, there is a significant difference in average delay as shown in Fig. 3.6(b). It is out of question that the baseline approach without traffic redistribution, called *evenly distributed*, has the worst average delay. AP proposals can achieve the same level of average delay as MPRTP by using only end-to-end delay statistics. The newly proposed comparison method *heuristic*, which uses only average end-to-end delay, performs much worse than AP proposals because using only the average delay cannot provide a good estimate of the path quality, i.e., congestion level.

Moreover, Fig. 3.7 indicates that the median of all methods generally follow the same tendency of the average, except the heuristic one. This is an effect from cases where the average delay is very high (capped and cannot be seen in the figure). Those cases are caused



(a) No background session

(b) Background CBR 12 sessions

(c) Background CBR 16 sessions

Figure 3.7: Average delay comparison under mobility scenario (y-axis is capped for visibility)

by the inappropriate traffic distribution that induced high congestion, which consequently causes failure in routing; hence, a much higher end-to-end delay.

According to these results, it can be understood that AP-based methods which use both average and fluctuation can perform relatively better than average value based methods like heuristic. Therefore, it is safe to claim that considering not only the average delay in the current interval but also the fluctuation is important for improving the performance of traffic distribution method.

Additionally, by using purely delay statistical information, AP-Com can achieve comparable throughput and end-to-end delay to MPRTP which uses more information of delivered bytes and loss rate. Hence, it is confirmed that AP-based method does not need the details

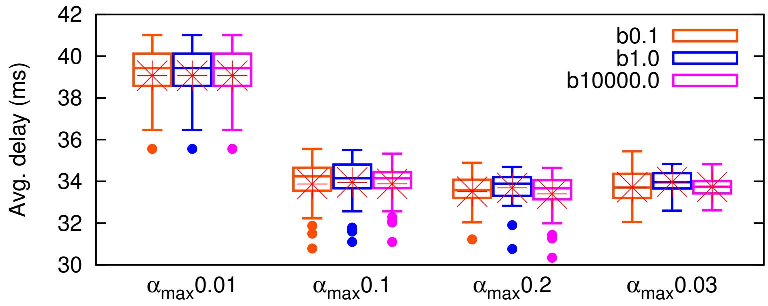


Figure 3.8: Results from different values of b

of the system under its control, which is preferable from an implementation viewpoint because a high processing overhead, energy consumption, and errors from actual measurement can be avoided.

3.5.4 Discussion on Bio-inspired Adaptability

From Fig. 3.6–3.7, it can be seen that AP-Com is the best among all approaches. Even though the throughput results of AP-Com in static ad hoc network scenario was slightly lower than other approaches, it can adapt well to scenario with higher dynamics. This result conforms with our previous assumption regarding rule-based bandwidth prediction of MPRTP and delay compensation of AP+Com, and shows that a bio-inspired method indeed has its adaptability over different scenarios without the need of fine-tuning parameters.

To further support this claim, we also added the results from bandwidth improvement scenario with different coefficients b in Fig. 3.8. It can also be seen that with inaccurate b for a specific scenario, the AP-based method can adapt to that situation and perform considerably well, due to its core bio-inspired model.

3.6 Summary

We presented a novel biologically inspired concurrent multipath traffic distribution method based on attractor perturbation (AP). Using AP, it is possible to regard the whole underlying system as a black box and perform control based on observed average and variance of the time series of the considered performance metric. Therefore, our proposal requires only end-to-end statistical information to perform traffic distribution. From simulation results, we have shown that our proposal (AP-Com) can achieve lower average end-to-end delay without sacrificing throughput when compared to *heuristic* method and evenly distributed traffic on all paths. Moreover, it can even achieve similar average end-to-end delay as MPRTP which uses delivered bytes and loss rate in addition to delay information.

In addition to the performance aspect, our proposal does not require parameter fine-tuning due to the nature of its bio-inspired core. The fluctuation, or noise, within the core gives it a flexibility to handle frequent changes in the network. It is also expected that with this adaptability, our proposal should be able to handle emerging problems better than traditional methods.

Chapter 4

Design Considerations for Future Applications of Noise-based Models

In this thesis, we have introduced two attractor-based mechanisms and proposed their applications, which are shown to be adaptive due to the characteristics of their original bio-inspired models. The two models have been used in only a few work, however, we found that they have noticeable advantages over existing bio-inspired models. Therefore, we dedicate this chapter to elaborate those advantages and considerations required when applying noise-based models in the future. Note that the possible applications are not limited to computer networking related as considered in this thesis.

4.1 Introduction

Since an introduction of bio-inspired mechanisms, there have been many research and a lot of achievements using these mechanisms [16], especially in the field of bio-inspired network systems [50]. Even though there are many bio-inspired mechanisms available, it is unclear to application designers which mechanism is better than another. In other words, there is no common framework in which self-organizing mechanisms can be compared, much less a common explanation for each and every component of self-organizing mechanisms. A similar argument regarding evaluation framework exists in [50], and recently there have been an attempt to classify a few bio-inspired mechanisms by their design patterns [28]. Sharing the same idea as the authors of [28], we would like to explain more about design factors of our noise-based models to lessen the complication that might arise whenever the models are reused.

Noise-based models used in this thesis are derived from cell biology. It is known that there are multiple sources of stochasticity and heterogeneity in biological systems, which are noise and its consequence respectively. Stochastic modeling of biological systems, therefore, incorporates intrinsic noise using stochastic chemical kinetics. Intrinsic noise in biochemical reactions is caused by randomness in many components, such as, DNA binding events, mRNA transcription and degradation processes, and other protein-metabolite interactions [67]. As a result, biological systems have developed to not only suppress noise but also exploiting it.

