ROUTING AND CHANNEL ASSIGNMENT SCHEMES
FOR INTERFERENCE AVOIDANCE IN
WIRELESS MESH NETWORKS

by

FAWAZ SALEEM BOKHARI

Presented to the Faculty of the Graduate School of

The University of Texas at Arlington in Partial Fulfillment

of the Requirements

for the Degree of

DOCTOR OF PHILOSOPHY

THE UNIVERSITY OF TEXAS AT ARLINGTON

May 2012

ACKNOWLEDGEMENTS

"Whoever does not thank people, has not thanked God", said the Prophet Muhammad (Peace be upon him). So I would like to express my sincere appreciation and gratitude to my supervising professor Prof. Gergely Záruba, for his leadership, knowledge, and encouragement which made the completion of this dissertation possible. I am also grateful to my other committee members Prof. Manfred Huber, Prof. Farhad Kamangar, and Prof. David Levine for their advice, time and effort expended to improve this research. Further, I wish to thank my colleagues in ASSIST Lab, Awais Iqbal, and Vijay Dixit, for their contribution and companionship. I am grateful to the helpful staff of CSE department, especially Sherri Gotcher, Pam McBride, and Camille Costabile.

I am eternally indebted to my parents Saleem A. Bokhari and Shehzadi Tahira for their prayers, worries, and support during my long years abroad. I would also like to thank my brothers and sisters for their moral support towards the success of my PhD studies. Finally, my immense gratitude goes to my wife Mehwish for her endurance, support and prayers during the course of this work, specially the way she brought up our three children and took care of them while I was busy in my studies.

It would be unjust to ignore the fact that I enjoyed the good company of friends at UTA such as: Raza ullah Khan, Taufique, Fahad Makhdoomi, Ahmed, Osama Ali, Mohsin Kareem, Mohsin Rizwan, Yasir, Fahd Bhatty, Abid, Hafiz Fahd, Sarang, Waseem and Azhar.

April 2, 2012

ABSTRACT

ROUTING AND CHANNEL ASSIGNMENT SCHEMES

FOR INTERFERENCE AVOIDANCE IN

WIRELESS MESH NETWORKS

Fawaz Saleem Bokhari, PhD.

The University of Texas at Arlington, 2012

Supervising Professor: Dr. Gergely Záruba

This dissertation presents efficient routing and channel assignment schemes for interference avoidance in wireless mesh networks (WMNs). The most significant contributions of this dissertation are the development and design of two routing algorithms that help in improving network throughput by selecting *less interference paths* both for single and multiple radio WMNs and the design of an intelligent channel assignment scheme which increases the overall network capacity by assigning partially overlapped channels having less interference among neighboring ones for multi radio multi channel wireless mesh networks (MRMC-WMNs).

For single radio single channel WMNs, we propose AMIRA (Ant Mesh routing for InteRference Avoidance), an interference-aware routing protocol designed to improve load balancing by avoiding inter and intra flow interference in a typical mesh backbone network. AMIRA is based on the framework of Ant Colony Optimization (ACO) which is a meta-heuristic approach for stochastically solving a problem. ACO is used together with our local heuristic technique to avoid interference within and among packet flows. In AMIRA, each node uses MAC level information to measure link qualities which helps in selecting reduced interference

paths thus resulting in improved load balancing in addition to the auto load balancing feature of the ACO framework. We demonstrate through simulations that AMIRA quickly converges to the best path when traffic characteristics change. We tune the parameters of AMIRA to study the effect on the performance of routing load and end-to-end delay. Our simulation results demonstrate that under congestion, AMIRA gives increased throughput and low end-to-end delay when compared to other existing ant-based routing protocols because of its interference aware technique and stochastic data forwarding nature.

We then extend our work of *AMIRA* to develop a forwarding architecture-AntMesh that is designed for both single and multiple radio infrastructure WMNs and take care of both inter and intra flow interferences. AntMesh is a distributed interference-aware data forwarding architecture based on smart ants. In addition, we also propose a novel routing metric called Ant Routing Metric (ARM) designed to effectively utilize the space/channel diversity typically common in infrastructure WMNs. One interesting result of our investigation is that *AntMesh* has the capability to discover high throughput paths with less inter-flow and intra-flow interference when conventional wireless network routing protocols and metrics fail to do so. This conclusion is based on extensive evaluation and testing of *AntMesh* under various network scenarios both on fixed nodes mesh networks and on mobile WMN scenarios. The results obtained show *AntMesh*'s advantages that make it a valuable candidate to operate in MRMC mesh networks.

In the design of any WMN channel assignment scheme, understanding and mitigating interference is one of the fundamental issues. Therefore, we address the problem of channel assignment considering partially overlapping channels (POCs) for interference avoidance in multi radio multi channel wireless mesh networks (MRMC-WMNs). A novel interference capture model is proposed which provides a systematic approach of measuring the interference caused by links operating on POCs. This model takes both the adjacent channel interference and the corresponding physical distance between mesh nodes into account. Based on this model, we design a centralized and a distributed interference-aware channel assignment algorithm called *i*-POCA which enables the use of smart ants for assigning orthogonal and non-orthogonal channels to radios in order to minimize total network interference. We evaluate our algorithms through extensive simulations and demonstrate that our proposed algorithms improve network throughput by efficient utilization of the available spectrum.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF TABLES

CHAPTER 1

INTRODUCTION

The concept of wireless mesh networks (WMN) has emerged as a promising technology for the provision of affordable and low cost solutions for a wide range of applications such as broadband wireless internet access in developing regions with no or limited wired infrastructure, security surveillance, and emergency networking; where public safety teams like firefighters can still be connected with the help of just-in-time mesh nodes mounted on street poles. The main reason for this vast acceptance of mesh networks in the industry and academia is because of its self-maintenance and resiliency feature and the low cost of wireless routers. In addition, the self-forming feature of WMN makes the deployment of a mesh network easy thereby affording scalability to the network. Mesh networks of most commercial interests are characterized as fixed backbone WMNs where mesh nodes (routers or access points) are generally static and are mostly supplied by a permanent power source. Such a wireless mesh network architecture is illustrated in Figure 1.1, consisting of mesh routers, clients, and gateway nodes. Mesh routers (MR) communicate with peers in a multi hop fashion such that packets are mostly transmitted over multiple wireless links (hops). Nodes forward packets to other nodes that are not within direct transmission range of each other through these MRs. Routers which are connected to the outside world are called gateway nodes (GWN). GWNs carry traffic in and out of the mesh network. The collection of such routers and gateway nodes connected together in a multi hop fashion form the basis for an infrastructure WMN (also called backbone mesh). Moreover, the multi hop packet transmission in an infrastructure WMN extends the area of wireless broadband coverage without wiring the network and therefore enables the integration of mesh networks with other existing wireless networks such as cellular and ad hoc networks (sensor, & vehicular), 802.11 WLAN (Wi-Fi) and 802.16 based broadband wireless (WiMax) [1].

WMNs can be classified based on the number of radios on a mesh router i.e. in single radio mesh networks each node in the mesh network is equipped with only one radio; in multi radio mesh networks multiple radios are installed on each mesh node in the backbone mesh. Depending upon the radio to channel allocation(also called interface to channel assignment), mesh networks can be further classified into single radio single channel (SRSC), single radio multi channel (SRMC), multi radio single channel (MRSC) and multi radio multi channel (MRMC) wireless mesh networks. In a SRSC-WMN, as the name suggests, all nodes are configured to the same channel; this ensures network connectivity. However, the capacity of the network is greatly affected as all nodes are competing for channel access and therefore, interference minimization becomes a major issue. SRMC-WMNs can achieve parallel transmissions by configuring different orthogonal channels (OCs) to radios belonging to different nodes, thus improving network capacity. However, such networks severely suffer from network disconnections due to having a single radio on each node and thus facing the *exposed terminal problem*. In MRSC-WMNs, nodes are equipped with multiple radios and all of radios in the network are configured to the same channel. This type of network does not give any improvement; it actually performs worse than SRSC because the number of contending links is increased due to multiple radios on a mesh node. In, MRMC-WMNs, with the availability of off-the-shelf, low cost, IEEE 802.11 based networking hardware, it is possible to incorporate multiple radio interfaces operating in different radio channels on a single mesh router. This enables a potential large improvement in the capacity of mesh networks compared to all the previously listed forms of mesh networks [2].

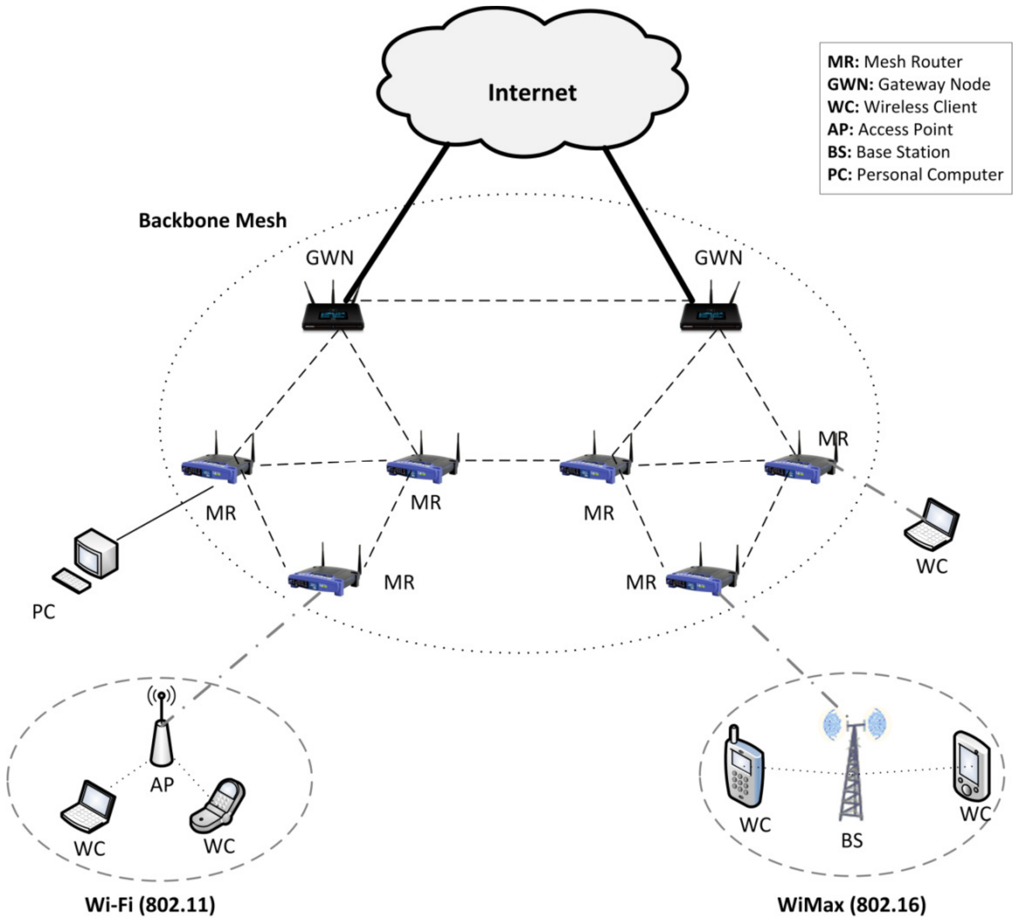


Figure 1.1 A typical wireless mesh network architecture

1.1. Interference in Wireless Mesh Networks

In a typical WMN, flows on links belonging to different nodes compete with each other in accessing the wireless medium. This results in possible interference among nodes and thus has the potential to severely affect WMN performance. Multiple types of interferences exist in a mesh network depending upon the flow characteristics and variable interface to channel configurations. We first explain what these different types of flow interferences are, particularly in an infrastructure WMN. Then, we will describe another classification of interferences that exists in mesh networks based on the configuration of the channels to radios and also on the number of radios installed in a particular node.

1.1.1 Flow based Interference

Inter-flow Interference: This type of interference occurs when neighboring nodes carrying different flows compete for channel access when they transmit on the same channel (as depicted in Figure 1.2.a). To reduce this type of interference, whenever a node is involved in a transmission, its neighboring nodes should not communicate at the same time with other nodes on the same channel.

Intra-flow Interference: Nodes on the path of a same flow compete with each other for channel access when they transmit (packets belonging to the same flow) on the same channel. This is referred to as intra-flow interference and is depicted in Figure 1.2 (b).

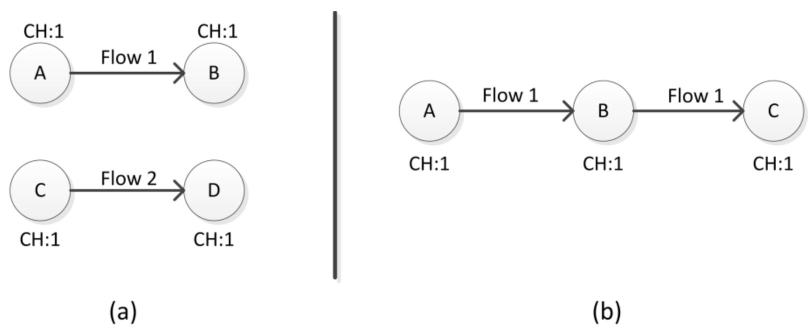


Figure 1.2 Flow based interference. (a) Inter-flow interference (b) Intra-flow interference

1.1.2 Interference based on Interface to Channel Configuration

A wireless mesh network utilizing both orthogonal and non-orthogonal channels suffers from interferences which can be characterized as follows.

Co-channel Interference (CCI): This is the most common type of interference, shown in Figure 1.3 (a) that exists in almost all wireless networks. It refers to the fact that radios belonging to two nodes, operating on the same channel would interfere with each other, if they are within the interference range of each other. This effectively means that parallel communication from two separate in-range nodes is not possible if such type of interference exist in a mesh network.

Adjacent Channel Interference (ACI): This happens mostly when radios on two separate nodes are configured to partially overlapping channels. For example, in Figure 1.3 (b), a radio on node *A* is configured to channel-4 while another radio at neighboring node *C* is configured to channel-1; then the transmission from either node would experience some sort of partial interference. This type of interference also restricts parallel communication depending on the channel separation and the physical distance between the two nodes.

Self Interference (SI): This is defined as transmission from a node interfering with one of its own transmissions. This is typically related to situations when nodes are equipped with multiple radios in a mesh network. Therefore, parallel communication cannot be achieved among multiple radios installed on a node, unless they are configured on completely orthogonal channels as shown in Figure 1.3 (c).

All of the above types of interference issues have to be considered when designing routing and channel assignment algorithms to exploit the full potential of the available wireless spectrum.

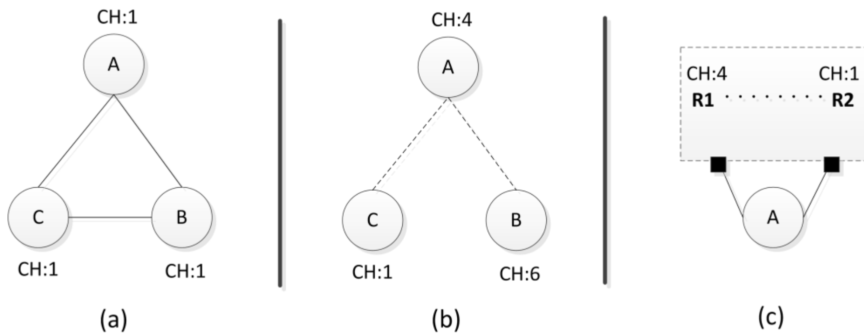


Figure 1.3 Types of interferences (a) co-channel interference (b) adjacent channel interference (c) self interference

1.2. Motivation and Objectives

Wireless mesh networks have some unique characteristics that set them apart from other existing wireless networks such as MANETs and sensor networks. For example, nodes(at least backbone router nodes) in a typical infrastructure mesh network are generally static and have no significant constraints on power consumption, as opposed to MANETs, where nodes have limited power capacity and are mostly mobile. Similarly, due to the shared nature of wireless medium, nodes compete with each other for channel access when they transmit on the same channel resulting in possible interference among the nodes. Unlike MANETs, where traffic could flow between any pair of mobile nodes, data flows in WMNs are typically between mesh nodes and Internet gateways. In general, traffic flows are saturated on certain paths in a WMN. Therefore load balancing is necessary to avoid hot spots and to increase network utilization. (Poor routes may exist for a long time in a static network and are likely to result in congestion and inefficient use of network resources). Furthermore, the fact that the nodal density in a typical WMN is high compared to other wireless networks, make the flow interference issue in WMNs more severe.

Routing and channel assignment in WMNs has been an active area of research for the last several years. In general, routing algorithms deal with discovering efficient data paths to forward network traffic and channel assignment provides link configuration to channels in such a way that the network throughput is improved by reducing link interference, thereby eventually increasing the network capacity. Designing efficient routing algorithms and channel assignment techniques in single and MRMC-WMNs are challenging problems mainly because of two reasons: i) the existence of network interference, and ii) the issue of load balancing in WMNs. Normally, in a typical multi radio mesh network, the total number of radios within the network is significantly higher than the number of available channels in the network (e.g. 11 channels in 802.11 b/g). This forces many links to operate on the same (set of) channels, resulting in possible interference among transmissions. The existence of such interference if not accounted for, can affect the capacity of the network by decreasing the throughput and may result in poor network performance. Therefore, understanding and mitigating interference has become one of the fundamental issues in MRMC-WMNs. Incorporating techniques to specifically address these challenges in WMN routing protocols and channel assignment schemes could improve the overall network flow capacity or performance of individual flows in the network. Apart from this, a routing or channel assignment algorithm for WMNs should also be adaptive, robust and self-healing which are common desired features for these types of algorithms in the wireless domain. A lot of research has been done in designing routing and channel assignment solutions to solve this problem [2, 5-8, 11, 14, 16, 21, 22, 34-48].

1.3. Contribution of this Dissertation

This thesis is focused specifically on *designing routing and channel assignment schemes for interference avoidance in WMNs*. In particular, we propose two routing algorithms that help improve network throughput by selecting less interference paths both for single and multiple radio WMNs. Moreover, we also present an intelligent channel assignment scheme which increases the overall network capacity by assigning partially overlapped channels (achieving less interference among neighboring nodes) for MRMC-WMNs. The main emphasis of our work is on how to improve throughput, provide efficient load balancing, achieve interference avoidance and increase network capacity in WMNs. In addition, this dissertation provides a comprehensive evaluation of our proposed routing and channel assignment schemes spanning from rigorous simulation testing to the actual evaluation of some of the proposed techniques on a real test-bed deployed at our university. In summary, the primary contributions of this thesis are as follows:

- By addressing the problem of packet routing for interference avoidance in infrastructure mesh networks, we designed a novel routing algorithm that incorporate techniques to specifically address the above mentioned unique characteristics of WMNs. The result is *AMIRA*; which is an interference-aware routing protocol designed to improve load balancing by avoiding interference in a typical single radio single channel (SRSC) mesh backbone network. AMIRA enables the use of smart ants acting as intelligent agents to probabilistically and concurrently perform the routing and data forwarding together with our local heuristic technique to avoid interference among packet flows. This custom designed local heuristic technique uses MAC level information to measure link qualities which helps in selecting reduced interference paths thus resulting in improved load balancing. One of the salient features of our proposed routing algorithm is that it is among the first work that studied the use of such bio-inspired agents i.e. ants for efficient packet routing in wireless mesh networks. (Almost all the previous research endeavors on ant-based routing were targeted on wired packet data networks and mobile ad-hoc networks).

- We studied the stability of *AMIRA*, i.e., how quickly it adapts itself to the changing dynamics of the network. We demonstrated that *AMIRA* quickly converges to the best path under situations when traffic characteristics change. We have also shown that under congestion, *AMIRA* gives increased throughput and low end-to-end delay when compared to other existing routing protocols because of its interference aware technique and stochastic data forwarding nature.
- One of the limitations of *AMIRA* is that it is designed for single radio single channel WMNs and therefore, only takes care of the inter-flow interference in WMNs. We extended our work to develop a forwarding architecture that is designed for both single- and multi-radio infrastructure WMNs and take care of both inter-flow and intra-flow interferences. We introduced *AntMesh*, a distributed interference-aware data forwarding architecture based on smart ants. For this, we designed a novel routing metric called Ant Routing Metric (ARM),with the sole purpose to effectively utilize the space/channel diversity typically common in infrastructure WMNs. One interesting result of our investigation is that *AntMesh* has the capability to discover high throughput paths with less inter- and intra-flow interferences while conventional wireless network routing protocols and metrics fail to do so. This conclusion is based on extensive evaluation and testing of *AntMesh* under various network scenarios, e.g., on both fixed and mobile node WMNs. The results obtained show *AntMesh*'s advantages that make it a valuable candidate to operate in MRMC mesh networks.
- We have developed a channel assignment scheme which considers the use of partially overlapping channels (POCs) for interference avoidance in the context of 802.11 b/g based MRMC-WMNs. The basic idea was to make the whole available wireless spectrum usable to nodes for channel selection based on the interference each channel

is experiencing. The first step in developing mechanisms which take advantage of the partial overlap is to build a model that captures this overlap in a quantitative fashion. Therefore, we proposed the partial overlapped interference graph (POIG) model, which is a weighted undirected interference graph and models an 802.11b based MRMC-WMN employing partially overlapped channels. Our custom designed weight function in the POIG models both the interference effects of POC and the corresponding physical distance between mesh nodes operating on those adjacent channels. This lead us to designing *i*-POCA; a channel assignment scheme which enables the use of smart ants for assigning orthogonal and partially overlapped channels (non-orthogonal) to radios in order to minimize total network interference and also to fully utilize the available 802.11 spectrum. Two versions of *i*-POCA, i.e. centralized (*i*-POCA-C) and distributed one (*i*-POCA-D) are presented in this thesis and compared with other existing partially overlapped channel assignment techniques. We showed that *i*-POCA performs well when compared to others in terms of throughput and channel utilization mainly because of the efficient allocation of partially overlapped channels resulting in more parallel communication thereby increasing overall network capacity. This also indirectly validates that our proposed POIG model models the interferences quite accurately most of the time.

Several results presented in this dissertation have also been published in peer-reviewed publications [23 – 27, 64, 65].

1.4. Organization of this Dissertation

Chapter 2 summarizes prior research in routing and channel assignment in WMNs. Two classes of routing algorithms are described: i) heuristics based techniques, and ii) swarm intelligence based techniques. A survey of existing channel assignment schemes exploiting partially overlapping (non-orthogonal) channels is also presented in this chapter.

