ROUTING METRICS FOR MULTI-HOP WIRELESS MESH NETWORKS

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

Bing Qi

Certificate of Approval:

Kai. H. Chang
Professor
Computer Science and Software Engineering

Saâd Biaz, Chair
Associate Professor
Computer Science and Software Engineering

Cheryl Seals
Assistant Professor
Computer Science and Software Engineering

George T. Flowers
Dean
Graduate School

ROUTING METRICS FOR MULTI-HOP WIRELESS MESH NETWORKS

Bing Qi

A Dissertation

Submitted to

the Graduate Faculty of

Auburn University

in Partial Fulfillment of the

Requirements for the

Degree of

Doctor of Philosophy

Auburn, Alabama
May 9th, 2009

ROUTING METRICS FOR MULTI-HOP WIRELESS MESH NETWORKS

Bing Qi

Signature of Author

Date of Graduation

DISSERTATION ABSTRACT

ROUTING METRICS FOR MULTI-HOP WIRELESS MESH NETWORKS

Bing Qi

Doctor of Philosophy, May 9th, 2009
(M.S., Capital University of Economics and Business, 1999)
(B.S., Nanchang University, 1996)

125 Typed Pages

Directed by Saâd Biaz

Wireless Mesh Networks (WMNs) have gained considerable popularity recently. Most research on WMNs focuses on improving network capacity, because WMNs suffer from severe throughput degradation due to interferences. Among them, routing has been an active area of research for a long time, and it is also the focus of this dissertation.

Routing metrics are critical for selection of a path with a maximum throughput in multi-hop wireless networks. Due to the nature of wireless losses and multi-rate support, wireless links in one single network may have a broad range of characteristics such as loss rates and data transmission rates. In addition, wireless nodes may be equipped with multiple radios tuned to non-overlapping channels in order to improve network performance. All these technologies bring new challenges to route selection in wireless multi-hop networks, and the traditional minimum hop count metric does not suffice in such a complicated environment, since it does not take any aforementioned characteristics into account.

To design an effective routing metric, it is necessary to accurately acquire the link properties that might affect path performance, most importantly, the loss rate. To better assess individual link loss rate, this study proposes several modifications to existing broadcast based probe method that

leads to a more accurate link loss rate measurement. In recent years, although many link quality-based routing metrics have been explored, none of the current routing metrics is capable of capturing the necessary link properties, which is critical for a comprehensive evaluation of routing path performances and the selection of the best route. In this study, we explore the drawbacks of current routing metrics and propose two improved strategies, improved Expected Time Transmission (iETT) and improved Bottleneck Aware Transmission Delay (iBATD), respectively, to address the challenges for single-radio and multi-radio wireless mesh networks. The proposed routing metrics will be explored in various wireless network scenarios through ns-2 simulations and their performances will be compared to current routing metrics, such as minimum hop count metric (HOP), Expected Transmission Count (ETX), Expected Transmission Time (ETT) and Weighted Cumulative ETT (WCETT) metrics. Additionally, an extended Dynamic Source Routing (DSR)-based routing protocol (eDSR), which works effectively in multi-radio scenarios and performs seamlessly with any quality aware routing metrics, has also been designed to aid the performance comparisons. Moreover, we develop iETT routing metric and three other selected representative routing metrics on a Linux based test bed and assess their performances with extensive experiments.

ACKNOWLEDGMENTS

First of all, I would like to express my hearty gratitude to my advisor, Dr. Saâd Biaz, for the professional assistance and sharp guidance during the pursuit of my Ph.D. degree. He provided an excellent environment and gave valuable feedbacks for my research. Without his help, I could not have completed this dissertation. Deep gratitude is also due to all my advisory committee members. I thank my committee members Drs. Kai H Chang and Cheryl Seals for valuable advice and guidance. Special thanks are also due to the professors and students of the Department of Computer Science and Software Engineering for all the advice, encouragement, and support I have received throughout my Ph.D studies. A special thank goes to Dr. Tin-Yau Tam in the Department of Mathematical and Statistics at Auburn University, the outside reader of this dissertation. I also want to thank Mr. Feng Wu for his help with the experiments.

I also want to express my thanks to my parents, my elder sister and my husband Ke Liu, who have always provided the inspiration and support during these years. Also I am also grateful to my friend Yihang Li, for continuous encouragement and reviewing my writing.

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

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

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Overview

Wireless Mesh Networks (WMNs) are dynamically self-organized and self-configured networks, with the nodes in the network automatically maintaining node connectivity among themselves [1]. Recently, WMNs have attracted considerable popularity both in research areas and application, since they provide efficient and cost-effective solutions to extend broadband services to places where establishing cables is difficult. Many interesting applications and commercial systems have been successfully developed based on such networks [2–6].

In WMNs, most of the wireless nodes are stationary or have minimal mobilities, and they do not rely on batteries. Therefore, a WMN is also known as a special type of wireless ad hoc network. Unlike traditional wireless ad hoc networks, in which research mainly emphasizes on the mobility and minimizing power consumption for each individual node, most research in WMNs focuses on improving the performance of individual data transfers and the overall network capacity [7].

In order to efficiently support individual data transfers between source and destination nodes in such networks, a routing metric plays a significant role. However, the design of an optimal routing algorithm is dependent on the specific characteristics of target networks. For example, wired networks have stable communication links and identical bandwidths to support data transfers, and a proactive routing protocol based on the simple minimum hop metric is sufficient in most cases. In infrastructure based wireless networks, where data packets are always transferred between an access point and a terminal node, only a simple routing method combined with a rate adaptive algorithm based on the existing channel conditions is needed. Multi-hop wireless mesh networks,

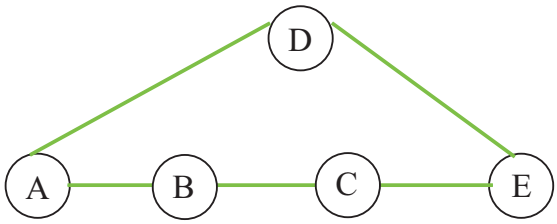

Figure 1.1: A Simple Example

however, are different from wired and infrastructure based wireless networks in that they are self-organized and all nodes take part in the process of routing packets. These networks require routing algorithms to select links from the set of available candidates to form a path between the source and the destination. Since each individual link may have different characteristics, the design of an appropriate routing algorithm in such a wireless network is much more challenging.

Another key challenge in wireless mesh networks is the limited capacity due to high levels of interference, which is typical for multi-hop wireless networks. This challenge has stimulated a number of comprehensive research projects. The most effective way to address this throughput deterioration is to configure each mesh node with multiple radio interfaces, which has been made highly feasible due to the inexpensive wireless hardware. By deploying multiple radios within one node and tuning them into non-overlapping channels, wireless nodes are able to transmit and receive packets simultaneously. Meanwhile, mesh nodes on different channels can work concurrently with minimal interference, even though they are in the direct interference range of each other. As a consequence, more radio spectrums can be effectively explored in one single WMN, which leads to enhancement of the overall network capacity. In search for an optimum network performance, this type of configuration has motivated different research challenges for wireless mesh networks, such as optimal channel assignments, new Medium Access Control (MAC) scheme design, load

balancing and optimal routing. The focus of this research is to study the routing issues, in which each node has identical channel assignment schemes in multi-radio mesh networks.

Currently, most routing protocols in wireless networks use minimum hop count (HOP) metric to select routing paths. However, this minimum hop count metric has been shown to have drawbacks in many previous research works [8–11]. It tends to select a path with physically more distant links, which generally suffers from higher loss rates and lower data transmission rates. For instance, as shown on Figure 1.1, path $A - D - E$ has only two hops, but the distance is longer between nodes A, D, and E than for the three hop path $A - B - C - E$. The HOP metric selects path $A - D - E$, but the throughput with path $A - B - C - E$ may be higher than with path $A - D - E$, since shorter links are in general more reliable. Therefore, it is crucial to consider the link/path quality when designing an efficient routing metric in wireless multi-hop mesh networks.

The objective of a routing metric is to locate a path with the highest throughput. To design such a metric, we need to capture all characteristics of the wireless links and paths that might affect the path performance. For one individual wireless link, a minimum of four characteristics need to be considered. (1) The first characteristic is the link loss rate. A lossy link may transmit multiple times to successfully deliver one data packet, leading to a lower throughput. (2) The second characteristic is the data transmission rate. Since most current wireless devices support multiple rates, a wide range of data transmission rates exist simultaneously in one single wireless network. Packets transmitted via a link with higher data transmission rate need less transmission time, usually resulting in a higher throughput. (3) To consider the transmission time via one hop link, the overheads along with each data packet transmission also need to be considered since they are transmitted at the basic data rate whenever the actual data transmission rate is. (4) The last

characteristic is the channel assignment. This channel diversity property only needs to be addressed when multi-radio/multi-channel is used.

A routing path is composed of individual links. For each entire path, the following characteristics may affect its performance. (1) Hop length. A longer path normally introduces more transmission time and packet delay. (2) Intra-path interference. Due to the shared nature of wireless links, the links on the same channel within one single path always interfere with each other when they are in the interference range. For example, in Figure 1.2, four wireless nodes S, A, B and D are placed in a single-radio network. Suppose that node S sends packets to node D, the transmissions via link S-A will interfere with the transmissions on the links A-B and B-D. As a result, only one link can successfully transmit packets at any given time. When links within one single path are in the same interference range, the path performance is limited not only by its own link quality, but also by other links. This interference caused by the same flow with different links on the same channel is called intra-path interference in this dissertation. When the link loss rates within one path are different, the path performance will be significantly affected. Therefore, to evaluate one path performance, only adding each individual link metric value separately is insufficient to guarantee a routing path with high performance. (3) Intra-flow interference. In multi-radio networks, nodes may have multiple radios tuned into different non-interference channels. Depending on which specific channel a link uses, the overall path throughput might be different. Therefore, the interference caused by the property of channel diversity, called intra-flow interference here, is very important in multi-radio wireless networks.

Currently, besides the traditional minimum hop count (HOP) metric, many other quality-aware routing metrics have recently been designed in wireless mesh networks, such as *RTT, ETX, ETT,*

Figure 1.2: Intra-path Interference

MTM, WCETT, *AETD* and *MIC* [8, 12–15]. Although each of them makes considerable improvements from the HOP metric, none of these routing metrics is able to capture all the aforementioned characteristics of one path. In this study, to address the limitations of current routing metrics, we propose two novel routing metrics, improved Expected Transmission Time *iETT* and improved Bottleneck Aware Transmission Delay *iBATD*, for single-radio and multi-radio wireless mesh networks respectively, to select paths with higher performance. Extensive experimental results show that both metrics can locate paths with significantly higher throughput and lower end-to-end delay compared to other metrics.

1.2 Accomplishments and Contributions

The main accomplishments and contributions of this dissertation are listed as follows:

- Most routing protocols on WMNs are based on the scenario of only one radio per node, and they usually depend on the number of hops to identify appropriate routing paths. However, due to the complex characteristics of wireless mesh networks, such as different link loss rates, diverse transmission rates within one single network, those existing protocols are not effective, especially for multi-radio support. Therefore, we designed a quality-aware dynamic source routing protocol by extending the traditional dynamic source routing protocol. This

extended dynamic source routing protocol is able to easily incorporate with different routing metrics and can be used to evaluate the performance of all routing metrics

- Accurate measuring of the wireless link loss rate is critical to the design of an effective routing metric, since the link loss rate is one of the most important factors to affect the path's performance. Most current routing metrics use a broadcast-based probe (BBP) method to measure the link loss rate. In this dissertation, we have explored the BBP method and found that this method leads to highly inaccurate loss rate measurement in wireless multi-hop network with IEEE802.11g radio. Corresponding solutions have been proposed to address the drawbacks of the BBP method. The experimental results indicate that our proposed modifications can measure the link loss rate more accurately.

- In single-radio wireless mesh networks, we summarize the limitations of current routing metrics and propose a new routing metric called *iETT*. By more thoroughly addressing the transmission overhead and intra-path interference, *iETT* is capable of finding a more efficient path with higher performance and lower packet delay.

- In multi-radio wireless mesh networks, in which each wireless node is configured with multiple radios tuned to non-interference channels, we first propose a new routing metric called *BATD*. The *BATD* metric not only takes the diverse channel distribution into consideration in case there are multiple non-overlapping channels within one path, but also recognizes that links on the different channels can transmit data packets concurrently. Along with *BATD*, we further design a routing metric called *iBATD* (improved BATD), which aims to combine the benefits of both *iETT* and *BATD*. After extensive testing under different network scenarios, it outperforms other routing metrics on different network topologies in most cases.

- To assess the effectiveness of our routing metric (iETT) and other most representative routing metrics in real world, we have implemented them on a Linux based testbed. Extensive experiments are conducted on the testbed to conform that *iETT* has a better performance than other routing metrics.

1.3 Organization of This Dissertation

The rest of this dissertation is organized as follows.

Chapter 2 investigates relevant literature work and introduces the background of this research. In Chapter 3, we describe the design principle of the extended Dynamic Source Routing Protocol and verify it with extensive simulations. Chapter 4 reveals the drawbacks of current methods due to probe link loss rates and proposes correspondent solutions to accurately measure link quality. Chapter 5 describes our novel high performance routing metric called *iETT* and demonstrates its effectiveness in single radio WMNs. In Chapter 6, we first propose the *BATD* metric to address the intra-flow interference in multi-radio WMNs in which multiple radios are configured within each wireless mesh node. And then, to combine the benefits of *BATD* and *iETT* , we introduce another new routing metric called *iBATD*. The effectiveness of both BATD and iBATD have been assessed under different simulations. Chapter 7 describes a Linux testbed to evaluate the performance of different routing metric. Chapter 8 concludes the dissertation and provides hints for future work.

CHAPTER 2

LITERATURE REVIEW

The related technology of our research is introduced in this section, and then the literature on the routing metrics for single-radio and multi-radio multi-hop wireless networks is reviewed.

2.1 Related Technology

Multi-hop wireless networks have been very popular in recent years because of their compatibility with most applications. Many efforts have been invested to improve their throughput and delay performances. Currently, the IEEE 802.11 technology [16], which is the most mature technology to support wireless networks, dominates the market of wireless products. The first version of 802.11 standards was released in 1997 and it supports single bit rates of 1 or 2 Mbps. The wireless link reliability depends on the distance, transmission power and environment [17]. Thus different from wired networks, in which a link either works or does not work, wireless links normally suffer some loss rates. However, as long as it successfully transmits routing control packets, it will be considered to deliver the data packets successfully.

With the demand for a higher data rate, the 802.11b [18] and 802.11a [19] amendments to the IEEE802.11 standard were extended for the original physical layer in the 1999. The basic architecture, feature and service of 802.11b are defined by the original 802.11 standard, with the change made only to the physical layer. The PHY layer of 802.11b exploits the Direct Sequence Spread Spectrum system in the 2.4 HZ band to support up to 11 Mbps date rate. The 802.11a PHY layer extends the existing IEEE 802.11 standard in the 5 HZ band and can provide data rates ranged from 6 Mbps to 54 Mbps. With the 802.11a/b/g [18–20] technologies, modern wireless devices

are able to support multiple transmission rates to accommodate a wide range of wireless channel conditions. A higher data rate can be achieved by adopting more efficient modulation schemes and coding techniques.

To select an appropriate modulation scheme, many rate-adaptation approaches have been proposed: Kamerman and Kamerman ARF [21]; B. Sadeghi *et al.* OAR [22]; Holland, Vaidya and Bahl RBAR [23]; Lacage, Manshaei and Turletti AARF [24]; Pang, Leung and Liew LD-ARF [25]. The key idea of the rate-adaptation is to dynamically select an appropriate data transmission rate to match the current channel conditions. Due to physical properties of communication channels, distance is one of the primary factors that limit wireless channel quality. Therefore, there is an inherent trade-off between high transmission data rate and effective transmission range. Although multi-rate devices enable applications to meet all kinds of demands with a great deal of flexibility [26], they cannot change the trade-off between data rates and transmission range. High data rates and long ranges of transmission cannot be achieved simultaneously.

This multi-rate characteristic has been shown to greatly improve wireless multi-hop network performance [22,23,26]. Meanwhile, it presents a new challenge when design an appropriate routing metric.

Besides the multi-rate strategy, many other technologies have been explored. As we all know, the poor performance of the multi-hop wireless networks is mainly due to the interference of wireless communication. For example, a flow with two hops away halves the throughput compared to a flow going via a single hop, because wireless interference dictates that only one of the two links can be active at one time. To reduce the interferences and therefore improve network performance, one of the approaches to alleviate this performance degradation is to exploit directional antennas [27, 28], which is advantageous over omni-directional antennas when applying in multi-hop

networks. For instance, directional antennas are able to focus energy and extend the transmission ranges in one given direction, thereby decreasing interferences and increasing the potential for spatial reuse in other directions. However, it usually suffers from disadvantages of having large size and expensive cost. Another approach is to allow each single wireless radio to switch among multiple non-overlapping channels [29–31]. By following the same channel hopping sequence scheme, all wireless nodes are capable of switching to the appropriate working channels according to network requirements. However, the nodes have to be locked on the same channel in order to communicate with each other. The performance of this scheme depends heavily on the channel switch latency, which can be very slow with currently commercial off-the-shelf products [32]. Due to the ever decreasing hardware cost, the switching latency can be eliminated by equipping each node with multiple radios. With this approach, the wireless nodes can send and receive packets simultaneously. The available spectrum is more efficiently shared and the network performance is greatly enhanced. In this dissertation, our research focus on this configuration for multi-radio mesh networks, in which each node can be equipped with multiple radio interfaces and the channels on each radio will not be changed dynamically.

However, due to the availability of multiple radios and multiple non-interference channels, the multi-radio wireless mesh network has a wide range of configuration flexibility. Much research work [33–35] has been stimulated to provide an intelligent channel assignment scheme to maximize the network performance. Overall there are three interface assignment strategies [36]: static assignment, dynamic assignment, and hybrid channel assignment. As our research focuses on the design of a routing metric, the multi-radio wireless mesh networks are used with the static assignment method. In such a channel assignment, the interfaces of all nodes are assigned to a common set of channels.

Routing protocol is one of the most important parts for the wireless mesh networks. Currently, most routing protocols have been proposed and exploited, such as Charles and Elizabeth M AODV [37], C. Perkins DSDV [38], T. Clausen *et al* OLSR [39], J.Garcia *et al* STAR [40], David B Johnson and David A Maltz DSR [41]. However, most of them use the minimum hop length to select the routing path in multi-hop wireless networks. Thus they are not effective in wireless mesh network scenarios. Since DSR is one of the best performing routing protocols in such wireless networks [42–44], we extend the current dynamic source routing protocol and develop DSR-based quality-aware routing protocol to evaluate our new routing metrics in this study.

2.2 Routing Metrics in Single-Radio WMN

In this section, we briefly review current routing metrics in the single-radio wireless mesh networks. In such networks, each node is configured with only one radio, and all the radios are tuned to the same channel to transmit/receive packets. Due to the propagation properties of the wireless link and its multi-rate characteristic, each individual link may have different attributes.

Minimum hop count metric (**HOP**) is one of the most popular routing metrics in multi-hop wireless networks. Many multi-hop routing protocols select paths based on HOP. With this metric, only the hop count is measured, and the path with the shortest hop length is selected. The primary advantages of this metric are its simplicity and low computing overheads. No additional measurement is required to assess the metric. However, it has been shown that a path with a minimal hop count does not necessarily yield a high throughput performance. The *HOP* metric tends to select the path with the smallest number of hop count, but this path usually includes long distance links with high loss rates. A lossy link requires averagely more transmissions for one successful data

delivery, resulting in lowered performance. Therefore, it is important to consider the link quality when designing a routing metric.

To take some link quality into account, Atul Adya *et al.* proposed the *per-hop round trip time metric* (**RTT**) [12], in which each node measures the hop round trip time and keeps an exponentially weighed average of the RTT samples to its neighbors. This routing metric selects the path with the least total sum of RTT and it is able to estimate how busy and lossy a link actually is. However, *RTT* is a load-dependent metric and easily leads to route instability. Furthermore, the high overheads of the round trip time measurement and lack of consideration of the link data rate are also its obvious drawbacks.

Expected Transmission Count (**ETX**) metric was introduced by De Couto *et al.* [8]. It considers the number of transmission times unicast packets need to be transmitted to successfully traverse a link. Defined as the number of expected transmission times needed to successfully send a packet over a hop, *ETX* is derived from the measurement of the packet loss rate. The paths are evaluated by the sum of ETX's on all hops, and the one with the lowest sum is selected. Let P_f and P_r be the hop forward and reverse direction loss rates respectively, and the probability of the unsuccessful delivery rate P for this link is:

$$P = 1 - (1 - P_f) \times (1 - P_r) \tag{2.1}$$

Therefore, the probability of one successful delivery is $(1 - P)$ and the expected number of transmissions of a packet on this hop is:

$$ETX = \frac{1}{1 - P} \tag{2.2}$$

The weight of a path with n links is defined as the sum of all link's *ETX*:

$$ETX = \sum_{i=1}^{n} ETX_i \tag{2.3}$$

Since the 802.11 MAC access mechanisms re-transmit lost packets, the data packets that go through the links with high loss rates might consume more medium time and reduce the overall throughput. The *ETX* metric captures the loss rate of each individual link and selects the paths with the lowest expected total number of transmissions. Draves, Padhye and Zill reported that *ETX* performs better than the *HOP* and *RTT* metrics. The disadvantage of *ETX* is that it does not recognize that wireless links with the same *ETX* may have very different data transmission rates in multi-rate networks. Apparently, *ETX* is unable to discriminate the difference between a link with a 11 Mbps data rate and a link with other data rates in case they have the same loss rates.

Draves, Padhye and Zill proposed the *Expected Transmission Time* (**ETT**) metric, which improves the *ETX* metric by taking the data transmission rate into account. It is calculated with the packet size, the data transmission rate and the link loss rate. It represents the expected amount of time needed to successfully deliver a data packet via one path. Let S be the packet size, B_i be the data rate and ETX_i be the expected transmission count for a given hop i, *ETT* is defined as:

$$ETT = \sum_{i=1}^{n} ETT_i = \sum_{i=1}^{n} ETX_i \times \frac{S}{B_i} = \sum_{i=1}^{n} \frac{1}{1 - P_i} \times \frac{S}{B_i} \tag{2.4}$$

In this equation, P_i represents the packet loss rate of link i.

The *ETT* metric for one path is the sum of all links' ETT for that path, indicating the total expected transmission time needed for a packet traveling along the specific path. It is also called "bandwidth-adjusted ETX" which multiplies the *ETX* by the S/B (packet size over bandwidth) to

approximately obtain the time spent on transmitting one packet. *ETT* captures link loss rates, path length as well as the data transmission rates, therefore outperforming *HOP* and *ETX* significantly in various scenarios. However, *ETT* might fail to differentiate some routing paths with very different throughput performances, since the *ETT* metric does not consider the overheads when estimating the expected transmission time over a hop. In a normal data packet transmission, the transmission time is not only related to the data rate of the link, but also to a considerable amount of control overheads transmitted at the basic data rate 1 Mbps. Especially for higher data rate links, the MAC layer overhead consumes a significant amount of transmission time, should not be ignored.

Medium Time Metric (**MTM**) was proposed by Awerbuch, Holmer and Rubens [45, 46]. It is a routing metric derived from a theoretical model of the attainable throughput in multi-rate multi-hop networks. *MTM* attempts to minimize the total consumed medium time to send packets from the source to the destination when it selects a routing path. It assigns a weight to each different link and the weight is proportional to the amount of the medium time used by transmitting a packet via that link. The *MTM* metric of any given path is thus a sum of the weights along that path. When the MTM metric calculates the medium transmission consumed by a data packet, it assumes a full RTS/CTS exchanges to take MAC layer overheads into account. Then MTM specifies a corresponding weight to each link with different data rates.

Although *MTM* considers the MAC layer overhead, the calculation is inaccurate for two reasons. Firstly, *MTM* assigns equal weights to links regardless of different packet sizes. In reality, packets with different sizes lead to different medium time consumption. This fact is ignored by *MTM*. Secondly, *MTM* assumes a full RTS/CTS exchange always along with a data packet transmission. In real applications, RTS/CTS may not always be enabled. These two assumptions make *MTM* metric ineffective under certain circumstances.

2.3 Routing Metrics in Multi-radio WMN

In this section, we focus on the current routing metrics in multi-radio multi-hop wireless networks. With the hardware cost decreasing, configuring each node with more than one wireless radio to improve the network performance has received considerable attention in recent years. This is also the basis of wireless multi-radio mesh networks [1]. In a wireless multi-radio mesh network, a collection of wireless nodes that acts as wireless access routers provide connectivity to mobile clients. The wireless access routers communicate with each other in multi-hop wireless mode. They are normally equipped with multiple radios, and each radio is configured to a different non-interference channel to route each other's traffic to the internet gateways. Nodes may also communicate with each other directly through the mesh network without going through the gateways. Also in such wireless mesh networks, wireless access nodes generally have minimized mobility and enough energy.