Our first model of attractor selection is derived from a common and well-studied of bistability in a reaction network which allows a single cell to select one of two phenotypic traits at random. In our case, the model is derived from a synthetic bistable gene switch in *Escherichia coli* in which mutually inhibitory operons govern the expression of two genes required in two alternative nutritional environments, cells reliably selected the adaptive attractor driven by gene expression noise [33]. With regards to the classification study in [28], the attractor selection model does not fit into any of the patterns discussed, hence we attempt to classify and further explain it here.

The second model of attractor perturbation is derived from a fluctuation-response relationship between a fluctuation of measured fluorescence intensity value and a change of phenotype by genetic mutation observed through mutation process of *E. coli* over multiple cloned generations [54]. The fluctuation and response relationship is similar to the concept of fluctuation-dissipation theorem of Thermodynamics [35], where the term response is used instead dissipation for a better understanding [32]. The model considers a system with an originally stable state which is a state that only fluctuates in a Gaussian distributed manner. If a perturbation, or force, is applied to such system, the amount of shifted state at a new stable state is proportional to the fluctuations, or variance, existed in the original stable state. We will discuss about the applicability of this model on computer networks in this chapter and point out difficulties in the process.

4.2 Advantages and Constraints of Noise-based Models

Through our proposed applications of attractor selection and attractor perturbation models, we have learned the their useful characteristics as follows.

4.2.1 Non-rule-based Functionality

Noise-based models do not have a strict set of rules for making decision on how they will operate but rather behave according to the current state and fluctuations that exist at that state. On the opposite to non-bio-inspired conventional approaches, which usually have a finite set of rules, the state space in noise-based models is infinite. Therefore, there is no such thing as unknown conditions (according to the noise-based models but some might arise as a result of implementation process), which makes noise-based models suitable for handling emerging problems where unpredictable conditions could occur

In attractor selection based application, the range-less flexibility is obtained through the definition of activity as a normalized ratio instead of range. On the other hand, attractor perturbation model is built around statistical information instead of raw measurement, therefore, it can handle infinite range of values.

However, such flexibility comes at a price. The concept of equilibrium state in bio-inspired models usually involves "time", which means that the system might not reach the stable state in a single iteration but rather gradually moving towards it through time. This particular characteristics has been discussed in [63] that, using our models, the system is not shifted directly towards the stable state but rather in a noisy manner, and requires some time to converge to it. Therefore, application designers need to be aware of this fact and use the outputs from the model with considerations. As an example from our implementation, instead of using the optimal solution from attractor perturbation based minimization problem as is, we gradually apply it in small steps α_{max} to the system to avoid results from the current state that might not be the target stable state and also to avoid large fluctuations that might occur after sudden change in the system.

4.2.2 Robustness against Uncertainties

Optimization algorithm is originally designed to solve problem without uncertainties. However, in real-world optimization, uncertainties are unavoidable. The term uncertainties can be further classified into aleatory uncertainty with stochastic nature which a randomness part within the real-world system, epistemic uncertainty due to incomplete or unknown knowledge, and error [2]. Robustness against uncertainties could be found generally in stochastic optimization algorithm. However, we consider alternatives of imitating the biological systems and handle uncertainties using the following approaches.

Let's first look at the proposed attractor selection based routing. It is obvious that the system has aleatory uncertainty, i.e. internal randomness, from the noise term $\vec{\eta}$ in attractor selection rate equation, Eqn. 2.1. However, the aleatory uncertainty is handled by the activity based noise suppression by adding the suppressing factor $(1 - \alpha)$ in the equation. Additionally, the epistemic uncertainty is handled by random walk process. In case of unknown path to the destination due to connection failure, the next best hop in the routing vector is first selected, and then the activity is evaluated using feedback packets. If a feedback packet causes lower activity, the random walk process starts. In case of incomplete knowledge due to loss of feedback packets, activity decay process automatically lowers the activity value over time and also triggers the random walk process.

Next, for the attractor perturbation model, microscopic aleatory uncertainties are not noticed because they are eliminated through averaging process. A significant aleatory uncertainty is taken care of in a form of variance. In complete information and error is quite a problem in attractor perturbation model as the coefficient b is estimated using measured data. In our application, we overcome the insufficient information using default coefficient b temporarily and once sufficient information is obtained, the more accurate coefficient b for the current state is calculated and used.

4.2.3 Simplified Design

Most optimization algorithms are made for specific kind of problems and it is generally difficult to understand and map each parameter/interaction to a proper component in computer networks. Moreover, a lot of optimization algorithms require a complete global information in order to operate, which is not suitable for computer networks, especially ad hoc networks, where each node normally has only access to a local information.

Both models used in this thesis have a less complicated design where only a few parameters are required and the global information is not needed. For instance, the attractor selection model only requires an appropriate definition of activity which is closely related to the performance metric that the application tries to achieve. In case of attractor perturbation, there are two parameters of attention, one is the target metric needed to be optimized and another one is the input control parameter.

Moreover, both models can operate with only a partial or local view of the whole system. For attractor selection, each node can use only local information, and the performance can be further enhanced with a wider view of the whole system. In addition, it is clear that attractor perturbation uses only end-to-end (end nodes) information and perform controls only from end nodes. The reason behind this ability is the closed loop feedback system that both models employ to gain an overview of the system through neighboring nodes.

However, the use of local information and feedback has drawbacks. In attractor selection, the system requires frequent feedback in order to maintain its performance, which requires a high overhead. The overhead can be lowered by adjusting the feedback frequency and the decay interval to avoid decreased activity through the decay process. This is considered a trade-off between response time of the algorithm and overhead which generally exists in most system. Therefore, the application designers need to carefully consider this trade-off or avoid using attractor selection where overhead is expensive. In case where applications cannot afford high control overhead, we suggest using attractor perturbation model instead.