Chapter 3 presents the formulation of our packet routing algorithm for interference avoidance in single radio WMNs. It starts with the description of smart ants, the rational on the use of such ants for routing in WMNs and what are some of the particular characteristics that those smart ants should have in order to perform the required task of routing. Then the design of the AMIRA routing approach is presented for smart ant based interference avoidance in WMNs. This is followed by a simulation evaluation, comparing results with other, competing approaches.

Chapter 4 extends Chapter 2 and presents AntMesh: a forwarding architecture based on smart ants for multi radio multi channel WMNs. We describe the AntMesh architecture by first explaining the smart ant packet structure and other data structures used for efficient data forwarding. Our novel ant routing metric (ARM) for calculating routing paths is presented and the necessary properties of this ARM metric are described. We then design rules for capturing inter and intra flow interference by the ARM metric and discuss why these rules are important by explaining the limitations of existing metrics. Finally, we provide extensive performance evaluation of AntMesh comparing it with existing approaches and show its viability.

Chapter 5 presents a novel channel assignment scheme considering partially overlapping channels for interference avoidance in MRMC-WMNs. We first propose an interference model called partially overlapping interference graph (POIG), which provides a systematic approach of measuring the interference caused by links operating on POCs (by taking into account both the adjacent channel interference and the corresponding physical distance between mesh nodes). We then design a centralized and a distributed interference-aware channel assignment algorithm called i-POCA which enables the use of smart ants for assigning orthogonal and non-orthogonal channels to radios in order to minimize total network interference. We evaluate our algorithms through extensive simulations and demonstrate that *i*-POCA improves network throughput by efficient utilization of the available spectrum.

Chapter 6 concludes this dissertation and discusses our future work.

CHAPTER 2

BACKGROUND AND RELATED WORK

This chapter discusses work related to routing and channel assignment algorithms in wireless mesh networks. Section 2.1 presents an overview of some of the popular routing algorithms in single and multi-radio wireless mesh networks. We also provide a brief introduction to the framework of ant colony optimization. Section 2.2 presents existing approaches to channel assignment in wireless networks based on partially overlapping channels.

2.1. Overview of Routing Algorithms

Generally, routing algorithms can be defined as multi-objective optimization problems in a dynamic stochastic environment. However, formalizing routing as such optimization problem requires complete knowledge of traffic flows between each node in the network; this is prohibitively difficult to model in the presence of rapidly changing network dynamics (found not only in typical WMNs but in any intra- or internet). Therefore, heuristic policies are often used to create quasi-optimal routing in WMNs. Most of the existing routing protocols in WMNs use traditional routing algorithms such as distance vector routing (AODV) [3] and link state routing (OLSR) [4] in order to calculate the shortest path among nodes in the network. Moreover, once the route is calculated in these protocols, data forwarding is performed deterministically. However, optimization criteria for determining the shortest path (also called *routing metric*) have received significant research attention in the past few years. There has been a significant body of research in designing efficient heuristic based routing protocols and metrics for WMNs (for a quick overview see [5 - 8]). In the wired networking domains, a new family of routing algorithms has been proposed based on swarm intelligence by Dorigo et al. called Ant Colony Optimization (ACO) framework [9, 10], which is a meta-heuristic approach for solving hard optimization problems.

2.1.1 Ant Colony Optimization

ACO algorithms draw their inspiration from the behavior of real ants, which are known to find the shortest path between their nest and a food source by a process where they deposit pheromones along trails (acting like a local message exchange in a communication network). Ants generally start out moving at random, however, when they encounter a previously laid trail, they can decide to follow it, thus reinforcing the trail with their own pheromone substance. This collective behavior is a form of an autocatalytic process where the more ants follow a trail, the more attractive that trail becomes to be followed by future ants. This process is thus characterized as a positive feedback loop, where the probability with which an ant chooses a path increases with the number of ants that previously chose the same path [10]. Hence, in ACO *artificial ants* probabilistically build a solution iteratively by taking into account pheromone trails or/and local heuristic information as well. Since, artificial ants move stochastically instead of deterministically, they explore a wide variety of possible solutions to a problem, independently, and in parallel.

In applying ACO to network routing, the problem is reduced to finding shortest path between each pair of nodes in a network much like in conventional distance-vector (DV) or link state routing (LS) protocols. Artificial ants are relatively small in size and are embodied by special routing packets. They can be piggybacked over data packets; in this way more frequent control information can be transmitted on the network when compared to traditional routing packets which could carry tables with them. Another promising feature of ACO based routing algorithms over traditional routing algorithms is their ability to produce good suboptimal solutions in a very short period of time [11]. Similarly, the way routing is performed, the forwarding tables are built and data packets are forwarded is different from those of conventional routing protocols (DV and LS).Previous research efforts concentrated on using ACO framework for efficient data forwarding in wired packet data networks [11, 12, 13] and mobile ad-hoc networks (MANETs) [14- 22]. For wireless mesh networks, we presented an interference-aware routing scheme based on ACO framework for single and multi-radio WMNs in [23 - 27].

2.1.1.1 Basic Algorithm

Applications of ACO generally fall into two categories namely i) static combinatorial optimization problems such as solutions for asymmetric traveling salesman problem [28, 29], graph coloring problem [30] and ii) dynamic combinatorial optimization problems such as routing in data networks [11, 12] which is the focus of our research.

Ants (small probing packets) are launched from each network node to a randomly selected destination node. When an ant *k* is at an intermediate node *i*, it selects the next node *j* according to a probabilistic decision rule which is a function of a local pheromone value (maintained by node *i*) and some heuristic information [31]. Equation 2.1 shows the function used for calculating this probability:

$$P_{ij}^{k} = \frac{[\tau_{i,j}]^{\alpha} \cdot [\eta_{i,j}]^{\beta}}{\sum_{l \in N_k} [\tau_{i,l}]^{\alpha} \cdot [\eta_{i,l}]^{\beta}} \tag{2.1}$$

where $\eta_{i,j}$ is the local heuristic value calculated a priori and models the time required to send a packet from node i to node j; α and β are the relative influence of the pheromone value and the heuristic information respectively; N_k is the list of neighbors of node *i* that ant *k* has not visited yet.

Each node maintains a pheromone table which contains the probability (or fitness) of selecting the next hop in order to reach a specific destination. Doing this, ants traveling to a specific destination collect information about the quality of their path and once they reach the destination, track back to the source, and update the pheromone table at each intermediate node thus emulating laying down the pheromone track. The update of the pheromone table during the backward trip is done by increasing the pheromone value of the link over which the ant arrived as shown in Equation 2.2:

$$\tau_{i,j} = \tau_{i,j} + \Delta\tau \qquad (2.2)$$

Similarly, the pheromone values of all the other neighbors of that node i are decreased as shown in Equation 2.3:

$$\tau_{i,j} = \frac{\tau_{i,j}}{1+\Delta\tau} \qquad for\ all\ \ j \in N_i^k \qquad (2.3)$$

Hence, the routing information is collected through the sampling of paths by the ants (generated independently at each node in the network). Since ants sample complete paths repeatedly and concurrently, routing information is updated in a pure Monte-Carlo fashion.

2.1.2 Ant-based Routing Protocols

This section provides an overview of some of the more prevalent, ant-inspired routing algorithms in communication networks.

AntNet: one of the first applications of ant colony optimization framework (ACO) for wired network routing is AntNet [11]. In AntNet, routing is achieved by generating forward ants at regular intervals from a source node to a destination node to discover a low cost path and by backward ants that travel from destination to source to update pheromone table at each intermediate node. Forward ants keep track of trip times from source to destination node using the data traffic queues in order to experience the same delay that data packets experience. Forward ants select the next hop by a probabilistic decision rule which takes into consideration the pheromone intensity which is reinforced by other backward ants and heuristic information which is based on the queue length of the intermediate node. Once, a forward ant reaches the destination node, a backward ant is generated that tracks back to the source using high priority queues (for timely delivery) reinforcing the selection probability of intermediate nodes according to the fitness of the trip times of forward ants. However, the fact that AntNet was proposed for wired networks makes it unsuitable for WMNs because of their unique characteristics, e.g., wireless interference, load balancing, etc. There are variations and extensions, e.g., [13, 32], to the original AntNet algorithm which targeted wired networks and thus are outside the scope of our research.

AntHocNet [16]: is a hybrid, multi-path, ant based algorithm. It consists of both a reactive and a proactive component. The reactive part is used for route establishment whereas the proactive component is used for route maintenance. The reactive component is used at the start of a data session where a reactive forward ant is broadcast to find multiple paths to the destination and upon reaching the destination; a reactive backward ant sets up the multiple paths to the destination using local heuristic information. While data packets are being routed, proactive forward ants are also generated. This helps in exploring new paths and getting up-to-date link quality information. One of the drawbacks of AntHocNet is the number of ants that need to be sent over the network for establishing routes to destinations as they are broadcast during a route discovery phase. Also, each ant stores list of visited nodes from source to current node and depending upon the distance to the destination, this list (and thus the ant's size) can grow long, increasing routing overhead.

POSANT: the authors of [14] present a position based ant colony routing algorithm for MANETs, in which they make use of position based instruments (e.g., GPS receivers) to combine with the ACO technique. POSANT is a reactive routing algorithm and [14] argues that the use of position information can greatly reduce the number of ant generations while reducing the route establishment time as well. Position information is used in the heuristic maintained at each node helping ants decide what next hop to take in the path discovery phase. However, POSANT makes an assumption that the transmission time of an ant to its neighbors is the same for all nodes therefore ignoring packet loss and flow interference, which are quite common in wireless networks.

SARA: Rosati *et al*., [21] have proposed a distributed ant routing algorithm for ad hoc networks where nodes are frequently joining and leaving the network and minimum signaling overhead is required. One of the design objectives of their proposed approach was to have low computational complexity and for this particular purpose, the algorithm creates routes on-demand making it a reactive protocol. Moreover, ants in SARA store only the node identity information to avoid extra control overhead and only the pheromone value is used to make a forwarding decision. The authors have conducted extensive simulations and showed the effectiveness of their algorithm in ad hoc networks with critical connectivity when compared with AODV. Since the approach aims at minimizing complexity in the nodes making it as simple as possible, the authors claim that SARA can be helpful in achieving seamless routing in heterogeneous networks. However, the same simplicity comes with the drawback that it does not capture the network dynamics in terms of link characteristics and interference. Therefore, it is not suitable for networks with high nodal densities and traffic flows moving to mostly one side of the network (gateways) as is typical in WMNs.

DAR: recently, the authors of [22] have presented a simple ant routing algorithm for ad hoc networks with the objective of minimizing protocol overhead. To achieve this objective, they have proposed a control neighbor broadcast method in which, a broadcast ant is unicast to the second-hop neighbor. Similarly, data packets are used to refresh paths of the active sessions in order to reduce the control overhead. Their third approach is implemented in link failure situation to repair that link locally using the two ends of the link rather than initiating a new source to destination path request. They have conducted extensive simulations to prove the main objective of their proposed approach i.e. less signaling overhead when compared with other existing ant-based routing algorithms.

2.1.3 Review of Mesh Routing Protocols and Metrics

Most existing research in mesh routing is focused on designing efficient routing metrics since they determine the creation of network paths in a mesh network. De Couto et al. have proposed a link metric called *Expected Transmission Count* (ETX) [5] to find high throughput paths by calculating link loss ratios. ETX calculates the link loss ratio by measuring the expected number of packet transmissions at the MAC layer needed to send a unicast packet on a link. However, ETX is designed for single-radio single-channel mesh networks and thus cannot take inter/intra flow interference into account. Furthermore, it does not consider variable transmission rates over different links as usual in WMNs. Another notable study is presented in [6] where a metric termed *Expected Transmission Time* (ETT) is proposed trying to overcome the limitations of ETX by considering the differences in link transmission rates. The authors of [6] have proposed another path metric for multi-radio mesh network called weighted cumulative ETT (WCETT) to capture the intra-flow interference within a path in a mesh network. The

WCETT metric reduces the number of nodes on a path that transmit on the same channel to reduce the intra-flow interference but it overlooks the situation when two sufficiently spatially far links belonging to the same path could be transmitting on the same channel without interference. Furthermore, WCETT does not consider the inter-flow interference among the nodes in a network. To capture both the inter-flow and intra-flow interference Yang et al. [7] have proposed a path metric called *metric of interference and channel switching* (MIC). It captures the inter-flow interference by considering the number of neighbors interfering with the transmission on both ends of the link. Similarly, to overcome some of the limitations of MIC, Subramanian et al. have proposed an Interference Aware (iAware) metric in [8] by using a more accurate *physical model* (as compared to *protocol models* being used by others) to characterize interference relationships in wireless networks. Although iAware follows the design of the MIC metric, it differs in measuring the interference among nodes by directly taking the average interference generated by neighboring nodes. In [23 - 17], we proposed ant routing algorithms based on the ACO framework specifically designed for single radio WMNs in order to capture the complex interaction between inter-flow and intra-flow interference typically common in WMNs.

2.2. Overview of Channel Assignment Algorithms

In the beginning, the problem of channel assignment (frequency assignment) focused on cellular networks[33]. However, with the proliferation of IEEE 802.11 based technologies in the wireless arena (WLANs, sensor networks, WMNs), channel assignment (CA) solutions were also sought based on the peculiar characteristics of such networks which are different from cellular networks. For example, base stations in a cellular network are typically connected via wired connections, whereas mesh nodes in WMNs are connected wirelessly. This brings up the issue of interference in mesh networks between mesh nodes, an issue not present in cellular networks, between base stations. Also, the bottleneck in cellular networks is from the base stations to the client device, whereas, in mesh networks, the bottleneck exists inside the mesh backbone network typically along the route from the mesh routers to the gateway nodes. In addition in cellular networks nearby base stations (cells) are configured on completely orthogonal channels (OCs) to avoid interference; this is not possible in a mesh backbone network, as i) the nodal density of a typical WMN is very high and the number of orthogonal channels available is limited and ii) time consuming engineering calculations to determine best frequency allocations and base station locations are unviable.

Most existing deployed mesh networks are based on IEEE 802.11 technologies; among the standards of 802.11, the most widely used ones are 802.11b/g, which support up to 14 channels in the unlicensed Industrial, Scientific, and Medical (ISM) radio band at 2.4 GHz [34]. Out of these 14 channels, only 11 channels are available for use in US (13 channels are open in EU, while Japan has made all of the channels available). In US, the 2.4 GHz band is divided into 11 channels as shown in Figure 2.1, where the channel numbers represents the center frequency of that channel. For example, channel 6 operates at *F=2.437* GHz. The channel bandwidth is 22 MHz and each channel's center frequency is separated from the next channel's center frequency by 5 MHz. Therefore, a channel overlaps with four of its neighboring channels which leave only three non overlapping (orthogonal) channels in the 2.4 GHz band (i.e. 1, 6, and 11 as depicted in Figure 2.1). Similarly, IEEE 802.11a operates in 5 GHz band and provides 12 orthogonal channels, but since it operates in higher frequency band, it has a shorter range as opposed to 802.11b/g (higher frequencies in general have higher inabilities to penetrate walls and obstructions). Recently, the IEEE 802.11n standard was proposed which operates in both the 2.4GHz and 5 GHz bands and provides legacy support to devices operating based on previous standards (b/g). It provides data rates of up to 600Mbps using

Multiple Input, Multiple Output (MIMO) technology with Orthogonal Frequency Division Multiplexing (OFDM).

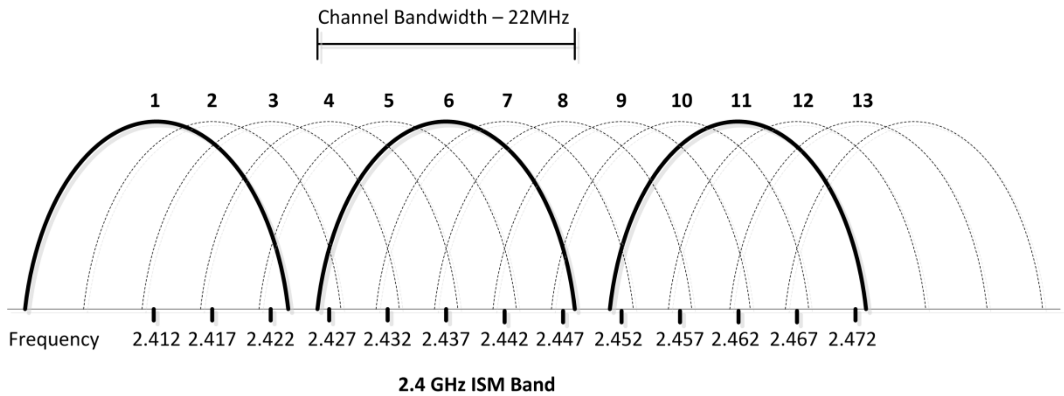


Figure 2.1 IEEE 802.11 b/g channel distribution showing the 3 orthogonal channels in bold color

The use of POCs for channel assignment in wireless networks has received some research attention in the past [2, 35-48]. We will quickly revisit some of these existing channel assignment algorithms based on POCs for wireless networks as we are going to refer to these approached in our comparative evaluation.

One of the first studies on exploiting partially overlapped channels appears in [35]; it shows that a systematic approach to exploit channels with partial spectrum overlap can lead to efficient spectrum utilization and improved network throughput in a wireless LAN environment. In addition, [35]provides empirical results showing that two links operating on partially overlapped channels with a separation of three-channels provides the same level of throughput as when they are operating only on two orthogonal channels.

In [48], Ding, *et al.*, proposed a genetic algorithm for POC-based channel assignment in WMNs. In order to model the interference accurately, they have extended the traditional conflict graph model to capture the interference by considering both channel separation as well as physical distance of the nodes. They have shown through simulations that POC works better in denser networks. Similarly, the authors in [43] formulated a linear mixed integer program of a joint channel assignment and link scheduling problem exploiting partially overlapping channels. Their proposed scheme achieves 90% improvement in aggregate network capacity when all 802.11b channels are employed.

In [44] a centralized heuristic based channel assignment algorithm for POCs is proposed. The authors develop an interference model called I-Matrix that measures the co-channel and adjacent channel interference among the nodes in order to assign channels to radios efficiently. Given the network topology their heuristic algorithm arranges the links according to their nodal degree. One of the drawbacks of this scheme is that it does into take into account the self-interference issue typically common in a MRMC-WMN, in which a node having multiple radios cannot transmit packets simultaneously on all of its radios unless they are configured on orthogonal channels. To overcome these issues the authors of [2] have extended the work of [44].

Recently, Cui, *et al.*, [36] have proposed an approximation algorithm for channel assignment in 802.11 based wireless networks. The objective of their approximation algorithm is to minimize total network interference for throughput maximization. The authors introduced the notion of node orthogonality to capture the fact that two nodes on adjacent channels are orthogonal only if they are physically sufficiently apart. They have demonstrated that the network throughput can be increased by using overlapping channels. However, their proposed solution addressed single radio WMNs. Our proposed algorithm is designed for multi-radio multi-channel WMNs and uses a more accurate partially overlapping interference graph model to capture all types of interferences that may exist in WMNs.

CHAPTER 3

AMIRA: AN INTERFERENCE-AWARE ROUTING ALGORITHM BASED ON SMART ANTS

In this chapter, we address the problem of packet routing for interference avoidance in wireless mesh networks (WMNs). We propose AMIRA(Ant Mesh routing for InteRference Avoidance), an interference-aware routing protocol designed to improve load balancing by avoiding inter and intra flow interference in a typical mesh backbone network. AMIRA is based on the framework of Ant Colony Optimization (ACO) which is a meta-heuristic approach for stochastically solving a problem together with our local heuristic technique to avoid interference within and among packet flows. In AMIRA, each node uses MAC level information to measure link qualities which help in selecting reduced interference paths thus resulting in improved load balancing in addition to the auto load balancing feature of the ACO framework. The stability of any routing protocol depends upon how quickly it adapts itself to the changing dynamics of the network. We demonstrate through simulations that AMIRA quickly converges to the best path when traffic characteristics change. We tune the parameters of AMIRA algorithm to study the effect of these on the performance in terms of the routing load and end-to-end delay. Our simulation results demonstrate that under congestion, AMIRA achieves increased throughput and low end-to-end delay when compared to other existing routing protocols because of its interference aware technique and stochastic data forwarding nature.

The rest of the chapter is organized as follows. In Section I, the concept of smart ants in mesh networks and the necessary properties that they should possess are discussed. We present AMIRA; our proposed interference-aware routing scheme based on ACO in Section II. Section III details a simulation based evaluation of AMIRA with two existing routing protocols along with our analysis of the results. Finally we list directions for future work and conclude our chapter in Section IV.

3.1. Smart Ants in Mesh Networks

This section explains the rational on the use of ants for routing in WMNs. We will first describe the characteristics of WMNs that makes them different from other existing wireless networks like MANETs. Then we will explain the concept of the smart ants, how they differ from regular ants used in previous techniques and the necessary properties that these smart ants should possess in order to efficiently perform routing in WMNs.

Although the ant-based protocols described in the previous chapter do provide good performance benefits compared to existing protocols, the fact that they are primarily designed for wired and ad hoc networks makes them unsuitable to be directly applied in WMNs. Some of the distinguishing characteristics of WMNs are described below:

- *Static nodes and no power constraint*: In WMNs, nodes (at least relay nodes) are generally static and have no significant constraints on power consumption, as opposed to MANETs, where nodes have limited power capacity and are mostly mobile.
- *Interference*: Due to the shared nature of wireless medium, nodes compete with each other for channel access when they transmit on the same channel resulting in possible interference among the nodes. There are two types of flow interferences i.e. inter-flow and intra-flow interference as explained in Chapter 1.
- *Load Balancing*: Unlike MANETs, where traffic flows between any pair of mobile nodes, data flows in WMNs are typically between mesh nodes and Internet gateways. In general, traffic flows are saturated on certain paths in a WMN. Therefore, load balancing is necessary to avoid hot spots and to increase network utilization (poor

routes may exist for a long time in a static network and are likely to result in congestion and inefficient use of network resources).

- *Channel diversity*: Since network nodes in a WMN can be equipped with multiple radios, simultaneous transmission and reception techniques can be used employing intelligent channel assignment schemes (AMIRA does not explicitly utilize this multi-channel feature of WMNs currently).
- It is because of these above characteristics of WMNs, which has given us an impetus to design a new routing protocol based on smart ants that is interference-aware and at the same time exploits the characteristics of such biological agents (ants) in WMNs.