These unique characteristics of multi-radio multi-hop network require additional consideration for the design of an appropriate routing metric. One of the important tasks is to take channel/radio information into account, because generally a route with more diverse channels will have less intra-flow interference than the route with less diverse channels

The *Weighted Cumulative ETT* metric (**WCETT**), proposed by Draves, Padhye and Zill, was designed specifically for multi-radio multi-hop wireless networks [13]. It assigns a weight to each individual link based on the Expected Transmission Time (ETT) of a packet over the link. Like the ETT metric in single radio wireless networks, ETT is a function of the loss rate, packet size and the data rate of the link. To take the intra-flow interference into account, *WCETT* selects a path to reduce the number of nodes that transmit on the same channel on that path. Combined with the sum of individual link's ETT weights, the *WCETT* metric explicitly accounts for the intra-flow

interference among links that use the same channel. We assume that a path with n hops exists in a wireless network, in which a total of k different non-overlapping channels are used. The *WCETT* metric can be represented as:

$$WCETT = (1 - \beta) \times \sum_{i=1}^{n} ETT_i + \beta \times \max_{1 \leq j \leq k} X_j \tag{2.5}$$

Where β is a tunable parameter subjected to $0 \leq \beta \leq 1$. For one path, X_j is the sum of transmission time consumed by links on channel j. The total path throughput will be greatly affected by the bottleneck channel, which has the largest X_j ($\max_{1 \leq j \leq k} X_j$).

For this specific routing metric, the first term is the sum of expected transmission time along all hops in the path, and it stands for the total individual links' resource consumption along that path. The second term $\max_{1 \leq j \leq k} X_j$ reflects the set of hops that might have the most impact on the path throughput. Apparently, *WCETT* essentially gives a low weight to paths that have more diverse channel assignments on their links. The β can be viewed as an attempt to balance the two components.

The *Adjusted Expected Transfer Delay* (**AETD**) metric was introduced by Zhou, Zhang and Qiao [14]. *AETD* is designed to consider both delays and jitters of candidate routes when making a routing decision. It attempts to select a route on which hops operating on the same frequency channel are separate as far as possible. When a packet is transmitted via a specific path, the achieved throughput is determined by the expected end-to-end transfer delay (ETD) and the lower bound of the expected delay jitter between consecutive packet transmissions (EDJ). A good route should have a small ETD as well as a small EDJ value. In this metric, its ETD represents the expected transmission time, and its EDJ actually stands for channel diversity information.

In detail, ETD is related to the hop length of the route and the link quality (including the data transmission rate and loss rate) of each hop. For a given path r, let $H_r = h_1, h_2, ...h_k$ denote the corresponding hop sequence along route r, h_i represent the hop between nodes $(i-1)$ and i, and ETT_{h_i} denote the expected packet transmission time over hop h_i. The expected end-to-end transfer delay for a single packet is therefore defined as:

$$ETD_r = \sum_{h_i \in H_r} ETT_{h_i} \tag{2.6}$$

The value of EDJ is mainly affected by the channel diversity of the route. A route with more diverse channels experiences less interference and has a small EDJ value, since packet transmissions on different channels do not interfere with each other. The calculation of EDJ relies on the interference model. In the work, *AETD* exploits the hop-distance-based interference model to compute the EDJ value. This interference model uses the hop distance to assess whether two links have interference; two links interfere with each other only if they operate on the same channel and are within 3-hop distance. With the hop-distance-based interference model, C_{h_i} denotes the frequency channel that nodes (i-1) and i used to communicate with each other. Suppose that $EDJ_r(i)$ is the expected extra delay from node i to the destination node k ($k > i$) along route r. For the route r, *AETD* is defined as follows:

$$EDJ_{r(i)} = \begin{cases} ETT_{h_k} & \text{if } i = k-1 \\ ETT_{h_{i+1}} + EDJ_{r(i+1)} & \text{if } \exists i+1 < j \leq \min(i+m+1, k) \text{such that } C_{h_{i+1}} = C_{h_j} \\ \max(ETT_{h_{i+1}}, EDJ_{r(i+1)}) & \text{else} \end{cases} \tag{2.7}$$

In the above equation, m is the interference distance (measured in hops, the AETD metric set it to 3). Clearly, for the two routes with the same ETD, the one with more channel diversity has a smaller EDJ. *AETD* is combined by ETD and EDJ:

$$AETD = (1 - \alpha) \times ETD + \alpha \times EDJ \tag{2.8}$$

Here, α is a tunable parameter between 0 and 1. The route with the smallest *AETD* metric will be preferred.

MIC, *metric of interference and channel-switching*, is a routing metric proposed by Yaling, Jun and Robin [15, 47, 48]. It is a novel topology-dependent path weight function and enhances the effectiveness of *WCETT* by considering the inter-flow interference between different data transmission flows. The key feature of this metric is to address the load-balancing problem, which is critical for improving performance in wireless mesh networks. The *MIC* metric is composed of two parts: Interference-aware Resource Usage (IRU) and Channel Switching Cost (CSC). These two components are used to capture different characteristics of multi-radio networks. The IRU component captures the effects of inter-flow interference as well as the different transmission rates and loss rates of wireless links, while the CSC component models the impact of diverse channels within one path.

For a path p, *MIC* can be denoted as follows:

$$MIC(p) = \frac{1}{N \times \min(ETT)} \sum_{\text{link l}\in p} IRU_l + \sum_{\text{node i}\in p} CSC_i \tag{2.9}$$

In this equation, N is the total number of nodes and $min(ETT)$ is the smallest ETT in the network. For one given link l from node $i \rightarrow j$, IRU_l is defined as $IRU_{ij}(c) = ETT_{ij}(c) \times$

$|N_i(c) \bigcup N_j(c)|$. $N_i(c)$ is the set of neighbor nodes that interfere with node i when it transmits on channel c. $|N_i(c) \bigcup N_j(c)|$ is the total number of nodes that might interfere with any transmission between node i and node j via channel c. $ETT_{ij}(c)$ is the expected transmission time over the link $i \rightarrow j$, which is used to capture the loss rates as well as the data transmission rates. In general, the IRU component simulates the aggregated channel time consumed by the transmissions at the link $i \rightarrow j$ over channel c. The second part of this formula is the CSC cost, which attempts to address intra-flow interference (channel diversity) within one single flow. Evidently, with the same IRU, the path with more diverse channels will have less intra-flow interference than the path that always uses the same channel to transmit packets. CSC is denoted as follows:

$$CSC(X) = \begin{cases} w_1 & \text{if } CH(prev(X)) \neq CH(X) \\ w_2 & \text{if } CH(prev(X)) = CH(X) \end{cases} \quad 0 \leq w_1 \ll w_2 \tag{2.10}$$

In this equation, prev(X) is the channel used by the previous hop of node X and CH(X) is the channel that node X uses to transmit to the next hop. The relationship $w_2 \gg w_1$ captures the fact that due to intra-flow interference, using the same channel at node X and prev(X) imposes a higher cost than using different channels. The *MIC* metric set w_1 to 0 since the cost of using multi-radio simultaneously is usually very small. The w_2 represents the cost when two consecutive hops using the same channel. However, *MIC* metric assumes that all the neighbor nodes within the interference range always interfere with the transmission, even if the neighbor nodes do not transmit any packets. This assumption is not always correct and it might select a path with worse performance and it also assumes that two links on the same channel interfere with each other only when they are consecutive. These two assumptions make *MIC* ineffective and unrealistic.

Routing Metrics	Single-radio					Multi-radio		
	HOP	RTT	ETX	ETT	MTM	WCETT	AETD	MIC
Path Length	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Link Loss Ratio	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Link Data Rate	No	Yes	No	Yes	Yes	Yes	Yes	Yes
Overheads	No	No	No	No	Partial	No	No	No
Channel assignment	N/A	N/A	N/A	N/A	N/A	Yes	Yes	Yes
Intra-flow interference	N/A	N/A	N/A	N/A	N/A	Yes	Yes	Yes
Intra-path interference	No	No	No	No	No	No	No	No

Table 2.1: Characteristic Summary of Routing Metrics

2.4 Summary of Current Routing Metrics

Most popular routing metrics in wireless networks have been examined and their limitations have also been briefly reviewed in previous two subsections. In summary, the *HOP* metric only considers the hop length characteristic and tends to select paths with higher loss rates. Then *ETX* was introduced to capture both the loss rate of each individual link and path length. After that, researchers proposed the *ETT* metric, which improved *ETX* metric by capturing the impact of link transmission data rate on the performance of the path. However, *ETT* fails to consider the overheads with the transmission of each packet. To compensate it, *MTM* was proposed by weighing each individual link according to its medium time usage by considering the overhead, but it ignore the packet size and only consider overhead value approximately(with weights). For some routing metrics considering multi-radio, in addition to taking link loss and transmission rates into account, *WCETT*, *AETD* and *MIC* also addressed the impact of channel diversity. *MIC* is the first metric to consider the inter-flow interferences, but it might lead to select non-optimum path. Table 2.4 summarized characteristics captured by current routing metrics in previous sections.

CHAPTER 3

EXTENDED DYNAMIC SOURCE ROUTING (E-DSR)

Wireless mesh networks have a great potential for a wide range of applications due to their self-organization, inexpensive deployment and independence of pre-existing infrastructures. Among many research challenges in such networks, routing protocols are one of the most critical roles to affect the overall performance. Currently, a number of different routing protocols have been proposed and most of them are based on single-radio scenario, in which each node is configured with only one radio. However, to enhance the overall wireless mesh networks' capacity, it is highly feasible to equip one wireless node with multiple radio interfaces operating on orthogonal channels. In such a multi-radio scenario mesh networks, a routing protocol is needed to be able to take advantage of the availability of multiple interfaces. Furthermore, most of current routing protocols utilize the number of hop count as metric to search routing paths and make a route selection. And this shortest hop count metric has been proved not effective in wireless mesh networks [9, 10]. It is necessary to enable routing protocols to work seamlessly with quality-aware routing metrics.

In this chapter, we describe an extended dynamic source routing protocol called E-DSR, which is able to work efficiently in multi-radio scenarios. This E-DSR routing protocol is designed based on the traditional Dynamic Source Routing (DSR) [41] protocol. The effectiveness of our E-DSR routing protocol has been verified through extensively simulations by comparing with DSR in single radio scenario. In addition, we will also briefly explain how it can work with any quality-aware routing metric.

3.1 Introduction

Wireless Mesh Networks is a new distributed access network. Compared with traditional ad-hoc multi-hop networks, it has no mobility and is not powered by the battery. The development of protocols for such wireless networks can be derived from protocols designed within the IETF Mobile Ad-hoc Network (MANET) working group [49] because most MANET protocols can provide self-configuration capability, which are highly desirable in the context of wireless mesh networks.

In general, routing protocols for wireless multi-hop networks are classified as *proative* and *reactive* [42]. Proactive routing protocols require wireless nodes to maintain routing information to all other nodes on the wireless network and they calculate routing paths in advance before wireless node actually need them. Routing protocols, such as Optimized Link State Routing (OLSR) protocol [39], Source Tree Adaptive Routing (STAR) protocol [40], Destination Sequence Distance Vector (DSDV) routing protocol [38], falls into this proactive category. In contrast, reactive routing protocols do not maintain routes to each destination of the network on a continual basis. Instead, routes are established on request by the source. Normally, when a route is needed by the source node, it floods a route request packet to search a route. Upon receiving route requests, the destination node selects the best route based on route selection metric and constructs a route reply packet. This route reply packet is then sent back to the source via the newly chosen route. Routing protocols, such as Dynamic Source Routing (DSR) protocol [41], Temporarily Ordered Routing Algorithm (TORA) protocol [50], Ad-Hoc On Demand Distance Vector (AODV) routing protocol [37], Associativity-Based Routing(ABR) protocol [51], are all typical on demand routing protocols. In on-demand routing protocols, control traffic overhead is greatly reduced since no periodic exchanges of route tables are required.

Among all different on-demand routing protocols, Dynamic Source Routing (DSR) [41] is one of the best routing protocols in terms of resources consumption and performance in wireless multi-hop network [44]. Therefore, we design a new routing protocol based on DSR protocol called E-DSR. This E-DSR protocol extends DSR to work with multi-radio scenarios. Its performance will be evaluated through various simulations.

3.2 Design of E-DSR

3.2.1 Traditional Dynamic Source Routing

Dynamic Source Routing (DSR) [41] is a typical reactive routing protocol specially designed for wireless multi-hop networks. It only attempts to search route when necessary and no periodic control packets are flooded in the network. Different from other reactive routing protocols, DSR is based on the concept of source routing instead of hop-by-hop packet routing. In detail, with DSR, every data packet follows the route stored in its packet header. This route provides the address of every node through which the packet must travel in order to reach its destination. This feature benefits DSR that the intermediate nodes do not need to keep any route information.

By using Dynamic Source Routing protocol for one pair communication between source and destination, source nodes normally get multiple routes to the same destination node and select the final path based on the applied routing metric. Due to source routing property, source nodes actually have the entire routing path information to destination nodes. Therefore, it is easy to apply different quality-aware routing metrics to select the most efficient routing path if we attach available routing information with the normal route message.

Source IP	Request ID	Destination IP	RouteRecord Series

Figure 3.1: RREQ Packet

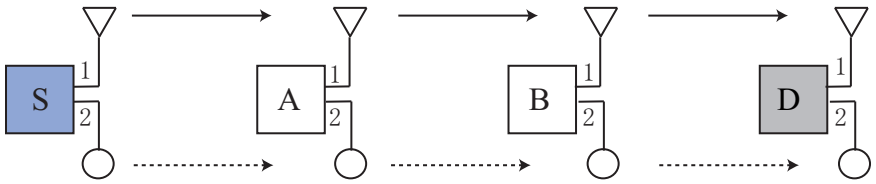

Figure 3.2: Wireless Multi-hop Network with Two Radio per Node

3.2.2 Dynamic Source Routing With Multi-radio Aware

As explained in previous section, the basic idea of DSR routing protocol lies in its route discovery process. When a source node S intends to communicate with a destination node D whose route is unknown, the source node S will initialize a route discovery process by flooding out a route request packet (RREQ) to all its neighbors. The simple format of RREQ packet is illustrated in Figure 3.1. On receiving a RREQ packet, the node A checks the destination address in RREQ packet header: if A is the target node or it have the routing information to the destination, it builds and returns a route reply packet (RREP) to the initiator node S by following the path, which is typically the reverse of RREQ Route-Record field. At the same time, The RREP will contain the sequence of nodes on the path from source S to target D. Otherwise, node A is just one of the intermediate nodes, thus node A caches the RREQ packet, appends its own identification to the RREQ's Route-Record field, and rebroadcasts the updated RREQ to its neighbours. Node A discards the RREQ packet in case the same RREQ packet has already been previously received. After the source S receives RREP, it caches the route and uses it to send subsequent data packets to the specific destination node.

Source IP	Request ID	Source Radio Index	Destination IP	RouteRecord Series

Figure 3.3: Modified RREQ Packet

When a node is configured with more than one single radio, radio indices are needed to make DSR aware of the existence of multiple radios. Figure 3.2 shows a simple network scenario with four nodes, in which every node has two independent radios (respectively represented by a triangle and a circle). The radios on each node are set to different non-overlapping channels such that they can independently transmit data packets without interference. Since each node may have one or more radios in the wireless multi-radio networks, the source route for a given path must include the radio index information on each hop. For example, one possible path in a network scenario like Figure 3.2, can be specified as: [(S,1)-(A,2)-(B,1)-D] where 1 and 2 indicate the radio index on each node.

Therefore some slight change should be made to traditional single radio DSR to perform properly in multi-radio scenarios: each node needs to broadcast RREQ packets on all its radios. When an intermediate node gets the RREQ packets from any of its radios, the node will append not only its own identification, but also the radio index to the route record to indicate on which radio it forwards the RREQ packet. As shown on Figure 3.3, a radio index field is added to DSR RREQ packet header.

Furthermore every radio within each node needs to send out a copy of RREQ packet via all its available radios in such multi-radio networks. Thus a relatively large amount of RREQ packets will be generated causing more RREQ packet collisions than single radio networks and increasing the likelihood to lose RREQ packets with good routes(broadcast packets are not retransmitted when

lost). In normal single radio multi-hop network, DSR forwards only the first RREQ packet and discards further RREQ with the same request id and source address. However, in multi-radio multi-hop networks, to increase the chances to get the best route, it is necessary to have wireless nodes to forward RREQ packets more greedily than single radio networks. Therefore, with Extebded Dynamic Source Routing (E-DSR), the intermediate node forwards *again* a RREQ packet previously seen as long as the hop count is lower.

In order to make E-DSR routing protocol work with quality-aware routing metrics, such as ETX, we can simply add one more field into the RREQ routing discovery packets and spread this information in the normal routing discovery process. For example, if we use ETX metric, this additional field can store links' loss rates. Therefore, when the source node gets the routing paths to its destination, it has the entire path information. It is easy for source node to make a route selection based on corresponding routing metric.

3.3 Performance Evaluation

3.3.1 Experimental Setting

The efficiency of E-DSR on a multi-radio network is evaluated using ns-2 simulator. In these simulations, all nodes have the same number of radios and those radios on each node are assigned to different non-overlapping channels. The same channel allocation scheme is used for all nodes.

The experimental multi-hop network consists of 50 nodes which are randomly positioned a field of $1500m^2$. Source node and destination node are randomly selected to start UDP connections. Each UDP connection sends CBR traffic with a packet size of 1000 byte every 30ms. To test the effects of various traffic loads, the number of connections is varied from 5 to 20 taking values 5,

Simulation Time	120 Seconds
Simulation Field	1500m*1500m
Propagation Mode	Two-ray Ground
Traffic Mode	CBR
Transmission Range	250 m
Number of Connections	5,10,15,20
Packet Size	1000 bytes
Traffic Interval	30 ms
Interface Queue	50

Table 3.1: Experimental Parameter for E-DSR

10, 15, 20. For a given set of parameters, the experiment is repeated for 50 times with different randomly starting times.

The performance results are obtained by averaging the results of 50 simulation runs. DSR on single radio network and E-DSR on multi-radio network are evaluated with the same simulation settings. The simulation parameters are summarized in Table 3.1.

3.3.2 Comparison Metrics

Average Throughput, Packet Delivery Rate, Average End-to-end Delay, and Average Path Length are the comparision metrics used to evaluate E-DSR routing protocol on multi-radio networks [44]:

- Average Throughput measures the total number of data packets successfully received during the unit experiment time.
- Packet Delivery Rate measures the number of end-to-end packets successfully received over the total number of data packets actually sent.
- Average End-to-end Delay measures the average packet latency time for all successfully received data packets.

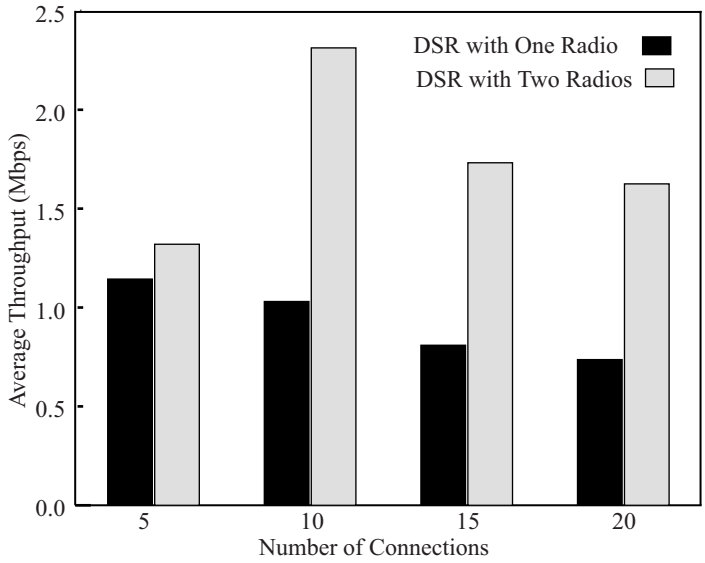

Figure 3.4: Average Throughput Vs. Connections

- Average Path Length is the average number of hops a packet traveled to reach its destination.

3.3.3 Performance Result

The simulations evaluated the impact of traffic load by varying the number of connections taking values 5, 10, 15, or 20. Promising performance results are obtained with extended DSR for multi-radio nodes.

All figures have the number of connections on the x-axis. Figure 3.4 plots the average throughput respectively achieved by DSR with single radio nodes and by our extended DSR with multi-radio nodes, which illustrates a significant throughput improvement of extended DSR with the multi-radio nodes over DSR with single-radio nodes. The average throughout improvement reaches up to 93% for different connections. As the number of connections increases, the throughput enhancement is more dramatic. This results confirms that network performance can be significantly enhances by configuring multiple radios within one single node.

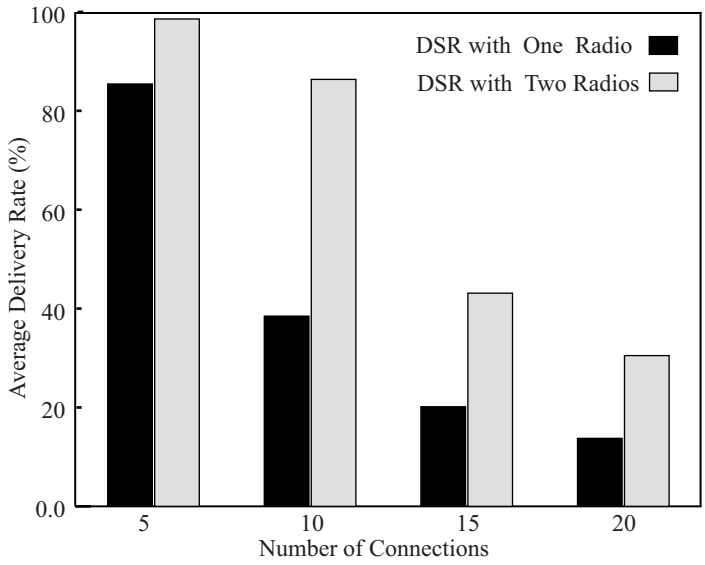

Figure 3.5: Average Packet Delivery Rate Vs. Connections

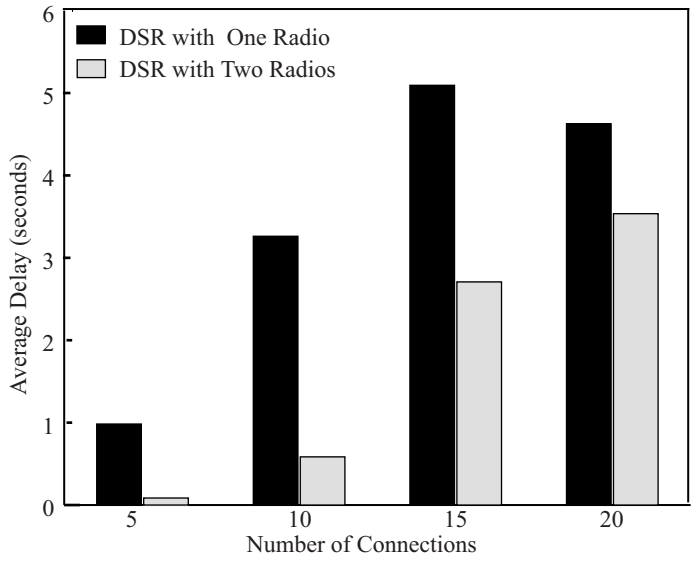

Figure 3.6: Average Delay Vs. Connections

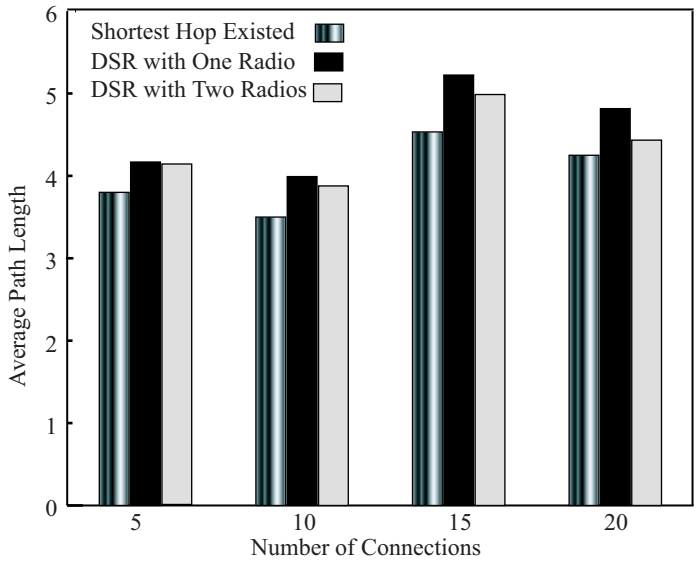

Figure 3.7: Average Path Length Vs. Connections

The experimental result for Packet Delivery Rate is plotted on Figure 3.5. As illustrated, more data packets are lost as the number of connections increases (traffic load increases). Losses may result from unavailable or incorrect routes, overflow of the buffers, and many other reasons. The Packet Delivery Rate for extended DSR on the multi-radio networks is usually better than DSR on single radio networks regardless of the number of connections since more spectrum can be used in multi-radio wireless network scenarios.

Figure 3.6 plots the Average End-to-end delay respectively for DSR with single radio nodes and for extended DSR over multi-radio nodes. The delays significantly vary with the number of connections. As the number of connections increase, medium contention increases causing more delay at each hop and more retransmissions. Although the average delay for extended DSR with multi-radio nodes is quite high for heavier traffic load, it remains significantly lower than with DSR on single radio nodes. On average, the delay is four times smaller.