On the other hand, attractor perturbation requires much less overhead since it uses only statistical information in a minimal amount over a periodic algorithm executing interval. However, such ability is a result of a lot of presumptions of the system. First, the distribution of observed variable x is required to be Gaussian. Second, the close relationship between the control variable a and the observed variable x is needed. Even though we have shown through our implementation and evaluation that the algorithm is still applicable to non-Gaussian system, the performance improvement is minimal. Therefore, strictly considering the presumptions of the model is recommended here in order to gain a higher performance.

4.3 Application Design Guidelines

4.3.1 Using Attractor Selection

Applying attractor selection on application can be separated into three parts.

Defining Attractors

Attractor selection is a mechanism to shift the system towards an attractor state. Therefore, application designers need to first have a clear definition of "state" for selection. In our implementation, the state space is the neighboring nodes where we use attractor selection to select a neighboring node which is a good next hop to the destination. The definition of "good" state, or the attractor, is defined in the next subsection.

Defining Activity

Activity, or the goodness of the current selection, is what drives the attractor selection mechanism. The application will know that the current selection is good or not by looking at this activity and the amount of randomness will also be controlled by it, as previously described in Section 1.2.3 using Figure 1.1. Therefore, activity of attractor selection can be viewed similarly to optimizing cost/fitness functions in other optimization algorithms.

Activity has a range of $[0, 1]$. Therefore, a normalized ratio or a mapping function is recommended for defining activity.

Adjusting Degree of Randomness

Another trade-off that exists in attractor selection mechanism is between the ability to search for a better state and the maintenance of the current good state. As the attractor selection mechanism is a mutually inhibitory mechanism, once a state is selected with high activity, other states have a much lower chance of getting selected, in other words, the selection of other states is suppressed. This natural ability is good for maintaining the good state selection but is undesirable where local minima exist. Therefore, application designers should consider how much randomness or fluctuation is tolerable in their applications and adjust the noise suppressing term accordingly. In our implementation, we decided to completely suppress noise in case of high activity, and therefore chose $(1 - \alpha)$ as the noise suppressing term in Eqn. 2.1.

4.3.2 Using Attractor Perturbation

There are two parameters that application designers need to define before starting using the model, a to-be-optimized (observed) variable x and a control parameter a. Since there are a few presumptions made during the derivation of there attractor perturbation model, application designers must make sure that the model is applicable in the considering system. First point to look at is the distribution of x when the system is stable. If the distribution of x is not Gaussian-like, we do not encourage the usage of attractor perturbation model. Second, application designers must confirm the effect of the control parameter a on the observed variable x since a close relationship between them is also required. These two points are deemed mandatory by the model creator and are the only two theoretical limitations in using the model.

Since the model and its requirements are quite simple, it is highly expected to be useful and applicable to any kind of systems. However, our implementation experience has shown otherwise. Here are the difficulties we have faced.

First of all, the coefficient b in the attractor perturbation relationship must be obtained before a control mechanism based on the model could be used. According to the model creators, the linear relationship exists if the two mandatory requirements are fulfilled. However, after a lot of attempts to obtain a constant b in either wired networks [64] or wireless ad hoc networks [9] have failed. In real world systems, the coefficient is not constant and seems to change over time. However, we have shown that the mechanism is robust enough to operate even with an inaccurate coefficient b and achieve considerably good performance.

Secondly, the distribution of x does not always follow Gaussian distribution. Even though the distribution of x is Gaussian-like when we first observe x in a stable condition. After applying a control force to the system, the distribution of x is somehow distorted and loses its Gaussian property. If we strictly obey the requirements, the control mechanism will not be applicable. Consequently, we attempted using the model without a fully fitted Gaussian distribution and are able to achieve considerably good performance out of the method.

It is important to note here that, the performance of our proposed control method based on attractor perturbation is not significantly better than existing methods. We believe that there is either a missing factor in the model or a misinterpretation of the model from our side. Since attractor perturbation is derived through the process of evolutionary molecular biology, it should be effectively applicable to any computer networking system that can be generalized to a evolutionary computation problem, in a similar manner to the genetic algorithm (GA) [29]. Hence, it is quite a surprise to us when there are so many difficulties in applying the model to network control problems. In this thesis, we have shown that we could achieve a certain kind of network control using attractor perturbation model but we left identification of the missing condition(s) in the model and the model modification as a future research work.

4.3.3 Multi-objective Applications

Single objective applications are considered simple with only one objective function, i.e., cost function or fitness function because various optimization algorithms can be used to find the optimal or sub-optimal solution to the given objective. However, there are applications with multiple objectives to fulfill and sometimes those objectives are conflicting with one another. Hence, there have been many research on multi-objective optimization algorithms for decades, where evolutionary algorithms, such as GA based algorithm [20], are popular.

In this section, we would like to provide an alternative approach to existing work which usually aims to optimize all objective functions using a single process. In our case, we consider network control applications and intend to divide the problem into multiple processes, or abstract layers. For example, instead of having a load-balancing routing protocol, one can split the problems into load-balancing which focuses on traffic load and routing. There are a few possible approaches to achieve this using only attractor selections, or a mixed model of attractor selection and attractor perturbation.

In case of using only attractor selection, application designers can choose between using a common objective function among layers, or using one sub-objective function per layer and share only the level of satisfactory of the current objective function with other layers. It is obvious that objective functions are activity definitions in this situation. On one hand, in a common objective function approach, there is only one activity which is shared among layers, taking inputs from all layers and reporting the overall satisfactory level of the current solution. On the other hand, we can also have one activity definition per layer where each layer does not use only its own activity but a combination of activity values from other layers.

The first approach is a straightforward translation from multi-objective optimization algorithms but reduces complication of having multiple control tasks in the same process. The second approach is an autonomous approach where a poor satisfactory level in one layer will trigger a random walk on another layer to search for better solutions for both (and all) layers until the satisfactory levels of all layers are met.