3.1.1 Desirable properties of smart ants

Let us now consider what kind of ants would be suitable for creating network paths in defining an ant based routing algorithm for MWNs. We argue that a routing algorithm based on smart ants designed for mesh networks should have the following desirable properties:

- Smart ants while creating paths should consider both types of interferences that inherently exist in mesh networks namely inter-flow interference among the nodes and intra-flow interference along the path of a flow.
- Smart ants should be able to evaluate the load on nodes in order to properly qualify the outgoing links. This would help in *detouring* the packets to new route and hence would result in a more load balanced network.

Smart ants are designed to exhibit the above mentioned desirable properties; it is because of these properties that our ants are “smart” (intelligent) and that is why we call them smart ants. In the following, we show how to incorporate these desirable properties in our smart ant-based routing algorithm (AMIRA).

3.2. AMIRA Approach

In this section, we present AMIRA; a distributed routing algorithm which incorporates smart ants to find high throughput paths with less interference and improved load balancing specifically designed for wireless mesh networks. We have transformed the basic ACO framework WMNs considering the distributed and dynamic characteristics of the later but retaining the basic ACO features (such as the need for a colony, the role of autocatalysis, the cooperative behavior mediated by artificial pheromone trails, the probabilistic construction of solutions biased by artificial pheromone trails and local heuristic information [31]).

3.2.1 Model

The basic operation of AMIRA is as follows: smart ants in the form of control packets are generated at regular intervals from each node towards destinations in the network. Indeed, two types of ants are generated: forward smart ants (FSA) which travel from source to destination and backward smart ants (BSA) traveling from the destination to the source much like the ants in [11].However, in AMIRA, both FSA and BSA use high priority queues so that the FSA do not need to carry their per hop experienced trip times, rather BSA will estimate their trip time using the link estimation table being maintained at each node. In AMIRA, every node maintains three types of data structures.

Pheromone Table: This table stores the fitness of choosing a specific neighbor as next hop to reach a particular destination in the form of a probability (similar to the table in [11]). The pheromone table at a node *k* contains *m* rows where $m \in N_k$ (neighbors of a node k) and each row contains *l* entries, where *l* is the total number of possible destinations in the network. So an entry P_{id} is the probability of sending a packet to destination *d* via link *i* and thus following relation holds for every column in the pheromone table at node *k* :

$$\sum_{i\epsilon N_k} P_{id} = 1 \; d\epsilon[1 \ldots N] \tag{3.1}$$

where N is the total number of nodes in the network.

Delay table: The second data structure that is maintained by each node is the delay table that stores the average trip time to each destination in the network from the current node (thus this table will have *m* entries one for each possible destinations in the network). The value stored is an average calculated from the delay value carried by the last W number of ants received.

Local Heuristic Table: Every node in AMIRA maintains a local heuristic table at the MAC layer which contains the quality/strength of the outgoing links of that particular node. This strength is calculated by the MAC layer. In this chapter, we have considered that the nodes are equipped with 802.11 [49] MAC with distributed coordination function (DCF). A more detailed explanation of how the quality of a link is calculated by MAC layer is provided later in this chapter.

3.2.2 Neighbor Initialization Rule

AMIRA gives higher pheromone values to neighbors of a node on initialization, as opposed to existing ant-based algorithms [14, 16, 21, 22] which assume a uniform probability distribution over all other nodes when a node is initialized. This means giving equal probability of selecting a specific neighbor as a next hop to reach a destination to all the neighbors of a node. However, by giving higher pheromone values to the neighboring nodes which would also be the destination nodes for some nodes in the network could result in fast network convergence and also reduce network resource usage. Therefore, in AMIRA, every node assigns the initial probabilities to all the destinations in the network as follows. For a destination node *d* that is not a neighbor node, the initial probability assignment to all the neighbors N_k of node k is the same (i.e. $1/N_k$). However, for a destination node d which is also a neighbor node the initial probability in the routing table is:

$$P_{d,t} = \frac{N_k+1}{2N_k} where d = t \tag{3.2}$$

The probabilities for the remaining neighboring nodes are:

$$P_{d,i} = \frac{1}{2N_k} where i = N_k \tag{3.3}$$

3.2.3 Node Transition Rule

In order to increase the chances of selecting paths with less interference and more throughput, AMIRA has adopted a pseudo-random node transition rule. A forward smart ant v at an intermediate node k chooses the next hop u to reach to a particular destination d according to Equation. 3.4 and Equation. 3.5.

$$u = \begin{cases} \arg\max_{u \in N_k}\{\tau_v(d,u)\} \; if p \leq p_0 \\ P_v(d,u) otherwise \end{cases} \tag{3.4}$$

$$P_v(d,u) = \frac{\tau_v(d,u)}{\sum_{i \epsilon N_k} \tau_v(d,i)} \tag{3.5}$$

where p is a random number uniformly distributed between [0, 1] and p_0 is a constant in the range [0, 1]. Similarly $\tau_v[d,$u] is the pheromone intensity on the link connecting next hop u with k to reach the destination d. Equation. 3.4 indicates that if $p \leq p_0$, the node with the maximum pheromone value among the neighbors will be selected, otherwise, a proportional selection will be made with the probability $P_v(d,u)$ based on the probability distribution of the neighbors as shown in Equation. 3.5. Note that the parameter p_0 determines the relative importance of exploitation versus exploration. In AMIRA, since pheromone table is the only data structure that contains routing information, all data packets are forwarded based on this pseudo-random node transition rule. Therefore, a high value of p_0 would direct all the traffic to the best path as the link with maximum pheromone value would be selected most of the time and a lower value of p_0 will allow the data traffic to be spread across multiple links thereby resulting in automatic load balancing. At what value p_0 should be set to depends highly on the offered load on the mesh network; we will study the effect of this parameter to AMIRA performance in detail in our simulation section. Setting p_0 values to 1 will always forward packets on links with the best current quality.

3.2.4 Interference Estimation Rule

This rule is paramount to AMIRA, not found in other ACO routing protocols. Our interference estimation rule is designed to capture the characteristics of WMNs including the types of interferences that exist in such networks and to incorporate the desired properties of smart ants into AMIRA.

During a backward smart ant's travel to the source, each node that encounters the "BSA" updates its own routing table. They do so by calculating estimates of the ant's trip time from the current node to the destination node. This trip time is the average transmission time it takes to send a data packet from the current node to the next hop node from where the smart ant has arrived and then by adding this to the estimated accumulated trip time calculated so far by the backward smart ant.

We believe that the most appropriate method for estimating trip time is to measure the delay required to transmit a packet at the MAC layer over a link, called link transmission delay. In other words, we want to make use of the information that has already been collected by 802.11 MAC layer using DCF. We define link transmission delay as follows:

$$Delay_{data} = N_{transmit} \times \left(MAC_{overhead} + \frac{L_{pkt}}{R_s} \right) \tag{3.6}$$

where $N_{transmit}$ is the number of transmissions including retransmissions needed to successfully receive a packet. We have set $N_{transmit}$ to be 1 in our simulations as almost all the current wireless devices are equipped with multi-rate feature which automatically adjusts the link rate according to the link quality, resulting in successful transmission of around 90% of the packets at the first time (as shown in [50]). L_{pkt} is the data packet size and R_s is the link speed. $MAC_{overhead}$ comes from the delays in the standard packet sequence of sending a data packet in 802.11 MAC standards, calculated as follows:

$$MAC_{overhad} = T_{rts} + T_{cts} + 3T_{sifs} + T_{difs} + T_{ack} \tag{3.7}$$

where T_{rts}, T_{cts}, T_{ack} are the time for the transmissions of RTS, CTS , and ACK frames respectively; T_{sifs} and T_{difs} are the inter-frame spacing times: SIFS and DIFS respectively.

Since the 802.11 MAC has a contention based nature, the link delay time is a random variable. Therefore, in order to mimic a real packet transmission delay, a smoothing window approach is used over a time interval to get the average value of the link transmission delay in AMIRA. If T^i_{data} is the current delay in sending one packet on a link then average link transmission delay is calculated by the following exponential running average:

$$Delay_{data}(avg) = \alpha\, T^{i}_{data} + (1-\alpha)T^{i-1}_{data} \tag{3.8}$$

where α represents the learning rate (valued between 0 and 1) controls how much relevance is given to old readings compared to new reading. The local estimate of the trip time is defined as the average link transmission delay times the total number of packets in the queue of a node *k* denoted as Q_k at the MAC layer:.

$$T_{data} = Delay_{data}(avg) \times Q_k \tag{3.9}$$

Since, nodes transmitting on the same wireless channel compete for the shared medium if a node is involved in a transmission on a wireless link, the neighboring nodes cannot communicate at the same time with other nodes (CSMA). The above MAC layer measurement also captures the interference caused due to the shared wireless channel access of the competing flows and therefore a high contention link would result in increased link transmission delay.

3.2.5 Pheromone Updating Rule

When the local trip time is calculated, the backward smart ant triggers the creation of an update of the delay table. Similarly, pheromone table values corresponding to a specific destination are also updated. However, there is a need for a mechanism to correctly integrate the local link quality estimates that our MAC layer has measured, into the pheromone table entries. This will improve the routing decision process with respect to the quality of the link.

Let us denote the average trip time for a particular destination *d* stored in the delay table at node *i* with $T_{i,d}$, and the current calculated trip by $T_{current}$, then:

$$\Delta p = \frac{1}{2} \times \left(\frac{T_{i,d}}{T_{current}} \right) \quad where \quad \alpha \leq \Delta p \leq \beta \tag{3.10}$$

where Δp is the reinforcement value that will be added to the pheromone table and it depends upon how good $T_{current}$ is. The value of Δp is restricted between α and β to prevent wild fluctuations in Δp (we have set the values of α and β to be 0.1 and 10 respectively). Since, AMIRA keeps up to date local link quality information, the trip time $T_{current}$ depicts the latest network condition in terms of traffic load and congestion.

Now, the pheromone values in the pheromone table corresponding to a particular destination *d* via next hop *i* can be updated as follows:

$$p(i,d) = \frac{p(i,d) + \Delta p}{1 + \Delta p} \tag{3.11}$$

Similarly, the other neighbors as next hops *j* for the same destination can be downgraded:

$$p(j,d) = \frac{p(j,d)}{1 + \Delta p} \qquad where \quad j \neq i \tag{3.12}$$

Note that Equation. 3.11 and Equation. 3.12 satisfy equation. 3.4.

3.3. Simulation Results

In this section, we discuss the experimental setups, motivation behind the setups, and results of our experiments. Our evaluations are divided into three parts. First, we study the stability of our smart ant-based routing algorithm (AMIRA) in terms of how quickly it manages to find new paths in dynamic networks by tuning the algorithm parameters in order to reach an optimized behavior. We also discuss the effect of smart ant generation rate on the overall performance of the mesh network by measuring normalized routing load (NRL) and packet delay. The second part of our evaluation consists of comparing AMIRA with some of the existing well-known routing protocols in wireless networks using different network topologies and traffic characteristics. Finally, the performance of AMIRA in mobile WMNs where some of the nodes are static and some of them are mobile is compared in the third part of our evaluation.

For the implementation of AMIRA, we have used the network simulator (ns-2) [51]. Our selection of ns-2 for implementation and simulation purposes is based on its popularity and its acceptance in the academic community. We have implemented our AMIRA routing agent as smart ants which are small fixed size control packets being sent periodically for finding paths and building routing and pheromone tables along the way in the network. In order to measure the local link qualities by the local heuristic table of AMIRA, a list of one-hop neighboring nodes and their corresponding link qualities (i.e., in the form of link delays) are maintained. The smart ant forwarding process is governed by our pseudo-random node transition rule according to which these smart ants are forwarded to next hop at each intermediate node with the highest pheromone values if a random variable is above a given threshold otherwise proportional strategy in the form of probability is adopted to decide the next hop selection. This enables the exploration of available paths in the network. Data packets are routed using this same transition rule which helps in spreading the packets across multiple links. Forwarding mostly selects the links with highest pheromone values because the pheromone values in the pheromone table are directly affected by the link quality that changes with the traffic load on the network thereby reducing interference.

Since, most of the traffic in a real WMN is either to or from a wired network [1, 52] (i.e., through Internet gateway points), in our simulations, flows are destined to one to four gateway nodes. The common configuration parameters for all simulation studies in this section are listed in Table 3.I. Although, some may argue that some of these values are low (e.g., link bandwidth), our purpose is to provide a comparative evaluation (relative results) and thus we believe that these values will not strongly influence our results. Each of our depicted data points in the results is an average over enough simulation runs to claim a 95% confidence that the relative error of them is less than 5%. Any topology related change in the simulation will be mentioned in the appropriate subsection.

The goals of our simulation are to evaluate the effectiveness of our proposed smart ant-based routing algorithm - AMIRA in finding high-throughput paths with less interference and to quantify the benefit of smart ants over existing ant based routing algorithms. We have compared AMIRA with two ant-based routing algorithms namely SARA [21] and DAR [22]. Our selection of these two protocols for comparative evaluation is because these are the most recent routing algorithms proposed that are based on ants to build routing paths. Both SARA and DAR are designed for MANETs; but the fact that AMIRA is the first ant-based routing approach to deal with WMNs, to the best of our knowledge, forces us to choose these two protocols. However, this does not imply that MANETs and WMNs have similar characteristics and therefore, a need to design a new routing algorithm for WMNs specially tailored for its unique characteristics was required as mentioned in section 3.1.

Table 3.1 Simulation Configurations

Simulation area	1000 x 1000 m^2
Transmission range	250 m
Propagation model	Two-ray ground
MAC protocol	802.11 CSMA (RTS/CTS disabled)
Link bandwidth	2 Mbps
Traffic type	CBR (UDP)
Packet size	512 bytes
Number of nodes	15 or 100
Buffer size	20 packets

3.3.1 Tuning AMIRA Parameters

We study two configuration parameters of AMIRA routing algorithm. The first is the parameter p_0 which is used in our pseudo-random node transition rule and it governs the relative behavior of smart ant forwarding. The parameter p_0 determines the probability of choosing a next hop for path discovery/data forwarding with maximum pheromone value or selecting the next hop based upon probability distributions using Equation 3.5. If we set the value of p_0 to be very high, AMIRA would select links with high pheromone values more often. Similarly, setting the value too low would spread the data traffic around the links and therefore, would fall back to the classical AntNet algorithm. So, there must exist a balance for this trade-off. Figure 3.2 demonstrates the performance of AMIRA in terms of average network throughput as a function of p_0 under varying network traffic. We have used a 15-node grid mesh topology shown in Figure 3.1 as the underlying network. Notice that the network throughput is almost the same for all values of p_0 when there is little traffic on the network. The p_0 value starts affecting the algorithm performance when more packets are pumped in the network (increasing network traffic). At that point, the higher the p_0 value, the better the throughput becomes; this is because, most of the time, links of good quality (highest pheromone) are selected for data forwarding. Since the network is still not saturated, the interference estimation rule perfectly captures the interference thereby resulting in increasing the pheromone values of links with less interference. However, when the network traffic is high, the gap between the AMIRA throughputs for different p_0 values becomes small. We believe the reason behind this pattern is due to the low p_0 value (i.e. 0.2, 0.5) spreading the data over multiple links with more chances of random next hop selection. However, for higher values of p_0, AMIRA still achieves better throughput than when a lower value is set. Therefore, in the rest of our simulations, we will select the value p_0=0.8.

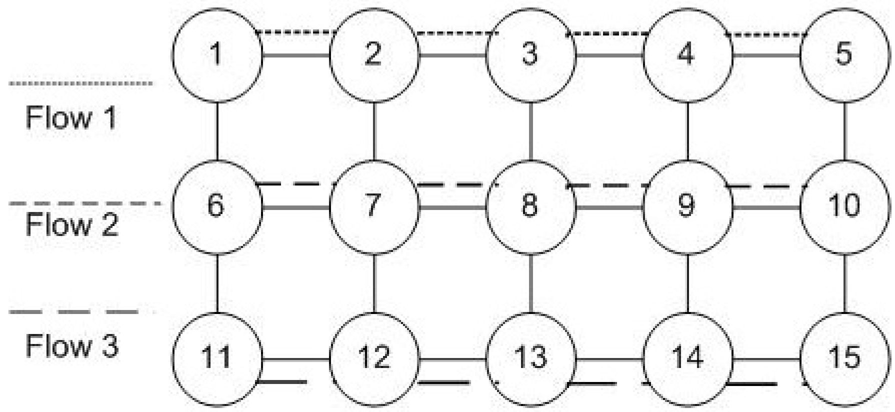


Figure 3.1 WMN grid topology

The second parameter we intend to tune is the smart ant generation rate which is related to AMIRA's stability. Recall, that the stability of any routing algorithm depends upon how quickly it adapts itself to the changing dynamics of the network. We define the time it takes for a routing algorithm to learn the best routing policies as the learning time (convergence time). We demonstrate the stability of our algorithm by measuring *path latency* in mesh networks under dynamic load situations when traffic characteristics change or load on the network is increased. In order to effectively demonstrate the quality of the learned policy, we have used the average packet delay as the evaluation metric.

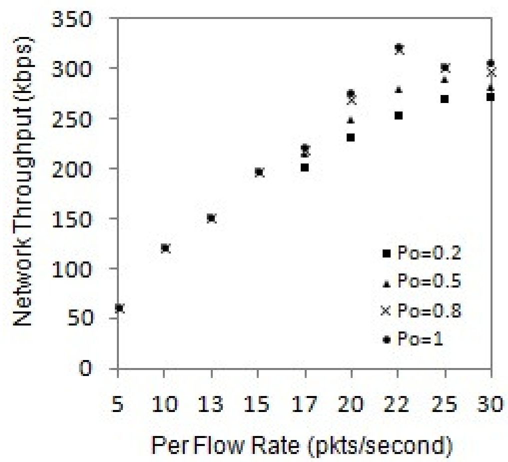


Figure 3.2 AMIRA throughput as a function of p_o

Figure 3.3 shows the learning times of the AMIRA algorithm under different smart ant generation rates. It relates the protocol's stability in the adaptation process for different ant rates during the simulation run time. In the beginning, one flow is generated to introduce a light load on the network. Then at $t = 10s$, 3 more flows are initiated in order to increase the traffic in the network. At $t = 20s$, these 3 flows are stopped in order to bring the network back to its normal state. It can be seen in Figure 3.3 that AMIRA adapts to this increase in network load at t=10s, by switching to a new path between the source and destination node with the help of its link and path estimation modules defined in the interference estimation rule. However, at $t = 20s$, the algorithm starts converging back to the previous best path that it had discovered. We believe that the convergence of AMIRA to the best available path is related to how fast smart ants are being generated by each node i.e. the ant generation rate. Figure 3.3 shows that a more frequent ant generation leads to less the time for the algorithm to find a better path. In other words, the time window for finding this path decreases with increasing number of smart ants in the network. However, these values of various ant rate is highly dependent upon the traffic characteristics as well as the network topology and therefore should not be taken as general criteria for applying AMIRA in WMNs.

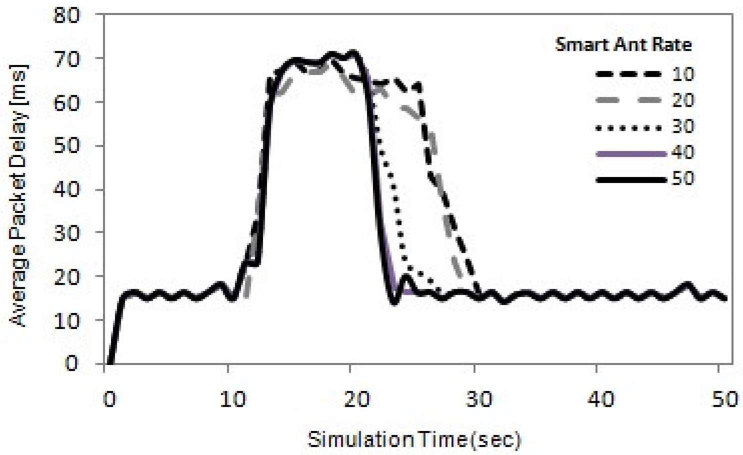


Figure 3.3 AMIRA learning time with various smart ant rates

Although the learning time of AMIRA is inversely proportional to the smart ant generation rate, a higher ant rate means increase control traffic in the network which in turn would limit the overall useful network capacity. Therefore there exists a tradeoff between the amount of routing traffic generated and the learning time. We investigate this tradeoff in AMIRA by using normalized routing load (NRL) as the evaluation metric which is the ratio of the control traffic in the network to the total amount of packets that were actually pumped into the network. Figure 3.4 shows the NRL for light and high network loads for varying smart ants' rates. One interesting observation is that the network load has almost negligible effect on the routing load when there are too few smart ants in the network i.e. the ant generation rate is small. Even for higher ant rates, one can observe that ant generation rates have a marginal effect on the NRL of our routing protocol under the conditions when the network is below saturated. On the other hand, there is a significant difference among the NRL values on the network when the network is overloaded. One reason behind this increase in NRL is due to the increase in packet collisions and the dropping of packets from the node queues when the network becomes overloaded, which in turn will generate more smart ants to discover new paths. Although AMIRA has a higher NRL as compared to the overhead of other routing protocols (i.e., distance vector or link state as shown in [16]), we argue that this is compensated by the efficient performance it provides in routing the traffic by finding less interference paths in WMNs (as shown later).

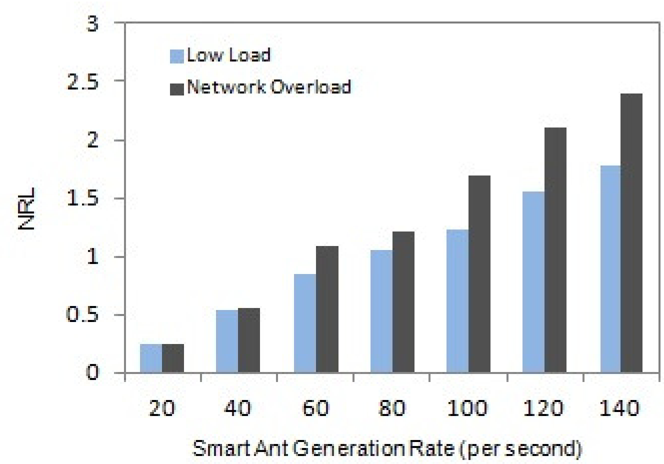


Figure 3.4 Effect of smart ant generation on protocol overhead

Figure 3.5 shows the impact on the network performance in terms of end-to-end delay which is depicted as a function of ant generation rates for different network loads. Notice, that the packet delay remains almost the same for increasing number of smart ants in the network.. This is because most of the traffic stays on the best path which keeps the packet delay almost constant among different network load conditions. It is because of this reason, for the rest of our simulation, we select the ant generation rate to 40 ants per second.