Figure 3.7 plots the Average Path Length. The leftmost column indicates the average shortest hop count that physically exists based on the perfect and global knowledge of the network topology. The middle column and the rightmost column display the average hop length respectively taken by extended DSR with multi-radio nodes and DSR with the single radio nodes. Results show that extended DSR with multi-radio nodes finds more often the existing shortest path especially when the number of connections is high.

3.4 Discussion

To address the limited capacity problems of wireless mesh networks, equipping multiple radio-interfaces within each node and operating them on orthogonal channels become very feasible to enhance the network performance. In order to route packets in a multi-radio networks, we describe in detail of how to design a new routing protocol called E-DSR based on the traditional Dynamic Source Routing Protocol. The E-DSR protocol is evaluated by comparing its single radio counterpart under the same conditions. Experimental results exhibit an overall performance improvement over DSR with single radio nodes in terms of throughput, end-to-end delay, and packet delivery ratio. Therefore E-DSR is a very encouraging solution to exploit advantages of multiple radios (more wireless spectrum).

Although, the E-DSR in this chapter deals with routing packets in wireless multi-radio mesh networks and uses the number of hop as metric to select a routing path. With the same design principle, it is also easily to be extended to work seamlessly with other quality-aware routing metrics, such as ETX. In this dissertation, the performance of some routing metrics will be evaluated based on this E-DSR routing protocol.

CHAPTER 4

ACCURACY OF LINK LOSS RATE MEASUREMENT

A number of routing metrics have been proposed to select a path with better performance in wireless mesh networks. Most of them partially rely on link loss rates when evaluating the path' performance. Thus accurately measuring the link loss rate is essential to design effective routing metrics identifying efficient paths. Currently, a broadcast based probe mechanism is commonly used by routing metrics to estimate the link loss rates. However, this mechanism often yields inaccurate link loss rates.

In this chapter, we analyze and evaluate the traditional method of measuring the link loss rate, and point out firstly its weaknesses and drawbacks: 1) the probes are sent out with the basic (lowest) data rate which is not the rate used for sending data packets; 2) this method assumes that links are symmetric, i.e., that loss rates are the same in both directions; 3) the frequency of sending out probe packets is one per second over 10 seconds generates a too small number of probes to get accurate link loss rates. For each limitation, appropriate solutions will be proposed in detail. Finally, our proposed modifications will be verified based on extensively experiments.

4.1 Introduction

To design an effective routing metric in wireless mesh networks, it is necessary to capture link characteristics that might affect the path performance. Among all these link characteristics, link loss rate is one of the key properties to affect link and path performance.

Due to the communication nature of these wireless links, the packet loss is ineluctable. Based on IEEE802.11 protocol [16], the MAC layer uses link-level retransmissions for frame losses. Thus

it takes more time to successfully send a packet over a lossy link. This retransmission time not only negatively affects this specific link performance, it also hurts the overall path throughput because other links on the route cannot send any packet while the sender is retransmitting. As a result, these link losses will be felt at application layer in forms of lower throughput and higher end-to-end delays. Therefore, it is essential to accurately capture the loss rates of the links, and avoid highly lossy links when selecting a routing path.

In order to accommodate different channel conditions, it is possible to use different data transmission rates based on different modulation schemes. This multi-rate characteristic has been shown to greatly improve wireless multi-hop network performance [22,23,26]. So, modern wireless radios typically support different rates. Therefore, a wide range of data transmission rates exists simultaneously within one wireless mesh network. Due to the physical properties of wireless radios, a higher data rate usually requires a better channel in order to transmit packets successfully. Thus on average, sending a packet with a lower data rate will more likely be delivered. Most existing routing protocols used in wireless networks, such as DSR [41], AODV [37], and LQSR [13], discover routing paths with a broadcast based method. These routing protocols broadcast route request packets into networks using the basic (lowest available) data transmission rate. As a result, the routing packets sent from the source node normally can be received successfully in the destination node. However, in a wireless mesh network which supports multi-rate, those paths acquired by routing protocols might not be efficient enough to support the actual data transmissions (in general higher than the basic data rate), because the actual higher data rate will yield higher loss rates resulting in higher delivery times. Thus it is necessary to consider an accurate link loss rate when determining a routing path.

Many link quality aware routing metrics have realized the importance of link loss rates, such as *Expected Transmission Count* (ETX) [8], *Expected Transmission Time* (ETT) [13], and improved Expected Transmission Time (iETT) [52]. All of them consider the link loss rate when performing path evaluations. For example, the Expected Transmission Count (ETX) evaluates paths largely based on the link loss rate, and Expected Transmission Time (ETT) actually is the function of the loss rate and the bandwidth on each link, which represents the total transmission time needed to send a data packet successfully over a path. Therefore, it is critical to assess the link loss rate accurately in order to guarantee the effectiveness of these routing metrics.

Currently, most routing metrics use a broadcast based probe method proposed by De Couto *et al.* to measure the link loss rate [8]. In this method, each node broadcasts a link probe packet every second, and every neighbor tracks the number of probes received in the last ten seconds to approximately measure the link loss rate. However, this method often yields inaccurate measurement of link loss rates. Therefore, it is necessary to find out the drawbacks of current measure method and make corresponding corrections since link loss rate is the most important characteristics to design an effective routing metric.

4.2 Observation and Solution

In detail, the simple method used by most current routing metrics to measure individual link loss rate was proposed by De Couto *et al.* We refer this method as Broadcast-Based Probing method (BBP) in this study. According to BBP, each node periodically broadcasts a link probe packet to its neighbor nodes in an average period of τ. In order to avoid synchronization, probes are sent out with a jitter of $0.01 * \tau$. Wireless nodes are required to record the number of probes they have received

during the last ω seconds. Therefore the delivery rate from the sender at time t can be represented as the follows:

$$d(t) = \frac{count(t-\omega, t)}{\omega/\tau} \tag{4.1}$$

Here $count(t-\omega, t)$ is the number of probes received during the window ω, and ω/τ is the number of probes that should have been received.

Since most wireless links are bidirectional, probes' loss rates on both directions are considered to measure the loss rates in this BBP method. Therefore, in addition to tracking the number of probe packets received, every node needs to put its measured loss rate in its own probe packets and send them to its neighbors. Since the broadcast packets are neither re-transmitted nor acknowledged according to IEEE802.11 technologies, a node can measure its link loss rates based on the information obtained from its neighbors and its own records.

Given one link A to B, suppose d_f is the forward probe packets' delivery ratio ($A \rightarrow B$) and d_r is the probability of the successfully reverse delivery rate ($B \rightarrow A$). Then the probability that a data packet can be successfully transmitted via link AB is:

$$p = \frac{1}{d_f * d_r} \tag{4.2}$$

Consequently, the link loss rate for this specific link is $1 - p$.

Although the BBP method is widely used in wireless mesh networks, it has some limitations that may yield inaccurate loss rates. In this section we demonstrate BBP drawbacks using the *ns-2* simulator [53] and explain the reasons for these drawbacks.

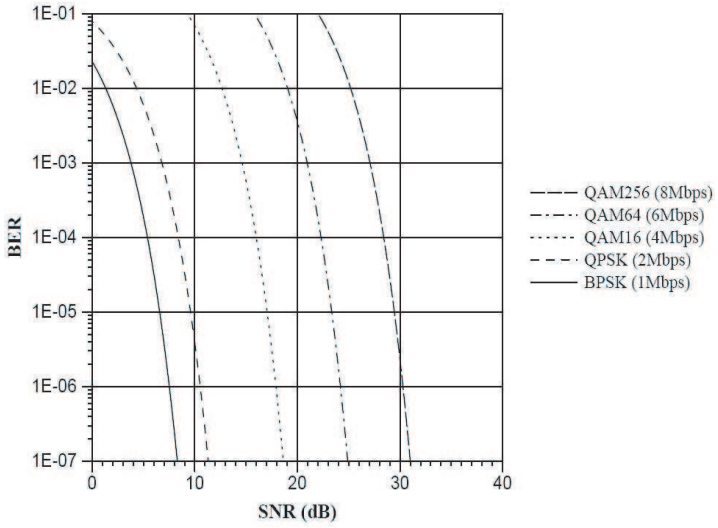

Figure 4.1: BER and As a F80211cardunction of SNR

The first drawback is that BBP assumes that the loss rate of packets sent by the basic data rate is the same as the loss rate for packets sent with other higher data transmission rates: BBP broadcasts probe packets with the basic data rate and measures the frame loss rate for that basic rate. This assumption is not valid. There is some relationship between the bit error rate (directly relevant to the loss rate of link) and signal-to-noise (SNR) for various modulation schemes or data rates [23]. For a given SNR, the modulation scheme with a higher data rate usually yields a higher loss rate, as shown in Figure 4.1 [23]. In other words, given a specific link, sending packets with lower data rates is more reliable (less link loss) than sending packets with a higher data transmission rate.

Wireless Mesh Networks use radios supporting multiple rates. Therefore, depending on channel conditions, distance and modulation schemes, different links will have very different transmission rates. Currently, the IEEE 802.11g standard offers data rates of 6, 9, 12, 18, 24, 36, 48 and 54 Mbps while the earlier IEEE 802.11b standard supports 1, 2, 5.5 and 11 Mbps data rates [18, 20]. The difference of data rates is more prominent in future IEEE 80211n standard, in which data rates

will be in the ranging of 6 Mps to 200 Mbps. According to the parameters from the published specifications of many wireless card providers, such as Lucent ORiNOCO PC card [54], a commonly used IEEE 802.11b wireless card, the sensitivities for successfully received data packets vary for different transmission rates. Therefore, using the basic data rate to probe the link loss rates is not effective on multi-rate networks. As a result, BBP tends to underestimate the loss rates for wireless links. To address this drawback, we propose to send probes with the actual data transmission rates when we probe the link loss rates for wireless links.

The second limitation is that BBP computes the link loss rate based on its probes' loss rates in both directions of that link. According to the equation 4.2 in previous section, BBP assumes that the link is symmetric. Thus for a bidirectional link AB, the link $A \rightarrow B$ and $B \rightarrow A$ hold exactly the same link loss rate with this method. Nonetheless, in real wireless network scenarios, a substantial percentage of wireless links are asymmetrical especially when the distance between nodes is large [55]. So BBP method omits the asymmetric nature of wireless channels.

Furthermore, the successful delivery of a packet requires successful forward delivery and successful acknowledgement delivery. The link loss rate depends on both directions. However, based on IEEE 802.11 standards [16], a receiver sends the ACK at the basic data rate. So ACKs have higher probabilities to be transmitted successfully than normal data packets because data packets are sent with the real data transmission rate. If a data packet can be transmitted successfully in forward link with higher data rates, the chance for the ACK to get through (on the reverse direction) is very high. To take wireless links' asymmetry and ACKs transmitted with basic data transmission rate issues into account, it is more appropriate to count only one way link probes to measure the link loss rates. By doing this, the link loss rate from A to B is different from the link loss rate from B to A.

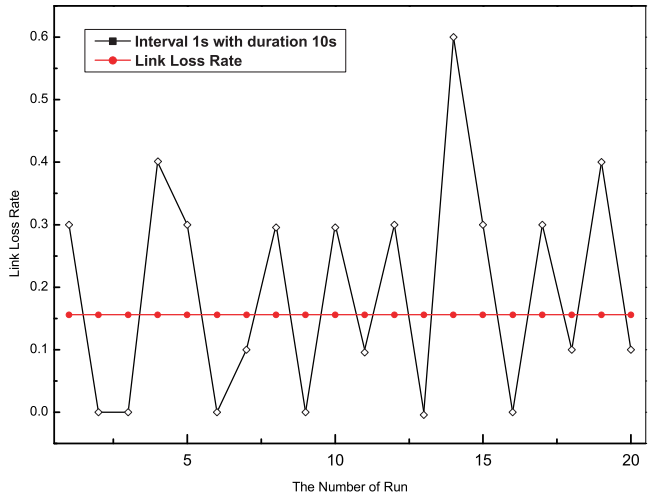

Figure 4.2: Measured Link Loss Rate with 1 Second Interval

The third limitation of BBP lies in its implementation. BBP sends probes every second (interval) to its neighbors, and the time (duration/window size) of sending probes is 10 seconds. To verify whether these parameters (interval and duration) are reasonable enough to measure the link loss rate accurately, the following experimental settings are established. Two wireless nodes (S and D) are placed at a distance of 250 meters. Both Node S and Node D are configured with one wireless radio supporting 802.11g. To find out whether interval one second with 10 seconds duration is appropriate to get accurate loss rates, we conduct experiments with interval 1 to probe the link loss rate from Node S to Node D. In this experiment, the actual data transmission rate is used to send probes and the probing duration is set to 10 seconds. 20 separate experiments are conducted. For each experiment, one individual link loss rate value is obtained. Figure 4.2 plots the link loss rates achieved from these 20 independent experiments, each point represents one experimental result. In this figure, the vertical axis represents the link loss rates and the horizontal axis stands for the experiment number. The line marked with • indicates the real loss rate of the link, the line marked with ◇ is the loss rates measured by these 20 independent experiments. According to the results, BBP

is prone to lead to a wide range of different loss rates from 0.0 to 0.6 for measuring one specific link. Apparently, the loss rates obtained by a one second interval with a duration of 10 seconds are inaccurate. Given the same link, the measured loss rate is not consistent and in real case only one measurement is used.

To detect how different intervals affect the accuracy of link loss rate measurements, various interval parameters are tested, which include 1s, 0.5s, 0.25s, 0.1s, 0.05s 0.025s, 0.01s, and 0. 005s. Like the former experiments, the actual data transmission rate is used to send probe packets and the duration is set to 10 seconds. According to our results, the interval 0.5s and 0.1s are still not effective enough to measure link loss rate accurately. Interval 0.005s is effective. To make our the figure clearer, Figure 4.3 only plots the correspondent link loss rates acquired by intervals 0.25s, 0.05s, 0.025s and 0.01s. By observing the experimental results from this figure, link loss rates can be calculated precisely when setting intervals to 0.025s or 0.01s. Although setting interval 0.01 is more precise than using interval 0.025, the result with 0.025s is already reliable enough since the maximal deviation getting from interval 0.025s is only 0.015.

Another way to measure the link loss rate more accurately is to use durations(windows) longer than 10 seconds. Actually, the accuracy of the link loss rate significantly depends on the number of probes sent out. If only a total of 10 probes is sent out (interval of 1s and the duration is 10s), whether one probe packet is lost greatly affects the final link loss rate. However if we send more probes in one unit time or in a longer duration, one or two single packet's loss will have a small effect on the overall result, which roughly leads to more accurate measurements. For example, Figure 4.4 demonstrates the link loss rate measured by using an interval of 1 second with a duration of 400 second to test the link loss rates with 20 separate experiments. Obviously, a longer duration yields better accuracy.

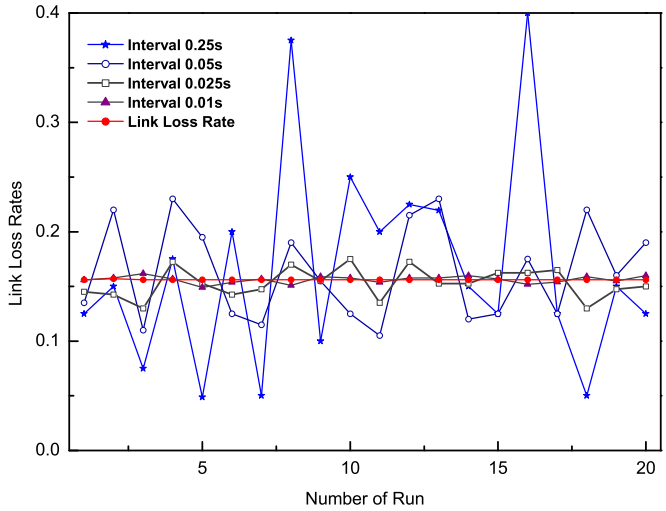

Figure 4.3: Measured Link Loss Rate with Different Intervals

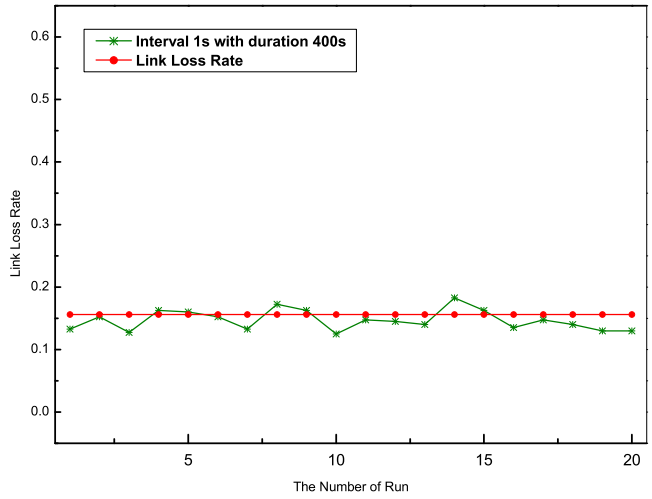

Figure 4.4: Measured Link Loss Rate with Longer Duration

4.3 Performance Evaluation

In this section, the performance of the enhanced versions of protocols with the new scheme is evaluated, and they are compared to the base protocols. The simulation environment is explained, and the simulation results are presented and discussed.

4.3.1 Experimental Setting

Basically, the effectiveness of our proposed solutions to address BBP's drawbacks are evaluated using *ns-2* simulator [53]. To make our experimental setting close to real systems, the parameters used in the simulator are based on the specification published on Cisco website [56]. Table 4.3.1 lists the sensitivities used for each individual transmission rate. Two wireless nodes (source node S and destination node D) are used in our simulation; the distance between the two nodes is set to vary from 210 to 300 meters in steps of 10 meters (according to the parameter shown in table 1, data rate transmitted with this range is typically 24 Mbps). Within each wireless node, an IEEE 802.11g radio is configured.

For each distance between Node S and Node D, we use the following five protocols to measure the link loss rates independently. 1) Original BBP: it uses the basic data rate to send probes with an interval of 1 second in a duration of 10 seconds, and the link loss rate is measured based on probe delivery rates in both directions; 2) BBP-I: it uses the basic data rate to send probes with an interval of 0.025 second in a duration of 10 seconds, and the link loss rate is measured based on both directions; 3) BBP-II: it uses the actual data rate to send probes with an interval of 1 second in a duration of 10 seconds and the link loss rate is measured based on the number of received packets in only the forward direction; 4) BBP-III: it uses the actual data rate to send probes with an interval of 0.025 second in a duration of 10 seconds and the final link loss rate is measured based on both

Parameter	Description
Transmission Power	10 dbm
54Mbps Sensitivity	-68 dbm
48Mbps Sensitivity	-71 dbm
36Mbps Sensitivity	-75 dbm
24Mbps Sensitivity	-79 dbm
18Mbps Sensitivity	-82 dbm
12Mbps Sensitivity	-84 dbm
9Mbps Sensitivity	-87 dbm
6Mbps Sensitivity	-89 dbm

Table 4.1: Wireless Interface Parameter for Cisco Card

directions of that link; 5) BBP-final, it is a BBP protocol with all modifications. BBP-final adopts all the proposed solutions explained in the previous section. So the difference between BBP-final and BBP-III is that BBP-final method measures the link loss rate only based on the number of received probes in forward direction.

The actual link loss rate is measured by a real CBR traffic initiated from Source node S to Destination node D. Since 802.11 MAC scheme uses link level acknowledgement and retransmission mechanisms to mask data packet losses for the application layer. So from the application layer point of view, links are packet-loss free with the price of low throughput. To calculate the actual link loss in real applications, we disabled this retransmission scheme. By checking the trace file, the actual link loss rate can be successfully and precisely measured.

4.3.2 Performance Result

The experimental results are shown in Figure 4.5. Here the vertical axis represents the link loss rates measured and the horizontal axis stands for the distances between these two nodes. Totally six curves are plotted in this figure, and the curve marked with • is the real link loss rates for each specific distance. The result measured with the original BBP sending probes with basic data rate

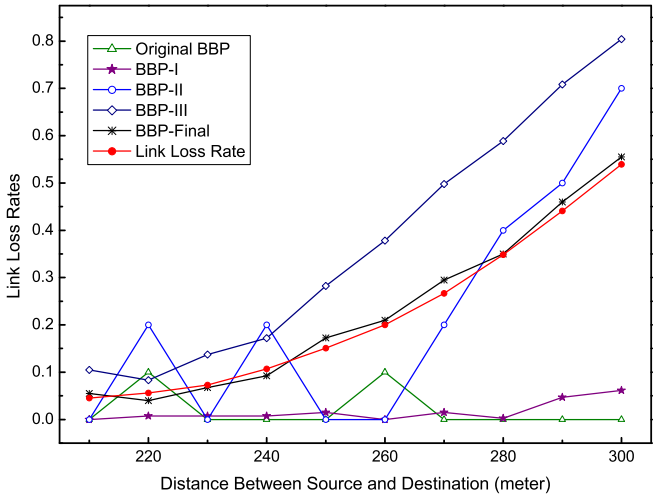

Figure 4.5: Link Loss Rates Based on Different Measure Protocols

is shown by the curve marked with $\star$ that obviously underestimates the link loss rates, because the required sensitivities to receive a packet with the basic data rate is much lower than the sensitivities needed to decode a packet sent with a higher data transmission rate. The same conclusion can be drawn from the curve marked with $\triangle$, which also sends probes with the basic data rate. Meanwhile, as revealed by the curve marked with $\circ$, when sending probes with actual data transmission rates, the number of probes is also an important factor to affect the accuracy of the link loss rates' assessment. The BBP implementation uses one second as the interval and 10 seconds as the duration. These values are not effective to get accurate link loss rates. We also find that calculating the loss rate based on the link quality of both directions leads to overestimating the real link loss rate according to the curve marked with $\diamond$. As a conclusion, to address all these drawbacks, our solution measures the link loss rate accurately, as shown by the curve marked with $*$.

4.4 Discussion

In this chapter, Broadcast Based Probe method (BBP), a commonly used method by many routing metrics to measure the link loss rates, has been extensively investigated. Based on *ns-2* simulations, we found that the accuracy of BBP is low in wireless networks supporting multiple rates. The limitations of the BBP method are mainly due to overall three aspects: 1) the probes are sent out with the basic data rate; 2) Deriving the link loss rate based on the link quality of both directions of the link is not necessary; 3) The number of probes(one second interval with window size 10 second) is not sufficient to deliver accurate loss rates.

To take all the limitations into consideration, we propose 1) to use actual data rates when sending probe packets to neighbors, 2) to send more probes, and 3) to compute the loss rate based on the forward link only. Our simulation results suggest that these modifications make BBP more accurate. In the following chapter, all routing metrics will use this modified BBP to calculate link loss rates.

CHAPTER 5

iETT: IMPROVED EXPECTED TRANSMISSION TIME METRIC

Different routing metrics have different performances in routing data packets and lead to different overall capacity for the same wireless mesh networks [11]. Therefore, routing metrics are critical for selection of a path with maximum throughput in wireless multi-radio multi-hop mesh networks. Although a number of different routing metrics have been proposed in single radio wireless multi-hop networks, most of them ignore three basic characteristics of a routing path that might greatly affect overall path performance.

In this chapter, we design a new routing metric called iETT, it aims to address three key characteristics of a routing path that other current routing metric ignores: 1) Transmission time needed to transmit corresponding overhead along with each data packet delivery; 2) a high discrepancy in packet loss rates of links; 3) the position of the links with different packet loss rates. By capturing these three characteristics, the iETT metric is able to choose a route with a better performance than other popular routing metrics, such as *shortest path* (HOP), *Expected Transmission Count* (ETX) and *Expected Transmission Time* (ETT) metrics.

5.1 Introduction

Locating a high performance routing path has been an active area of research in single-radio wireless mesh networks for many years. To find such an efficient path, a good routing metric is significant. Currently, the most widely used routing metric in multi-hop wireless networks is the minimum hop count metric, which is also used by most of existing routing protocols such as DSR [41], AODV [37]. The primary advantage of the minimum hop count metric is its simplicity;

and no extra measurements or overhead are needed for selecting the appropriate route. However, it has been shown that a route with minimized hop count does not necessarily guarantee the high throughput [8, 9]. This is due to the fact that the minimum hop count metric does not consider any link quality of a routing path.

Researchers have proposed routing metrics that attempt to capture the link quality. Adya *et al.* proposed the *Per-hop Round Trip Time* (**RTT**) [12] metric that measures the round trip time between pairs of neighboring nodes. De Couto *et al.* proposed the *Expected Transmission Count* (**ETX**) as a measure of link quality of a route based on the packet loss rate on each individual link. Draves *et al.* developed the *Expected Transmission Time* (**ETT**) [13], which is the function of the loss rate and the bandwidth on each link. The sum of *ETT* at each link over a path is used to evaluate the path. *RTT*, *ETX* and *ETT* metrics all capture the quality of a path better than the simple hop count metric and yield better performances. However, these metrics are not effective when there is a high discrepancy in the quality of the links on a given path.

A path with many excellent links and one bad link may appear good with *shortest path*, *RTT*, *ETX* or *ETT* metrics: the scores of the excellent links may smooth out the score from the bad link. The problem is that the bad link will negatively impact the quality of the path more than what the metric captures in reality. Additionally, the way to calculate the metric value is not accurate because most of routing metrics do not consider the MAC overhead that are transmitted along with each actual data packet.