Moreover, application designers can also choose to use a mixed model approach of both attractor selection and attractor perturbation. In this case, the objective functions are split into activity definition of attractor selection and the target average of the observed variable x of attractor perturbation. A hierarchy is assumed in this model where the underlying attractor selection is the base mechanism and the attractor perturbation is a higher layer control mechanism. The interaction is only top-down in this model. When the upper layer target average value is not met, the random walk in the lower attractor selection layer is triggered by applying force which directly affects the lower layer behavior.

4.4 Example of Multi-objective Applications

There are a few existing work in the literature which attempted using attractor selection mechanism on two layers, called layered attractor selection approach. We introduce those proposals in Section 4.4.1. However, the mixed model approach has not been investigated before. Therefore, we propose a novel network control using the mixed model in Section 4.4.2 but left the implementation and evaluation as a future research work.

4.4.1 Layered Attractor Selection

Cooperative Routing and Clustering Controls

Sakhaee et al. [53] proposed a clustering and data gathering scheme for wireless sensor networks, where the proposal consists of two seemingly independent layers of clustering and routing, as shown in Figure 4.1. However, the clustering layer's activity uses the routing layer's activity in its activity calculation. Therefore, the performance of routing layer has an effect on triggering the re-selection process on clustering layer.

The routing activity is a combination of currently selected gateway node's residual energy ratio, charging rate (using solar power), and the available cache size ratio. This is the attempt to maximize energy usage efficient of the gateway node, and switching gateway is not preferred as long as the energy and the cache size are still available.

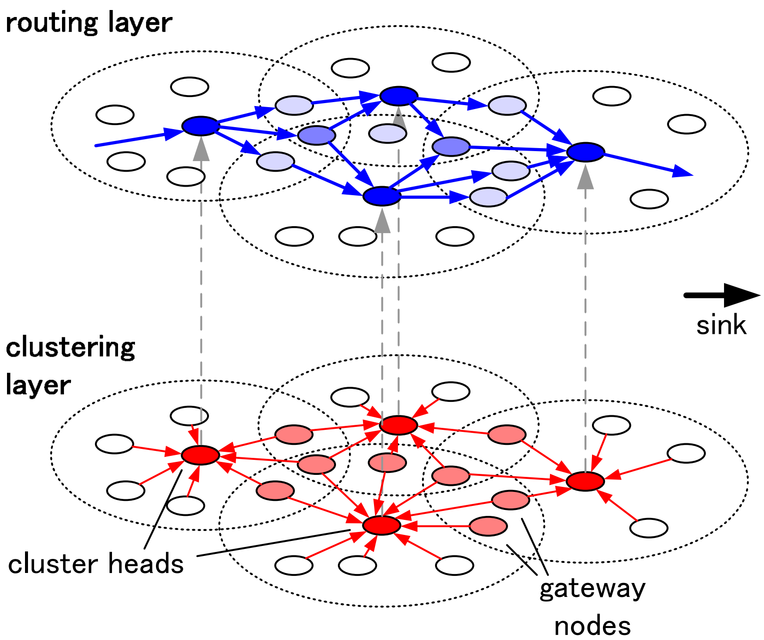


Figure 4.1: Overview of cooperative routing and clustering controls [53]

The clustering activity is a combination of routing activity and variance of energy among neighboring nodes, where an effect of routing activity is controlled by coefficient ρ, as described in the following equation:

$$\alpha_i^* = \rho A_i + (1-\rho)\sum_{j=0}^{M_i}(e_{i,j} - \bar{e}_i)^2 \tag{4.1}$$

The variance term in Eqn. 4.1 is used to select cluster head with highest residual energy, considering that such node exists when the variance is high and random selection is acceptable when the variance is low.

An interaction between routing layer and clustering layer occurs when the routing activity cannot be recovered via switching gateway alone, for example, when all gateway candidates have low residual energy rate. In such condition, re-clustering is desirable, hence, low routing activity over a period of time will eventually trigger re-clustering on clustering layer through the reuse of routing activity in Eqn. 4.1.

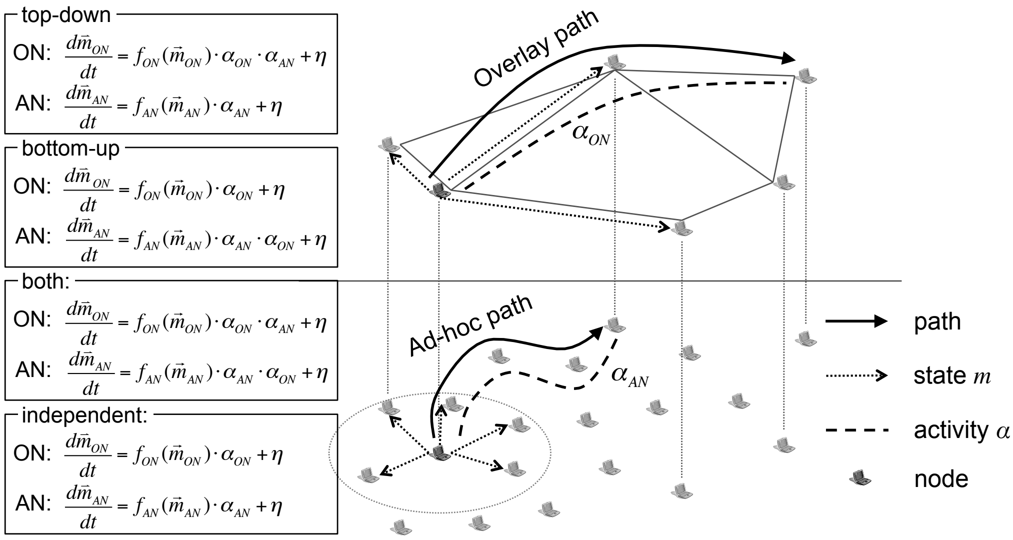


Figure 4.2: Layered routing based on attractor selection model [69]

Cooperative Overlay Routing and Network Routing Controls

Yamamoto et al. [69] has studied a cooperative routing between overlay layer and ad hoc network layer. A comparative investigation of the usage of activity of one layer on another (top-down and bottom up), mutual interactions (both), and independent cases have been made as shown in Figure 4.2. It has been shown through simulations that the bottom-up variant, where the ad hoc routing is aware of the performance of overlay routing, achieve the fastest convergence to a path with lowest delay in the overlay layer at the price of the stability and the delivery ratio during path recovery.