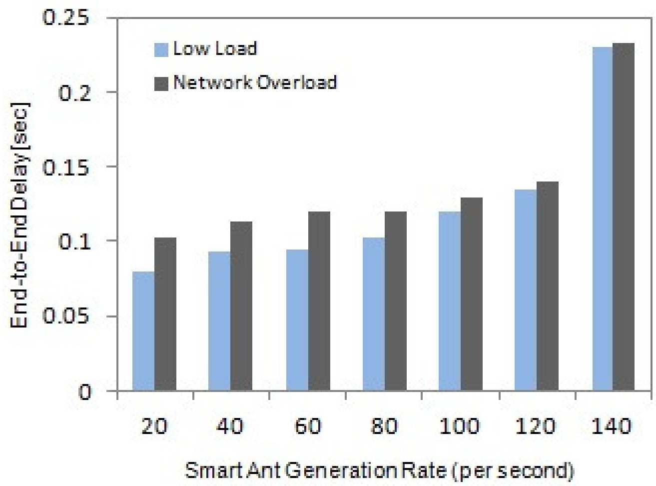


Figure 3.5 Effect of smart ant generation on AMIRA end-to-end delay

3.3.2 Mesh Network with Static Nodes

We now evaluate the performance of AMIRA in a mesh network scenario when all nodes are static. We divide our simulations into two sets of experiments based on the topology, namely grid topology simulations and random topology simulations.

3.3.2.1 Grid Topology

In the first set of experiments we use a 15-node grid network as shown in Figure 3.1 with nodes placed over a 1000m x 1000m area, arranged in a 3 x 5 row-column grid fashion. Three constant-bit-rate flows of UDP packets are deployed on the grid with traffic destined to each end of the grid's rows. Figures 3.6(a,b) show total network throughput and average end-to-end packet delay, respectively. It can be observed that AMIRA outperforms the other two protocols in terms of both the average end-to-end delay and network throughput. In Figure 3.6 (a), at low network load condition all algorithms perform similarly, but as the traffic on the network increases throughput of SARA starts to degrade, but DAR and AMIRA still provide better throughput; when the network reaches the saturation point DAR starts to decline as it cannot deal with interference among the flows. AMIRA maintains better throughput because of the link quality information that it stores in order to re-route the packets in case of increased interference among the flows. AMIRA's pseudo-random node transition rule for data forwarding also helps distribute the data traffic in the network thus resulting in automatic load balancing. Similarly, in Figure 3.6 (b), notice that under low load conditions, all the protocols have almost the same average end-to-end packet delay. However, as the network becomes saturated, packets start to drop due to flow interference. AMIRA outperforms the two other algorithms.

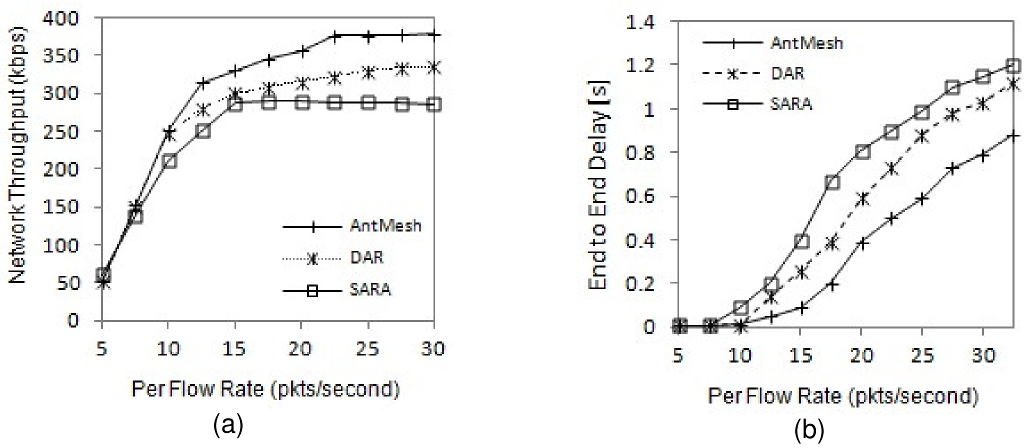
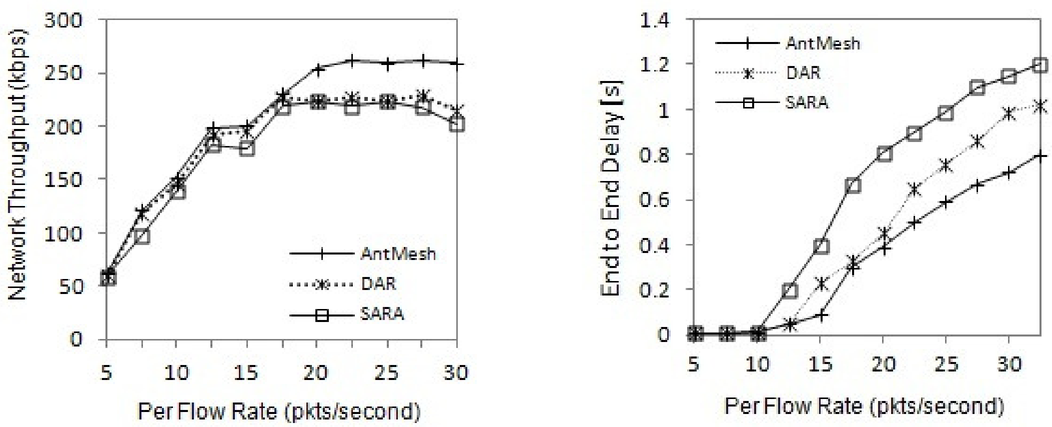


Figure 3.6 Grid Topology (a) total network throughput (b) end-to-end delay

3.3.2.2 Random Topology

In the second set of simulations we randomly generate networks of size 1000m x 1000m each having 100 nodes, and 10 flows, destined to 3 gateway nodes. (Node locations are uniform random I.I.D. over the area.) Figures 3.7 (a,b) show total network throughput and average end-to-end packet delay respectively. AMIRA once again outperforms all the other routing algorithms because of its interference-avoidance strategy, both in terms of throughput and average end-to-end packet delay. In Figure 3.7 (b): even though, the interference avoidance feature of our smart ant-based routing algorithm might force a packet to take longer path, the fact that the path chosen would be less prone to packet losses results in lower end-to-end delay. Similarly, to the previous case, the throughput of all investigated protocols were comparable until the saturation point from which on AMIRA outperforms the others as shown in Figure 3.7 (a). We believe that the better performance of AMIRA against SARA and DAR particularly in situations of more network traffic is because of its accurate estimation of link interferences (inter/intra flow) which is specifically designed in consideration for unique characteristics of WMNs.

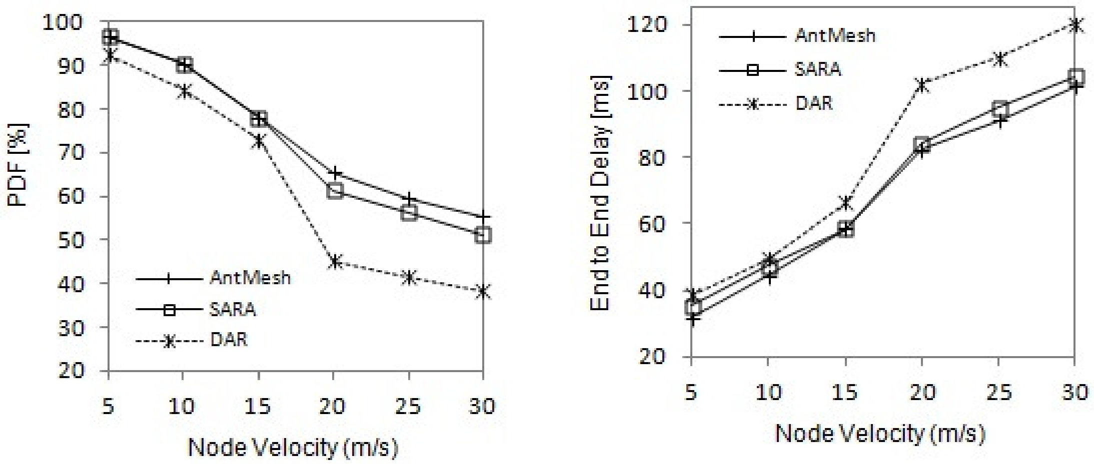


(a) (b)

Figure 3.7 Random Topology (a) Total network throughput (b) End-to-End delay

3.3.3 Mesh Network with Mobile Nodes

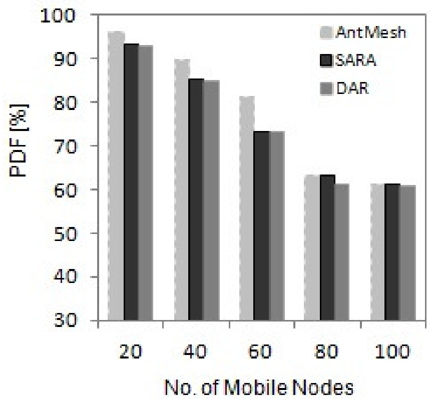
In this part of our evaluation, we test the performance of AMIRA in a mobile WMN scenario with static backbone and mobile client nodes. We consider a network of 100 nodes randomly distributed in a 500m x 500m region. All mobile nodes in each simulation run (of 60s) are configured with the same speed and the random waypoint model [53]. The data traffic consists of 6 CBR flows of random source destination pair. In order to evaluate the effectiveness of our smart ant-based routing algorithm, we have use packet delivery ratio and average end-to-end delay as evaluation metrics. Figures 3.8 (a,b) show the delivery ratio and average end-to-end delay as a function of different node speeds ranging from 0 m/s to 30 m/s. As it can be easily seen from the delivery ratio graph, AMIRA outperforms SARA and DAR clearly and this performance gap between the schemes is more evident when the nodes are moving at a faster rate. The performance difference for average packet delay in Figure 3.8 (b) is less but again it increases for higher node speeds. Notice the dip in the delivery ratio for all the algorithms in Figure 3.8 (a), we believe that this is because when nodes are moving at high speeds, it is possible that some nodes get out of reach from the rest of the network (network graph becomes separated) and therefore packets cannot be delivered to them resulting in possible low delivery ratio. Similar arguments can be made for the sharp rise in packet delay of the algorithms in Figure 3.8 (b). Furthermore, in the mobility model (RWP) that we have used in our experiments, nodes tend to make sudden and uncorrelated changes in their movement direction at the pause points; this is captured in a timely manner by our adaptive smart ant-based routing algorithm resulting in improved performance compared to the SARA and DAR protocols which are reactive in nature. Another interesting result is that SARA performs better than DAR and very close to AMIRA in both the graphs. This is due to the design objective of SARA which is targeted for networks having highly mobile nodes with critical connectivity. However, it still under performs than AMIRA because of its inability in effectively measuring the interference.



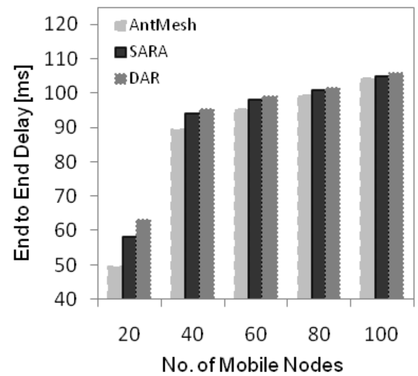
(a) (b)

Figure 3.8 AMIRA as a function of node speed (a) PDF (b) Packet delay

In order to simulate the true nature of a WMN with mobile nodes, we have collected another set of results by gradually increasing the number of mobile nodes in the network among the fixed nodes in the networks. The results are collected with 20% to 100% of the total nodes in the network given a random movement. We have selected the speed as a simulation factor set to 10 m/s. Figures 3.9 (a,b) show the same performance measures for all algorithms as a function of number of mobile nodes in the network. The graphs point to AMIRA exhibiting a better performance in terms of higher delivery ratio and less packet delay particularly when most nodes in the network are mobile. The stochastic forwarding of smart ants for path exploration and timely capture of inter and intra flow interference by our interference estimation rule of smart ants pay off more when the number of mobile nodes is increased. The reason for the comparatively poor performance of SARA and DAR lies in the fact that they use minimum hop count metric for path construction, and thus they fail to cope with the sudden and dynamic changes in the network. These graph point out as well a better performance of SARA compared to DAR because of the previously mentioned design objective of SARA.



(a)



(b)

Figure 3.9 (a) Packet delivery ratio and (b) Packet delay vs. percentage of mobile nodes

3.4. Summary

In this chapter, we addressed routing for interference avoidance in infrastructure mesh networks. We proposed AMIRA which is an interference-aware routing protocol designed to improve load balancing by avoiding inter- and intra-flow interference in a typical mesh backbone network. AMIRA has its roots in Ant Colony Optimization which is a meta-heuristic approach for stochastically solving an optimization problem. We added a local heuristic technique to avoid interference within and among the flows. In AMIRA, each node uses MAC level information to measure the link qualities which helps in selecting reduced interference paths and thus resulting in improved load balancing in addition to the auto load balancing feature. Our simulation results demonstrated that under network congestion, AMIRA achieves better throughput and lower end-to-end delay compared to other existing routing protocols because of its interference aware technique and stochastic data forwarding.

CHAPTER 4

SMART ANTS FOR EFFICIENT ROUTING IN MULTI-RADIO MULTI-CHANNEL WIRELESS MESH NETWORKS

The availability of off-the-shelf, low cost, commodity networking hardware enables the incorporation of multiple radio interfaces operating in different radio channels in to a single mesh router; thus, forming a multi-radio mesh network. This can lead to a potential large improvement in the capacity of mesh networks compared to single-radio mesh networks. In this chapter, we address the problem of packet routing for efficient data forwarding in WMNs with the help of smart ants acting as intelligent agents. The aim is to study the use of such biology inspired agents to effectively route the packets in multi-radio multi-channel wireless mesh networks (MRMC-WMNs). In particular, we propose AntMesh, a distributed interference-aware data forwarding architecture which enables the use of smart ants to probabilistically and concurrently perform the routing and data forwarding in order to provide a stochastic solution to the dynamic network routing problem. AntMesh belongs to the class of routing algorithms inspired by the behavior of real ants which are known to find a shortest path between their nest and a food source. In addition, we also propose a novel routing metric called Ant Routing Metric (ARM) which is designed to effectively utilize the space/channel diversity typically common in infrastructure WMNs. One interesting result of our investigation is that AntMesh has the capability to discover high throughput paths with less inter- and intra-flow interference when conventional wireless network routing protocols and metrics fail to do so. We implemented the AntMesh metric in ns-2 and will show extensive simulation evaluation results. We have tested the performance of our proposed routing architecture under various network scenarios, i.e., in fixed and mobile node WMNs. The results obtained show AntMesh's advantages that make it a valuable candidate to operate in MRMC mesh networks.

4.1. Problem Addressed

Most of the traffic in a mesh network usually flows between regular nodes and a few Internet gateways (i.e., rarely end-to-end between regular nodes). This can result in an uneven loading of links and can cause certain paths to be saturated. Similarly, the existence of inter-flow interference among the nodes and intra-flow interference within a transmission path may affect traffic loads on mesh nodes in a multi-radio WMN. The main objective of any mesh routing protocol is thus to effectively distribute the traffic by selecting channel diverse paths with less inter- and intra-flow interference.

4.2. Contributions

For the above described WMN packet routing problem, we have developed an efficient data forwarding architecture (AntMesh) based on smart ants acting as intelligent agents. The aim of this chapter is to study the use of such biologically inspired agents (smart ants) to effectively route the packets in MRMC WMNs. To achieve this, we propose a novel routing metric called Ant Routing Metric (ARM) which captures inter- and intra-flow interference by selecting reduced interference paths with increased channel diversity; thus, intuitively resulting in improved network performance. We will support this intuition by a thorough performance evaluation. We implemented the proposed ARM metric of AntMesh in ns-2 and carried out detailed and extensive simulation studies which we will use to compare our proposed metric to competing approaches. The salient features of our work that set our approach apart from the existing routing protocols and metrics in WMNs are listed as follows:

- We propose a distributed routing mechanism which enables the use of smart ants to probabilistically and concurrently perform the routing and data forwarding in order to solve a dynamic network routing problem.
- We design a novel routing metric called Ant Routing Metric (ARM), and formally define the properties and conditions necessary for the design of such ant based metric for WMNs. The ARM helps to effectively utilize the space/channel diversity typically common in infrastructure WMNs.
- We will show that ARM has the capability to discover high throughput paths with less inter- and intra-flow interferences when conventional wireless network routing protocols and metrics fail to do so.
- To the best of our knowledge, our proposed forwarding architecture is among the first works that investigate the use of smart ants in WMNs while demonstrating a performance advantage.

The rest of the chapter is organized as follows. We start with presenting the architecture of our proposed smart ant routing algorithm in Section 3 together with the introduction of our custom designed Ant Routing Metric (ARM). We describe the implementation details of AntMesh in Section 4 followed by simulation results to evaluate its performance, comparing it to existing routing schemes in Section 5. Finally, we conclude the chapter in Section 6.

4.3. AntMesh Forwarding Architecture

In this section, we describe the details of AntMesh, a distributed routing algorithm which incorporates smart ants to find high-throughput paths (i.e., less interference paths) and improved load balancing specifically designed for WMNs. We will also formally introduce the Ant

Routing Metric (ARM) which helps to effectively handle the space/channel diversity typically common in WMNs.

4.3.1 Data Structures

The packet structure of a smart ant in our AntMesh forwarding architecture is shown in Table 4.I. The *type* field captures the type of ant that is being generated. *ID* is a uniquely generated smart ant identifier; for FSA it is generated by the source node and it is copied into BSA upon reaching the destination. The *Src* and *Dst* fields hold the addresses of the source node which generated the FSA and destination node for which a route is desired respectively; the addresses are swapped when an ant becomes a BSA. *Hop* holds the number of hops that a smart ant has covered so far. The *CH* field lists the channel information of the last two hops carried by the BSA and is used to measure the intra-flow interference in a mesh network. The *Trip Time* field reflects the accumulated trip time calculated so far by the BSA from the next hop node to the destination. Similarly, *Q.Sizes* represents the queue lengths of the last hop's neighboring nodes from where the smart ant has arrived. This field helps in estimating inter-flow interference of a link in a mesh network.

Table 4.1 Smart Ant Structure

Type	ID	Src	Dst	Hop	CH	Trip Time	Q. Sizes
FSA/BSA/Hello	1	S	D	2	2-hop list	0.1ms	Nbr. Q. size list

In AntMesh, every node maintains three types of data structures explained as below:

*Pheromone table (Probabilistic routing table).*This data structure stores the fitness of choosing a specific neighbor as next hop to reach a particular destination in the form of a probability. In other words, it contains pheromone trail information for routing from current node to destination node via the next hop. Thus, the pheromone table at any particular node *k* contains m_k rows where $m_k = |N_k|$ and N_k is the set of neighboring nodes to node k. Each row contains N columns, where N is the total number of possible destinations in the network (total number of nodes, or population). So an entry P_{id} is the probability of sending a packet to destination d via link i and thus following relation holds for every column in the pheromone table at node k:

$$\sum_{i \epsilon N_k} P_{id} = 1 \ \forall d \epsilon [1 \dots N] \qquad (4.1)$$

Delay table. The second data structure that is maintained by each node is the delay table that stores the average trip time to each destination in the network from the current node (thus this table will have m entries, one for each possible destination in the network). The value stored is an average calculated from the delay value carried by the last W number of smart ants received.

Link estimation table. The third table maintained by AntMesh is the local estimation table which contains the quality/strength of the outgoing links of that particular node to its neighbors. AntMesh uses hello smart ants (HSA) to measure these local link statistics in terms of link level packet transmission delays.

A more detailed explanation on how these data structures are maintained will be provided later in this section.

4.3.2 Route Discovery

Smart ants in the form of control packets are generated at regular intervals from each node towards possible destinations in the network. Indeed, three types of ants are generated: forward smart ants (FSA) which travel from source to destination to discover paths, backward smart ants (BSA) traveling from the destination to the source to update the routing tables and hello smart ants (HSA) which collect the local link quality information to populate link estimation table. In AntMesh, both FSA and BSA use high priority queues so that FSAs do not need to carry their per-hop experienced trip times, rather BSAs will estimate their trip time using the Ant Routing Metric (with the help of link estimation table being maintained at each node).

AntMesh pheromone values to neighbors differently on initialization compared to other existing ant-based algorithms [14, 16, 21, 22]. The latter assume a uniform probability distribution when nodes are initialized. This means giving equal probability of selecting a specific neighbor as a next hop to reach a destination to all the neighbors of a node. However, assigning higher pheromone values to the neighboring nodes, which are also destination nodes for some other nodes in the network, results in faster initial network convergence while reducing network resource usage. In AntMesh, every node assigns the initial probabilities to all the destinations in the network as follows. For a destination node *d* that is not a neighbor node, the initial probability assignment (to all the neighbors N_k of node *k*) is the same: $1/N_k$. However, for a destination node *d* which is also a neighbor node the initial probability in the routing table is:

$$P_{d,t} = \frac{N_k+1}{2N_k} where d = t \tag{4.2}$$

The probabilities for the remaining neighboring nodes are:

$$P_{d,i} = \frac{1}{2N_k} where i = N_k \tag{4.3}$$

Similarly, the selection of the next hop node by AntMesh is performed by the same principle as discussed in our previously proposed AMIRA algorithm (Chapter 3). An FSA *v* at an intermediate node *k* chooses the next hop *u* to reach a particular destination *d* according to Equation. 4.4 and Equation. 4.5.

$$u = \begin{cases} \arg\max_{u \in N_k} \{\tau_v(d,u)\} \; if p \leq p_0 \\ P_v(d,u) otherwise \end{cases} \tag{4.4}$$

$$P_v(d,u) = \frac{\tau_v(d,u)}{\sum_{i \epsilon N_k} \tau_v(d,i)} \quad (4.5)$$

where p is a uniform random number between [0, 1] and p_0 is a constant in the range [0, 1]. Similarly $\tau_v[d,u]$ is the pheromone intensity on the link connecting next hop u with k to reach destination d. Equation. 4.4 indicates that if $p \leq p_0$, the node with the maximum pheromone value among neighbors will be selected, otherwise, a proportional selection will be made based on the distribution $P_v(d,u)$ over the neighbors (as shown in Equation. 4.5). Note, that the parameter p_0 can be seen as the relative importance of exploitation versus exploration.