Therefore, we proposes a new routing metric named *Improved Expected Transmission Time* (**iETT**). The iETT metric is designed to take into account the discrepancy of link loss rates within one path, as well as the MAC layer overheads when computing an expected packet transmission time (instead of simply using packet/bandwidth).

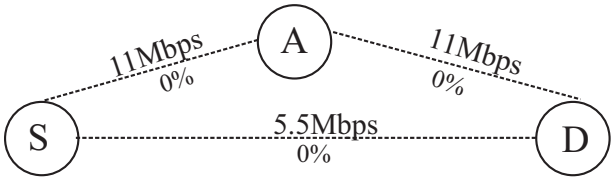

Figure 5.1: Basic Wireless Network Scenario

5.2 Design of iETT

5.2.1 Transmission Time for Overhead

In multi-hop wireless networks, the expected transmission time to send one data packet is very important to evaluate the performance of one link. Most current routing metrics use this value to assess different paths before making route selections. In most cases, these routing metrics simply use S/B (where S is the packet size and B is the data transmission rate) to compute the expected transmission time for one hop transmission. However, in IEEE 802.11 networks, the time to successfully transmit a data packet is related to not only the packet size and transmission rate of the link, but also control frame overhead that are sent along with each data packet. Without considering the overhead, the expected transmission time may not be an accurate indicator to evaluate the path performance.

For example, Figure 5.1 deploys a simple network scenario. In this scenarios, if we simply use S/B to computer the expected transmission time, Path $S - A - D$ has the same expected transmission time as path $S - D$ and these two paths are expected to have similar performances. However, according to on*ns-2* simulations [53], with a CBR traffic of 1500 bytes packet, the path $S - D$ yields more than a 30% higher throughput against path $S - A - D$. This performance discrepancy indicates that taking overhead into account is necessary to estimate the expected transmission time.

To consider the overhead when computing the expected time for one data transmission, we need to find out the exact overhead along with each data packet. For a UDP application data packet, we need to add 8 bytes of UDP header, then 20 bytes of IP header and 4 bytes of link layer header. When the encapsulated data packet goes to the MAC layer, a 34 bytes of MAC frame header is appended. With the IEEE802.11 standards, the MAC layer access mechanism requires extra time to access the medium. For instance, any node needs to sense the medium first and waits a DIFS time slot before sending out any frames; a node also needs to wait an IFS time slot to send back an ACK packet. In case the RTS/CTS scheme is enabled, the time to transmit RTS/CTS packets also has to be considered. In multi-rate wireless networks, except for the data packets, the basic control frames including RTS/CTS, PLCP header are all sent with the basic data rate of 1 Mbps regardless of the actual data transmission rate. Figure 5.2 and Figure 5.3 shows the overhead in an IEEE 802.11b wireless network with a data packet size of 1000 bytes. Apparently, the almost fixed amount of medium time caused by overhead introduce a considerable dependency on the whole data packet transmission time, especially for a higher data rate.

Here, let TTPD be the expected successful transmission time per data packet. Based on IEEE 802.11b MAC scheme, *TTPD* can be represented as follows:

$$TTPD = \begin{cases} T_{difs} + T_{rts} + T_{cts} + T_{data} + T_{ack} + T_{backoff} + 3 * T_{sifs} & \text{with RTS/CTS enabled} \\ T_{difs} + T_{data} + T_{ack} + T_{backoff} + T_{sifs} & \text{with RTS/CTS disabled} \end{cases} \tag{5.1}$$

In the above equation, $T_{data} = PLCP/1M + 8 * (\text{MAC header} + \text{other headers} + data)/B$, B denotes the actual data transmission rate, and PLCP is the physical layer header transmitted at the basic data rate of 1 Mbps. Since all the components in the two equations are known based on

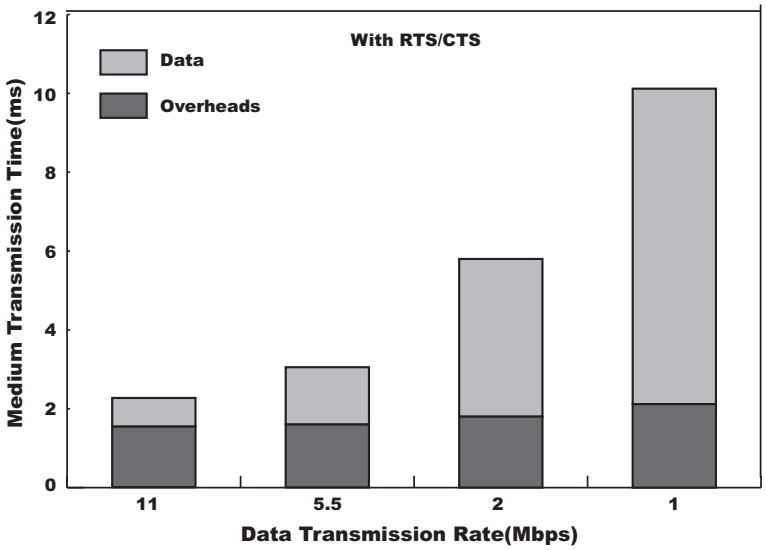

Figure 5.2: Overhead Per Data Packet With RTS/CTS

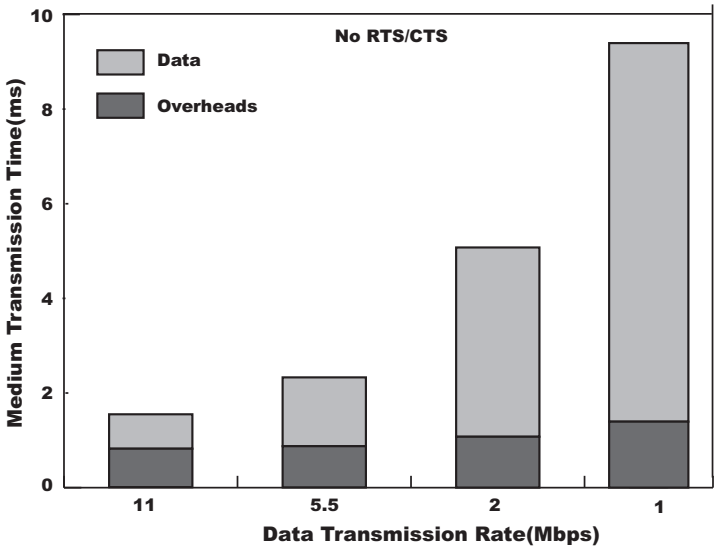

Figure 5.3: Overhead Per Data Packet Without RTS/CTS

Data Rates	RTS/CTS		CSMA/CA	
	α	β	α	β
11Mbps	0.727	1536	0.727	812
5.5 Mbps	1.455	1594	1.455	870
2 Mbps	4	1798	4	1074
1 Mbps	8	2118	8	1394

Table 5.1: Parameters for Transmission Time Based on IEEE802.11b

IEEE802.11 standard, the theoretical transmission time to send a packet can be denoted:

$$TTPD = \alpha \times S + \beta \tag{5.2}$$

where S is the actual data packet size. Table 5.1 lists the parameter sets for different MAC access schemes and data transmission rates. Instead of simply using S/B to get expected transmission time to evaluate one hop link, we propose to use TTPD to more accurately calculate the transmission time for the packet. Note that [57] also uses a similar formula to calculate the expected transmission time, but their calculation is not as accurate as ours, since it incorrectly considers ACK and overlooks other layer overhead.

5.2.2 Intra-path Interference

In a single-radio wireless network, all nodes share the same channel to communicate with each other. Thus, for a routing path with multiple links, those links in each other's interference ranges compete for the same medium resource to send their data packets. This intra-path interference is unavoidable since one data packet has to travel through all the links within one path to its destination. Current routing metrics do not consider the intra-path interference, because they only evaluate

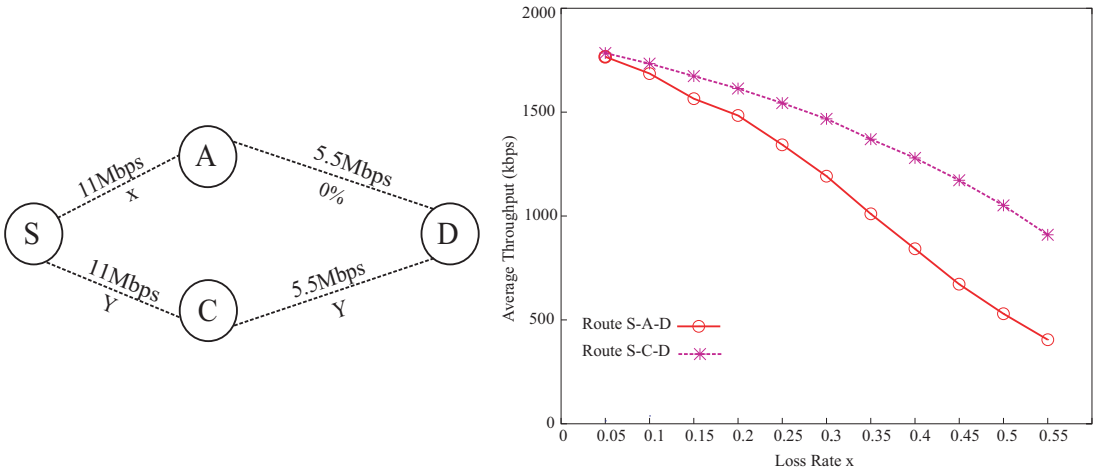

Figure 5.4: Intra-path Interference-Scenario I

individual links independently and simply sum these individual metrics up to compose a path metric. Therefore, a path with many excellent links and one bad link may appear good since the scores of the excellent links may smooth out the score from the bad link. In reality, however, due to the intra-path interference, the bad link will negatively impact the quality of the path more than what the metric captures.

Two simple network scenarios are explored to underline the importance of the intra-path interference when evaluating a route performance. The first scenario is shown in Figure 5.4, in which X and Y represent loss rates of the indicated wireless links. Two different paths exist from node S to node D: path $S - A - D$ and $S - C - D$. Path $S - C - D$ has equal loss rates Y on its both links. We intentionally set loss rate Y based on X value so that the path $S - C - D$ has the same expected transmission time as the path $S - A - D$ ($Y = \frac{X}{3-2X}$). Thus, according to the *ETT/MTM* metrics, we expect these two routes to have similar performances. Using *ns-2* simulator [53], we have established one Constant Bit Rate (CBR) traffic between nodes S and D with a packet size of 1500 bytes. For each experiment, the traffic was exclusively forced through one given path. The

throughput achieved by routing packets via path $S - A - D$ and path $S - C - D$ are independently recorded. For each set of experiments, we varied the loss rates X and Y and plotted our result in Figure 5.4 on the right panel, with the loss rate and the average throughput plotted on the x- and y-axis, respectively. The curve marked with $\star$ indicates the throughput obtained by the path $S - C - D$ while the curve with $\circ$ is for path S-A-D. Apparently, path $S - C - D$ outperforms path $S - A - D$ by achieving a higher throughput, especially when the loss rate X is high.

Another network scenario example is shown in Figure 5.5. Like our previous example, two different routes from node S to node D exist. These two routes have equal *ETT* metric value, indicative of identical/similar performance. However, the simulation results show that these two paths actually have very different throughput performance. Same as Scenario 1, we respectively collected the throughput achieved by the paths $S - A - D$ and $S - B - D$ while we varied the link loss rate X value in the link with 11 Mbps data transmission rate. The curve marked with $\bullet$ indicates the throughput obtained by the route $S - B - D$ and the curve with $\circ$ represents the throughput achieved by route $S - A - D$. Thus, the result shows that without considering the intra-path interference, ETT/MTT metrics can not differentiate these two routes. In reality, however, the discrepancy between the two paths' performances becomes dramatically as the loss rate X increases.

The two examples above indicate that: (1) the path containing links with different loss rates might achieve lower throughput performance than the path containing links with similar loss rates, (2) the position of the link with different loss rates might affect the path performance. Therefore, intra-path interference is essential for the wise selection of a routing metric.

The IEEE802.11 MAC protocol provides throughput fairness to those wireless links within each other's interference range in the absence of packet loss [58]. Regardless of data transmission

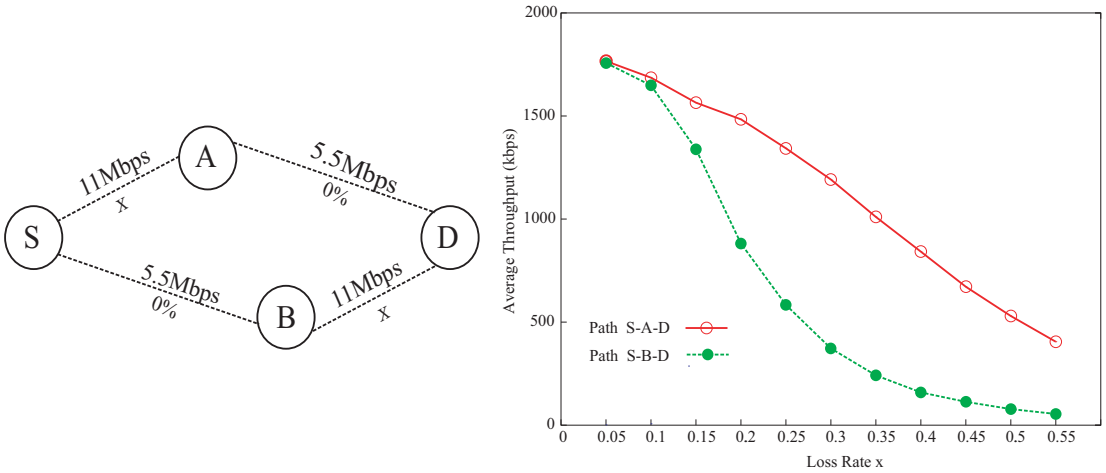

Figure 5.5: Intra-path Interference-Scenario II

rates, the throughput fairness guarantees that individual senders within the same interference range have the same chance to send data packets. However, if the links within one single path have very different loss rates, the one with a lower loss rate has a larger chance to grab the medium than the links with higher loss rates. This is due to the nature of the MAC access scheme: when a node fails to send one packet, it will always double its contention window size since it can not differentiate whether the loss is caused by collision or channel loss. In the end, the link with the higher loss rate has a larger contention window size leading to a low chance of getting the channel. Since a packet needs to go over all the links within one path, the link with the highest loss rate may become a bottleneck that hurts the overall path performance. When all the links within one single path have the similar loss rates, they will have the statistical equal chance of accessing the medium. Thus no adverse bottleneck will be presented and higher path performance is achieved. Thus, Path $S-C-D$ has an averagely better performance than path $S-A-D$.

As shown in Figure 5.5, the position of link with different loss rate may affect the path performance differently. Apparently, path $S-B-D$ has the link with loss rate closer to the destination

than path $S - A - D$, $S - B - D$ achieves a lower throughput because packets lost on the link $B - D$ already (unnecessarily) use the medium on link $S - B$: there is a waste of resources on link $S - B$ when a lossy link is downstream. Reversely, on path $S - A - D$, the packets that succeed on link $S - A$ will succeed on link $A - D$ (0% loss rate). Thus path $S - A - D$ performs better than path $S - B - D$. We use the position of the link with the highest loss rate and the position of the link with the lowest loss rate to approximately simulate this impact later.

In summary, in order to address these intra-path interferences caused by the uneven loss rate links, we propose to add extra medium delay to approximately simulate these two effects. We call this link interference delay caused by intra-path interference as *LID*. The *LID* is denoted as follows:

$$LID = \begin{cases} (P_{max} - P_{min}) \times TTPD_{max} & \text{if } l_{max} \geq l_{min} \\ (P_{max} - P_{min}) \times TTPD_{max} + (P_{max} - P_{min}) \times TTPD_{min} & \text{if } l_{max} < l_{min} \end{cases} \tag{5.3}$$

where P_{max} represents the maximal loss rate and P_{min} stands for the minimal loss rate in the path. When Link $l_{max} \geq l_{min}$, it means that the position of link max with the highest loss rate is before the position of link min with the lowest loss rate in the path. On the other hand, if Link $l_{max} < l_{min}$, it stands for that the position of link max is after that of link min in the path.

Combined with individual link' *TTPD* , we propose a routing metric named improved Expected Transmission Time metric (iETT) that considers the intra-path interference, which can be presented as follows:

$$iETT = \sum_{i=1}^{n} (TTPD_i \times ETX_i) + LID \tag{5.4}$$

LID captures the position of the link with the highest loss rate and the position of the link with the lowest loss rate to approximately quantify the extra delay caused by the intra-path interference.

Parameter	Description
Transmission Power	15 dbm
11Mbps Sensitivity	-82 dbm
5.5Mbps Sensitivity	-87 dbm
2Mbps Sensitivity	-91 dbm
1Mbps Sensitivity	-94 dbm
Carrier Sense Threshold	-108 dbm
Capture Threshold	10

Table 5.2: Wireless Interface Parameter for ORiNOCO Card

With *LID*, the link loss rate distribution on all links as well as the approximate bottleneck position within one single path are addressed.

5.3 Performance Evaluation

This section evaluates the performance of the proposed *iETT* metric and compares iETT with other existing routing algorithms under different network scenarios. The simulation tool we used is *ns-2* on version 2.30 [53].

Currently *ns-2* simulator have no built-in multi-rate support for wireless mesh network simulation. We implement a basic multi-rate on MAC layer scheme based on RBAR algorithm [59]. The default physical layer parameter on *ns-2* are from old release of Lucent Wavelan card working at 900 MHz. To make our simulation more realistic, we apply the real physical interface parameters from one of current vendor providers. Thus, the 802.11 MAC and physical wireless parameters in table 5.3 will be modified to match the published specifications of a Lucent ORiNOCO PC Card [54].

Compared with the simulations in wired networks, it is more complicated to do simulations in wireless networks since an appropriate wireless channel model is very important. Wireless channel models normally are refereed as propagation models. When a data packet is transmitted via wireless channel, the signal strength is declined from wireless radio propagation phenomena by a number of

effects. These effects include distance and other multi-path fading, such as reflection, diffractions, and scattering. Therefore the received signal strength is actually a composition of a line-of-sight (LOS) component and the multi-path fading. Currently, different radio propagation models are used to predict or model the received signal strength at a known distance in a specific environment. The most widely used propagation models are two-ray ground [60, 61] and ricean model [62]. Two-ray ground propagation model measures the received power based on the direct line of sight path as well as the path including one reflection on ground between sender and receiver. And ricean propagation model models time-correlated variations of the receiver's signal strength. In official ns-2 release versions, ricean fading model has not been included. Therefore, we embed ricean model support based on an implementation available from here [63].

Before we start to run our experiments with multi-rate support, we use single rates to obtain the correspondent transmission range in each single data transmission rate based on parameters shown in table 5.3.To achieve this, we set up two nodes, one is the receiver and the other node acts as sender. We enable the sender to keep sending CBR data packets with a constant data transmission rate at a time. The rates we use include 11Mbps, 5.5Mbps, 2Mbps and 1Mbps in turn. For each experiment, we specify the distance of these two nodes. And the distance will be varied from 200 meters to 800 meters. Both two way ground propagation model and ricean model are applied.

For two ray ground propagation model, Figure 5.6 plots the experiments' result and shows the correspondent number of data packets received by different transmission rates based on the different distances between sender and receiver pair. Obviously, the two-ray ground model is deterministic in the possibility of receiving a packet, which means with one specific range all packets sent from the sender are either all received or all discarded. There is no any other situation in between. For

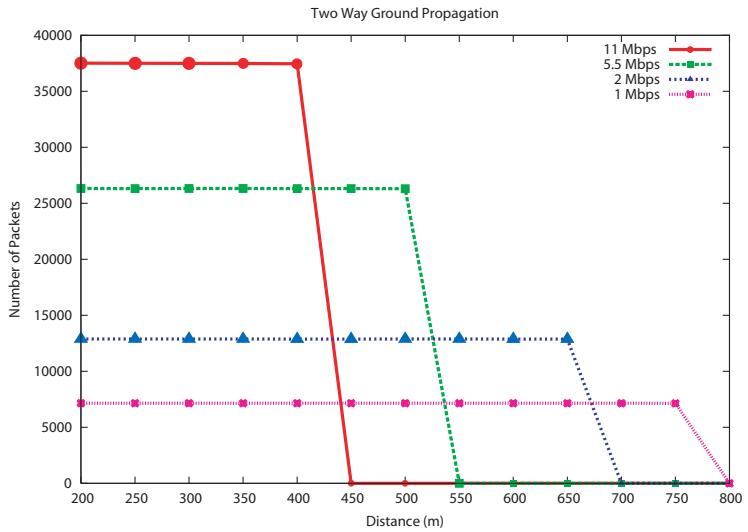

Figure 5.6: Two Ray Ground Propagation Model

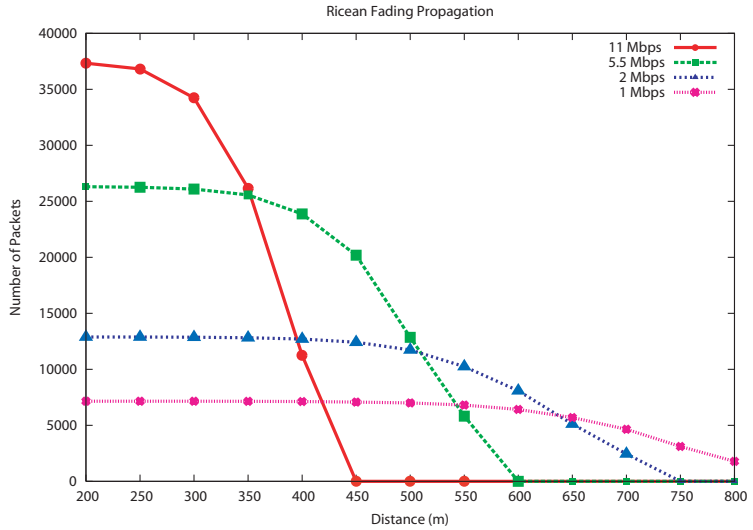

Figure 5.7: Ricean Fading Propagation Model

example, when sender sends packets with a data rate of 11 Mbps, the packets can be received successfully when the distance between the sender and the receiver is less than 350 meters, however no packet can be received successfully starting from 400 meters distance. Apparently, two ray ground propagation model can be considered to be a deterministic propagation model. It always predicts the same signal strength for a specific distance. Therefore, when two nodes are in transmitting, they are either in the transmission range which means they can always receive all packets successfully or out of the transmission range which means they always fail to receive any packet successfully. No packet loss will be automatically occurred in this propagation model. However, wireless packet losses cannot be avoided in a real wireless communication. So, two ray ground propagation model does not have the loss in nature. To simulate wireless losses, a corresponding wireless loss model working with two ray ground model is necessary.

Figure 5.7 plots the experimental result under ricean fading model. Observed from this figure, different from two-ray ground model, ricean fading model is a probabilistic model. It has a random component leading to a possible variation for the received signal strength at any specific distance. Therefore, for a receiver and sender pair with specific distance in between, one packet only can be received successful with some possibilities. For example, when the sender transmits packets with a data rate of 5.5Mbps, the packets can be possible acquired successfully at a distance of 600 meters with some packet loss rates. Therefore, to initiate our experiments on ricean fading models, we do not need to manually add packet loss rates.

5.3.1 Experimental Setting

In our experiments, we utilized Constant Bit Rate (CBR) as our traffic generation model. Data packets will be sent out with a deterministic interval rate (0.5 ms). Only one CBR traffic

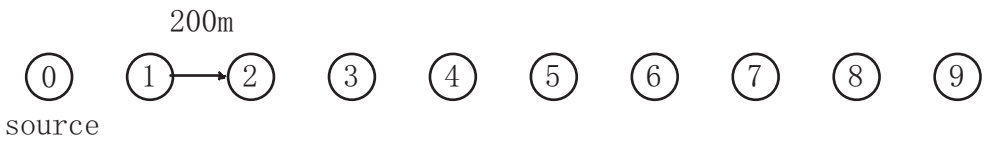

Figure 5.8: iETT: Network Topology for Experiment I

transmission is active at any time. All wireless nodes in our experiments are equipped with one IEEE80211b [18] radio interface. Each radio interface is able to operate at one of the four available IEEE802.11b transmission rates: 1, 2, 5.5, or 11 Mbps. The transmission power is fixed at 15 dbm shown in Table 5.3. *HOP* metric, *ETX*, *ETT*, and *iETT* will be seamlessly incorporated with E-DSR routing protocol described in Chapter 3. The propagation model we used is ricean model (We also applied two ray ground model with random generated link loss rates and achieved similar conclusions).

Two important performance metrics are measured to evaluate the effectiveness of these routing metrics. One is average throughput, it represents the total amount of data packets successfully received at destination node during the unit experiment time. The other performance metric is the average end-to-end delay, which measures the average packet delay for all received data packets from source to destination.