4.4.2 Attractor Perturbation over Attractor Selection

In this section, we propose a novel application which uses an attractor perturbation control over an attractor selection system. The application is a cooperative concurrent traffic distribution control over multipath routing in wired networks. We will separate the application description into two parts, the base layer of multipath routing and the upper layer traffic distribution control.

Base Layer: Proactive Multipath Routing over Wired Networks

Multipath routing could provide higher achieving bandwidth, faster route failure recovery, and load balancing ability. However, the widely used OSPF does not offer real multipath routing; only an equal cost multipath routing (ECMP) in a special case when there are more than one path with equal cost. Therefore, we propose another multipath routing protocol which takes into account link utilizations and end-to-end delays, which are both important factors for connection oriented protocol like TCP. We aim to provide multipath routing ability while also avoiding congestion and keeping the fairness among flows.

The proposed attractor selection based multipath routing protocol uses the same control messages (e.g. LSA) as OSPF. However, the exchanged metric is not the hop count or the raw bandwidth, but the link utilization on each interface (note that today's routers have SNMP enabled and the link utilization can be easily derived from provided statistical information [3, 19]).

After obtaining the link state database with link utilization metric, we use Yen's k-shortest path algorithm [71] to find k loopless paths to each destination. These k paths are the states in attractor selection mechanism, and the activity is calculated using end-to-end delay ratio. The routing will make a decision on which path to use based on the normalized state value. In contrast to how MARAS (see Chapter 2) chooses only the state with the highest state value, our multipath routing chooses a path out of k paths probabilistically using each the normalized state value as a probability of choosing each path. After the path is selected, packets are routed to the destination using source routing by embedding the whole selected path in the data packet header. Each packet's end-to-end delay is sent back to the source to calculate the activity.

According to the above explanation, we have two separated route maintenance methods. The less frequent one is the link utilization based shortest path calculation which provides less congested candidates. The more frequent one is the attractor selection mechanism which uses end-to-end delay feedback to evaluate the current condition of each path, avoid congestions and recover from link errors.

Based on attractor selection mechanism, the selection tends to converge to using only the path with the lowest end-to-end delay, which we will call a main path. From then on, switching to another path rarely occurs unless the activity decreases to a very low value. As a result, if there is a path with a slightly better end-to-end delay, that path will not be selected as a main path even though the overall end-to-end delay can be further improved. Therefore, in order to trigger such switching, we need another control method to assist the routing protocol, which we will propose in the next subsection.

Cooperative Concurrent Traffic Distribution Control over Multipath Routing

The attractor selection based routing protocol tends to use the currently selected path over performing random walk to switch to a slightly better path because the random walk process is costly and temporarily lower overall performance. Therefore, we need another method to assist such switching without triggering random walk. For this specific purpose, we consider using attractor perturbation model.

Since the attractor perturbation will be applied on traffic distribution control, the control variable has to be related to path selection mechanism of the routing layer. In this case, we choose the normalized state values as control variable a because it is directly related to traffic rate on each path. The observed variable x is obviously the end-to-end delay on each path. Similar to how we perform traffic distribution in Chapter 3, the minimization problem is formulated and solved. The optimal solution is then applied to the lower layer normalized state values to quickly change the traffic load in a way that minimizes overall end-to-end delay.

Actually, the attractor perturbation model has been proposed for this specific use, which is to speed up the attractor selection based control. The core concept is to perturb (or shift) the current attractor (selected state) to another state which is better according to fluctuation and response relationship. Hence, the name attractor perturbation is given to the model. We have mentioned earlier that the attractor selection model shifts the system in a noisy manner and without specific direction. The attractor perturbation model has been proposed to give such direction to the attractor selection model and enhance its performance. This combined application is proposed as a suggestion for future research direction, which is very plausible with regards to our two main proposals in this thesis.

4.5 Summary

In this chapter, we have provided a summarized observations and facts regarding both models from our own experience in implementing them. First, we explained the advantages and constraints of each model, together with our strategies used to overcome those constraints. We also emphasized certain points which required application designers' considerations when using our models. Afterward, we describe steps needed for implementing our models and raised a few examples on layered attractor selection to clarify those steps in real applications. Finally, we proposed a novel concept of mixing the use of attractor perturbation model over attractor selection model to achieve a multi-objective application and left the implementation as a future research work.

Chapter 5

Conclusion

There have been numerous attempts to achieve adaptive and autonomous network control. We used bio-inspired models and noise to achieve such network control in opposite to conventional methods, which usually use average value in performing network control and ignoring noise. Based on this research, we have shown that fluctuations should not be ignored and proposed a control method that uses meaning behind fluctuations of end-to-end delay. As for the future research direction, more studies should be done in attempts to understand the meaning of these fluctuations and apply the knowledge to network control.

5.1 Contributions of This Thesis

In this thesis, we have presented two network control methods for ad hoc networks. Both methods share a common approach of utilizing noise-based models because we realized that the current communications networks need new adaptive and robust mechanisms to overcome those challenges, and the noise-based models are capable of providing such features. Our contributions of this thesis consist of improvements achieved by two proposed methods and the provided design considerations as follows.

1. In our study of a MANET routing protocol, we proposed a routing protocol based on attractor selection mechanism to overcome changing environment conditions, e.g.,

network topology and traffic load. By using attractor selection with path length based activity for next hop selection process in MANET routing, the following features:

- adaptability without repeating parameter-tuning in new scenarios,
- resilience against node failures,
- scalability for high node density, and
- survivability for heavy traffic conditions.