4.3.3 Designing the Ant Routing Metric (ARM)

Previous research has addressed the design of efficient routing metrics in WMNs; we have provided an overview of some of these metrics in Section 2.1.3. However, routing metrics such as MIC [7], WCETT [6] and iAware [8] suffer from certain limitations when trying to capture interference in the network. For example, WCETT does not explicitly consider inter-flow interference which might result in directing traffic to an already saturated node thereby affecting overall network utilization [54]. MIC tries to reduce the interflow interference by counting the total number of neighboring nodes on both ends of a wireless link, however, this approach can prove to be over-conservative; this is due to assuming that just because a node is a neighbor, it will be definitely interfering (which is untrue if it has no data to send) [8]. iAware uses a physical model to capture the interference in WMNs; however, scalability issues may arise due to its complexity, and thus its application may be limited to small mesh networks [55]. These limitations to the existing routing metrics have given us an impetus to design a novel routing metric that is interference-aware and can capture the characteristics of ACO approaches in terms of robustness, efficiency, self-organization and automatic load balancing in WMNs. In order to understand ARM, we first have to look at what properties a routing metric should have in order to be suitable for creating network paths in an ant based routing algorithm.

4.3.3.1 Properties of Ant Routing Metric (ARM)

We believe that a routing metric based on smart ants designed for WMNs should have the following properties:

- The metric should capture the two types of interferences that inherently exist in mesh networks (namely inter- and intra-flow interference).
- The metric should be able to evaluate the load on nodes in order to properly qualify the outgoing links. This would help in *detouring* packets to new route when needed and hence would result in a more load balanced network.
- Since network nodes in a WMN can be equipped with multiple radios, a metric should be able to discover *channel-diverse paths* in order to reduce the interference and thus effectively improve the overall network throughput.

The Ant Routing Metric was designed to exhibit the above mentioned desirable properties; ARM is a significant component of the AntMesh interference estimation rule. This rule is designed to fully capture the characteristics of WMNs including the types of interferences and to incorporate the desired properties of ARM in AntMesh forwarding architecture. ARM consists of two modules, i.e., the link estimation module (LEM) and the path estimation module (PEM)as shown in Figure 4.1. These two modules help ants to accurately measure the inter- and intra-flow interferences.

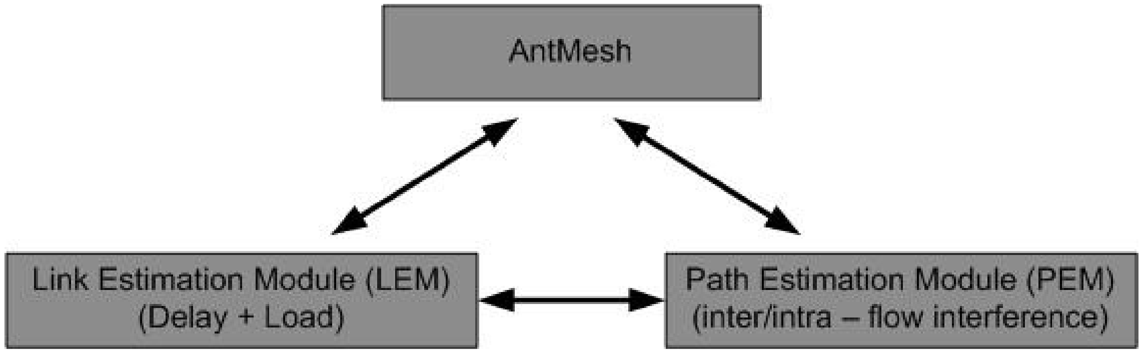


Figure 4.1 The AntMesh Architecture showing the two ARM modules

4.3.3.2 Link Estimation Module (LEM)

The link estimation module of AntMesh determines the quality/cost of a wireless link in terms of the average transmission time (delay), i.e., the time the MAC layer needs to transmit a packet on a particular outgoing link. We define this *link transmission delay* as the time from when a packet starts to be serviced by the MAC layer to the instant that it is successfully transmitted (thus including the time required by any or all retransmissions). Let T_i denote the transmission delay over link *i,* and N_{tx} denote the number of transmissions including retransmissions needed to successfully receive a packet. (In most practical cases, $N_{tx} \cong 1$, as almost all the current wireless devices are equipped with multi-rate feature which automatically adjusts the link rate according to the link quality, resulting in successful transmission of around 90% packets at the first time - see [50]). L_{pkt} is the data packet size and R_s is the link speed. Then the link transmission delay can be defined as follows:

$$E[T_i] = N_{tx} \times \left(MAC_{oh} + \frac{L_{pkt}}{R_s}\right) \qquad (4.6)$$

where MAC_{oh} is the standard packet sequence of sending a data packet. For example in the IEEE 802.11 DCF, MAC_{oh} is calculated as follows:

$$MAC_{oh} = T_{rts} + T_{cts} + 3T_{sifs} + T_{difs} + T_{ack} \quad (4.7)$$

T_{rts}, T_{cts} , T_{ack} are the times required for the transmissions of RTS, CTS, and ACK frames respectively; T_{sifs} and T_{difs} are the inter-frame spaces: SIFS and DIFS.

Since the 802.11 DCF MAC has a contention based nature, the link delay time needs to be modeled as a random variable. Therefore, in order to mimic a real packet transmission delay and to avoid any wide fluctuations in the measurement due to traffic bursts, a smoothing window approach is used over a time interval to get the average value of the link transmission delay. If T_i^k is the current delay in sending one packet on a link then we calculate the average link transmission delay using the following exponential running average:

$$Delay(T_i) = \beta T_i^k + (1 - \beta) T_i^{k-1} \quad (4.8)$$

whereβ is the *learning rate* that controls how much relevance is given to old samples compared to new samples.

However, $E[T_i]$ of Equation. 4.6 is just the MAC layer transmission delay which does not capture the traffic load, i.e., it does not encapsulate the queuing delay, which depends on the number of packets waiting in the buffer for transmission. Packets that are already in the queue must be served by the MAC layer before any newly arrived packet is served. Therefore, the link estimation module (LEM) calculates the total transmission delay of a packet on a particular outgoing link as the packet transmission delay of the number of packets in the buffer plus the transmission delay of the newly arrived packet:

$$LQ_i = E[T_i] \times Q_k + E[T_i] \quad (4.9)$$

where Q_k is the queue size at node *k* and thus LQ_i captures the link quality of link *i*. Note that LQ_i includes the load on node *k* and is therefore commensurate to the total time it would take for a newly arrived packet at a node *k* to be transmitted to the next hop.

4.3.3.3 Path Estimation Module (PEM)

The path estimation module of AntMesh has been designed to cater for the remaining desirable properties of smart ants, i.e., it will help discover channel diverse paths and capture inter- and intra-flow interference.

Recall, that after the completion of the path discovery an FSA, a backward smart ant (BSA) is created and is sent back to the source to update the data structures maintained by each node. During the BSA's travel to the source, each node that encounters the BSA updates its delay table data structure. They do so by estimating the smart ant's trip time from the current node to the destination node; this trip time is estimated as the sum of the average transmission time it takes to send a data packet from the current node to the next hop node (from where the smart ant has arrived) and the estimated accumulated trip time encountered so far by the BSA. The local link transmission delay of the next hop node is provided by the link estimation module using the link estimation table.

Inter-flow Interference. Since, nodes transmitting on the same wireless channel compete for the shared medium, whenever a node is involved in a transmission; its neighboring nodes should not communicate at the same time with other nodes on the same channel. The PEM of AntMesh captures inter-flow interference at each node and incorporates it into the local link estimation table. Let $I_{(k)}$ denote the set of queue sizes of node *k's* interfering nodes; we define $IFLD_{k,i,c}$, "inter-flow link delay" as follows:

$$IFLD_{k,i,c} = LQ_i \times max\left[I_{(k)}\right] \quad (4.10)$$

where $IFLD_{k,i,c}$ is the link transmission delay of a particular link *i* transmitting on channel *c* when inter-flow interference is considered. The path estimation module (PEM) only uses the maximum queue size of the neighboring node among all the neighbors (because a node with a very short queue length can still be congested if its interfering nodes have a lot of packets to send out); This makes sense as the current activities of all neighboring nodes are important in order to find out if they are indeed contending or not. AntMesh uses hello smart ants (HSA) to capture these node statistics in terms of queue sizes among the neighboring nodes. Therefore a high-contention link that would result in increased link transmission delay is detected timely by PEM.

More precisely, the rationale behind using queue lengths of neighboring nodes when measuring contention on a link belonging to a particular node can be understood with the help of a simple example (as illustrated in Figure 4.2). Let us assume that each node is configured to use a single channel; let *S* be the source node and *D* the destination node. So there are two paths to reach to *D*: *S-A-D* and *S-C-D.* Node *S* calculates the inter-flow interference of its neighboring nodes *A* and *C* by taking into account their queue sizes on links *i* and *j* respectively. It is clear that neighboring node *A* has more neighbors than node *C*, which initially gives the impression that the contention on link *i* is going to be higher than on link *j*. However, the amount of interference really depends on the number of packets waiting to be served in the queues of those neighboring nodes. So the neighbors around link *j*(although less in number) have more packets waiting in the queue than the neighbors of link *i*. Therefore, the contention on link *i* is less than that at link *j*. This is what is captured by the PEM, which will eventually lead to improving the overall network performance.

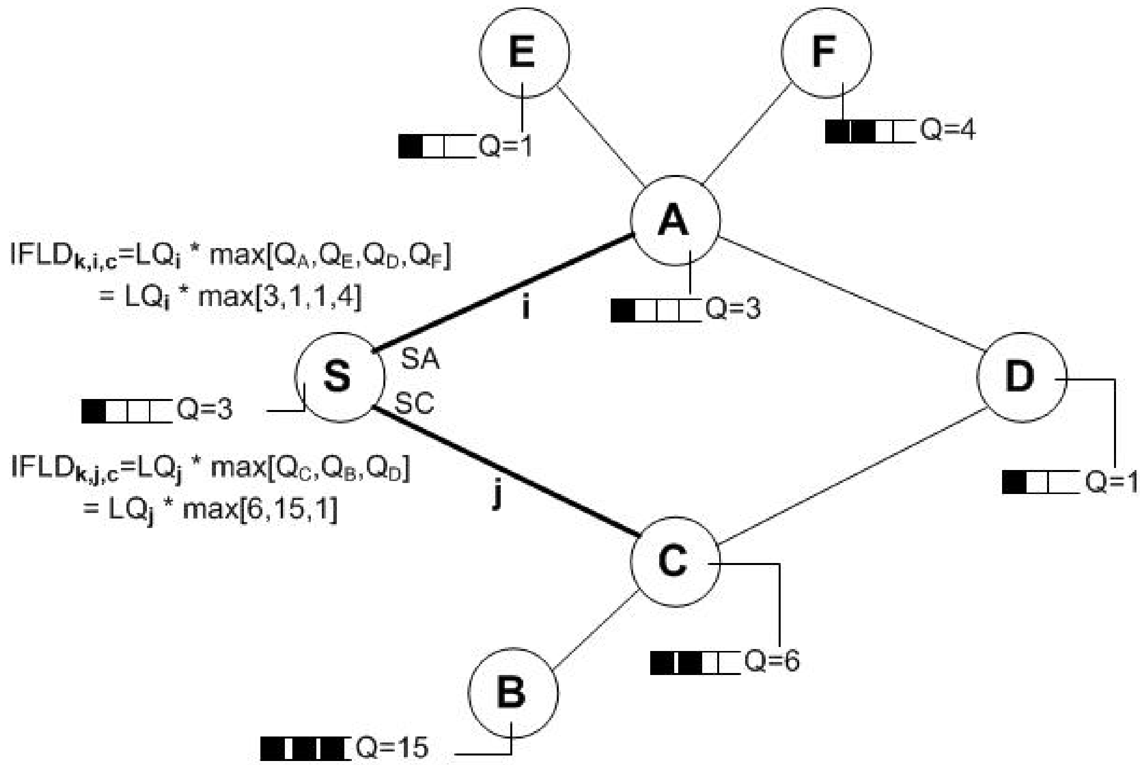


Figure 4.2 AntMesh PEM inter-flow interference capturing example

Intra-flow Interference. BSAs in AntMesh follow a deterministic path from destination to source; they keep a record of the channel properties of the links on which it has traversed together with the accumulated trip time. When the BSA is received by a node (intermediate, or destination) the PEM checks the smart ant for the links it has traversed; if the two latest hops used the same radio channel then it increases the total transmission time calculated in Equation 4.10 by a cost factor α::

$$ITT_{i,c} = IFLD_{k,i,c} \times \alpha \tag{4.11}$$

where $ITT_{i,c}$ (inter + intra-flow transmission time) is the total link transmission delay of a link i on channel c of a particular node. The cost factor α is calculated as:

$$\alpha = \frac{2Q_{next}L}{B_i} \tag{4.12}$$

where Q_{next} is the number of packets waiting in the queue at the next hop node, L is the packet size, and B_i is the bandwidth of the link. Since only one link can be active at any time, if both the consecutive links are on the same channel then the effective bandwidth will be B_i / 2 (i.e., the transmission time should be doubled for packets using the same channel on consecutive links). Similarly, no cost will be added to the total transmission delay when the two last-hop channels are different along the path, i.e., $\alpha=0.ITT_{i,c}$ will be added to the trip time that the BSA has accumulated so far from the next hop neighbor node j to the destination node d denoted as $Trip_{j,d}$. The BSA updates the trip time field of the delay table and hence the total trip time from a node i to destination node $d$is cumulative and is calculated as:

$$Trip_{i,d} = ITT_{i,c} + Trip_{j,d} \tag{4.13}$$

where $Trip_{j,d}$ is the trip time from next hop node j to the destination d and is taken from the *Trip Time* field of the BSA packet structure.

4.3.4 Route Maintenance

After the calculation of trip times $Trip_{i,d}$, BSAs trigger an update of the pheromone table values corresponding to a specific destination. However, there is a need for a mechanism to correctly integrate the transmission time measured by the AntMesh, into the pheromone table. Such mechanism will improve the routing decision process with respect to the quality of the link. Let us denote the average trip time for a particular destination d stored in the delay table at node i with $T_{i,d}$, (and the current calculated trip by $Trip_{i,d}$) then:

$$\Delta p = \frac{1}{2} \times \left(\frac{T_{i,d}}{Trip_{i,d}}\right) \tag{4.14}$$

where Δp is the reinforcement value that will be added to the pheromone table;Δp depends on how good $Trip_{i,d}$ is. Since, AntMesh keeps up to date local link quality information, the trip time $Trip_{i,d}$ captures the latest network condition in terms of traffic load and congestion. Now, the pheromone values in the pheromone table corresponding to a particular destination *d* via next hop *i* can be updated as follows:

$$P_{i,d} = \frac{P_{i,d} + \Delta p}{1 + \Delta p} \quad (4.15)$$

Similarly, the other neighbors as next hops *j* for the same destination can be downgraded:

$$P_{j,d} = \frac{P_{j,d}}{1 + \Delta p} where j \neq i \quad (4.16)$$

Note that Equation. 4.15 and Equation. 4.16 satisfy Equation. 4.1 (they sum up to 1).

4.3.5 Smart Ant Flow Diagram

The processes involved in AntMesh are fully distributed in nature and they are performed concurrently on each node of the WMN. Figure 4.3 shows a flow chart for smart ants in AntMesh as experienced at any particular node. We can observe that when a smart ant arrives at a node, the following sequence of steps are taken.

- First, AntMesh checks whether the smart ant that has just arrived is a duplicate (based on its ID) and if so then the packet gets dropped.
- If the incoming ant is an FSA, and the node is an intermediate node then a pseudo-random node transition rule is applied. If the FSA arrives to the destination node, then necessary information from the FSA is copied into BSA and the FSA is destroyed.
- If the incoming ant is an HSA, then the ARM metric is applied (as explained in Section 4.3.1.)

- If the incoming ant is a BSA, then path estimation module of the ARM metric is used to capture the inter-flow and intra-flow interference. Also, for each BSA received, the pheromone updating rule is applied to update corresponding pheromone table entries.

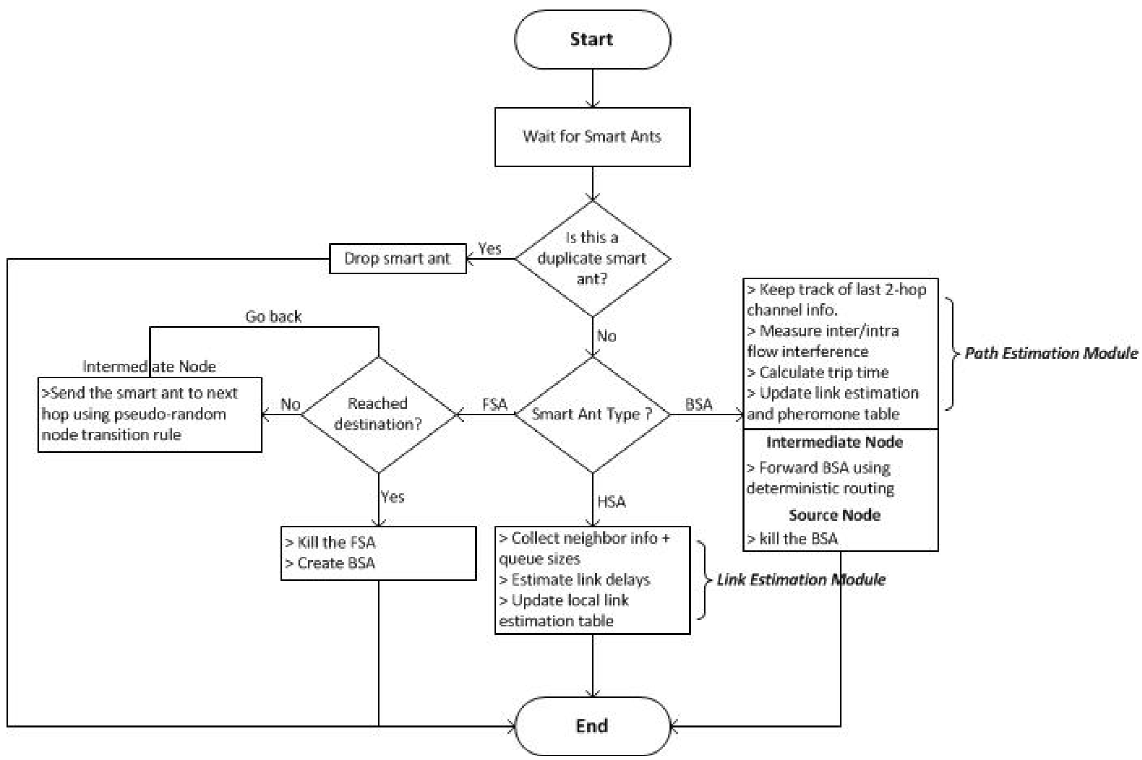


Figure 4.3 AntMesh flow-chart for smart ant processing

4.4. An Illustrative Example

Here we will use a simple wireless network (depicted in Figure 4.4) to illustrate how AntMesh and particularly its ARM work. We denote the source and destination nodes by *S* and *D* respectively; *Ch*i s the channel number a link is using. The number of packets waiting in the queues is depicted next to each node in the circles. Similarly, the link estimation and delay tables of each node are shown in the figure and the path computation is shown above each table to calculate the trip time. The path discovery process works as follows: in order to compute our path metric when the FSA traverses the network, we need the link metric for each link traversed and the channel id of the last two hops on which they were transmitted. The FSA carries the link metric and the channel ids of the links traversed. The sequence of steps performed by the AntMesh are as follows:.

- Node *S* initiates route discovery by generating an FSA, which reaches *D* while using the pheromone tables on each node by applying pseudo-random node transition rule.
- The FSA can reach to destination node *D* through two paths i.e. *S-A-F-D* (path 1) or *S-G-C-F-D* (path 2). Now, destination node *D* generates the BSA which follows the same path as the FSA; the BSA updates the pheromone table, the local link estimation table and the delay table (as explained before).
- When the BSA reaches node *F*, it records both inter- and intra-flow interferences of the link it has traversed by taking the maximum of the queue sizes of the neighboring nodes of *D* (max[Q_D]=1) and incorporating α. It then merges these measurements into the link quality of the link (link estimation module - maintained by link quality table). This step is the same for either paths.
- Now we have to distinguish between the two paths. For path-1 (*S-A-F-D*), as the backward smart ant travels to *A*, it carries with it the accumulated trip time of *F-D*. On reaching node *A*, the inter-flow interference is measured (max[Q_F,Q_C,Q_D,Q_E]=5) and then is incorporated into the local link quality of the link A-*F* stored at link quality table using Equation. 4.10. The last two hops' channels were different (AF = 2, FD = 4), therefore no intra-flow interference exists along these links $\alpha=0$.

  Similar calculations are carried for the BSA traversing on path-2 (*S-G-C-F-D),* i.e., when reaching node *C*, the trip time for a packet to reach node *D* from *C* is calculated. This trip time is then integrated into the pheromone table. The same step is repeated for node *G* on path-2 and eventually upon reaching source node *S*, the metric captures the inter- and intra-flow interferences of the links on which the smart ant has arrived (updating the data structures accordingly). Note, that for path-1, there is an intra-flow interference along links *SA* and *AF* as they both are on the same channel and this has been taken care of by the ARM metric using its path estimation module (by adding a factor $\alpha=0.5$).

The example points out that although path-1 contains less number of hops than path-2, the contention in terms of inter- and intra-flow interference along the path is higher. Therefore, the trip time to reach designation *D* on node *S* via the outgoing link *G* is likely less than through neighbor node *A*. This is shown in the delay table of node *S*, in Figure 4.4. Although this discussion concludes our example on how our smart ants capture interferences along routes, we will show shortly how this behavior will eventually result in an overall improved performance.