5.3.2 Performance Result

Experimental Scenario I

First we intentionally set up a simple topology with 10 wireless nodes placed in a chain line, where the neighbor nodes are separated with a distance of 200 meters. In this experimental scenario shown in Figure 5.8, neighboring nodes can communicate with a data rate of 11 Mbps, a long link with distance 400 meters can support 5.5 Mbps, and a longer link with distance 600 meters is able to

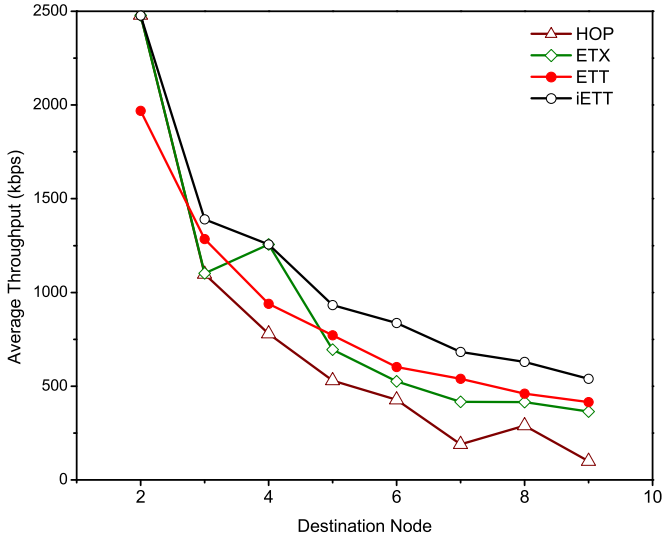

Figure 5.9: iETT: Throughput on Linear Topology (kbps)

give 2 Mbps data transmission rate. Here we always set node 0 as source node, and initiate a CBR connection to the destination node, where destination nodes start from node 3 to node 9. For each experiment, we use *HOP*, *ETX*, *ETT*, and *iETT* metric to carry out our experiments. The packet size of 1000 bytes is evaluated in these experiments. Similar experimental results are obtained with different packet size values. For each pair of connections, we repeat experiments for 5 times for each individual routing metric and average the results.

Figure 5.9 and Figure 5.10 demonstrate the experimental results based on ricean model for average throughput and average end-to-end delay, respectively. In both figures, four curves are plotted. The curves marked with $\triangle$ always stand for performance achieved by HOP metric, and the curves marked with $\diamond$ is for *ETX* metric, the curves along with $\bullet$ represent the result of *ETT* metric and the ones with $\circ$ correspond to the *iETT* metric. Here x-axis represents the destination node (corresponding to different distance). As observed in these figures, Our *iETT* metric is always

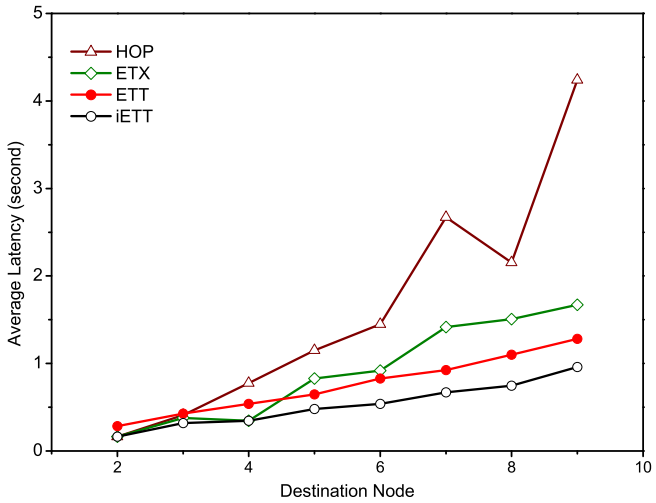

Figure 5.10: iETT: Average Latency on Linear Topology (s)

able to find paths with the highest throughput and the lowest average end-to-end latency in all connection pairs whereas HOP metric has averagely the worst performance since it only considers the path length.

Experiment Scenario II

A randomly generated topology is used as our second experiment's scenario, in which 50 wireless nodes are spread freely in a field of 5000m*5000m. To vary the load in the experiment, the CBR data packet sizes are varied from 500 to 2000 bytes. In this experiment, we totally have $50 * 49 = 2450$ different sender and receiver pairs. Among those pairs, 50 pairs (sender-receiver) are randomly selected and one CBR traffic is established for each pair once a time. Using exactly the same conditions, *HOP*, *ETX*, *ETT*, and *iETT* metrics are separately evaluated. In the following

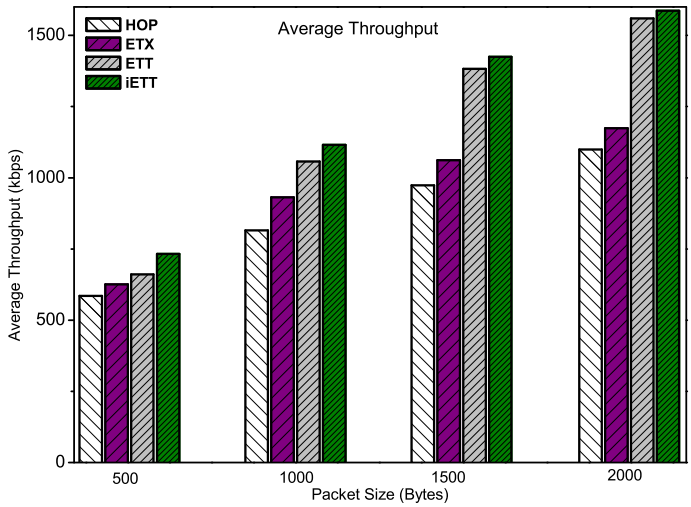

Figure 5.11: iETT: Throughput on Random Topology (kbps)

figures, each data point represents the average value achieved by those 50 connection pairs with different metrics. For fair comparison, identical conditions are used for all metrics.

The experimental results based on ricean fading model are shown in Figure 5.11 and Figure 5.12 with x-axis standing for the packet size. Obviously, these results confirmed that *iETT* metric has the best performance in terms of throughput performance and average packet latency. Since *iETT* captures more link quality information, it has better performance than other metrics, and as usual, the worst routing metric is *HOP* metric.

However, observed from these two figures, it seems that *iETT* has not much improvement compared with *ETT* metric, especially when the packet size is large. This is due to the fact that *iETT* and *ETT* metrics select the same path for some pairs among the 50 randomly connections. For instance, when using a CBR traffic with a packet size of 500 bytes, we found that 23 pairs out of 50 were using the same paths based on *ETT* and *iETT* metrics. To see what exactly improvement we have

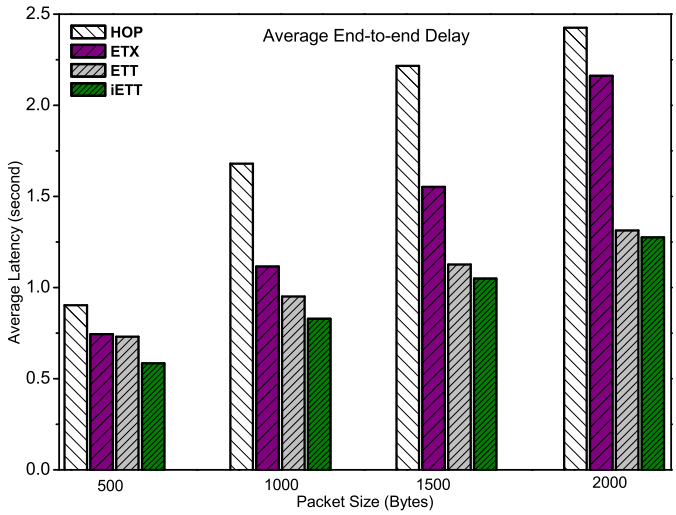

Figure 5.12: iETT: Average Latency on Random Topology (s)

obtained with the *iETT* metric over *ETT* metric, we only focus on those pairs that select different paths to evaluate the performance. Therefore, in the following figure, each data point represents the average throughput improvement (100%) value achieved by our iETT metric (only for those connection pairs choosing different routes with *iETT* and *ETT*). The average highest throughput we can obtained from *iETT* metric against *ETT* can be as high as 35% when we set packet size to 500 bytes. Apparently, as the packet size is larger, the less throughput enhancement we can achieved by using *iETT*. This is because *iETT* considers the overhead to calculate the path's performances. When the data packet size is small, the overhead will take a higher percentage of the overall transmission time compared with large data packet size and the time of transmitting overhead is more important. Therefore, *iETT* can get more performance achievements to select more efficient path in case of small data packets.

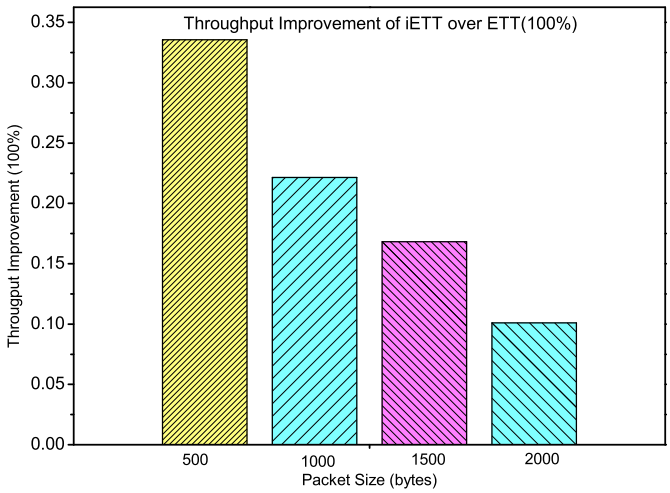

Figure 5.13: iETT: Average Throughput Improvement over ETT (100%)

5.4 Discussion

This chapter discusses the weaknesses of major routing metrics and illustrates them using *ns-2* simulations in single radio networks. Existing routing metrics ignore two characteristics of a path that may dramatically affect its performance: the discrepancy in loss rates and the position of links with different loss rates. Also most of them ignore transmission time to send MAC overhead along with each data packet.

To take these three factors into account, we designed a new quality routing metric called *improved Expected Transmission Time*(iETT) and compared its performance against *shortest path* (HOP), *Expected Transmission Count* (ETX) and *Expected Transmission Time* (ETT) metrics on a chain topology as well as a random topology. Extensive *ns-2* simulation results prove that the *iETT* metric outperforms all other metrics by improving average network throughput and reducing the average packet latency.

CHAPTER 6

iBATD: IMPROVED BOTTLENECK AWARE TRANSMISSION DELAY METRIC

An easy and relatively low-cost approach to address limit capacity problem in wireless mesh networks that has recently been proposed is to equip multiple wireless radios per wireless node. Operating the radios within each node on non-overlapping channels can use the radio spectrum more greedily and thereby reduce interference and contention. In such a multi-radio wireless mesh network, an efficient routing metric should deal with channel diversity (intra-flow interference) in order to make a routing path selection with high performance.

The work in this chapter is inspired by the observation that current routing metrics are not effective enough in differentiating candidate routes having different performance, especially when mesh nodes are configured with more than two radio interfaces. Thus, we propose a new routing metric called Bottleneck Aware Transmission Delay (BATD) in multi-radio wireless mesh network. For EACH channel within one path, the *BATD* metric calculates the summation of transmission time from the links within the same carrier sense range, and assigns a weight to the path based on the transmission delay of the channel that hold the maximal transmission time. The path with the least weight is preferred. Therefore, *BATD* not only takes the diverse channel distribution into considerate when there are multiple non-overlapping channels within one path, but it also considers that the links on the different frequencies can transmit data packets concurrently.

To verify the effectiveness of the *BATD* routing metric, along with other routing metrics, such as *HOP*, *ETX*, *ETT*, and *WCETT*, *BATD* has been evaluated through in-depth ns-2 simulations. The performances achieved by those routing metrics are compared under simple chain scenarios as well as randomly generated scenarios. The experimental results show that our *BATD* metric outperforms

the other routing metrics, especially for scenarios with more than two radios. It achieves up to 35% throughput improvement from the *WCETT* metric.

Moreover, the proposed *BATD* metric can be further improved by considering the overhead sending along with each data packets and intra-path interference that we already explained in Chapter 5. Therefore based on *BATD* metric,an improvement bottleneck aware transmission delay metrics called *iBATD* is designed. By comparing *iBATD* with *BATD* under various network configurations, the *iBATD* metric has been proved to be more effective in terms of obtaining high throughput and averagely low packet latency.

6.1 Introduction

In multi-radio wireless mesh networks, each wireless node is configured with multiple radio devices tuning into different non-overlapping channels. Therefore, to locate a high performance routing path in such a multi-radio network, routing algorithms should consider the fact that links on non-overlapping channels can transmit packets simultaneously.

Together with routing metrics in single radio scenario, some routing metrics have been proposed to take care of multi-radio scenarios in wireless mesh networks [13, 14, 48]. Most of them have certain drawbacks, such as metric of interference and channel-switching(MIC) [15, 47, 48]. MIC is a novel topology-dependent path weight function and claims to be the first metric to consider the inter-flow interference between different data transmission flows. The key feature of this metric is to address the load-balancing or inter-flow interference. It assumes that all the neighbor nodes within the interference range always interfere with the transmissions, even if the neighbor nodes do not transmit any packets. This assumption is not always correct and it might select a path

with worse performance and it also assumes that two links on the same channel interfere with each other only when they are consecutive. These two assumptions make *MIC* ineffective and unrealistic.

Among all current routing metrics, WCETT (Weighted Cumulative Expected Transmission Time) is one of most popular routing metrics used in multi-radio mesh network [13]. In addition to accounting the link loss rates and different transmission rates, the *WCETT* metric is the first metric to explicitly consider the interference among links that use the same channel and favors paths with more diverse channels, but it may not be adequate to reflect the actual effects of diversity level due to the following reasons: (1) *WCETT* does not fully address the fact that links on different non-overlapping channel can work concurrently without any interference when it calculates the expected transmission time. (2) *WCETT* ignores the possible spatial reuse that is links on the same channel can work simultaneously if they are out of carrier sense range of each other.

To address these two drawbacks, we have proposed a new high performance routing metric, called BATD (Bottleneck Aware Transmission Delay), for multi-radio wireless mesh networks. The basic idea of *BATD* can be summarized as follows. For each independent channels within one path, the summation of transmission delay time on the links within the same carrier sense range is measured. And the channel with the largest transmission delay time is considered as bottleneck channel. The path's performance is evaluated based on the transmission delay time on this bottleneck channel. Based on the *BATD* metric, we have also designed an improved BATD metric called (iBATD). By combining the design principles of both *BATD* and *iETT*, the *iBATD* metric can improve the performance of BATD greatly in most cases.

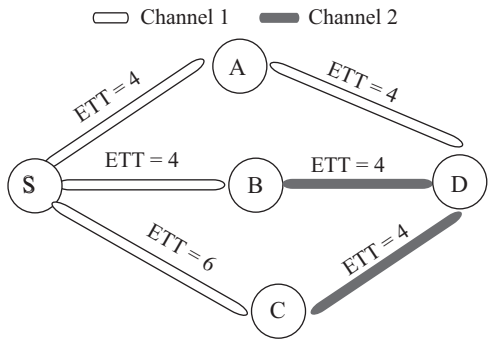

Figure 6.1: Wireless Network Scenario for Three Paths

6.2 Design of BATD

When a mesh node has only one radio interface, the expected transmission time represents approximately the actual ”air time” used by a packet traversing the path. The path with the least *ETT* metric value normally has a better performance than paths with higher ETT values since more data packets on average can be transmitted in the same amount time. However, when each node has multiple radios, links on different channels exist simultaneously within one path. Those channels are non-overlapping channels and can work concurrently without interference, which means those links can transmit packets at the same time within one path. For example, Figure 6.1 shows a simple scenario, in which three different paths exist from source S to destination D, where the links are labeled with their individual transmission unit time. The figure on the right shows the detail of how these three paths deliver their data packets respectively. Here $pkt1$ stands for the first injected data packet and $pkt2$ represents the second injected data packet and so on. Due to non-interference channels, path $S - B - D$ and path $S - C - D$ can send new packets in the first hops while the second links are sending their previous data packets, and for path $S - A - D$, only one link can work at one time. So for the same amount of transmission time (40 unit time in this scenario), path

$S-B-D$ can transmit 9 data packets, path $S-C-D$ can send 6 transmission and path $S-A-D$ only can deliver 5 packets. Therefore these three paths have different performances.

However, currently routing metrics are not always capable of differentiating them successfully. For instance, based on the *ETT* metric, the ETT metric value for any path is the summation of all links's transmission time. For path $S-A-D$, every data packet needs around 8 unit times to reach the destination node D. For path $S-B-D$, the transmission time is also the sum of all links (8 unit time in this case) according to the ETT metric. However, in path $S-B-D$, Link $S-B$ uses channel 1 and link $B-D$ occupies channel 2, then packet transmission on first hop may proceed successfully with the transmission on the second hop simultaneously since these two links do not interfere. Therefore, path $S-B-D$ actually can transmit more data packets during the same time compared with path $S-A-D$ in Figure 6.1, which leads to higher throughput performance. *ETT* cannot differentiate these two paths. In addition, the *ett* metric value for path $S-C-D$ is 10, which is considered by *ETT* to be the worst path. This is also not true in real case.

Based on the *WCETT* metric, the wcett metric values for the three paths shown in Figure 6.1 can be represented respectively as follows:

$$WCETT = \begin{cases} (1-\beta)*8+\beta*(8) & \text{path S-A-D} \\ (1-\beta)*8+\beta*(4) & \text{path S-B-D} \\ (1-\beta)*10+\beta*(6) & \text{path S-C-D} \end{cases} \tag{6.1}$$

Obviously, *WCETT* evaluates path $S-A-D$ and path $S-B-D$ correctly because the $wcett$ metric value for path $S-A-D$ is always larger than the $wcett$ metric value of path $S-B-D$, no matter which β value. However, depending on β value, *WCETT* might not be effective to differentiate path $S-A-D$ and path $S-C-D$. The reason is because *WCETT* metric

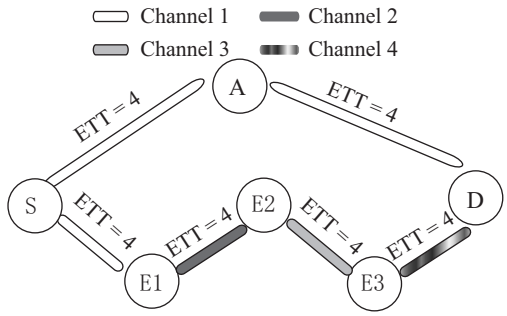

Figure 6.2: Wireless Network Scenario for Channel Diversity

ignores the fact that links on different channels along one path can transmit packets concurrently. In detail, when *WCETT* calculate its first term (the expected transmission time along all hops in the path), it simply sums up each individual link's etts one by one (actually the same time slot can be used by links on different channels to transmit different packets). Although *WCETT* applies its second term to represent channel diversity in some degree, it does not account for this factor appropriately. Especially when one node has over two radios, *WCETT* is not very effective. A simple example is shown in Figure 6.2. Under real circumstances, the performance of path $S - E1 - E2 - E3 - D$ is better than path $S - A - D$ because all links ($S - E1, E1 - E2, E2 - E3, E3 - D$) on path $S - E1 - E2 - E3 - D$ can transmit different data packets at the same time slot. Therefore, observed from figures above, except for the first data packet, the transmission delay time between two consecutive data packets is largely determined by the transmission time on the bottleneck channel. The bottleneck channel here is the channel taking the maximum transmission delay along one path. For example, in Figure 6.1, the bottleneck channel transmission times for paths $S - A - B$, $S - B - D$, $S - C - D$, and $S - E1 - E2 - E3 - D$ are 8, 4, 6, and 4 *ETT* unit time, respectively. The bottleneck channel time is a good indicator of paths' performance.

Due to shared nature of wireless communications, when one link is sending packets, all other links using the same channel should not transmit packets. For instance, Figure 6.3 shows a simple

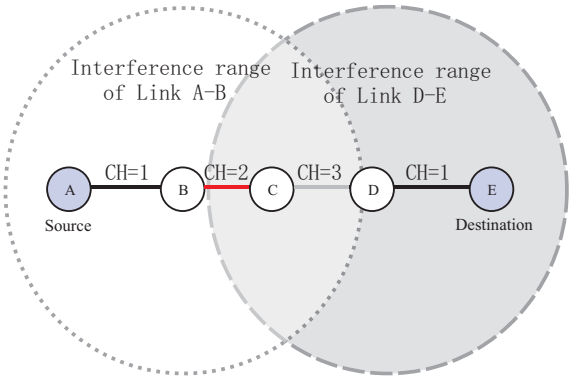

Figure 6.3: Spacial Reuse for a long Path

scenario, in which Nodes A, B, C, D and E are positioned on a chain topology with their channel usage information indicated. Within path $S - A - B - C - D$, link $A - B$ and link $D - E$ are far apart and not in interference range of each other. Although they are on the same channel, these two links can transmit data packets simultaneously. So the intra-flow interference might not always exist between links within one path. The spatial reuse along one single path is highly possible especially for a long path.

To determine whether two links interfere, some of the current routing metrics, such as *WCETT* metric, use a pessimistic model. It assumes two links on the same channel always interfere, which is not true(see Figure 6.3) especially for a longer path in multi-radio mesh networks. To attenuate such a pessimist view, some use a hop-distance-based interference model that defines that two links interfere if they are less than three hops away from each other. However, when a path has a zigzag pattern, this definition may not hold [14]. The two definitions are not realistic.

Therefore, to detect whether two links on the same channel within one path interfere or not, we exploit a simple measurement method proposed by J. Padhye *et al.* [64]. The method is based on the following observation: if two nodes are in each other's interference range, their carrier sense mechanisms will prevent them from transmitting simultaneously. Since mesh networks are static,

it is highly feasible to measure whether two nodes are in the carrier sense range of the other when the network is established. Consequently, we use the following rule to determine whether two links interfere or not: if any node in one link is in the carrier sense range of any node of another link, interference between the two links will be accounted; otherwise, we consider the two links are interference free.

Based on the discussion above, for each individual channel within one path, we compute the maximum summation of ETTs for those links within carrier sense range of each other. This is a more accurate indicator of expected transmission delay time consumed by sending packets over that specific path. Simply adding all ETTs for links on the same channel within one path overestimates the intra-flow interference and omits the possibility of time-reuse. Among all channels, the channel with the largest value of ETTs within one path is considered as the bottleneck channel, which is the major factor to affect the whole path performance.

Therefore, given one path p with k channels (channel 0,1...,k), let ETD_c be the expected time transmission delay for channel c on path p, N_c be the number of links on channel c with path p (those links within the same carrier sense range), our *BATD* is defined as:

$$BATD(p) = \begin{cases} \max(ETD_1, ETD_2, \ldots, ETD_k) & \\ ETD_c = \sum_{i=1}^{N_c} ETT_i & 0 \leq c \leq k \end{cases} \tag{6.2}$$

6.3 Performance Evaluation of BATD

This section evaluates the effectiveness of the proposed metric *BATD* using ns-2 simulator tool, which is a discrete event simulator targeted at a number of networking research [53]. Since

this study aims to address intra-flow interference rather than inter-flow interference, we use *HOP, ETX, ETT, WCETT* for comparison. The ns-2 used her is on version 2.30 .

6.3.1 Experimental Setting

Like iETT metric evaluations, we use extended DSR explained in chapter 3 as our routing protocol in our experiments. Constant Bit Rate (CBR) is utilized as the traffic generation model. Data packets will be sent out with a deterministic interval rate (1ms). The packet size is set to 500 bytes. Only one CBR traffic transfer is active at any time. All wireless nodes in our experiments are equipped with multiple IEEE80211b radio interfaces tuned to non-overlapping channels. Each radio interface is able to operate at one of the four available IEEE802.11b transmission rates, which are 1, 2, 5.5, or 11 Mbps. The transmission powers for all radios are fixed at 15dbm. Detail parameters we used for our radio interface can be found in Table 5.3.

For each set of experiments, The number of radio interfaces configured to each individual node is varied from 2 to 4. Static channel assignment scheme is applied to all wireless mesh node. And all node have the equal channel allocation.

The simulations are conducted with two different node deployment patterns: Simple chain network scenarios and randomly generated scenarios. In each simulation, the data rate sent from the source node is high enough to saturate the network. Under the same circumstances, Average throughput and average end-to-end delay achieved by different routing metrics are measured. We set β value to 0.5 for *WCETT* metric.