2. In our study of traffic distribution over multiple paths in ad hoc networks, we proposed a concurrent traffic distribution method using attractor perturbation to improve performance by adjusting traffic rate on each path according to observed end-to-end information, not using additional paths only as backups or using all paths randomly. The contributions of the novel traffic distribution method are the following features:

    - its pure end-to-end nature where only end-to-end statistical information is used,
    - simplification of the traffic distribution process by considering the underlying ad hoc network as a blackbox,
    - overall bandwidth improvement by using multiple paths concurrently, and
    - average end-to-end delay improvement by adjusting traffic rate on each path to avoid using paths with high end-to-end delay and attempt to minimize the expected delay at the same time.

In addition to proposed network control methods, we have also explained the design considerations required to apply our models in other applications. Based on advantages and constraints of our models, application design guidelines, and implementation examples raised in Chapter 4, it should be sufficient for application designers to adopt our models with ease.

5.2 Future Research Direction

A future research direction is as follows:

- Further investigation of attractor perturbation model to identify the possibly missing factor(s) in the model and make a model modification since we had difficulties in achieving good performance of attractor perturbation based model even though the requirements were fulfilled.

- Implementation and evaluation of an attractor perturbation over attractor selection network control protocol as a proof of concept of a mixed model.

Bibliography

[1] M. Abolhasan, T. Wysocki, and E. Dutkiewicz. A review of routing protocols for mobile ad hoc networks. *Ad Hoc Networks*, 2(1):1–22, Jan. 2004.

[2] H. Agarwal, J. E. Renaud, E. L. Preston, and D. Padmanabhan. Uncertainty quantification using evidence theory in multidisciplinary design optimization. *Reliability Engineering & System Safety*, 85(13):281–294, July 2004.

[3] S. Amarnath and A. Kumar. A new technique for link utilization estimation in packet data networks using snmp variables. In *IEEE Global Telecommunications Conference (GLOBECOM '97)*, volume 1, pages 212–216, 1997.

[4] H. A. Amri, M. Abolhasan, and T. Wysocki. Scalability of MANET routing protocols for heterogeneous and homogenous networks. *Computers and Electrical Engineering*, 36(4):752–765, 2010. Signal Processing and Communication Systems.

[5] N. Asvarujanon, K. Leibnitz, N. Wakamiya, and M. Murata. Extension and evaluation of biologically-inspired routing protocol for MANETs. In *Technical Report of IEICE (NS2009-52)*, July 2009.

[6] N. Asvarujanon, K. Leibnitz, N. Wakamiya, and M. Murata. Evaluation of robustness and adaptability of a biologically-inspired MANET routing protocol. In *Technical Report of IEICE (NS2009-52)*, Jan. 2010.

[7] N. Asvarujanon, K. Leibnitz, N. Wakamiya, and M. Murata. Robust and adaptive mobile ad hoc routing with attractor selection. In *Proceedings of the 4th International*

workshop on Adaptive and DependAble Mobile Ubiquitous Systems (ADAMUS), Berlin, Germany, July 2010.

[8] N. Asvarujanon, K. Leibnitz, N. Wakamiya, and M. Murata. Noise-assisted traffic distribution over multi-path ad hoc routing. In *Proceedings of 4th International Symposium on Applied Sciences in Biomedical and Communication Technologies (ISABEL)*, Barcelona, Spain, Oct. 2011.

[9] N. Asvarujanon, K. Leibnitz, N. Wakamiya, and M. Murata. Noise-assisted concurrent multipath traffic distribution in ad hoc networks. conditionally accepted for IEICE Transactions on Communications (Special Section on Progress in Information Network Science), Nov. 2013.

[10] N. Asvarujanon, K. Leibnitz, N. Wakamiya, and M. Murata. Resilient mobile ad hoc routing with attractor selection for dense and heavy traffic scenarios. to appear in Special Issue of International Journal on Autonomous and Adaptive Communications Systems (IJAACS) on Self-* Systems, 2013.

[11] C. Avin and C. Brito. Efficient and robust query processing in dynamic environments using random walk techniques. In *Proceedings of the 3rd International Symposium on Information Processing in Sensor Networks (IPSN)*, pages 277–286, Apr. 2004.

[12] I. Aydin and C.-C. Shen. Performance evaluation of concurrent multipath transfer using SCTP multihoming in multihop wireless networks. In *the Eighth IEEE International Symposium on Network Computing and Applications (NCA 2009)*, pages 234–241, 2009.

[13] E. Bonabeau, M. Dorigo, and G. Theraulaz. *Swarm Intelligence: From Natural to Artificial Systems*. Oxford University Press, 1999.

[14] E. Bonabeau, M. Dorigo, and G. Theraulaz. Inspiration for optimization from social insect behaviour. *Nature*, 406(6791):39–42, July 2000.

[15] O. Brousse, J. Guillot, G. Sassatelli, T. Gil, M. Robert, J. M. Moreno, A. Villa, and E. Sanchez. A bio-inspired agent framework for hardware accelerated distributed pervasive applications. *Adaptive Hardware and Systems, NASA/ESA Conference on*, 0:415–422, 2009.

[16] O. Castillo and P. Melin. Optimization of type-2 fuzzy systems based on bio-inspired methods: A concise review. *Information Sciences*, 205:1–19, Nov. 2012.

[17] C. Cetinkaya. Improving the efficiency of multipath traffic via opportunistic traffic scheduling. *Computer Networks*, 51(8):2181–2197, June 2007.

[18] I. Chlamtac, M. Conti, and J. J.-N. Liu. Mobile ad hoc networking: imperatives and challenges. *Ad Hoc Networks*, 1(1):13–64, July 2003.

[19] Cisco. How to calculate bandwidth utilization using SNMP. Document ID: 8141, Oct. 2005. Updated: Oct 26, 2005.