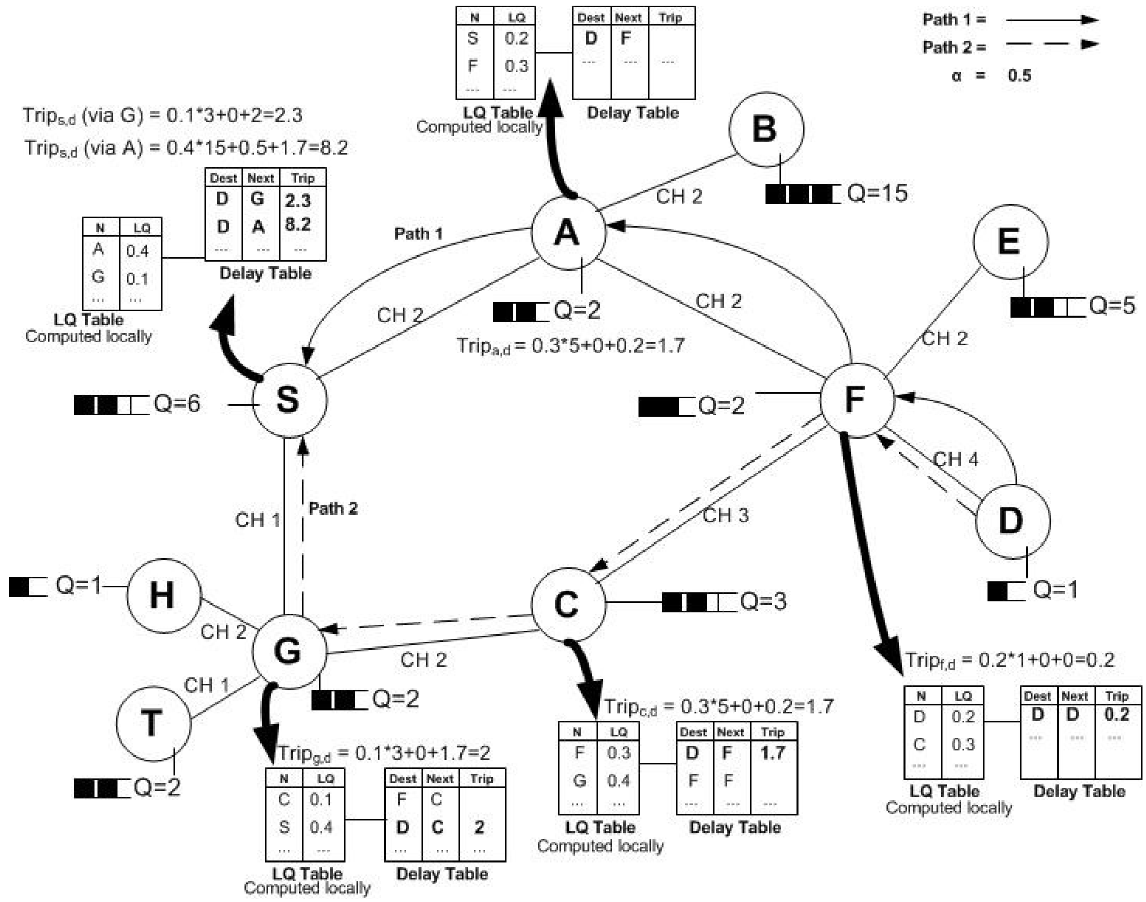


Figure 4.4 Illustrative Topology - An example illustrating the working of Smart Ants in AntMesh.

4.5. Experimental Setup

In this section we describe implementation details of AntMesh as incorporated into ns-2 [51]. We have selected ns-2 for implementation and simulation purposes based on its popularity and acceptance by the academic community. In order to measure link qualities by our link estimation module (LEM) of the ARM metric, a list of one-hop neighboring nodes and their corresponding link qualities (i.e., in the form of link delays) are maintained in a local heuristic table. This table is receives input from periodically broadcast HELLO messages. Upon receiving a HELLO message, each node responds with its queue size along with the list of its one-hop neighbors and their corresponding queue sizes (this is used to calculate the inter-flow interference by the PEM). This creates an inherent message overhead of AntMesh. However, periodical HELLO exchange has been widely adopted by most of the routing protocols reviewed in section 2.1.3 [5, 6, 56]. The smart ant forwarding process is probabilistic by nature meaning that ants are forwarded at each intermediate node according to the next hop probabilities in the pheromone table. This enables the exploration of available paths in the network. However, data packets are routed deterministically (based on the highest pheromone value in pheromone table) at each node due to the fact that the pheromone values in the pheromone table are directly affected by the link quality that changes with the traffic load on the network.

The actual code, i.e., the implementation of the AntNet algorithm for ns-2 is available at [57] (recall, that AntNet was originally designed for wired networks). In order to make it comparable with our proposed forwarding architecture for evaluation purposes, we modified the AntNet code for wireless scenarios. We call this modified AntNet implementation WAntNet (Wireless AntNet). Similarly, we have implemented some well-known link-quality metrics in WMNs such as MIC and WCETT in ns-2 for comparative studies.

Since, most of the traffic in a real WMN is either to or from a wired network [1, 52] (e.g., through Internet gateway points), in our simulations, flows are destined to one to four gateway nodes. The common configuration parameters for all simulation studies in this section are listed in Table 4.2. Although some may argue that some of these values do not represent current state of the art (e.g., link bandwidth is low), our purpose is to provide a comparative evaluation (relative results) and thus we believe that the absolute values of simulation parameters will not strongly influence our results. Any topology related change in the simulation will be mentioned in the appropriate subsection.

The goals of our simulations are to evaluate the effectiveness of AntMesh (particularly its ARM metric) in finding high-throughput paths(paths with less interference) and to quantify the benefit of AntMesh over existing mesh routing metrics and protocols.

Table 4.2 Simulation Configurations

Simulation area	1000 x 1000 m^2
Transmission range	250 m
Propagation model	Two-ray ground
MAC protocol	802.11 CSMA (RTS/CTS disabled)
Link bandwidth	2 Mbps
Traffic type	CBR (UDP)
Packet size	512 bytes
Number of nodes	15 or 100
Number of radios	≤ 3
Hello interval	1 second
Buffer size	20 packets

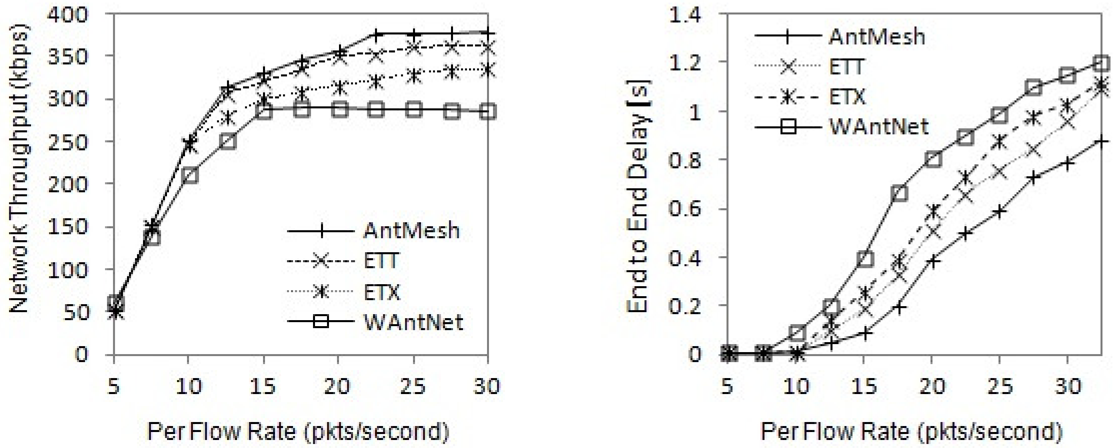
4.6. Simulation Results

4.6.1 Single Radio Single Channel WMNs

Here we show evaluation results on the performance of AntMesh in single radio, single channel WMN environments. Since, each radio is configured to the same channel, there is no reason to capture intra-flow interference along the flows. Therefore, the path estimation module of ARM is disabled in these sets of simulations. As in Chapter 3, we divide our simulations into two sets of experiments namely grid topology simulations and random topology simulations. We compare the ARM metric of AntMesh with two well-known mesh routing metrics: ETX [5] and ETT [6]. We also compare AntMesh with WAntNet.

4.6.1.1 Grid Topology

In the first set of experiments we use a regular grid network as shown in Figure 3.1 with nodes placed over a 1000m x 1000m area, arranged in a 3 x 5 row-column grid fashion. Three constant-bit-rate flows of UDP packets are deployed on the grid with traffic destined to nodes at each end of the grid's rows. Figures 4.5a) and b) show total network throughput and average end-to-end packet delay, respectively. We can observe that ARM outperforms the other metrics in terms of both the average end-to-end delay and network throughput. At low network load conditions (when the packet generation rate is low), all metrics perform similarly, but as the load increases the throughput of WAntNet and ETX start to degrade, but ETT and AntMesh still go strong. When the network reaches the saturation point ETT starts to decline as it cannot deal with interference among the flows. The ARM metric of AntMesh keeps providing better throughput because of the link quality information that it stores in order to re-route the packets in case of increased interference among the flows. AntMesh's stochastic data forwarding helps distribute (or load balance) the data traffic in the network. Similarly, looking at Figure4.5b), we can notice that under low load conditions, all protocols have similar average end-to-end packet delays. However, as the network becomes saturated, packets start to drop due to flow interference; In these situations, as expected, AntMesh/ARM outperforms the three other metrics.

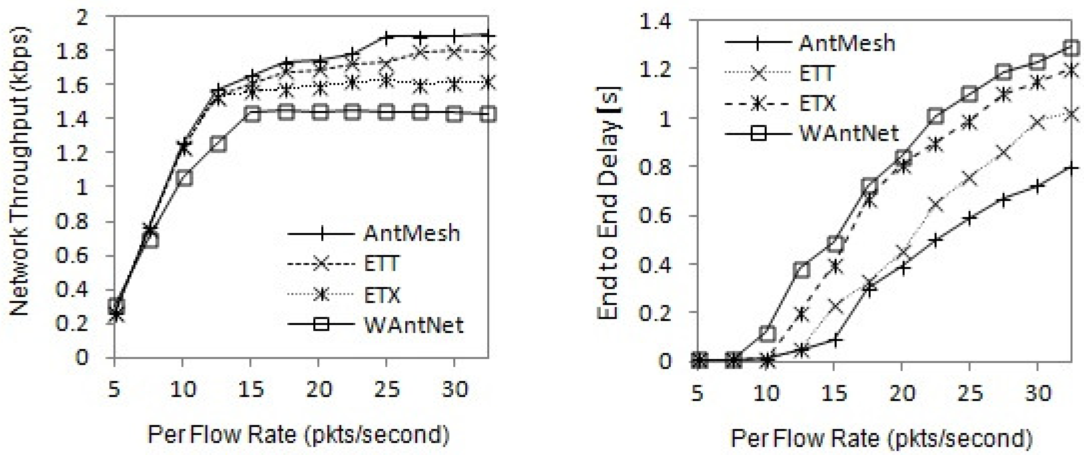


(a) (b)

Figure 4.5 Grid Topology (a) Network throughput (b) End-to-End delay

4.6.1.2 Random Topology

In the second set of simulations we randomly generate networks of size 1000m x 1000m each having 100 nodes, and 10 flows, destined to 3 gateway nodes. Figures 4.6 a) and b) show total network throughput and average end-to-end packet delay respectively. AntMesh once again outperforms all the other routing metrics because of its interference-avoidance strategy, both in terms of throughput and average end-to-end packet delay.Figure4.6 b) captures the situation when even though, the interference avoidance feature of ARM metric of AntMesh might force a packet to take longer path, the fact that the path chosen would be less prone to packet losses results in lower end-to-end delay. Similarly, to the previous case, the throughput performances of all investigated protocols were comparable until the saturation point from which on AntMesh/ARM has a significantly better performance.



(a) (b)

Figure 4.6 Random Topology (a) Network throughput (b) End-to-End delay

4.6.2 Multi Radio Multi Channel WMNs

For evaluating AntMesh in a multi-radio WMN environment, we modified the node models in our studies to contain two radios. 802.11b DCF is used as the underlying MAC protocol. We have considered two different topologies for our evaluation. The first topology is that of Figure 3.1 while the second topology is an infrastructure grid mesh topology with random location nodes added. In this semi-random topology we place 20 nodes uniform randomly in a 1000m by 1000m area, and four traffic flows are generated destined to one to four gateway nodes. We compare the performance of AntMesh/ARM with two of the already described routing metrics: WCETT and MIC.

4.6.2.1 MIC Topology and Traffic Pattern

We want to observe the effectiveness of our proposed ARM in overcoming the MIC limitation of not capturing inter-flow interference. We therefore compare the performance of AntMesh with MIC on the topology shown in Figure 4.4. We first start a flow from node S to node D and then we add four more flows each having different source and destination pairs. Figure 4.7 shows the end to end throughput of flow (S-D) with varying offered load when ARM/AntMesh, MIC, and WCETT are used in the illustrative topology. When the load on the network increases, node A experiences more contention on its outgoing links (captured by PEM). Clearly, the benefit of considering the queue sizes of neighboring nodes to measure the actual interference rather than just a mere number of neighboring nodes (MIC) is evident from the increased throughput in the graph. Notice, that WCETT performs almost equally well as ARM; we believe this is because the best path in the illustrative topology is path-2 ($S - G - C - F - D$) (a channel diverse path) therefore, the channel cost component and its use of the ETT metric managed to choose the "right path". However, we will see later that WCETT underperforms AntMesh and even MIC on large scale mesh networks because of its inability in effectively measuring inter-flow interference.

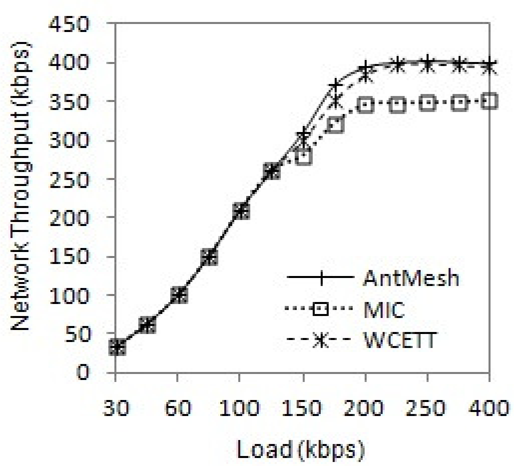
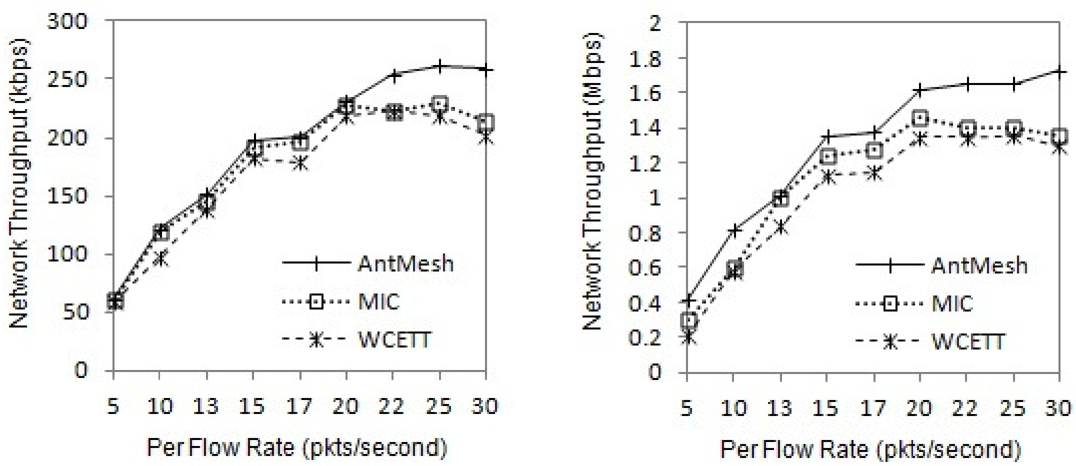


Figure 4.7 Throughput in the Illustrative MIC-tuned Topology

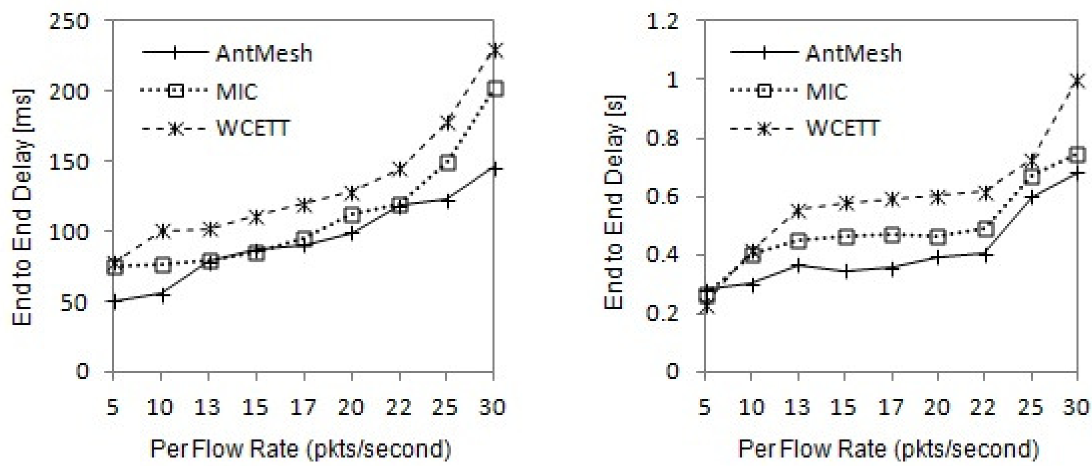
4.6.2.2 Mesh Topology

Figures 4.8 a) and b) present the total network throughput versus the flow rate for the grid and the random topologies, respectively. We can observe that AntMesh provides increased throughput, outperforming the other schemes by as much as 35% especially when the network becomes congested. WCETT performs the poorest among the schemes since it does not capture the inter-flow interference among the nodes. Again, the increased throughput of AntMesh is due to the capturing of inter- and intra-flow interferences by its PEM of ARM. Although, MIC performs equally well when the network load is light, as the network starts to saturate, nodes' queues start filling up, affecting the link quality; this is only recognized by ARM's PEM module. The involvement of the PEM module of ARM results in earlier detouring of the flow than with the MIC where only neighboring node count is considered.



(a) (b)
Figure 4.8 AntMesh Throughput (a) Grid Topology (b) Random Topology

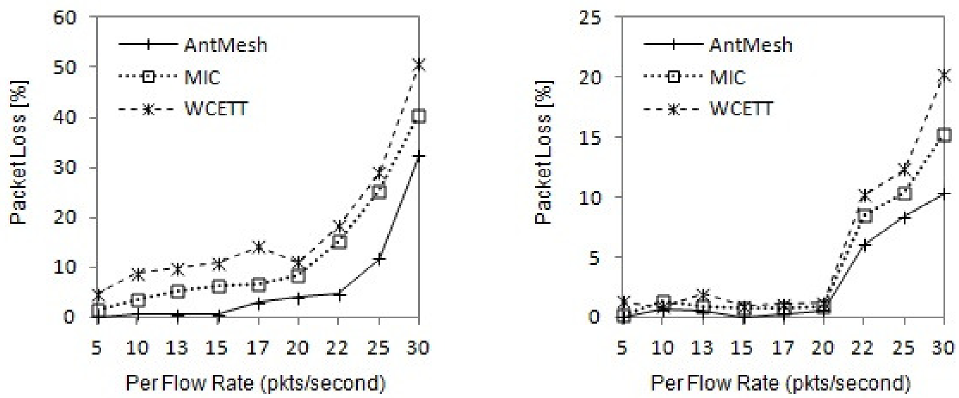
Figures 4.9 a) and b) demonstrate the average end-to-end delay in the grid and random topologies respectively. AntMesh outperforms other schemes in terms of end-to-end delay of packets by as much as 30%. However, and in particular in the random topology, the end-to-end delay of ARM/AntMesh and MIC are comparable when the network is heavily loaded; this is because of overhead experienced by AntMesh due to the path change (influenced by the stochastic behavior of queue lengths).



(a) (b)
Figure 4.9 End-to-End delay (a) Grid (b) Random

The packet loss ratios of all three approaches in the grid and random topologies are shown in Figures 4.19 a) and b) respectively. When the network load is low, the packet loss ratios, as expected, for all of the three schemes are low; as the network becomes more and more saturated, AntMesh outperforms WCETT and MIC by avoiding inter- and intra-flow interference paths. Particularly, in grid topology, the difference in packet loss ratio among AntMesh and other two schemes is very evident and can result in as much as a 60% reduction.

Table 4.3 summarizes our multi-radio mesh network results of ARM/AntMesh and the two comparative approaches (MIC, WCETT). It lists the average packet latency of the end-to-end packet delay. It also shows the packet delivery fraction (PDF) which is the ratio of the number of packets received at all the destinations to the total number of packets sent in the network. One interesting observation in Table 4.3 is that the latency of almost all the metrics on our MIC-illustrative topology is significantly less when compared to the latency of these metrics in a mesh topology. This is due to the size of the MIC-illustrative topology which contains fewer mesh nodes as compared to the larger mesh topologies that we have considered in the results. AntMesh reduces packet drops and maintains smaller average packet latencies and PDFs as compared to both the MIC and WCETT. Also, note that in our MIC-illustrative topology, the PDF of MIC is less than WCETT whose PDF is close to ARM. This is due to the timely estimation of link qualities by taking the queue sizes into account with the help of hello smart ants and the ability of backward smart ants to select the best path in terms of channel diversity.



(a) (b)

Figure 4.10 Packet Loss (a) Grid (b) Random

Table 4.3 Multi radio Mesh Network Results

Topology	Illustrative		Mesh Topology			
			Grid		Random	
	Avg. Latency (ms)	Packet Delivery Fraction [%]	Avg. Latency (ms)	Packet Delivery Fraction [%]	Avg. Latency (ms)	Packet Delivery Fraction [%]
AntMesh	13.772	91	93.96	93.2	0.41	94.33
MIC	21.34	82	110.81	85.79	0.49	90.99
WCETT	15.34	90.1	132.35	81.23	0.59	86.56

4.7. Summary

This chapter studied the problem of packet routing in WMNs with a specific emphasis on a smart ant framework. To enable the use of such intelligent and distributed agents we proposed an interference-aware data forwarding architecture called AntMesh. AntMesh provides a distributed, stochastic heuristic to solve a dynamic network routing problem. In addition, we also proposed a novel routing metric called Ant Routing Metric (ARM) which was designed to enable effective utilization of the space/channel diversity typically common in infrastructure WMNs. We demonstrated the stability of our proposed ARM metric through simulations; we have shown that AntMesh/ARM quickly converges to favorable path under situations when traffic characteristics change (among others when load on the network is increased). We have shown that with an appropriate tuning of the parameters, ARM behaves better when compared to other competing approaches in both single- as well as multi-radio mesh networks.