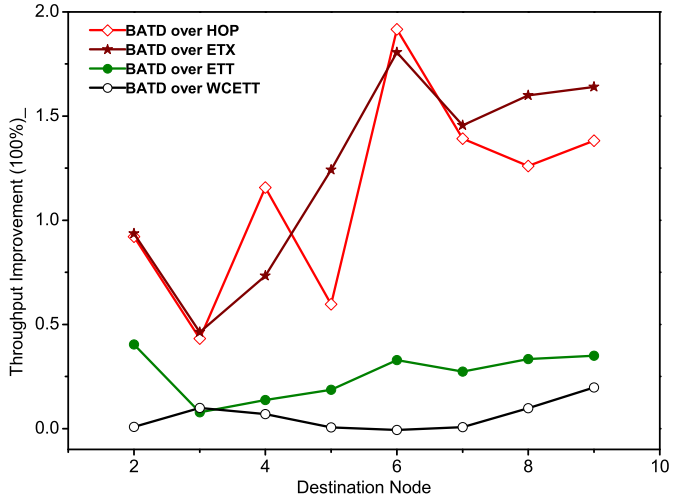

Figure 6.4: BATD: Throughput Improvement on Linear Topology (Two Radios per Node)

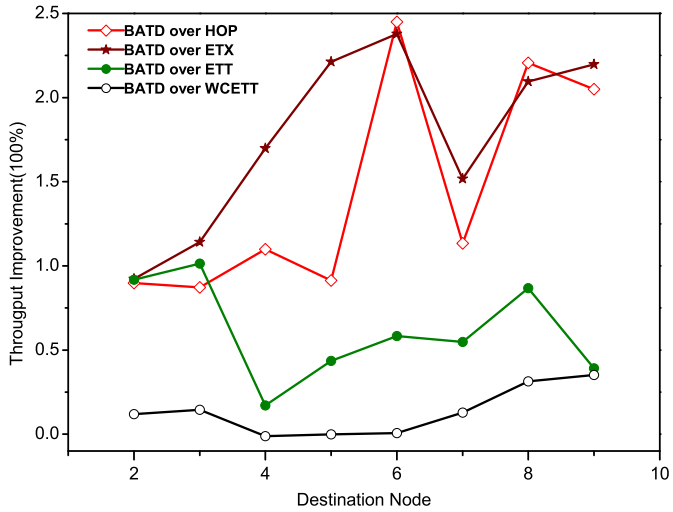

Figure 6.5: BATD: Throughput Improvement on Linear Topology (Three Radios per Node)

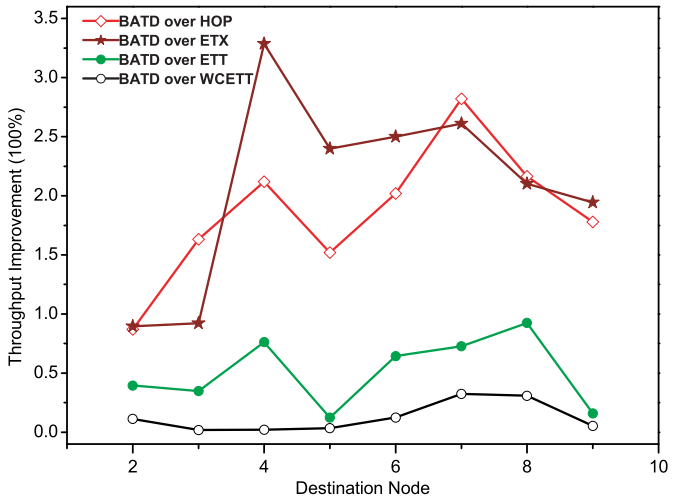

Figure 6.6: BATD: Throughput Improvement on Linear Topology (Four Radios per Node)

6.3.2 Performance Result

Experimental Scenario I

In our first experiment, ten wireless nodes are lined as a chain of nodes 250m apart. The first node 0 is always designated as the source node, and nodes 2, 3 to 9 are respectively assigned to be the destination nodes. A CBR connection is started between the source node and one of the destination nodes.

For each source-destination pair, we run the CBR traffic and measure the throughput with *HOP*, *ETX*, *ETT*, *WCETT*, and *BATD* metrics separately. The number of radio interfaces for each individual node is varied from 2 to 4. For each specific experiment setting, we repeat the experiment 5 times.

Figure 6.4, 6.5 and 6.6 present simulation results. For all these figures, the vertical axis represents the average throughput improvement(*100) achieved by *BATD* against other metrics and the horizontal axis stands for the destination nodes. In each figure, four curves are plotted, in which

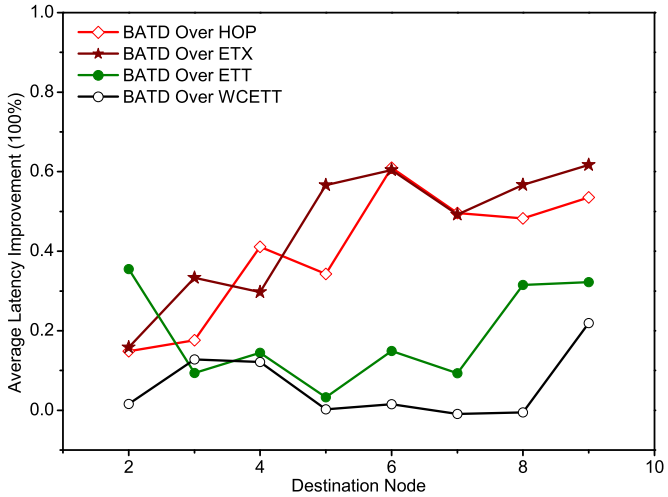

Figure 6.7: BATD: Latency Improvement on Linear Topology (Two Radios per Node)

the curve marked with ⋄ is the throughput improvement achieved by *BATD* over *HOP*, the curve with ⋆ is for the improvement against *ETX* metric, the one marked with • is for the improvement against *ETT* based on *BATD*, and the curve with ∘ stands for improvement obtained by *BTAD* over *WCETT*. The results show that *HOP*, *ETX* and *ETT* are not effective in finding a high throughput path since they ignore the availability of multiple radios. Comparing our *BATD* metric with *WCETT*, the throughput does not improve in most cases especially when the number of radios is 2. This is because these two metrics actually select the same path in those destination-source pairs. For example, when the number of radios is 3, *WCETT* and *BATD* select the same path when destination nodes are 4, 5 and 6, respectively. However, when these two metrics select different paths, *BATD* metric always performs better than *WCETT* with an improvement as high as 135%.

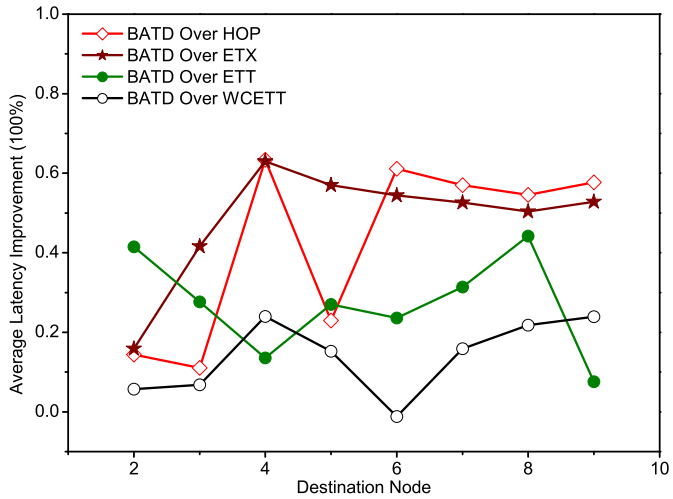

Figure 6.8: BATD: Latency Improvement on Linear Topology (Three Radios per Node)

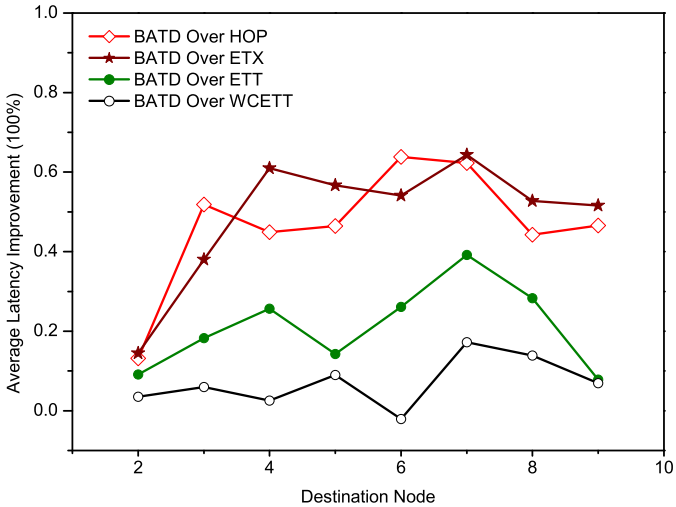

Figure 6.9: BATD: Latency Improvement on Linear Topology (Four Radios per Node)

The average latency improvement obtained by our *BATD* metrics is plotted on Figure 6.7, 6.8, and 6.9, respectively. On all these figures, the vertical axis represents the average latency improvement achieved by the *BATD* metric against other routing metrics and the horizontal axis stands for the destination nodes. As on the throughput improvement figures, four curves are plotted in each figure in which the $\diamond$ curve is the throughput improvement achieved by *BATD* over *HOP*, the $\star$ curve is for the improvement against *ETX* metric, the $\bullet$ curve is for the improvement against *ETT* based on *BATD*, and the $\circ$ curve stands for improvement obtained by *BATD* over *WCETT*. As expected, in most cases, *BATD* can achieve on average a lower latency than other routing metrics under same circumstances. The *BATD* metric achieves up to 125% average latency obtained with the *WCETT* metric.

Experimental Scenario II

The second experiment is conducted based on a randomly generated topology. In this experiment, 25 wireless nodes are created and spread out on a 5000m*5000m field. Therefore, a total of 25*24=600 different sender,receiver pairs exist within this setting. Among those pairs, 20 are randomly selected, and a CBR flow is established once at a time. As in our first experiment, we change the number of radios from 2 to 4 and carry out each experiment with *HOP*, *ETX*, *ETT*, *WCETT*, and *BATD* separately under the same conditions.

The throughput achieved is shown in Figure 6.10. In this figure, each column represents the average value achieved by 20 randomly selected pairs. As expected, we get similar results to those obtained on a chain topology. *BATD* selects on average higher performance paths than other routing metrics. Although for some pairs, *BATD* and *WCETT* chose the same paths, *BATD* still achieves a throughput improvement of approximately 7%, 11% and 16%, respectively, over *WCETT* for

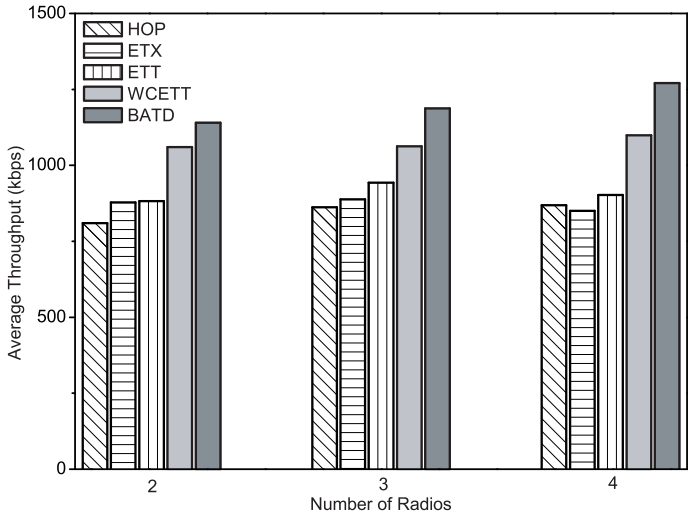

Figure 6.10: BATD: Average Throughput on Multi-radio WMNs (kbps)

two, three and four-radio scenarios. If we compare the performance improvement of these 20 pairs individually, the average throughput performance improvement that *BATD* achieves against *WCETT* reaches up to 26% in the case of four-radio scenarios. Figure 6.11 shows the average latency achieved by these five different routing metrics and demonstrates that the *BATD* metric can on average select the path with the smallest end-to-end delay. Apparently, the throughput and latency improvement steadily increases as the number of radios per node increases.

In all the previous experiments, a single CBR connection is active at any one time. To observe the performance of the *BATD* metrics when there are concurrent flows in wireless mesh networks, we consider two simultaneous CBR transfers in our randomly generated topology used in our second experiment. For each experiment, we randomly select two source-destination pairs and make sure that these two CBR connections are active at the same time. The experiments are repeated 20 times in case of two radios per node scenario, and the performance of total network throughput and average end-to-end delay achieved by different routing metrics are collected and compared.

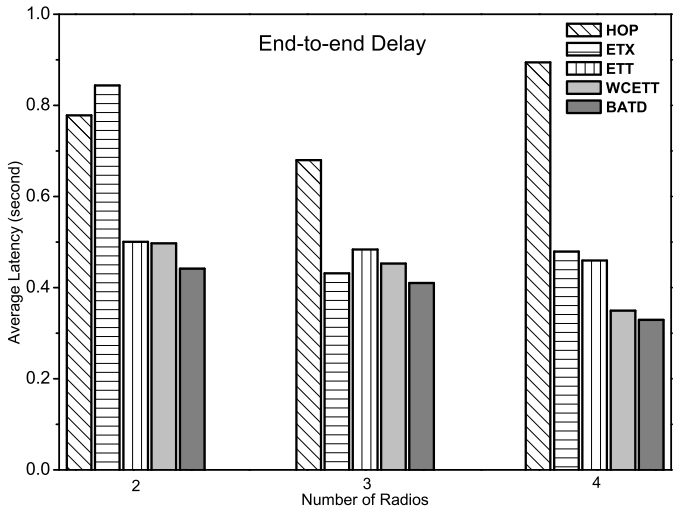

Figure 6.11: BATD::Average Latency on Multi-radio WMNs (s)

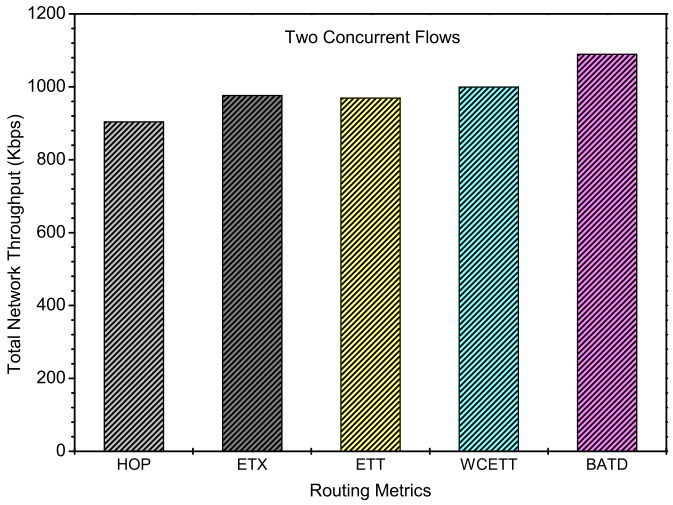

Figure 6.12: BATD:The Total Network Throughput for Multiple Flows

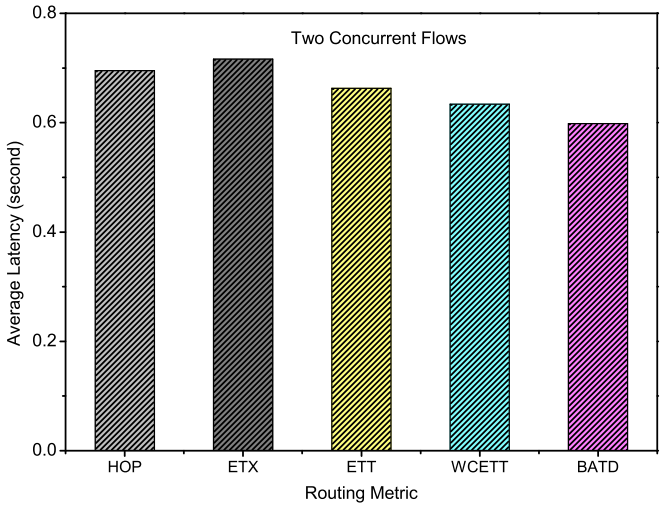

Figure 6.13: BATD:Average End-to-end Latency for Multiple Flows

Figure 6.12 and Figure 6.13 show the performance results for the total network throughput and average end-to-end packet delay. In both figures, five columns are plotted and each column represents the actual performance values achieved by different routing metrics. These two figures again confirm that *BATD* has the best performance in terms of high network overall throughput and average low end-to-end packet delay. It can achieve up to 11% throughput improvement over *WCETT* in this specific scenario. Since *WCETT* captures intra-flow interference in some degree, it has better performance than other routing metrics. Hence, *BATD* is effective for WMNs with various channel/radio distributions.

6.4 Design of iBATD

In previous section, we demonstrated that the performance of one path's performance is largely determined by the total transmission delay time on the bottleneck channel along the path in multi-radio scenarios. Based on this observation, we already proposed the *BATD* routing metric. It calculates the summations of transmission time for each different channels along one path and assigns the maximum transmission time to the path as an indicator of the path performance.

According to the design principle of *BATD*, for a specific path p with n different non-overlapping channel, *BATD* computes the summation of expected transmission delay (ETD) for each channel. And the ETD for one specific channel is defined as the total ETT values for links within the same carrier sense range. The ETT for one individual link is actually S/B, in which S represents the frame size and the B stands for the data rate. However, Chapter 5 have proved that simply summing these individual ETT metrics up to compose a path metric is not efficient to represent intra-path interference, which might lead to poor path selections since a bad link can be smoothed by a good link. In addition, as also explained in chapter 5, using S/B to calculate expected transmission time on one link is not accurate since it does not consider the necessary MAC overhead along with each packet transmissions. The MAC overhead can affect a path's performance greatly especially when data packets is small.

Therefore, to combine the benefits of *iETT* in chapter 5 and *BATD* metric in previous section, we design an improved Bottleneck Aware Transmission Delay metric, called iBATD. The basic idea of *iBATD* is very similar to *BATD*, except that it uses *iETT* value instead of *ETT* value to represent the transmission delay time of each individual channels. It can be defined as follows:

$$iBATD(p) = \begin{cases} \max(ETD_1, ETD_2, \ldots, ETD_k) \\ ETD_c = \sum_{i=1}^{N_c} iETT_i \qquad\qquad 0 \leq c \leq k \end{cases} \tag{6.3}$$

Where *iETT* can be found in equation 5.4.

6.5 Performance Evaluation of iBATD

In this section, we will use different network scenarios to evaluate the performance of our *iBATD* routing metric. Since our extensive experiments result have confirmed that *BATD* metric is better than other routing metrics in previous sections, such as *HOP*, *ETX*, *ETT* and *WCETT* metrics, we will only compare our *iBATD* routing metric with the *BATD* metric.

6.5.1 Experimental Setting

The scenarios and experimental parameters in this section will be almost the same as we used in previous section when we evaluate *BATD* against other routing metrics. In summary, the traffic generation model is CBR with a packet size of 500 bytes, each wireless node can be configured to 2,3 and 4 wireless radio cards operating in different channels, and each radio can support a range of different data rates. We use extended DSR routing protocol under ns-2 simulation environments.

6.5.2 Performance Result

Experimental Scenario I

Like experiments with *BATD* routing metric, our first experiment will utilize a simple linear network scenario, which is actually a chain composed by ten independent wireless nodes. Here,

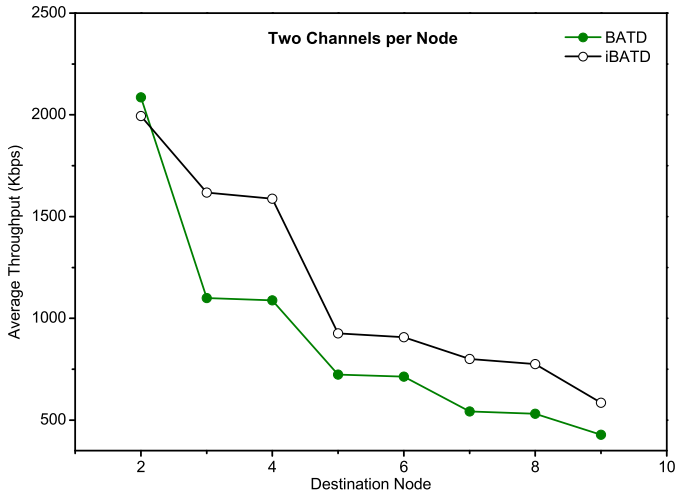

Figure 6.14: iBATD: Average Throughput on Linear Topology (Two Radios per Node)

neighboring nodes are 200 meters apart from each other. The experiment for each source and destination pair connections will be repeated for 5 times under the same condition and their throughput and packet delays will be recorded and averaged. In addition, every experiments will be conducted with both *BATD* and *iBATD* metrics.

The experimental results about the average throughput are shown respectively in Figure 6.14, 6.15 and 6.16. In each figure, two curves are plotted, one represents the results for *BATD* routing metric and the other is for *iBATD* metrics. Apparently, *iBATD* metric is able to improve *BATD* greatly in terms of higher throughput performance. As shown in these figures, *BATD* and *iBATD* select different paths in most cases and *iBATD* is always able to select a better performance path. The highest throughput performance *iBATD* achieved over *BATD* is around 50%.

As experiments in *BATD* metrics, we also plot the average packet delay achieved by these two routing metrics in Figure 6.17, 6.18 and 6.19, respectively. As figures related to throughput, in each figure, 2 curves are plotted, in which the curve marked with ∘ is the throughput achieved by *iBATD*,

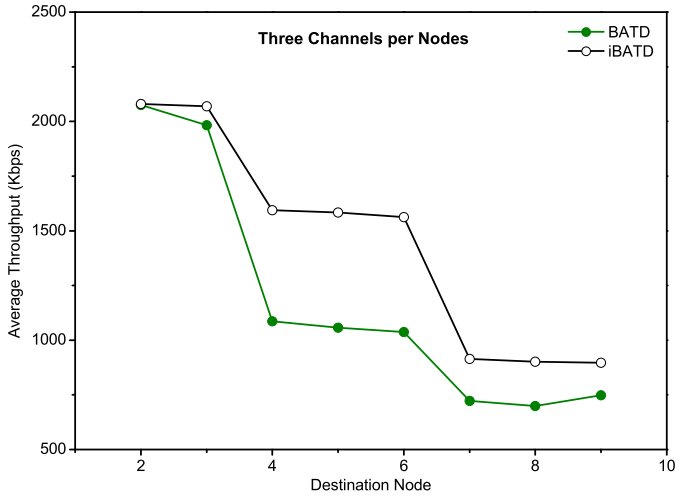

Figure 6.15: iBATD: Average Throughput on Linear Topology (Three Radios per Node)

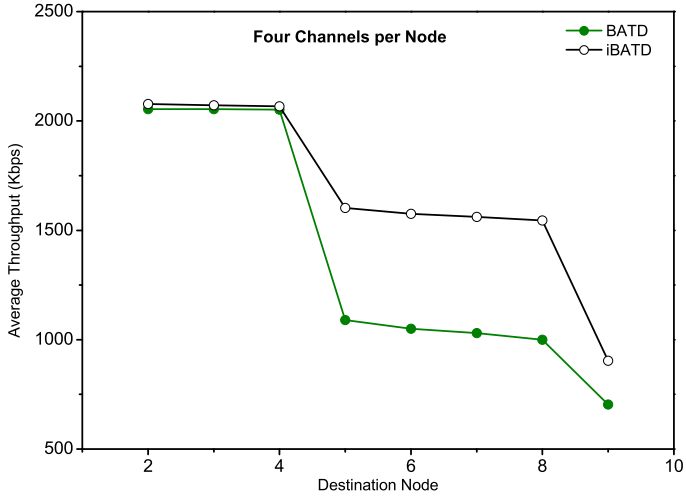

Figure 6.16: iBATD: Average Throughput on Linear Topology (Four Radios per Node)

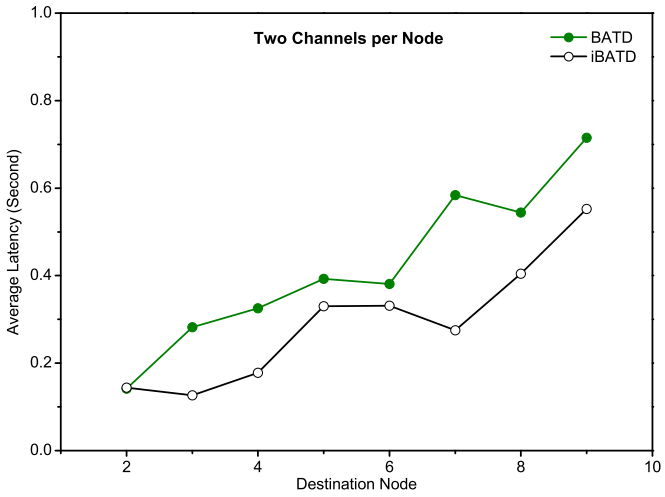

Figure 6.17: iBATD: Average Latency on Linear Topology (Two Radios per Node)

the curve with • is for the *BATD* metric. As expected, in most cases, *iBATD* can achieve averagely low packet delay compared to *BATD* metric.

Experimental Scenario II

By using the same randomly generated scenarios like in previous section, we randomly select 40 pairs to establish a UDP connection once at a time. Each experiment will be carried by both *BATD* and *iBATD* routing metrics. The number of radios/channels in each scenario will be changed from 2 to 4.