[20] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: Nsga-ii. *Evolutionary Computation, IEEE Transactions on*, 6(2):182–197, 2002.

[21] G. Di Caro and M. Dorigo. AntNet: distributed stigmergetic control for communications networks. *Journal of Artificial Intelligence Research*, 9(1):317–365, Aug. 1998.

[22] G. Di Caro, F. Ducatelle, and L. Gambardella. AntHocNet: An adaptive nature-inspired algorithm for routing in mobile ad hoc networks. *European Transactions on Telecommunications*, 16(5):443–455, Sept. 2005.

[23] M. Dorigo, G. Di Caro, and L. M. Gambardella. Ant algorithms for discrete optimization. *Artificial Life*, 5(2):137–172, Apr. 1999.

[24] F. Dressler and O. B. Akan. Bio-inspired networking: From theory to practice. *IEEE Communications Magazine*, 48(11):176–183, November 2010.

[25] F. Dressler and O. B. Akan. A survey on bio-inspired networking. *Computer Networks*, 54(6):881–900, 2010.

[26] F. Ducatelle. AntHocNet4.zip: code for the use of AntHocNet in the QualNet 4.0 network simulator. Available at `http://www.idsia.ch/~frederick/anthocnet/anthocnet.html`.

[27] A. Eldar and M. B. Elowitz. Functional roles for noise in genetic circuits. *Nature*, 467(7312):167–173, Sept. 2010.

[28] J. L. Fernandez-Marquez, G. D. M. Serugendo, S. Montagna, M. Viroli, and J. L. Arcos. Description and composition of bio-inspired design patterns: a complete overview. *Natural Computing*, 12(1):43–67, Mar. 2013.

[29] D. E. Goldberg and J. H. Holland. Genetic algorithms and machine learning. *Machine learning*, 3(2):95–99, 1988.

[30] M. Günes, U. Sorges, and I. Bouazizi. ARA - the ant-colony based routing algorithm for MANETs. In *Proceedings of the 2002 International Conference on Parallel Processing Workshops (ICPPW)*, pages 79–85, Aug. 2002.

[31] J. Iyengar, P. Amer, and R. Stewart. Concurrent multipath transfer using SCTP multi-homing over independent end-to-end paths. *IEEE/ACM Transactions on Networking*, 14(5):951–964, Oct. 2006.

[32] K. Kaneko. *Life: An Introduction to Complex Systems Biology.* Springer Publishing Company, Incorporated, 1st edition, 2010.

[33] A. Kashiwagi, I. Urabe, K. Kaneko, and T. Yomo. Adaptive response of a gene network to environmental changes by fitness-induced attractor selection. *PLoS ONE*, 1(1):e49(1–10), Dec. 2006.

[34] J. Kilpi and I. Norros. Testing the gaussian approximation of aggregate traffic. In *2nd ACM SIGCOMM Workshop on Internet measurement (IMW)*, pages 49–61, Marseille, France, 2002.

[35] R. Kubo. The fluctuation-dissipation theorem. *Reports on Progress in Physics*, 29(1):255, 1966.

[36] K. S. Lee and Z. W. Geem. A new meta-heuristic algorithm for continuous engineering optimization: harmony search theory and practice. *Computer Methods in Applied Mechanics and Engineering*, 194(36-38):3902–3933, Sept. 2005.

[37] S.-J. Lee and M. Gerla. Split multipath routing with maximally disjoint paths in ad hoc networks. In *Communications, 2001. ICC 2001. IEEE International Conference on*, volume 10, pages 3201–3205, 2001.

[38] K. Leibnitz, C. Furusawa, and M. Murata. On attractor perturbation through system-inherent fluctuations and its response. In *Proceedings of the 2009 International Symposium on Nonlinear Theory and its Applications (NOLTA 2009)*, Sapporo, Japan, October 2009.

[39] K. Leibnitz and M. Murata. Attractor selection and perturbation for robust networks in fluctuating environments. *IEEE Network*, 24(3):14–18, May-June 2010.

[40] K. Leibnitz, N. Wakamiya, and M. Murata. Biologically inspired self-adaptive multi-path routing in overlay networks. *Communications of the ACM*, 49:62–67, March 2006.

[41] K. Leibnitz, N. Wakamiya, and M. Murata. Self-adaptive ad-hoc/sensor network routing with attractor-selection. In *Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM)*, pages 1–5, Nov. 2006.

[42] K. Leibnitz, N. Wakamiya, and M. Murata. A bio-inspired robust routing protocol for mobile ad hoc networks. In *Proceedings of the 16th International Conference on Computer Communications and Networks (ICCCN)*, pages 321–326, Aug. 2007.

[43] R. Leung, J. Liu, E. Poon, A.-L. Chan, and B. Li. MP-DSR: a QoS-aware multi-path dynamic source routing protocol for wireless ad-hoc networks. In *Local Computer Networks, 2001. Proceedings. LCN 2001. 26th Annual IEEE Conference on*, pages 132–141, 2001.

[44] J. Li, C. Blake, D. S. De Couto, H. I. Lee, and R. Morris. Capacity of ad hoc wireless networks. In *Proceedings of the 7th annual international conference on Mobile computing and networking*, MobiCom '01, pages 61–69, Rome, Italy, 2001.

[45] M. Lima, A. dos Santos, and G. Pujolle. A survey of survivability in mobile ad hoc networks. *IEEE Communications Surveys and Tutorials*, 11(1):66–77, First Quarter 2009.

[46] L. Lpez-Maury, S. Marguerat, and J. Bhler. Tuning gene expression to changing environments: from rapid responses to evolutionary adaptation. *Nature Reviews Genetics*, 9(8):583–593, Aug. 2008.

[47] M. K. Marina and S. R. Das. On-demand multipath distance vector routing in ad hoc networks. In *Network Protocols, 2001. Ninth International Conference on*, pages 14–23, 2001.