CHAPTER 5

INTERFERENCE-AWARE CHANNEL ASSIGNMENTEXPLOITING PARTIALLY OVERLAPPED CHANNELS IN 802.11-BASED MESHES

In this chapter, we address the problem of channel assignment considering partially overlapping channels (POCs) for interference avoidance in multi-radio multi-channel wireless mesh networks (MRMC-WMNs). A novel interference model is proposed which provides a systematic approach of measuring the interference caused by links operating on POCs by considering both adjacent channel interference as well as corresponding physical distances between mesh nodes. Based on this model, we design an interference-aware channel assignment algorithm called *i*-POCA which enables the use of smart ants for assigning orthogonal and non-orthogonal channels to radios in order to minimize total network interference. More precisely, we will show a centralized and a distributed design for *i*-POCA. We evaluate our algorithms through extensive simulations and demonstrate that our proposed algorithms improve network throughput by efficient utilization of the available spectrum.

5.1. Introduction

Mesh nodes incorporating multiple radios operating in different radio channels form the basis of multi-radio multi-channel wireless mesh networks (MRMC-WMN). This technology enables a potential large improvement in bandwidth capacity compared to single radio mesh networks. However, in a typical multi-radio mesh network, the total number of radios within the network is usually significantly higher than the number of available channels in the network (e.g., the total number of channels in the U.S. for IEEE802.11b/g is 11). This forces many links to operate on the same (set of) channels, resulting in interference among transmissions. The existence of such interference if not designed for, can affect the capacity of the network by decreasing the throughput. Therefore, understanding and mitigating interference has become one of the fundamental issues in MRMC-WMNs and recently a number of channel assignment (CA) solutions have been proposed to solve this problem [2, 35-47].

Most of the existing research [45-47]on channel assignment is focused on assigning orthogonal (non-overlapping) channels to links belonging to neighboring nodes in order to minimize the interference in the network. Since, links operating on orthogonal channels do not interfere at all, multiple parallel transmission can be possible resulting in overall network throughput improvement. The number of non-overlapping channels in commodity wireless platforms such as 802.11b/g is very small (only 3 orthogonal channels out of total 11 channels), while nodal density in a typical MRMC-WMN is high. This realization has recently drawn attention to the study of partially overlapped channels (POC) for channel assignment in such networks [35]. The basic idea is to make the whole wireless spectrum available to nodes for channel selection as a result of which, partially overlapped channels may be used. This enables multiple concurrent transmissions on radios configured on POCs and therefore may increase network capacity (in terms of throughput) by more efficient spectrum utilization.

Previously, an algorithm for channel assignment based solely on orthogonal channels had to deal with only co-channel interference. However, one of the major issues in designing efficient channel assignment schemes using POCs is the adjacent channel interference, which is the interference between two neighbors configured on adjacent channels. The effect of such adjacent channel interference has a direct relationship with the geographical location of these two nodes, i.e., the farther two nodes are apart, the less interference is created on adjacent channels. Nonetheless, the assignment of orthogonal and non-orthogonal channels in high density mesh networks needs to be carefully coordinated; the key issue lies in the fact that the interference between adjacent channels has to be considered. This needs to be done intelligently so that channel capacity is maximized, otherwise the shared nature of wireless medium can lead to serious performance degradation of the whole mesh network.

This chapter focuses on the problem of channel assignment using partially overlapping channels for interference avoidance in the context of 802.11 b/g based MRMC-WMNs. The salient features of our work are as follows:

- We propose a partially overlapped interference graph (POIG) model to capture the interference caused by links operating on non-overlapping and overlapping channels.
- Based on this model, we design *i*-POCA; a channel assignment algorithm which enables the use of smart ants for assigning POCs to radios.
- *i*-POCA utilizes all the available channels effectively and achieves a significant improvement in network throughput by reducing various interference effects (e.g., self, co-channel, and adjacent channel interference) that are typically common in a MRMC-WMN.
- Simulation results demonstrate that our proposed *i*-POCA algorithm outperforms other existing schemes by as much as 36% when the network is saturated.

The rest of this chapter is organized as follows. In Section 2, we describe the benefits of using POCs in WMNs with the help of our test bed experiments, followed by the description of our proposed interference model in Section 3. Our interference-aware channel assignment algorithm *i*-POCA is presented in Section 4(both its centralized and distributed versions). Section 5 evaluates the performance of the *i*-POCA algorithms. Section 6 concludes this chapter.

5.2. Benefits of Using Partially Overlapped Channels (POCs)

In this section, we will discuss the benefits of using POCs in WMNs. First, we will explain what the different scenarios are and where the use of partial overlap among channels can be useful. This will be followed by results from our testbed experiment to demonstrate the effectiveness of POCs for WMN.

Mishra et. al,. in [58] have performed detailed experiments to demonstrate the effectiveness of using partial overlap among channels in WMNs. The authors have measured the signal to noise ratio (SNR) of two communicating nodes configured on adjacent channels and mapped them on to a normalized [0,1] scale with 0 representing the minimum signal received; Their results are shown in Table 5.I.

Table 5.1 SNR of transmission made on channel 6.

Channel	1	2	3	4	5	6	7	8	9	10	11
Normalized SNR (I-factor)	0	0.22	0.60	0.72	0.77	1.0	0.96	0.77	0.66	0.39	0

A typical bandwidth of an IEEE802.11b channel which uses direct sequence spread spectrum (DSSS) is 44MHz. It is distributed equally on each side of the center frequency of that channel i.e. 22MHz on each side. A transmit spectrum mask as shown in Figure 5.1 is applied to the signal at the transmitting station which is basically used by the transmitter to limit the output power on different frequencies. As it can be seen from the figure, the mask is set to 0 dB at the center frequency where signal is passed without any attenuation. However, at frequencies beyond 11MHz on either side of the center frequency, the signal's power is attenuated by as much as 30 dB and at 22MHz as much as50 dB. The receiver also uses a band pass filter centered around the nominal transmission frequency of the channel.

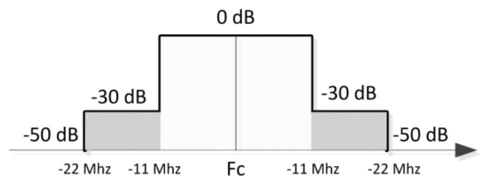


Figure 5.1 IEEE 802.11b transmit spectrum mask

Three scenarios are discussed in [58] where the use of partial overlap among the channels can be useful in the context of wireless mesh networks.

Multi-channel communication: The first scenario is when a node can communicate with two of its neighboring nodes configured on orthogonal channels (OCs) by operating on a partial overlapping channel. Basically, for a little reduction in throughput, one can use partially overlapping channels and this can give flexibility in topology construction while reducing the extra overhead in channel switching to enable communication.

Throughput improvement: The second scenario is when nodes in a mesh network have only one radio and therefore, can be configured to one channel at a time. there is a possibility of network disconnection while assigning different channels to nodes in the network. Channels with partial overlap can be assigned to nodes in such a manner that improves the overall network throughput capacity. In this way, the assignment of partial overlapping channels has to be intelligent enough to utilize the maximum bandwidth available and therefore can result in significant throughput improvements.

Channel re-use: Shorter ranges for frequency reuse can be obtained if two interfering links are assigned partially overlapping channels rather than orthogonal channels. It is possible to significantly improve the overall channel re-use by careful assignment of channels which results in better throughput.

Later, in [59], the same authors have shown the advantage of using POCs in two different types of networks i.e. WLANs and WMNs. In a WLAN setup, close by access points (APs) can be assigned POCs such that the signal attenuation due to the overlap degrades to a tolerable level. In other words, the interference range of the APs is reduced as perceived by the neighboring AP operating on a partial overlapping channel. This provides efficient spatial re-use of channels and more APs can operate concurrently providing better service to clients. Similarly, in a single radio WMN environment throughput can be improved when nodes can be configured to overlapping channels in order to avoid network disconnection and also to avoid channel switching overhead.

Next we will show results from experiments performed on a real testbed in order to evaluate the benefits of using partially overlapping channels in mesh networks. Our experimental testbed consists of four LinksysWRT54GLv1.1 wireless routers, each equipped with one radio. We installed the Freifunk firmware [60] on these routers for more freedom in our experiments. We created two point-to-point networks between two router pairs and thus formed two links each consisting of two routers as shown in Figure 5.2; Link-1 belongs to Pair-1 and Link-2 belongs to Pair-2. Each radio onLink-1 is fixed on channel 6; we varied the channels Pair-2 from 1 to 6. The distance between nodes belonging to the same link is kept constant throughout the experiment. Pair-1 is fixed at a particular location whilePair-2 is moved to various locations with distances from Pair-1 ranging from 5 to 30 meters. UDP and TCP traffic is generated on both links lasting for 10 seconds. The throughput on Link-2 is measured and the results are averaged over several runs. Three different IEEE802.11b defined data rates are used for conducting the experiments, i.e. 2, 5.5 and 11 Mbps.

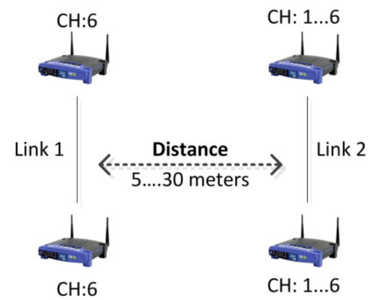


Figure 5.2 POC measurement testbed

Figures 5.3 a), b), and c) show the UDP throughput on Link-2 with the different channel separations for the three data rates. It can be seen that as the distance between the two interfering links are increased, the throughput increases due to the reduced amount of interference. In this setup we did not see any further improvements when nodes were more than 30 meters apart. However, the same maximum throughput can be achieved at a much lower distance with increased channel separation between the two links. For example, at about 20 meters, Link-2 achieves the maximum benchmark throughput, when the channel separation between the two links is three. For data rates 5.5Mbps and 11Mbps, we notice similar results; however, maximum throughput can be achieved by eliminating interference at a much lower distances i.e. about 15 meters, when the channel separation is 3 as compared to 30 meters, when both the channels are separated by 1 only. Figures 5.4a), b), and c) show the comparable results when TCP traffic is used on all the three 802.11b data rates.

From these results, we can extrapolate the interference ranges of nodes with varying channel separation and at different data rates; this comprehension is shown in Figure 5.5. Each point in the graph represents the minimum distance that is required between the two links in order for them to experience no interference and achieve maximum throughput when they are on particular partially overlapping channels (with a given channel separation). We can observe that the interference ranges are decreasing with increasing channel separation and increasing data rates. From these measurements, we can empirically conclude that the interference range of nodes operating on POCs is significantly less than the range when they are on the same channel. (Similar experiments have been performed before in [35, 41, 48];however, those experiments were done either on wireless card equipped computers or a computer attached to an access point. We believe, that our setup is more representative for a WMN and thus provides a better understanding of POCs in mesh networks.)Therefore, there is a tradeoff between efficient utilization of the wireless spectrum and a slight decrease in throughput. An intelligent assignment of partially overlapping channels can decrease the impact of interference, eventually resulting in more efficient utilization of the spectrum.

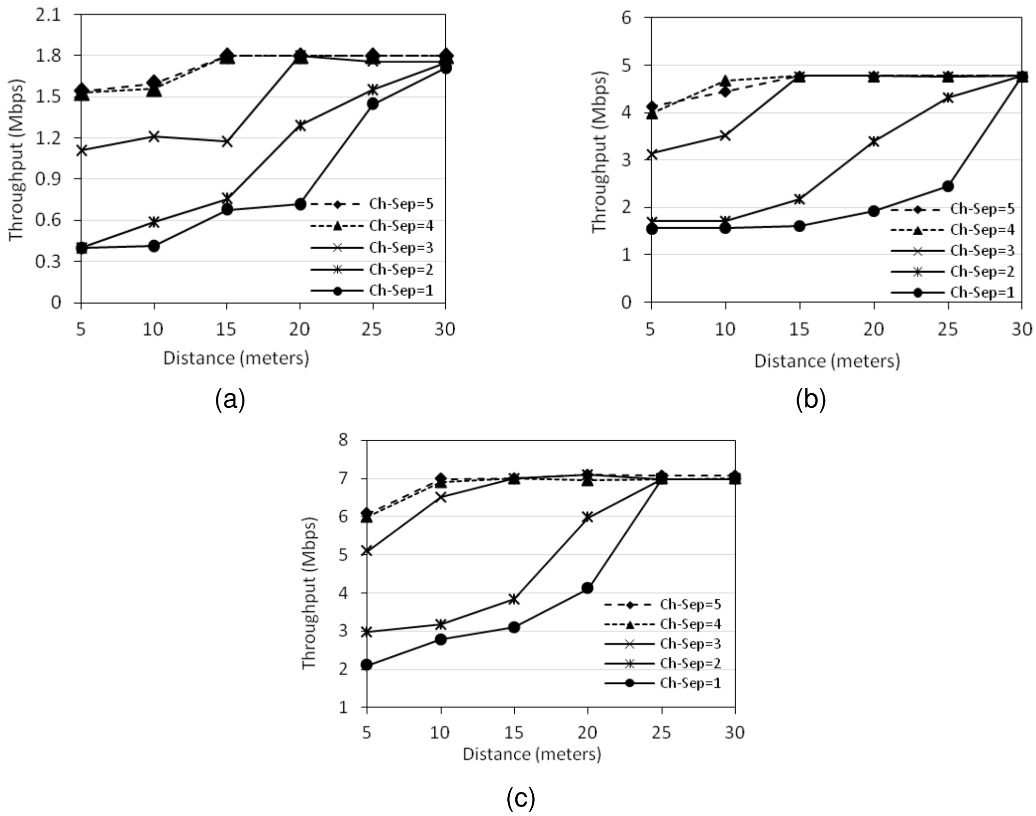
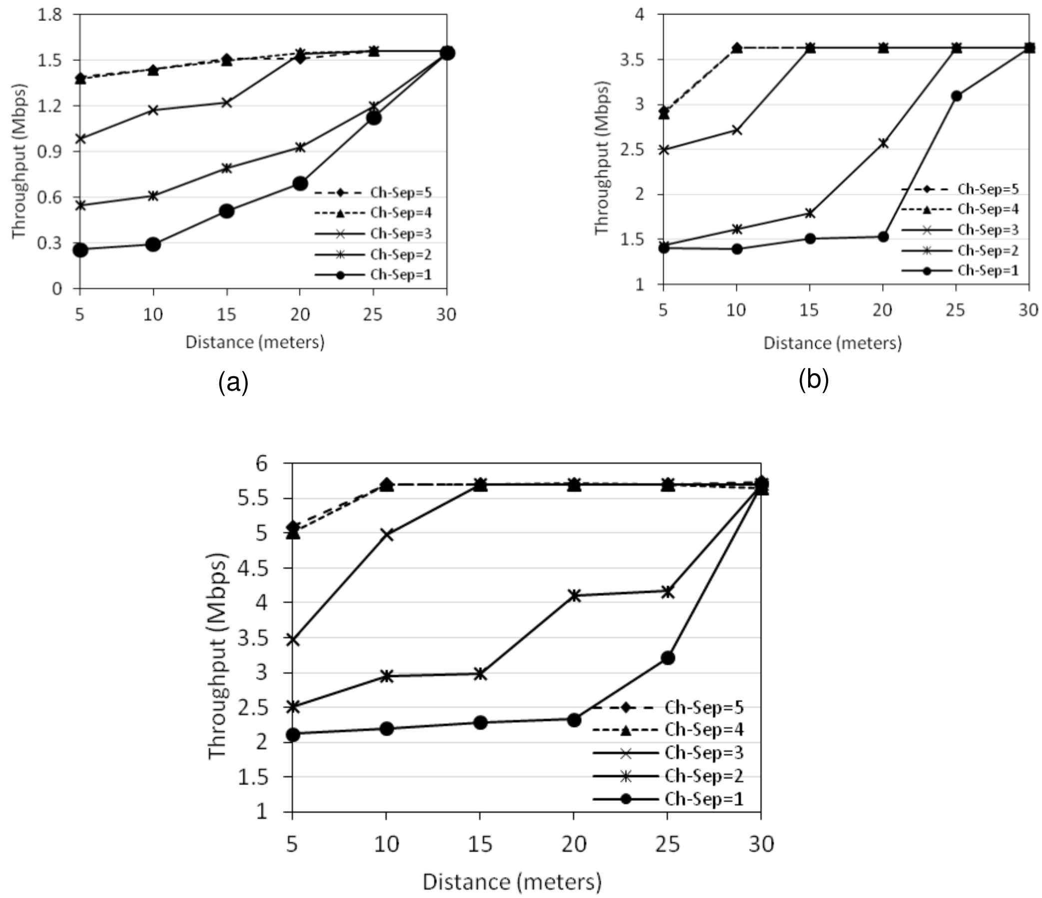


Figure 5.3 UDP throughput of two interfering links as a function of channel separation. (a) 2.2 Mbps (b) 5.5 Mbps (c) 11 Mbps



(a) (b)

(c)

Figure 5.4 TCP throughput of two interfering links as a function of channel separation. (a) 2.2 Mbps (b) 5.5 Mbps (c) 11 Mbps

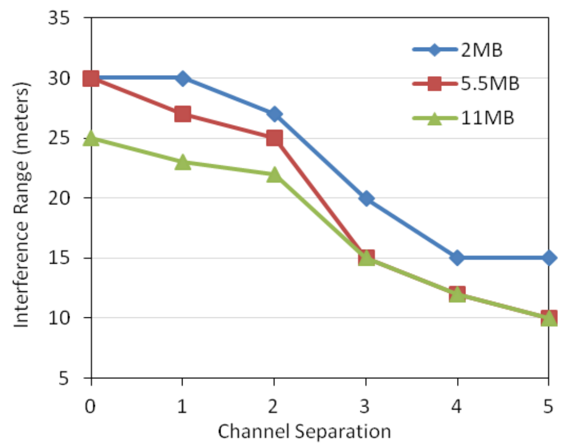


Figure 5.5 Interference range as a function of data rates.

5.3. Interference Model

In this section, we discuss our first contribution for POCs, the partially overlapped interference graph (POIG), which can effectively model interferences in a WMN.

5.3.1 Partially Overlapped Interference Graph (POIG) Model

The first step in developing mechanisms which take advantage of the partial overlap is to build a model that captures this overlap in a quantitative fashion. We introduce POIG, modeling co-channel, adjacent channel and self-interference. Previously we have shown that the interference range of two nodes communicating using POCs is much less than that of two nodes communicating on same channel. This is because only a part of signal's power from the sender is picked up by the receiver. The decrease in interference range is related to the amount of channel separation that exists between two neighboring POCs. To introduce our interference model for POCs, let us take a simple case where a sender node *s* is transmitting at channel *i* and a receiver node *u* is receiving at channel *j*. We define *Th* as the threshold that specifies the tolerance level of interference for successful communication. Let P_t be the transmitted power of the signal and $D_{s,u}$ the physical distance between the sender and receiver, then under the two-ray ground propagation model of [61]:

$$Th < \frac{o(i,j)P_tK}{ND_{s,u}^{\propto}} \tag{5.1}$$

where *K* is a constant reflecting the effect of antenna gain and channel attenuation, *N* is the ambient noise experienced by receiver *u* and α is the path loss exponent; *o(i,j)* is simply the convolution of power spectrum densities (PSDs) of the transmission and reception channels and thus the extent of overlap between channels *i* and *j*. Therefore, Equation 5.1 indicates the possibility of correctly receiving a transmission on a partially overlapping channel *j* that was sent on channel *i* as long as the received power is above the threshold.

Now we can define the POC interference range (adjacent channel interference range) between two nodes that are configured on two adjacent channels *i* and *j* as:

$$ACIR(i,j) = \delta\left(\frac{o(i,j)P_tK}{NTh}\right)^{\frac{1}{\propto}} \tag{5.2}$$

where $\left(\frac{o(i,j)P_tK}{NTh}\right)^{\frac{1}{\propto}}$ is effectively the transmission range between two nodes and *δ* is the coefficient that characterizes the impact of channel separation on the interference range. (The authors of [48] have conducted an experimental study to calculate the values of *δ* under different channel separations and transmission bit rates in 802.11b networks.) Similarly, the co-channel interference range between two nodes configured on same channel can be defined as:

$$CCIR(i,j) = \left(\frac{P_tK}{NTh}\right)^{\frac{1}{\propto}} wheni = j \tag{5.3}$$

Note that *CCIR(i,j)* is the regular interference range of a transmitting node typically taken as twice its transmission range.

We have sufficient background now to present our partially overlapped interference graph (POIG) model which is a weighted undirected interference graph and models an 802.11b based MRMC-WMN employing POCs. The basic idea is to assign weights to the edges of our interference graph in such a way that they represent the amount of adjacent or co-channel interference that exists between two radios configured on POCs belonging to neighboring nodes.

We need to measure the adjacent channel interference by taking into account the amount of channel overlap between two radios on adjacent nodes together with the physical distance between the corresponding nodes. We consider a wireless mesh network with stationary wireless routers (nodes), where each node is equipped with a certain number of transceivers that can work on any channel provided by the IEEE 802.11b standard. For the sake of simplicity, we assume that all nodes transmit with the same transmission power. This type of mesh network can be modeled by an undirected graph G(V,E), where nodes are represented by the vertices *V* in the graph and the communication links between different nodes correspond to the set of edges *E*. Two nodes *s* and *d* are connected by a link $l \in E$ if they are within the regular interference range of each other. The link weight *w(l)* is defined as:

$$w(l) = \frac{ACIR(i,j)}{CCIR(i,j)} \qquad (5.4)$$

where *ACIR(i,j)* is the interference range of a POC; it is always less than *CCIR(i,i)* which is the co-channel interference range. Thus, *w(l)*is always mapped to a real number between [0..1]. The link weight is the ratio of the adjacent channel interference range to the co-channel interference when both the channels are same. Hence, our weight function quantifies the physical distance separation between nodes along with the adjacent and co-channel interference among the radios in a mesh network.

We illustrate the concept of our POIG model in Figure 5.6. The wireless network in the figure has three nodes *A, B* and *C*, as shown in the communication graph (Figure 5.6 a). Node *A* has two radios indicated by the rectangular shapes on top of that node; nodes *B* and *C* have only one radio each. Figure 5.6 b) shows the partially overlapped interference graph G(V,E); each node's radio in the communication graph is represented by a vertex $v \in V$; an edge $e \in E$ between two nodes exist in G if and only if $w(e)>0$. In other words, if there is any interference between two radios in a network, it will be represented by a weighted edge in our interference graph with the weight showing the amount of interference experienced by these two radios based on the channel separation. If the two radios were on the same channel, then the edge weight would be 1, resulting in complete channel overlap and therefore maximum interference.