The experimental results are shown in Figure 6.20 and Figure 6.21. In both figures, two columns are plotted, one for *BATD* and the other stands for the performance obtained by iBATD. Apparently, the same conclusion can be reached: *iBATD* is able to find a better path with higher performance path and averagely low packet delay than *BATD* routing metric. In average, *iBATD* can

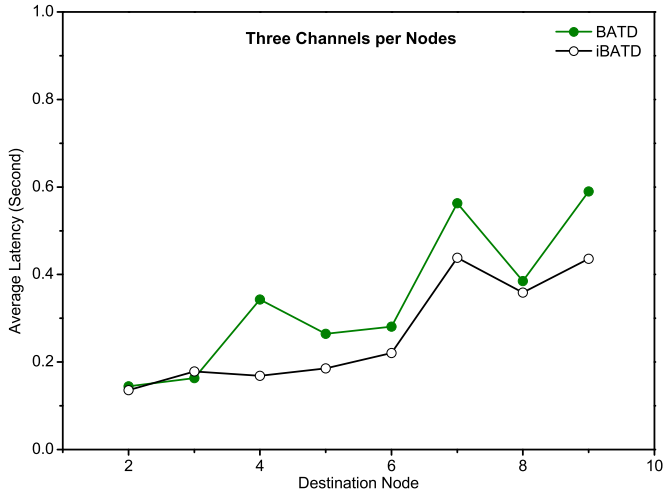

Figure 6.18: iBATD: Average Latency on Linear Topology (Three Radios per Node)

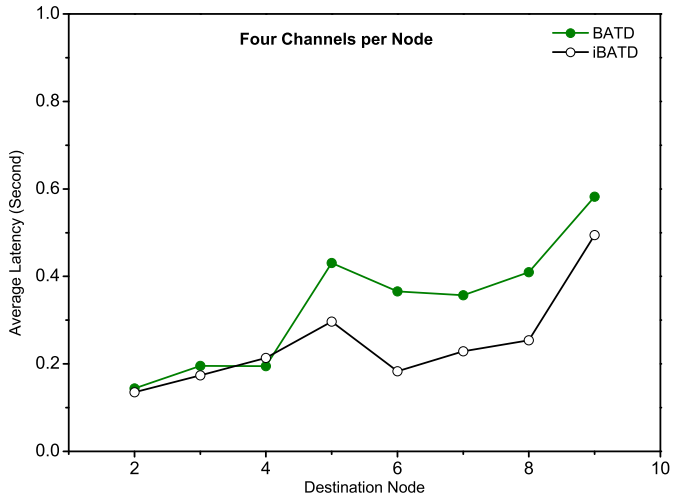

Figure 6.19: iBATD: Average Latency on Linear Topology (Four Radios per Node)

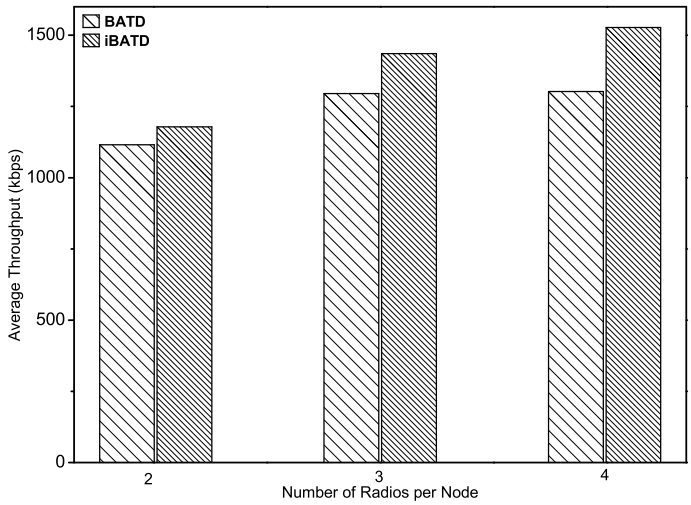

Figure 6.20: iBATD: Average Throughput on Multi-radio WMNs (kbps)

get a performance enhancement over the *BATD* routing metric as high as 117% in case of four radios per node. And furthermore, *iBATD* can achieve a latency decreasing up to 25%. Since we have proved that *BATD* metric is better than other routing metrics in previous sections, we can conclude that *iBATD* is very effectively to identify a higher performance paths with multiple channels/radios.

6.6 Discussion

In this chapter, we study the routing metric issue in multi-radio multi-channel wireless mesh networks. A new Bottleneck Aware Time Delay (*BATD*) metric has been proposed and analyzed. The *BATD* routing metric selects a path based on the maximum transmission delay from interfering links on the bottleneck channel along one path. It considers the possible concurrent data transmission on different channels as well as diverse channel distributions within one path in case of multiple radio mesh network, which has been proved to be a good indicator to evaluate paths.

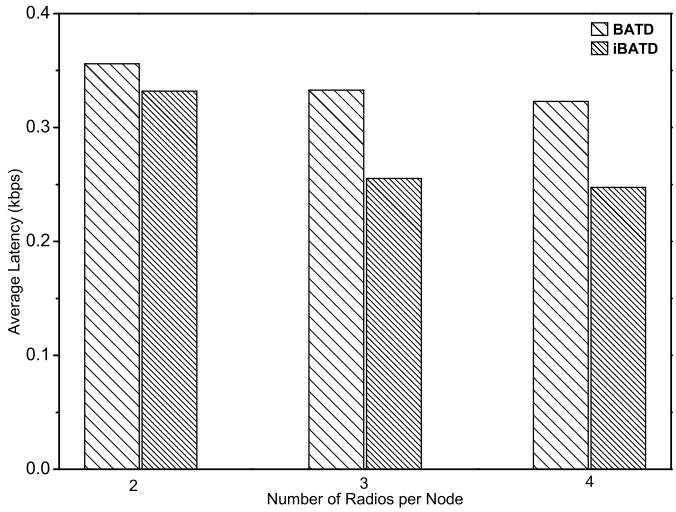

Figure 6.21: iBATD::Average Latency on Multi-radio WMNs (s)

The throughput and end-to-end delay performance of *BATD* has been compared against four other routing metrics, including *HOP*, *ETX*. *ETT*, and *WCETT* metrics on both a simple chain network scenario and a randomly generated topology. Simulations are conducted under different number of available channels/radios. The experimental results showed that our *BATD* metric outperforms other metrics by improving average network throughput. Based on the design principle of *BATD* routing metric, we further design another more efficient routing metric by embedding the idea of *iETT* explained in chapter 5, called iBATD (improved BATD), it holds the advantages of both *BATD* and *iETT* metric and has been proved its effectiveness in difference scenarios. We conclude that the *BATD* metric is capable of selecting effective paths in multi-hop multi-radio mesh networks.

CHAPTER 7

LINUX-BASED TESTBED

In this section, we implement and use a Linux testbed to evaluate actual performances of routing metrics. As in chapter [52], along with *iETT* routing metric, we will also develop Expected Transmission Count (ETX), Expected Transmission Time (ETT) and traditional HOP routing metric, all of them are used as our benchmark to compare the performances.

This section first presents the detailed architecture of our Linux testbed on wireless multi-hop network, then discusses the experimental environments and methodology. Finally the results gathered from the testbed are presented and analyzed.

7.1 Testbed Architecture

Currently, most research in WMNs about routing issues focuses on simulations since it is complicated to deploy a real testbed. This is due to the following two reason: 1) the environment of WMNs is highly dynamic and unpredictable environment; 2) Most routing algorithms in WMNs need to gather link properties to make the right path selections. However, most of the link properties are not available directly at network layer.

Based on traditional wired networking principles, network components are organized as a stack of layers, each layer is built upon the one below it. These different layers have various functions and one of the main purposes of each layer is to offer certain services to the layer above, while hiding details of how the services are actually implemented by the layer above [65]. However, under a wireless environment, in which link connections over wireless channel are not always stable and have a wide range of different characteristics, the traditional layer isolation model is no longer

applicable. For instance, to design an effective routing algorithm in a wireless mesh network, the routing protocol in the IP layer needs additional information about the layer below or its peer layer, such as current link transmission rate, besides its predefined task of routing data packets.

As we explained in Chapter 5, most quality aware routing metrics, such as *ETX*, *ETT* and the new designed routing metric *iETT*, consider link loss rates as well as the transmission data rates to assess paths' performance. Both link loss rates and transmission rates are normally measurable at the MAC layers. Therefore, to design a quality-aware routing metric, we need to collect link quality information from lower layers. Recent researchers have explored a number of flexible cross-layer approaches to challenge the normal network stack design [66, 67]. To implement our testbed to evaluate different routing metrics, we select XIAN (Cross-layer Interface for wireless Ad hoc Networks) as our testbed framework [68].

XIAN is a generic interface for experimenting cross-layer designs with legacy 802.11 networking cards on a Linux platform. The code is freely accessible under the GPL license. The basic idea of the XIAN framework is to provide an easy method of exchanging state information between different layers or neighboring nodes, especially MAC layer and Routing layer. Therefore the XIAN framework can be considered as a service by network layers or system components to access information about configuration and performance from MAC/PHY layers. To expose MAC/PHY state information, such as data transmission rate or received signal strength indicator to higher layers, the XIAN framework first needs to acquire that information itself. So its implementation requires to be combined with the MAC layer driver. Currently, XIAN framework only works with a Madwifi driver.

Currently, the Madwifi driver is one of the most advanced wireless network drivers available for Linux. It is also an open source driver comprehensively supporting adaptors with the Atheros

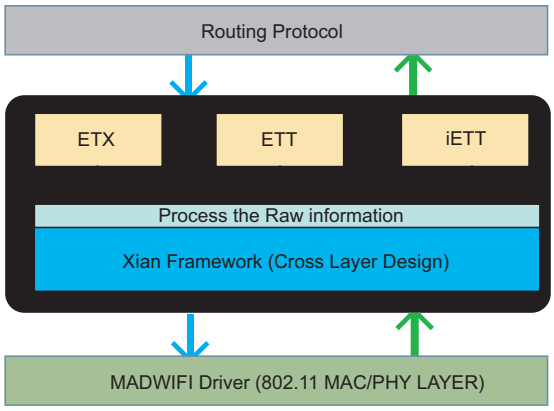

Figure 7.1: Architecture of Implementation

chipset, and can be downloaded from the MadWifi website [69]. The driver is a multi-core driver module that includes the PCI hardware module for interfacing with the PCI bus, the Atheros chipset specific module to connect the hardware registers with the actual driver, and the device independent module. Therefore, it has been delicately designed and layered in architecture to work with any additional development. Most importantly, the Madwifi driver has already built an established mechanism to gather MAC layer state information as well as diverse statistics directly.

However, the raw information XIAN obtains from the MAC layer through the Madwifi driver can not be directly used by our testbed. Thus, we have implemented additional kernel modules in order to provide actual link characteristics or metric values to our routing layer. The architecture of the implementation is shown in Figure 7.1. To make the testbed independent of any applications and easily to be deployed, all the testbed solutions are develop as loadable/unloadable Linux kernel modules.

To incorporate different routing metrics with routing protocol as we did in our simulations, we adopt a DSR based routing protocol. Currently, a number of open-sources for Dynamic source routing protocol implementation are available on line. The one we use to build our Linux testbed is

called DSR-UU, which can be downloaded from [70]. DSR-UU has developed most DSR features specified in the DSR draft except flow extensions. One of its important features is that it implements a virtual network interface (dsr0) to enables DSR based network to coexist with the regular non multi-hop ad hoc network. However, DSR-UU only uses the minimal hop count metric for path selection. We have modified/revised it by adding necessary fields on its routing packets to support our routing metrics, which are explained in detail in Chapter 3.

7.2 Testbed Implementation

To compute the actual values of different routing metrics such as *ETX*, *ETT* and *iETT*, we measure the link loss rate as well as data transmission rate for each individual link.

7.2.1 Link Loss Rate

To assess the link loss rate, forward and reverse loss rates (P_f and P_r) must be calculated/addressed. The values P_f and P_r can be approximated by the broadcast based packet technique. Basically, each node periodically sends out a broadcast probe packet. Meanwhile, it also tracks the number of probes received from neighboring nodes, and includes this information in its own probes. One drawback of this approach in implementation is that the result of loss rate measurement is not accurate if two nodes are not starting the probe process simultaneously. For example, Node A and Node B can communicate with each other directly. To measure the link loss rate, Node A sends its probing process at second "1", so it sends out its first probe packet, and at the same time, it is ready and expects probes from neighbors at second "1". Suppose Node B starts its probing process at time "1.2". Thus when Node A's first probe reaches Node B at "1.1", Node B is not ready, which leads to the loss of the first probing packet from Node A (not due to actual link quality). This non-synchronized

probing process obviously results in inaccurate measurement. However, it is practically impossible to synchronize all nodes within one wireless network.

To address this problem, we implement a counter buffer scheme in our testbed. In this scheme, every node keeps a buffer to always record the number of probes received from each neighbor in the past ten seconds. In every second interval, node sums the numbers in this buffer and broadcasts this information with a probe to its next neighbors. As a result, nodes are able to derive their loss rates easily by computing P_r from corresponding counter buffer of a neighbor node and obtaining P_f information directly from the last probe from that specific neighbor. With this scheme, a node can start its probing process at any time, and get accurate information of link loss rate at any time after initial ten second.

7.2.2 Data Transmission Rate

Determining the data transmission rate of each link is harder. One possibility is to set the bandwidth of each 802.11 radio to some given value. For example, De Couto *et al.* restricted transmission rate of 802.11b radios only to 1Mbps when they evaluate *ETX* [8]. However, in reality, wireless cards automatically select data transmission rates for each transmitted data packet. This feature is known as rate adaptation, and is supported by most modern 802.11 drivers. Since 802.11 standard does not specify which rate adaptation algorithm is actually used to set transmission rate, a number of rate adaption algorithms have been proposed, such as AARF, LD-ARF and RBAR [23–25]. To empirically obtain the data transmission rate, Draves *et al.* uses a technique called packet pairs to evaluate *WCETT* routing metric [13]. This technique requires each node to send two back-to-back probe packets to each of its neighbors every minute. The two probe packets are of different size, with the second being larger than the first. The neighbor node measures the difference between

the reception time of these two data packets and communicates the value back to the sender. The sender takes a minimum of 10 consecutive samples and then estimates the bandwidth by dividing the size of the second probe packet by the minimum sample [71]. However this method is not very accurate because it ignores the other factors that affect the packet transmission time [13]. The reason *WCETT* uses it is that it does not know the rate adaptation algorithm used by its 802.11 cards and the drivers do not supply bandwidth information.

In our experiments, we use wireless cards based on Atheros chipset AR5212 and have full access to the driver. With this chipset, Onoe is the default rate adaptation mechanism. It is a credit-based strategy and very conservative in that it maintains credits for the currently used rate on a per-destination basis to aid in the decision to increase the data rate [72]. Since our solution is built based on Linux Kernel Module and Onoe adaptation algorithms is also a relatively independent Linux Kernel Module, it is possible to get data transmission rate information directly by accessing Onoe module. However, most of rate adaptation algorithms within a wireless node updates its actual data transmission rate only when there are data packets traversing it. Otherwise, the derived transmission rate is not changed. When a node starts up, it keeps a default transmission rate to all its neighbors. For instance, all nodes' default transmission rate is 11Mbps under Onoe rate adaptation scheme. Therefore, to get an accurate measurement of data transmission rate, we frequently send out "warm-up" data-rate probe packet. In our experiments, we propose the following approach: 1) each node in turn sends out data rate probe packets around 40 seconds to its one hop neighbor before we start to transmit real data flow. 2) after that, each node sends a data packet to its next neighbor every 1 second to keep updating the transmission rate. As a consequence, each node can keep a relatively meaningful data transmission rate for all its neighbor nodes.

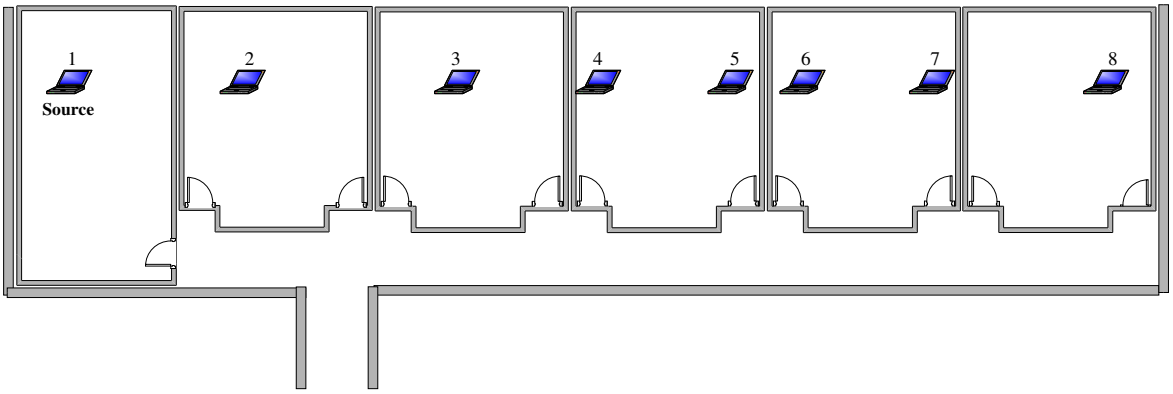

Figure 7.2: Testbed Setting for Experiment I

7.3 Performance Evaluation on Testbed

7.3.1 Experimental Setting

In this experiment, we use Iperf to initialize UDP traffic with a packet size of 500 bytes [73]. To guarantee Iperf always sends at its fastest rate, we set the sending rate to 20 Mbps and collect the results every 10 second.With one connection pair, we apply different routing metrics to select routing paths. The routing metrics used here are *HOP*, *ETX*, *ETT*, and *iETT*. For each experiment with one specific routing metric, 17 runs are repeated, and each run of experiments lasts sixty seconds. We set every two experiments around 40 seconds apart so that the outgoing data packets do not affect the following experiments. When we collect the trace result, the maximum and the minimum values of those experimental runs are ignored to avoid large variation. Thus the final results are averaged by 15 individual experiments.

Eight IBM Thinkpads are used in this project, including six T60s, an X61 and a T61. All of them are equipped with wireless notebook adapters. In the testbed, we use two different types of cardbus wireless cards, D-Link DWL-AG660 Wireless Cardbus Adapter and Linksys WPC55AG

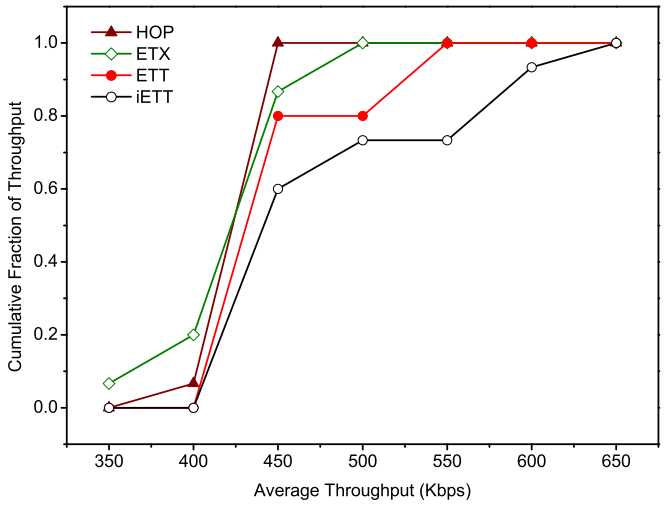

Figure 7.3: Throughput CDFs for Destination Node 3

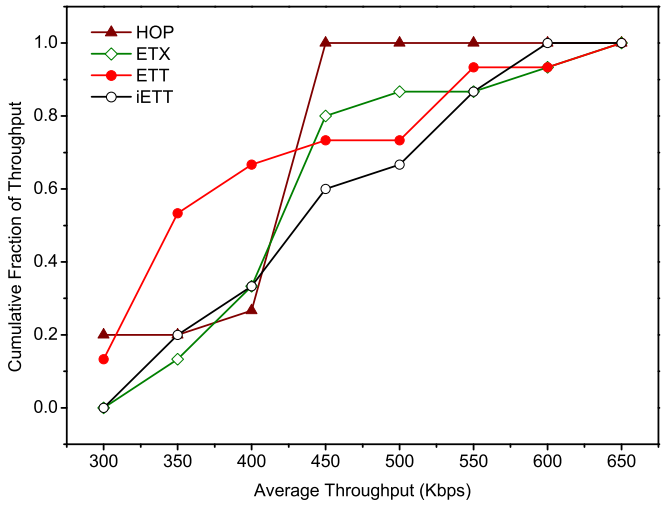

Figure 7.4: Throughput CDFs for Destination Node 4

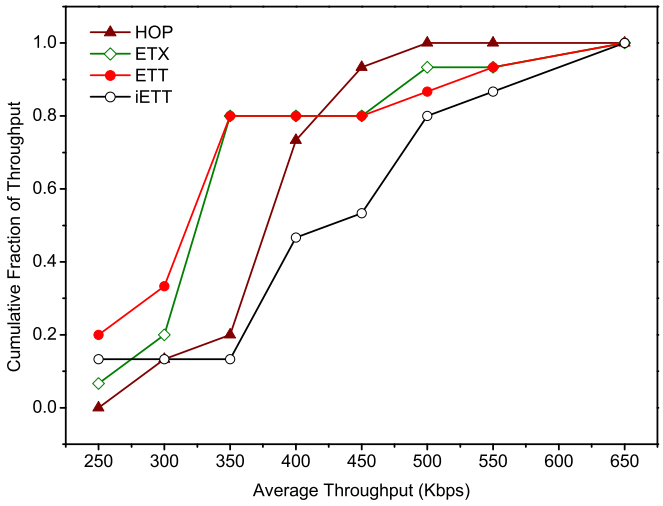

Figure 7.5: Throughput CDFs for Destination Node 5

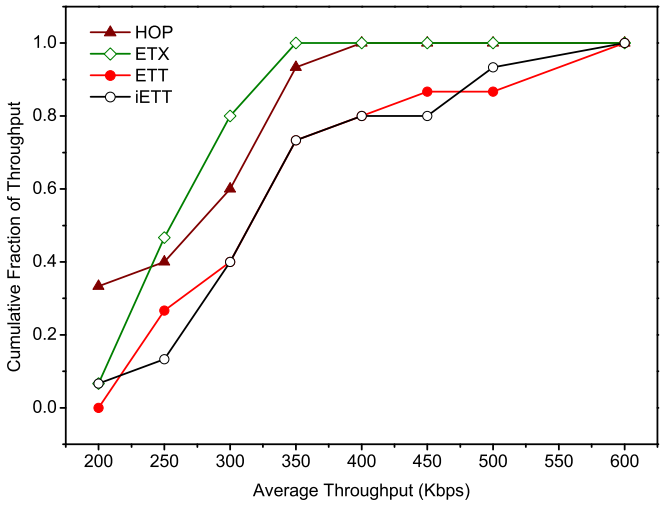

Figure 7.6: Throughput CDFs for Destination Node 6

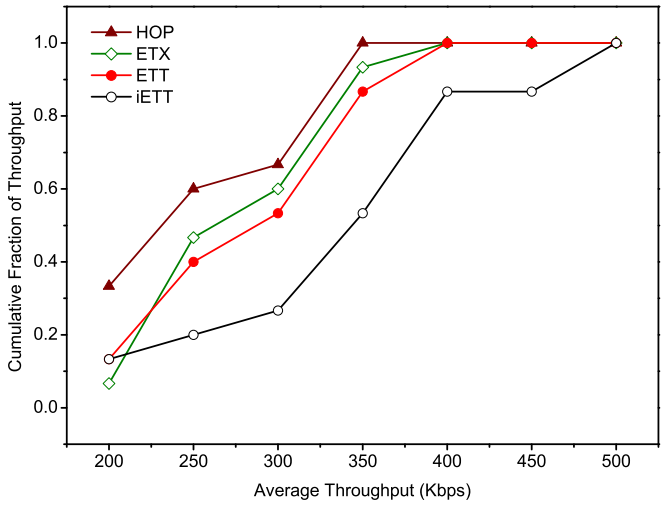

Figure 7.7: Throughput CDFs for Destination Node 7

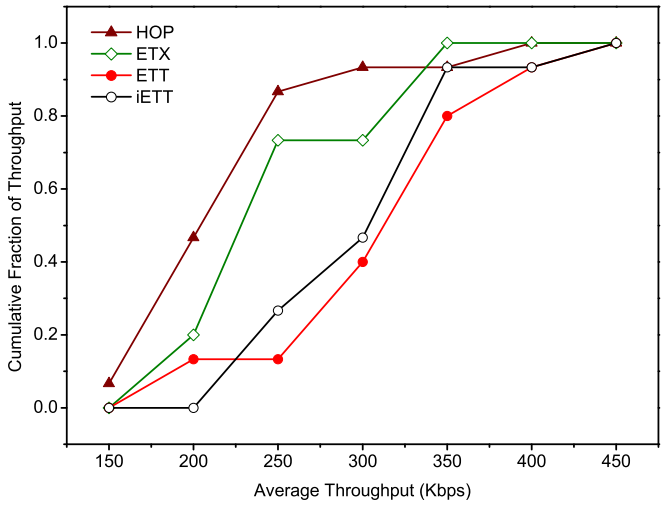

Figure 7.8: Throughput CDFs for Destination Node 8

adapter. Both adaptor cards use Atheros R5212 chipset. These wireless cards are able to operate at one of the four available IEEE802.11b transmission rates: 1, 2, 5.5, or 11 Mbps. To compare the performance of different routing metrics, we collect the average throughput achieved by different routing metrics.

7.3.2 Experimental Result

To evaluate the performances of different routing metrics, we firstly set up a simple scenario, in which 8 wireless nodes are placed as a chain in different rooms on the second floor of our research building (shelby center). As shown in Figure 7.2, the source node, Node 1, is the node on one end. The destination nodes start form node 3 to node 8. In this scenario, the source node has many routing paths to reach its destinations. For instance, when destination is node 4, there are at least 4 paths between source and destination: node 1-node 4, node 1-node 2-node 4, node 1- node 3- node 4, node 1-node 2-node 3- node 4.