[48] M. Meisel, V. Pappas, and L. Zhang. A taxonomy of biologically inspired research in computer networking. *Computer Networks*, 54(6):901–916, 2010.

[49] S. Mueller, R. P. Tsang, and D. Ghosal. Multipath routing in mobile ad hoc networks: Issues and challenges. *Computer*, 2965(May):209–234, 2004.

[50] T. Nakano. Biologically inspired network systems: A review and future prospects. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 41(5):630–643, 2011.

[51] C. Perkins, E. Belding-Royer, and S. Das. Ad hoc on-demand distance vector (AODV) routing. RFC 3561, July 2003.

[52] R. Ramanathan and J. Redi. A brief overview of ad hoc networks: challenges and directions. *IEEE communications Magazine*, 40(5):20–22, 2002.

[53] E. Sakhaee, K. Leibnitz, N. Wakamiya, and M. Murata. Layered attractor selection for clustering and data gathering in wireless sensor networks. In *Wireless Communications and Networking Conference (WCNC), 2010 IEEE*, pages 1–6, 2010.

[54] K. Sato, Y. Ito, T. Yomo, and K. Kaneko. On the relation between fluctuation and response in biological systems. In *Proceedings of the National Academy of Sciences*, volume 100, pages 14086–14090, 2003.

[55] B. Selman, H. Kautz, and B. Cohen. Noise strategies for improving local search. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, pages 337–343, July 1994.

[56] D. W. Sims, E. J. Southall, N. E. Humphries, G. C. Hays, C. J. A. Bradshaw, J. W. Pitchford, A. James, M. Z. Ahmed, A. S. Brierley, M. A. Hindell, D. Morritt, M. K. Musyl, D. Righton, E. L. C. Shepard, V. J. Wearmouth, R. P. Wilson, M. J. Witt, and J. D. Metcalfe. Scaling laws of marine predator search behaviour. *Nature*, 451(7182):1098–1102, Feb. 2008.

[57] V. Singh, S. Ahsan, and J. Ott. MPRTP: multipath considerations for real-time media. In *Proceedings of the 4th ACM Multimedia Systems Conference*, MMSys '13, pages 190–201, New York, NY, USA, 2013. ACM.

[58] R. Stewart and C. Metz. Sctp: new transport protocol for tcp/ip. *IEEE Internet Computing*, 5(6):64–69, 2001.

[59] J. Suzuki and T. Suda. A middleware platform for a biologically inspired network architecture supporting autonomous and adaptive applications. *IEEE Journal on Selected Areas in Communications*, 23(2):249 – 260, feb. 2005.

[60] C. K. Toh, A.-N. Le, and Y.-Z. Cho. Load balanced routing protocols for ad hoc mobile wireless networks. *IEEE Communications Magazine*, 47(8):78–84, 2009.

[61] A. Tyrrell and G. Auer. Imposing a reference timing onto firefly synchronization in wireless networks. In *Proceedings of the IEEE 65th Vehicular Technology Conference (VTC2007-Spring)*, pages 222–226, 2007.

[62] R. van de Meent, M. Mandjes, and A. Pras. Gaussian traffic everywhere? In *IEEE International Conference on Communications (ICC)*, pages 573–578, 2006.

[63] N. Wakamiya, K. Leibnitz, and M. Murata. Noise-assisted control in information networks. In *Frontiers in the Convergence of Bioscience and Information Technologies, 2007. FBIT 2007*, pages 833–838, 2007.

[64] M. Waki, N. Wakamiya, and M. Murata. Proposal and evaluation of attractor perturbation-based rate control for stable end-to-end delay. In *7th International Conference on Bio-Inspired Models of Network, Information, and Computing Systems (BIONETICS)*, pages 1–15, Dec. 2012.

[65] J. Wang, E. Osagie, P. Thulasiraman, and R. K. Thulasiram. HOPNET: A hybrid ant colony optimization routing algorithm for mobile ad hoc network. *Ad Hoc Networks*, 7(4):690–705, June 2009.

[66] H. Wedde, M. Farooq, T. Pannenbaecker, B. Vogel, C. Mueller, J. Meth, and R. Jeruschkat. BeeAdHoc: an energy efficient routing algorithm for mobile ad hoc networks inspired by bee behavior. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO)*, pages 153–160, June 2005.

[67] D. J. Wilkinson. Stochastic modelling for quantitative description of heterogeneous biological systems. *Nature Reviews Genetics*, 10(2):122–133, Feb. 2009.

[68] H. Yamamoto, K. Hyodo, N. Wakamiya, and M. Murata. A reaction-diffusion-based coding rate control mechanism for camera sensor networks. *Sensors*, 10(8):7651–7673, 2010.

[69] H. Yamamoto, N. Wakamiya, and M. Murata. Study on interaction between layered self-organization based control. In *Proceedings of the 4th International Symposium on Applied Sciences in Biomedical and Communication Technologies*, page 118, 2011.

[70] X.-S. Yang. *Nature-inspired metaheuristic algorithms*. Luniver Press, 2011.

[71] J. Y. Yen. Finding the k shortest loopless paths in a network. *management Science*, 17(11):712–716, 1971.

[72] J. Yi, A. Adnane, S. David, and B. Parrein. Multipath optimized link state routing for mobile ad hoc networks. *Ad Hoc Networks*, 9(1):28–47, Jan. 2011.

[73] S. Yi, Y. Pei, and S. Kalyanaraman. On the capacity improvement of ad hoc wireless networks using directional antennas. In *Proceedings of the 4th ACM international symposium on Mobile ad hoc networking & computing*, MobiHoc '03, pages 108–116, San Diego, California, USA, 2003.

[74] F. Zhong, C. K. Yeo, and B. S. Lee. Adaptive load balancing algorithm for multi-homing mobile nodes in local domain. In *2011 IEEE Consumer Communications and Networking Conference (CCNC)*, pages 482–486, Jan. 2011.