Figure 7.3, Figure 7.4, Figure 7.5, Figure 7.6, Figure 7.7, and Figure 7.8 illustrate the performance of different routing metrics when the destination is node 3, node 4, node 5, node 6, node 7 and node 8 respectively. In all figures, four curves are plotted, the one marked with $\circ$ represents the result of routing metric *iETT*, the curve marked with $\bullet$ stands for the performance obtained by *ETT*, the line with $\diamond$ represents *ETX* and the last one marked with $\triangle$ is for performance of *HOP* metric. X-axis stands for the throughput while Y-axis represents the cumulative fraction of throughput. Apparently, as the conclusion we have obtained from simulations, in most cases, *iETT* and *ETT* are able to find a better path than *HOP* and *ETX* because both *iETT* and *ETT* consider the link transmission rate to select a routing path. Overall, *iETT* is more capable of locating a high throughput

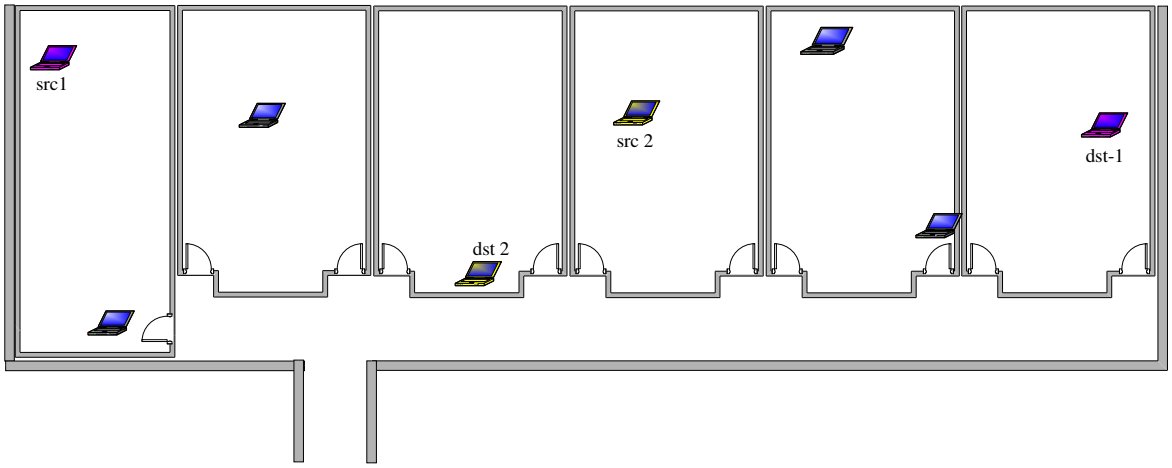

Figure 7.9: Testbed Setting for Experiment II

path than *ETT*. On average, the performance enhancement of *iETT* over *ETT* in some connection pairs can be as high as 121%.

The second experimental scenario is a randomly generated topology, which also contains 8 wireless laptop nodes. These 8 wireless nodes are spread freely in the second floor of shelby center as shown in Figure 7.9. In this experiment, we initiate two concurrent flows at the same time. These two flows are indicated in Figure 7.9 as pair src1-dst1 and pair src2-dst2. For these two specific sender-receiver pairs, we separately apply *HOP*, *ETX*, *ETT*, and *iETT* metrics to select routing paths to carry traffic.

Like our first experiments, we repeat experimental runs for 17 times and ignore the maximum and the minimum values of those experimental runs. Then we collect the trace file from 15 experimental runs for each different routing metrics with both pairs. Figure 7.10 illustrates the result of cumulative fraction of throughput obtained by different routing metrics. The throughput here means the total network throughput (we sum up the average throughput achieved by both flows). Since none of these four routing metrics considers the inter-flow interference (interference due to

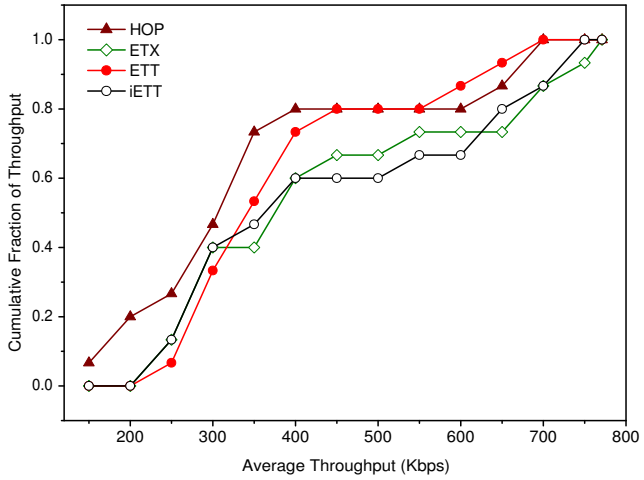

Figure 7.10: Throughput CDFs for Two Flows

contention from different flows) directly, these routing metrics sometimes route packets to congested area, which leads to a wide range of total throughput. Although, *iETT* does not achieve a significant throughput improvement compared with one flow scenario, it still performs slightly better than other routing metrics.

CHAPTER 8

CONCLUSIONS AND FUTURE WORK

In this chapter, we briefly conclude our research work and provide some direction for our future work.

8.1 Conclusions

Wireless mesh network has gained considerable notice due to its flexibility to support diverse applications. A number of research works have been involved in order to improve its performance. In this dissertation, we mainly focus on routing issues in such networks.

An efficient and effective routing metric to locate a high performance path is significant to maximal exploit wireless medium resource in wireless mesh network. In this dissertation, we reviewed current routing metrics in literature for wireless mesh networks and briefly explained their advantages and limitations. and after that, since current popular routing protocols only use HOP metric to select path and they only work on single-radio wireless mesh network scenario, thus we developed an extend dynamic source routing (E-DSR) protocol based on traditional DSR to evaluate the performance of different routing metrics. This E-DSR has been designed to effectively consider multi-radio scenario in wireless mesh network and it also can easily work with any quality aware routing metrics. Since link loss rate is one of the most important factor to affect path's performance, accurate measuring it is a very important task for most of routing metrics. To probe a relatively accurate link loss rate, we found the widely used broadcast based probe method is not effective enough. Thus we advised corresponding strategies to measure link loss rate. The strategies include sending more probes, using actual data rates to send probe to neighbors and only deriving link

loss rate based on the forward link, which has been proved their effectiveness by our experimental results.

In our effort to design an effective routing algorithm in single radio mesh networks, we found that most of existing routing algorithms simply sum individual link metric along one path without considering the following two issues related with intra-path interferences: (1) a path with loss rates evenly distributed on all links yields better throughout than those path with heterogeneous loss rates, (2) the position of the link with the different loss rate on a path is an important factor to affect the throughput achieved. In addition to taking these two issues into account, we also consider the unavoidable MAC overhead sending along with each data packets and design a novel routing metric called *iETT*. The performance of our *iETT* metric is compared against *HOP*, *ETX* and *ETT* metrics on different network scenarios. And the extensive results confirm that *iETT* outperforms other metrics by improving average network throughput and reducing the average packet latency in most cases.

A highly feasible way of reducing limit capacity problem of wireless mesh networks is to use multiple radio interfaces within each node. With this method, more wireless spectrum can be greedily used within one wireless mesh network. Such a multi-radio configuration requires a routing metric to consider the existence of non-overlapping channel within one path. This dissertation proposed and analyzed a bottleneck-aware routing metric (BATD). By considering the possible concurrent data transmissions on non-interference channels, our *BATD* metric suggests to select a path based on the maximum transmission delay from interfering links on bottleneck channel along one path. With different network scenario and different number of configured radios, *BATD* can be acted as an effective indicator to evaluate the performance of the path. Furthermore, by taking into consideration of the idea of *iETT* metric, we further upgraded our *BATD* routing metric by

proposing an routing metric called *iBATD* and proved that *iBATD* is very effective in finding a high performance routing path in WMNs.

8.2 Future Work

One of our future works is to take inter-flow interference into account. When multiple flows operate simultaneously in wireless mesh networks, different flows in the same interference range always compete with each other to send out data packets. The interference caused by different flows is called inter-flow interference. It is not easy to consider the inter-flow interference when designing a routing metric, since considering inter-flow interference is easy to lead to route instability. Currently, only MIC metric considers this effect by giving preference to those paths with less interference to its neighboring nodes. However, the *MIC* metric assumes that all the neighbor nodes within the interference range of that specific link always interfere with the transmission, even if the neighbor nodes do not transmit any packets. By static calculating possible inter-flow interference, it can avoid route instability but it might lead to select a non-optimum routing path. A better solution to consider inter-flow interference should be carefully explored. One possible way is to take the traffic pattern into consideration.

As we mentioned in our dissertation, the assessment of one link's loss rate is an important task to aid an efficient routing metric design. We already addressed most of the limitations of current broadcast based probe to measure the actual link loss rate, However, we assumes that the loss rate of broadcasting probe packets is the same as loss rate of actual data packets, which is not always true in real circumstances, since a large data packet usually has a higher probability of getting lost than a small packet. Also, our modifications of BBP method averagely cause some measuring overhead. In our future work, we will try to let each wireless node keeps tracking of all the data

packet transmission events in its MAC layer. The transmission events include the normal successful and unsuccessful data transmission, as well as successful and unsuccessful data re-transmission. For certain amount time, we record how many transmissions are successful and how many are unsuccessful. In addition to the broadcasting based probe method, the instantaneous information collected is exploited to calculate the link loss rate for every link.

Part of our future work also includes dealing with the broadcast routing request packets in multi-radio wireless mesh networks. When single node is configured with multiple radio devices, node can send out its broadcast based packets from all its radios. In a DSR based routing protocol, when source node initiates a route request packet, every intermediate node will broadcast a copy of this RREQ packet out. As a consequence, many RREQ packets will be generated. For example, in a multi-radio network with each node configured with m independent radio, for a path having n hops away between source and destination, a single route request will generate m^n RREQ packets around. When m or n becomes large, it is important to suppress some redundant routing packets. Traditional DSR only broadcast the first received RREQ to reduce the possible flooding. However, this method will cost the loss some useful routing information. Therefore, we would like to discover a novel method to address this situation.

BIBLIOGRAPHY

[1] I. F. Akyildiz, X. Wang, and W. Wang, "Wireless mesh networks: a survey," *Computer Networks*, vol. 47, no. 4, pp. 445–487, March 2005. [Online]. Available: http://portal.acm.org/citation.cfm?id=1071646

[2] R. Karrer, A. Sabharwal, and E. Knightly, "Enabling large-scale wireless broadband: The case for taps," 2003. [Online]. Available: citeseer.ist.psu.edu/karrer03enabling.html

[3] "Bay area wireless user group." [Online]. Available: http://www.bawug.org/

[4] "Mit roofnet." [Online]. Available: http://www.pdos.lcs.mit.edu/roofnet

[5] "Radiant networks." [Online]. Available: http://www.radiantnetworks.com

[6] A. Raniwala and T. cker Chiueh, "Architecture and algorithms for an ieee 802.11-based multi-channel wireless mesh network," *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE*, vol. 3, pp. 2223–2234 vol. 3, March 2005.

[7] *Capacity of Ad Hoc wireless networks*. New York, NY, USA: ACM Press, 2001.

[8] D. DeCouto, D. Aguayo, J. Bicket, and R. Morris, "A high-throughput path metric for multi-hop wireless routing," in *MobiCom '03: Proceedings of the 9th annual international conference on Mobile computing and networking*. New York, NY, USA: ACM Press, 2003, pp. 134–146.

[9] D. S. J. D. Couto, D. Aguayo, B. A. Chambers, and R. Morris, "Performance of multihop wireless networks: shortest path is not enough," *SIGCOMM Comput. Commun. Rev.*, vol. 33, no. 1, pp. 83–88, 2003.

[10] Y.-C. Hu and D. B. Johnson, "Design and demonstration of live audio and video over multihop wireless ad hoc networks," vol. 2, 2002, pp. 1211–1216 vol.2. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1179651

[11] R. Draves, J. Padhye, and B. Zill, "Comparison of routing metrics for static multi-hop wireless networks," in *SIGCOMM '04: Proceedings of the 2004 conference on Applications, technologies, architectures, and protocols for computer communications*. New York, NY, USA: ACM Press, 2004, pp. 133–144.

[12] A. Adya, P. Bahl, J. Padhye, A. Wolman, and L. Zhou, "A multi-radio unification protocol for ieee 802.11 wireless networks," in *BROADNETS '04: Proceedings of the First International Conference on Broadband Networks (BROADNETS'04)*. Washington, DC, USA: IEEE Computer Society, 2004, pp. 344–354.

[13] R. Draves, J. Padhye, and B. Zill, "Routing in multi-radio, multi-hop wireless mesh networks," in *MobiCom '04: Proceedings of the 10th annual international conference on Mobile computing and networking*. New York, NY, USA: ACM Press, 2004, pp. 114–128.

[14] W. Z. D. Z. D. Qiao, "Comparative study of routing metrics for multi-radio multi-channel wireless networks," in *Wireless Communications and Networking Conference, 2006*, ser. 3-6, vol. 1. IEEE, April 2006, pp. 270–275.

[15] Y. Yang, J. Wang, and R. Kravets, "Interference-aware loop-free routing for mesh networks," 2006.

[16] "IEEE 802.11 Local and Metropolitan Area Networks: Wireless LAN Medium Acess Control (MAC) and Physical (PHY) Specifications." 1999. [Online]. Available: http://grouper.ieee.org/groups/802/11/main.html

[17] D. S. J. De Couto, D. Aguayo, B. A. Chambers, and R. Morris, "Effects of loss rate on ad hoc wireless routing," MIT Laboratory for Computer Science, Tech. Rep. MIT-LCS-TR-836, March 2002. [Online]. Available: citeseer.ist.psu.edu/decouto02effects.html

[18] "Supplement to IEEE standard for information technology telecommunications and information exchange between systems - local and metropolitan area networks - specific requirements. part 11: wireless LAN Medium Access Control (MAC) and Physical layer (PHY) specifications. Amendment 2: higher-speed physical layer (PHY) extension in the 2.4 GHz band." 1999.

[19] "Supplement to IEEE standard for information technology telecommunications and information exchange between systems - local and metropolitan area networks - specific requirements. Part 11: wireless LAN Medium Access Control (MAC) and Physical layer (PHY) specifications: high-speed physical layer in the 5 Ghz band." 1999.

[20] "IEEE Std 802.11h-2003 (Amendment to IEEE Std 802.11, 1999 edn. (Reaff 2003)) as amended by IEEE Stds 802.11a-1999, 802.11b-1999, 802.11b-1999/cor 1-2001, and 802.11d-2001)." 2003.

[21] A. Kamerman and L. Monteban, "Wavelan(c)-ii: a high-performance wireless lan for the unlicensed band," *Bell Labs Technical Journal*, vol. 2, no. 3, pp. 118–133, 1997. [Online]. Available: http://dx.doi.org/10.1002/bltj.2069

[22] B. Sadeghi, V. Kanodia, A. Sabharwal, and E. Knightly, "Opportunistic media access for multirate ad hoc networks," 2002. [Online]. Available: citeseer.ist.psu.edu/sadeghi02opportunistic.html

[23] G. Holland, N. H. Vaidya, and P. Bahl, "A rate-adaptive MAC protocol for multi-hop wireless networks," in *Mobile Computing and Networking*, 2001, pp. 236–251. [Online]. Available: citeseer.ist.psu.edu/holland01rateadaptive.html

[24] M. Lacage, M. H. Manshaei, and T. Turletti, "Ieee 802.11 rate adaptation: A practical approach." [Online]. Available: citeseer.ist.psu.edu/719742.html

[25] S. Pang, Q.; Leung ; Liew, "A rate adaptation algorithm for ieee 802.11 wlans based on mac-layer loss differentiation," *Broadband Networks, 2005 2nd International Conference on*, pp. 659–667 Vol. 1, 3-7 Oct. 2005.

[26] B. Awerbuch, D. Holmer, and H. Rubens, "Effects of multi-rate in ad hoc wireless networks." [Online]. Available: citeseer.ist.psu.edu/676530.html

[27] Y.-B. Ko, V. Shankarkumar, and N. H. Vaidya, "Medium access control protocols using directional antennas in ad hoc networks," in *INFOCOM (1)*, 2000, pp. 13–21. [Online]. Available: citeseer.ist.psu.edu/ko99medium.html

[28] R. Choudhury and N. Vaidya, "On ad hoc routing using directional antennas," 2002. [Online]. Available: citeseer.ist.psu.edu/choudhury02ad.html

[29] Y. Lee, K. Kim, and Y. Choi, "Optimization of ap placement and channel assignment in wireless lans," *lcn*, vol. 00, p. 0831, 2002.

[30] A. Tyamaloukas and J. J. Garcia-Luna-Aceves, "Channel-hopping multiple access," in *ICC (1)*, 2000, pp. 415–419. [Online]. Available: citeseer.ist.psu.edu/459675.html

[31] A. Tzamaloukas and J. J. Garcia-Luna-Aceves, "A receiver-initiated collision-avoidance protocol for multi-channel networks," in *INFOCOM*, 2001, pp. 189–198. [Online]. Available: citeseer.ist.psu.edu/457950.html

[32] M. I. P. Inc., *MAX2820, MAX2820A, MAX2821, MAX2821A 2.4GHz 802.11b Zero-IF Transceivers Data Sheet rev. 04/2004*, California, USA, 2004.

[33] K. N. Ramachandran, E. M. Belding, K. C. Almeroth, and M. M. Buddhikot, "Interference-aware channel assignment in multi-radio wireless mesh networks," 2006, pp. 1–12. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4146830

[34] P. Kyasanur and N. H. Vaidya, "Routing in multi-channel multi-interface ad hoc wireless networks," Tech. Rep., December 2004. [Online]. Available: http://www.crhc.uiuc.edu/wireless/papers/pradeep-routing-dec2004.pdf

[35] S.-L. Wu, C.-Y. Lin, Y.-C. Tseng, and J.-P. Sheu, "A new multi-channel mac protocol with on-demand channel assignment for multi-hop mobile ad hoc networks," *ispan*, vol. 00, p. 232, 2000.

[36] P. Kyasanur and N. H. Vaidya, "Routing and interface assignment in multi-channel multi-interface wireless networks," vol. 4, 2005, pp. 2051–2056 Vol. 4. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1424834

[37] C. E. Perkins and E. M. Royer, "Ad-hoc on-demand distance vector routing," *wmcsa*, vol. 00, p. 90, 1999.

[38] C. E. Perkins and P. Bhagwat, "Highly dynamic destination-sequenced distance-vector routing (dsdv) for mobile computers," 1994, pp. 234–244.

[39] T. Clausen, P. J. (editors), C. Adjih, A. Laouiti, P. Minet, P. Muhlethaler, A. Qayyum, and L.Viennot, "Optimized link state routing protocol (olsr)," RFC 3626, October 2003, network Working Group. [Online]. Available: http://ietf.org/rfc/rfc3626.txt

[40] J. J. Garcia-Luna-Aceves and M. Spohn, "Source-tree routing in wireless networks," in *ICNP*, 1999, pp. 273–282. [Online]. Available: citeseer.ist.psu.edu/garcia-luna-aceves99sourcetree.html

[41] D. B. Johnson and D. A. Maltz, "Dynamic source routing in ad hoc wireless networks," in *Mobile Computing*, Imielinski and Korth, Eds. Kluwer Academic Publishers, 1996, vol. 353. [Online]. Available: citeseer.ist.psu.edu/johnson96dynamic.html

[42] E. Royer and C. Toh, "A review of current routing protocols for ad-hoc mobile wireless networks," 1999. [Online]. Available: citeseer.ist.psu.edu/royer99review.html

[43] A. J. A. K. A. Patwardhan, J. Parker and M. Iorga., "Secure routing and intrusion detection in ad hoc networks," March 8-12, 2005.

[44] J. Broch, D. A. Maltz, D. B. Johnson, Y.-C. Hu, and J. Jetcheva, "A performance comparison of multi-hop wireless ad hoc network routing protocols," in *Mobile Computing and Networking*, 1998, pp. 85–97. [Online]. Available: citeseer.ist.psu.edu/broch98performance.html

[45] B. Awerbuch, D. Holmer, and H. Rubens, "The medium time metric: high throughput route selection in multi-rate ad hoc wireless networks," *Mob. Netw. Appl.*, vol. 11, no. 2, pp. 253–266, 2006.

[46] ——, "High throughput route selection in multi-rate ad hoc wireless networks," Tech. Rep., 2003. [Online]. Available: citeseer.ist.psu.edu/awerbuch04high.html

[47] Y. Yang, J. Wang, and R. Kravets, "Interference-aware load balancing for multihop wireless networks," University of Illinois at Urbana-Champaign, Tech. Rep., 2005. [Online]. Available: http://www.cs.uiuc.edu/research/techreports.php?report=UIUCDCS-R-2005-2526

[48] ——, "Designing routing metrics for mesh networks," September 2005. [Online]. Available: http://info.library.unsw.edu.au/cgi-bin/local/access/access.cgi?url=http://www.ieee.org/ieeexplore

[49] [Online]. Available: http://www.ietf/org/html.charters/manet-charter.html

[50] V. D. Park and M. S. Corson, "A highly adaptive distributed routing algorithm for mobile wireless networks," *infocom*, vol. 00, p. 1405, 1997.

[51] C.-K. Toh, "Associativity-based routing for ad hoc mobile networks," *Wireless Personal Communications*, vol. 4, no. 2, pp. 103–139, March 1997. [Online]. Available: http://dx.doi.org/10.1023/A:1008812928561

[52] S. Biaz, B. Qi, and Y. Ji, "Improving expected transmission time metric in multi-rate multi-hop networks," *Consumer Communications and Networking Conference, 2008. CCNC 2008. 5th IEEE*, pp. 533–537, Jan. 2008.

[53] "The ns-2 simulator." [Online]. Available: http://www.isi.edu/nsnam/ns/

[54] "The hardware specification of orinoco card." [Online]. Available: http://www.agere.com/client/wlan.html

[55] D. Kotz, C. Newport, and C. Elliott, "The mistaken axioms of wireless-network research," Dartmouth College, Tech. Rep., July 2003.

[56] "Cisco aironet 802.11a/b/g wireless cardbus adapter." [Online]. Available: http://www.cisco.com/en/US/prod/collateral/wireless/ps6442/ps4555/ps5818/

[57] J. Jun, P. Peddabachagari, and M. Sichitiu, "Theoretical maximum throughput of ieee 802.11 and its applications," in *NCA '03: Proceedings of the Second IEEE International Symposium on Network Computing and Applications*. Washington, DC, USA: IEEE Computer Society, 2003, p. 249.

[58] M. Heusse, F. Rousseau, G. Berger-Sabbatel, and A. Duda, "Performance anomaly of 802.11b," in *Proceedings of IEEE INFOCOM 2003*, San Francisco, USA, March-April 2003. [Online]. Available: citeseer.ist.psu.edu/561001.html

[59] M. Li, "An extension of rate-adaptive mac protocol for ns2 simulator." [Online]. Available: citeseer.ist.psu.edu/702473.html

[60] T. S. Rappaport and T. Rappaport, *Wireless Communications: Principles and Practice (2nd Edition)*. Prentice Hall PTR, December 2001. [Online]. Available: http://www.amazon.ca/exec/obidos/redirect?tag=citeulike09-20\&path=ASIN/0130422320

[61] "The ns manual." [Online]. Available: http://www.isi.edu/nsnam/ns/ns-documentation.html

[62] R. Punnoose, P. Nikitin, and D. Stancil, "Efficient simulation of ricean fading within a packet simulator," *Vehicular Technology Conference, 2000. IEEE VTS-Fall VTC 2000. 52nd*, vol. 2, pp. 764–767 vol.2, 2000.

[63] "Cmu additions to the ns to handle ricean and rayleigh fading." [Online]. Available: http://www.ece.cmu.edu/wireless/downloads.html

[64] J. Padhye, S. Agarwal, V. N. Padmanabhan, L. Qiu, A. Rao, and B. Zill, "Estimation of link interference in static multi-hop wireless networks," in *IMC'05: Proceedings of the Internet Measurement Conference 2005 on Internet Measurement Conference*. Berkeley, CA, USA: USENIX Association, 2005, pp. 28–28.

[65] A. S. Tanenbaum, *Computer Networks, Fourth Edition*. Prentice Hall PTR, August 2002. [Online]. Available: http://www.amazon.ca/exec/obidos/redirect?tag=citeulike09-20\&path=ASIN/0130661023

[66] M. Conti, G. Maselli, G. Turi, and S. Giordano, "Cross-layering in mobile ad hoc network design," *Computer*, vol. 37, no. 2, pp. 48–51, 2004.

[67] V. Kawadia and P. Kumar, "A cautionary perspective on cross-layer design," *Wireless Communications, IEEE*, vol. 12, no. 1, pp. 3–11, Feb. 2005.

[68] "Xian framework." [Online]. Available: http://xian.sourceforge.net/archi/archi.html

[69] "Madwifi driver." [Online]. Available: http://madwifi.org/

[70] "Dsr-uu." [Online]. Available: http://core.it.uu.se/core/index.php/DSR-UU

[71] S. Keshav, "A control-theoretic approach to flow control," *SIGCOMM Comput. Commun. Rev.*, vol. 21, no. 4, pp. 3–15, 1991.

[72] S. H. Y. Wong, H. Yang, S. Lu, and V. Bharghavan, "Robust rate adaptation for 802.11 wireless networks," pp. 146–157, 2006.

[73] [Online]. Available: http://dast.nlanr.net/Projects/Iperf/