Channel assignment (CA) problem. Based on our POIG model, the CA problem can be stated as follows. To assign a POC to each link in the graph such that the channel separation between two adjacent links are maximized by choosing light weight links in the graph which would eventually results in reducing the overall network interference. Our objective therefore lies in minimizing the overall network interference which will result in an overall improved network capacity.

Relationship with set-t-coloring. Our CA problem for MRMC-WMNs can also be modeled as a graph coloring problem, specifically a set-t-coloring problem [62] in which colors (channels) are assigned to a single vertex (with multiple radios) such that they enforce a certain distance (orthogonal means a channel separation of 5 in 802.11b) between radios of that vertex together with the constraint that the channel separation between adjacent vertices in the graph should not be less than their edge weight (should not belong to set *T*). This set-t-coloring formulation captures all the three types of interferences that typically exist in a MRMC-WMN, i.e., self-, co-channel and adjacent channel interference. In our POIG model, since an edge weight represents the ratio of POC to co-channel interference, we can define a channel separation mapping function $C_{s,u}$ between two vertices s and u from the weight function in Equation 5.4 as follows:

$$C_{s,u} = \begin{cases} 2 & 0 > w(l) \leq 0.5 \\ 5 & 0.5 > w(l) \leq 1 \end{cases} \quad (5.5)$$

[35] showed that two nodes configured on POC with channel separation larger than 2 provide the same throughput as two orthogonal channels in 802.11b networks. Similarly, a channel separation of 5 means that the adjacent vertices should be assigned orthogonal channels. For our example (in Figure 5.6), we construct a set-t-coloring graph from the corresponding POIG graph as shown in Figure. 5.6 c). (The numbers inside the vertices correspond to radio ids.) The link weights in our interference graph are mapped to the channel numbers as link weights in the set-t-color graph. These channel numbers define the minimum separation that is required for a conflict free communication between two adjacent nodes. Note, that the channel set on top of the nodes avoids self-interference issue by implementing the constraint that no two radios on a single node can have partially overlapped channels. Finding solutions for the set-t-coloring problem is known to be NP-hard, therefore, an optimal solution of our channel assignment problem is also NP-hard. The focus of this chapter is therefore to define scalable and fault tolerant CA heuristics using POCs in MRMC-WMNs.

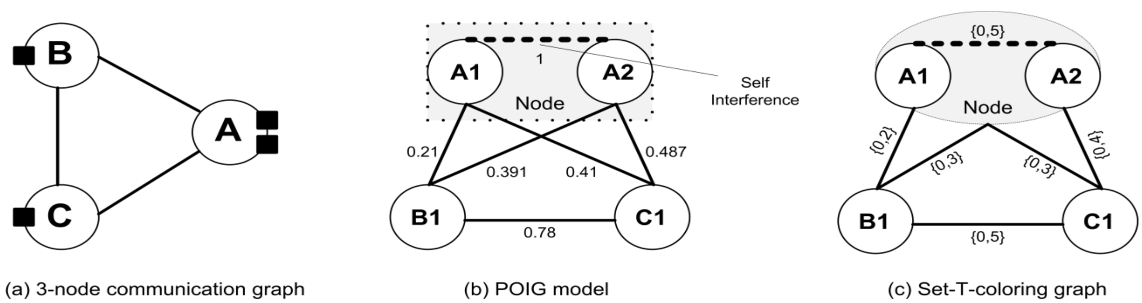


Figure 5.6 POIG Model (a) Three node communication graph (b) partially overlapping interference graph (POIG) (c) set-t-coloring graph.

5.4. *i*-POCA - Channel Assignment Algorithm

In this section, we propose *i*-POCA, a channel assignment algorithm which utilizes POCs. *i-POCA* makes full use of the whole available 802.11b spectrum by assigning non-overlapping (orthogonal) and partially overlapped channels (non-orthogonal) to radios in order to minimize total network interference. We present two versions of *i*-POCA, the first is a centralized algorithm for channel assignment. The second algorithm is a distributed interference-aware partially overlapping channel assignment algorithm based on smart ants following the train we have established in the previous chapters.

5.4.1 Centralized Algorithm (i-POCA-C)

Centralized algorithms are (arguably) realistic in infrastructure WMNs where they are normally controlled by a single entity. The *i*-POCA-C algorithm consists of three components, namely interference graph (POIG) construction, link ordering and channel assignment.

POIG construction. The first step of our centralized algorithm involves the construction of the partially overlapped interference graph (POIG). The interference graph of the whole network is constructed by a designated node (server) in the mesh network and each node measures the interference values for all channels supported by its radios and periodically sends this information to the server. The server first constructs the interference graph using the above information and then uses the interference estimates to assign weights to the edges in the graph (see Equation 5.4).

Link ordering. After the construction of the POIG, the server arranges the communication links based on their weights. The main purpose of this component is to describe in which priority the links should be assigned channels. The links are arranged in the descending order of link weight values.

Channel assignment. Finally, the last component of the centralized CA algorithm assigns channels to the links of the interference graph (POIG). We use a greedy strategy of selecting the channels based on the link weights in the graph. The server assigns a particular channel to a link using the channel separation mapping function described in Equation 5.5. Note that with this type of channel assignment, if the link weight is high (since they are already ordered in descending order), orthogonal channels are being assigned first and when they are exhausted then partial overlap channels are assigned by the algorithm. This would eventually result in less interference and increased channel utilization.

The pseudo-code for the centralized CA algorithm using POCs is shown in Algorithm 1. Line 3 is responsible to order the edges according to the edge weights in descending fashion. Lines 5-9 encode the assignment of channels to the edges based on their weight (by first trying to assign any free orthogonal channel to the link, if it succeeds then it moves to the next edge, otherwise it will pick a POC).

5.4.2 Distributed Algorithm (i-POCA-D)

i-POCA-D performs the channel assignment with the help of smart ants. There are a number of ACO approaches that have been proposed for solving optimization problems but to the best of our knowledge our proposed distributed interference-aware CA algorithm is among the first works that investigates the use of smart ants in MRMC-WMNs and demonstrates a possible performance advantage.

Algorithm 1: *i*-POCA-C

1: Let $E=\{e|e\in POIG\}$

2: Let $K= C^o \; U \; C^p$ List of all available channels

where $C^o =\{ c_1, c_2, ..., c_n\}$ List of free orthogonal channels

and $C^p =\{ c_1, c_2, ..., c_m\}$ List of free (POCs)

3: Order_Link(E)

4: **while** sizeof(E) > 0 do

5: E = removeHead(E)

6: If w(e) > 0.5 then

7: $e \leftarrow c\in C^o$ assign orthogonal channel *c* from the set of free channels to the edge, making the channel separation either one of these c+5,c,c-5

8: Else

9: $e \leftarrow c\in C^p$ assign POC *c* to the edge, making the channel separation either one of these c+δ,c,c- δ

10: end while

The above proposed POIG model needs to be derived based on the communication topology, therefore, the first step in designing a distributed channel assignment algorithm is to build the interference graph which can be constructed only based on local topology information. Next we explain how this local interference graph is constructed in our distributed *i*-POCA algorithm.

Each node computes its local interference graph by periodically sending small control packets, i.e., *smart ants* to its two-hop neighbors. The ants contain the connectivity and all channel interference information that a particular node has measured so far. Based on this information, nodes construct their local view of the network in the form of partially overlapped interference graph (POIG). After the construction of the POIG, nodes generate special ants called *channel ants* whose role is to iteratively assign POCs to nodes of the interference graph by traversing a path with the objective of minimizing the local network interference at each iteration. At any node *i*, channel-ant *k* selects the next node to assign POCs according to a probabilistic decision rule which is a function of a local pheromone value (maintained by node *i*) and a heuristic. Equation 6 shows the probability with which ant *k* performs the transition from node *i* to node *j*:

$$P_{i,j}^{k} = \frac{[\varepsilon_{i,j}]^{\alpha}.[w_{i,j}]^{\beta}}{\sum_{l \in N_k}[\varepsilon_{i,l}]^{\alpha}.[w_{i,l}]^{\beta}} \tag{5.6}$$

where $\varepsilon_{i,j}$ is the value of the pheromone intensity associated with the edges and $w_{i,j}$ is the local heuristic calculated a priori; $w_{i,j}$ indeed represents the link weights in the interference graph which models the adjacent channel interference between the two nodes on both ends of the link together with the physical separation of these nodes. $\alpha$and β are two parameters which determine the relative influence of the pheromone intensity and the link weight respectively; N_k is the list of neighbors of node *i* that ant *k* has not visited yet. *Channel-ants* keep record of the POCs that they assigned to the nodes so far in a list (also called a tabu list) which will help in making sure that a channel does not get re-assigned a second time during the ants' travel.

After a *channel-ant* completes its channel assignment path, it triggers the update of pheromone values according to the quality of the constructed solution in terms of how much interference it has successfully minimized. However, before describing the pheromone update process we need to define what a pheromone trail is. Our definition of pheromone trail relies on the fact that a trail should correctly integrate the edge weight estimates calculated by our interference model into the pheromone table entries. This will reduce interference with respect to the channel assignment process.

Let us denote the channel separation between two radios belonging to two neighboring nodes, assigned on POCs by a recent channel-ant *k* as S_k; then:

$$\varepsilon_{i,j}^{l} = 0.5 \times \left(\frac{1}{S_k}\right) \tag{5.7}$$

where $\varepsilon^{l}_{i,j}$ now represents the new pheromone trail belonging to link *l* between node *i* and *j*. It depicts the reinforcement value that will be added to the pheromone table; it is proportional to how wide the channel separation is. The channel separation can be calculated using Equation 5.5. Now, the pheromone values can be updated as:

$$\varepsilon_{i,j} = (i - \rho)\varepsilon_{i,j} + \sum_{k=1}^{N} \varepsilon_{i,j}^{k} \tag{5.8}$$

where $\rho \in (0,1)$ is the evaporation rate of previous pheromone trails and *N* is the number of channel ants used in the algorithm. The probability of selecting a particular node for channel assignment changes over time according to a pheromone trail factor, which is determined by the quality of the previous solutions.

i-POCA-D is summarized in Algorithm 2. Each channel ant belonging to a node in a mesh network performs the following task locally in a cooperative manner to support the global optimal partially overlapped channel assignment problem.

Algorithm 2: *i*-POCA-D

1: Construct "Local" partially overlapped interference graph (POIG)
2: Initialize the pheromone trails of each edge by a constant
3: Repeat
4:     Move to the next node according to probability function (pheromone table)
Among all the node's outgoing links *(i,j)* from the list, choose the one to assign channel that produces largest decrease in local interference.
5:     For each edge do
6:         Update the pheromone intensity using the pheromone updating rule in Equation 8
7:     End for
8: **Until** a maximum number of iterations in terms of channel ant generation is achieved

5.5. Performance Evaluation

In this section we show simulation results to evaluate both versions of our *i*-POCA algorithms, i.e., *i*-POCA-C and *i*-POCA-D. We compare the performance of our proposed schemes to a centralized channel assignment algorithm using only orthogonal channels (C-3 channels) and with CAEPO [63] which utilizes both orthogonal and overlapping channels for channel assignment. Briefly, in CAEPO, each node has two radios, one is fixed and the other one can be switched to different channels depending upon the traffic characteristics. Basically, it consists of a metric to calculate interference and that metric is then used to assign channels to the links with the objective of minimizing network interference.

The goals of our evaluations are to quantify i) the effectiveness of the interference-aware partially overlapping channel assignment algorithm in mitigating interference and ii) the benefit of using POCs over orthogonal channels. In particular, we study the impact of *i*-POCA in improving throughput in an 802.11b based MRMC-WMN. The algorithms are compared in terms of their performance in network throughput, packet delivery ratio and channel utilization ratio. Our simulations were performed using ns-2. Each of our depicted data points is an average over enough simulation runs to claim a 95% confidence that the relative error is less than 5%.

Like in previous chapters, in our simulations, all flows are destined to multiple gateway nodes. IEEE 802.11b is used as the wireless technology. We randomly generate networks of size 1000m x 1000m each having 20 nodes. All nodes in our studies are equipped with one to three radios. We generate a certain number of UDP flows on the network destined to 3 gateway nodes. The packet size in all the flows is fixed to 512 bytes. For our distributed smart ant based channel assignment algorithm (*i*-POCA-D), the values of parameter α and β are set to be 2 and 9 respectively. The evaporation rate ρ is set to ρ=0.2, channel-ants are generated at a rate of 5 ants per second and the algorithm terminates after 20 iterations.

Figure 5.7 shows the network throughput of the various schemes as a function of load on the network. We can observe that both the centralized and distributed versions of *i*-POCA outperform the other schemes. In particular, *i*-POCA-C achieves up to 58.9% and 36.3% higher throughput than its orthogonal and CAEPO counterpart respectively. The reason for this performance improvement lies in the intelligent interference modeling of our proposed algorithm in the form of partially overlapped interference graph (POIG). Although, *i*-POCA-D performs better than CAEPO and C-3,it (expectedly) underperforms *i*-POCA-C by11.2%. The reason for this slightly degraded performance of our smart ant based algorithm is because of its computational complexity in calculating POIG in a distributed manner. For example, the cost of message exchange in terms of smart ant generations, and assignment of POCs to radios by channel ants would naturally eat network resources that could have been used for sending data traffic. We will see this pattern of lower *i*-POCA-D performance over *i*-POCA-C in our other simulation results also.

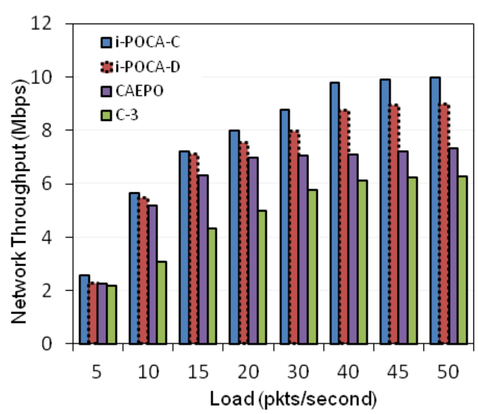


Figure 5.7 Network throughput for various channel assignments

We now investigate the packet delivery ratio (PDR) with varying number of nodes in the network. We measure PDR as the number of packets received at the destinations (gateways) to the total number of packets pumped into the network. Figure 5.8shows that the PDR decreases as the number of nodes in the network increases (as the number of flows increase). When the network size/load is small, all the schemes perform almost equally well. However, as the network size grows (and thus the number of flows grow), nodal density increases, which results in severe contention on the links; this causes performance degradation in the network. The use of partially overlapping channels in our proposed channel assignment schemes relieves the contention by assigning POCs to links and therefore, the PDR of *i*-POCA-C is up to 13.9% higher than that of CAEPO and consequently up to 37.3% higher than C-3. Although, both *i*-POCA and CAEPO use orthogonal and non-orthogonal channels, the fact that *i*-POCA captures both the partial interference and the physical distance between the interfering nodes helps it to accurately measure the interference thereby resulting in improved performance.

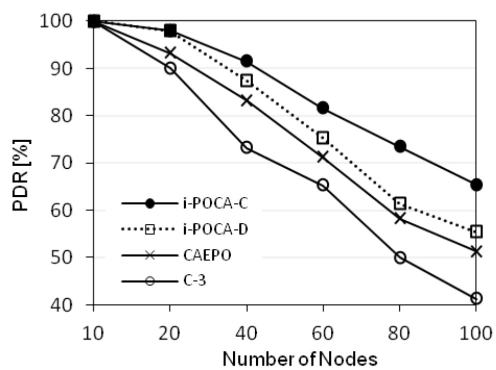


Figure 5.8 Packet delivery ratio for various channel assignment schemes.

Finally, we will use a metric called channel utilization ratio which is defined as the fraction of the number of radios configured to a particular channel to the total number of radios in the MRMC-WMN. Figure 5.9 shows the distribution of each individual channel in the network. It is evident that *i*-POCA-C&D and CAEPO utilize more channels more uniformly than C-3, which only uses three orthogonal channels. Based on the other results we can conclude that the efficient allocation of partially overlapped channels minimizes the interference resulting in more parallel communications.

Figure 5.10.showschannel utilization ratio versus the number of nodes in the network. The formula to calculate channel utilization ratio has also been changed to: the number of radios configured to a particular channel to the total number of channels in the network (i.e., 11 channels). We can observe that when there are less number of nodes in the network, assigning POCs does not provide much performance advantage unlike when the number of nodes is high. One possible reason for this behavior is that when nodes are distributed sparsely, they might be out of interference range of each other, and can be assigned orthogonal channels thus resulting in parallel transmissions without exploiting overlapping channels. Therefore, the benefit of *i*-POCA algorithm or any other partially overlapping channel assignment algorithm can only be truly appreciated when the nodal density is high (as is a typical WMN).

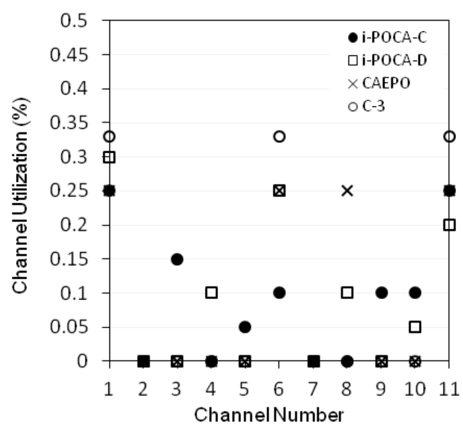


Figure 5.9 Channel utilization vs. channel number

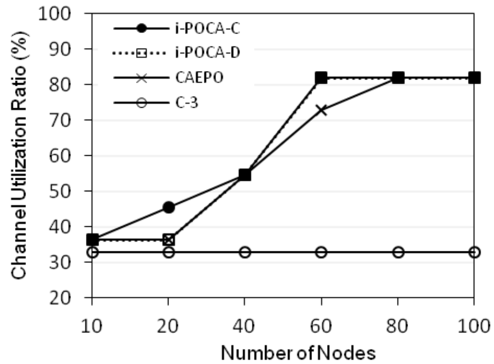


Figure 5.10 Channel utilization vs. number of nodes

5.6. Conclusion

IN this chapter, we proposed a POC based channel assignment algorithm called *i*--POCA for interference avoidance in IEEE 802.11b based MRMC-WMNs. A novel partially overlapped interference graph (POIG) model is presented with the goal of capturing all types of interference that is typical in MRMC-WMNs. Based on this model, we designed a centralized and a distributed version of *i*-POCA employing smart ants for assigning POCs to radios with the objective of minimizing overall network interference. Our simulation results showed that *i*-POCA using POCs can reach better throughputs than other existing channel assignment schemes.

CHAPTER 6

CONCLUSIONS AND FUTURE WORK

Routing in WMNs has been an active area of research for the last several years. Most of the traffic in a WMN usually flows between regular nodes and a few Internet gateways (i.e., rarely end-to-end between regular nodes). This can result in an uneven loading of links and can cause certain paths to be saturated. Similarly, due to the shared nature of the wireless medium, nodes compete with each other to access the medium resulting in possible interference. There are two typical types of interferences in a MRMC-WMN, namely: i) intra-flow interference: nodes on the path of the same flow compete with each other to access the wireless channel, and ii) inter-flow interference: this occurs when neighboring nodes carrying different flows compete for channel access when they transmit on the same channel. Both intra- and inter-flow interferences can severely affect network behavior.

We therefore, have addressed the problem of packet routing for interference avoidance in infrastructure mesh networks and designed a novel routing algorithm to incorporate techniques that specifically address these unique characteristics of WMNs. The result was *AMIRA*, an interference-aware routing protocol designed to improve load balancing by avoiding interference in a typical single radio single channel mesh backbone network. *AMIRA*, enables the use of smart ants acting as intelligent agents to probabilistically and concurrently perform the routing and data forwarding. *AMIRA* relied on a local heuristic technique to avoid interference among packet flows. This custom designed local heuristic technique employs MAC level information to measure link qualities which helps selecting reduced interference paths, thus resulting in improved load balancing. One of the salient features of our proposed routing algorithm was that it is among the first works studying the use of such biology inspired agents for efficient packet routing in wireless mesh networks (most previous research on ant-based routing was targeted at wired packet data networks and mobile ad-hoc networks). We studied the stability of *AMIRA,* i.e., how quickly it adapts itself to the changing dynamics of the network. Our research work showed that *AMIRA* quickly converges to the best path when traffic characteristics change. Also, our results demonstrated that under congestion, *AMIRA* achieves increased throughput and lower end-to-end delay when compared to other existing routing protocols (because of its interference aware technique and stochastic data forwarding nature).

One of the limitations of *AMIRA* was that it was designed for single radio, single channel WMNs and therefore, it only took care of the inter-flow interference. Thus, we extended our work and developed a forwarding architecture that is designed for both single and multiple radio infrastructure WMNs which can incorporate both inter- and intra-flow interferences. This lead to the introduction of *AntMesh*, a distributed interference-aware data forwarding architecture based on smart ants, with a built in novel routing metric called Ant Routing Metric (ARM). ARM is used as the optimization criterion when determining the best path in a mesh network. The rationale behind the development of ARM was based on the observation that most existing routing metrics for mesh networks were designed to tackle only a particular problem in WMN routing. For example, metrics like ETX and ETT were targeted for single radio mesh networks. Although MIC and iAware metrics were designed for multi radio WMNs and can handle both inter- and intra-flow interference, these metrics suffer from scalability issues because of their complexity in capturing the interference and therefore are not suitable for large scale mesh deployments. Thus, our goal was to design a metric that can capture the complex interaction between inter- and intra-flow interference while being simple, robust, and scalable at the same time. The ARM metric's sole purpose was to effectively utilize the space/channel diversity typically common in infrastructure WMNs.

One interesting result of our investigation was that *AntMesh* has the capability to discover high throughput paths with less inter- and intra-flow interference while conventional wireless network routing protocols and metrics fail to do so. This conclusion was based on extensive evaluation and testing of

AntMesh under various network scenarios, i.e., in both fixed and mobile node WMNs. The obtained results showed *AntMesh*'s advantages, making it a valuable candidate to operate in MRMC mesh networks.

We then turned our attention to partially overlapping channels (POCs) for interference avoidance in the context of 802.11 b/g based MRMC-WMNs. Our basic idea was to make the whole wireless spectrum available to nodes for channel selection based on the interference each channel is experiencing. Partially overlapped channels can be used when performing channel assignment, not only to increase spatial reuse but also to increase network capacity in terms of throughput (and therefore achieving more efficient spectrum utilization).

Previously, an algorithm for channel assignment based solely on orthogonal channels had to deal with only one type of interference, i.e., co-channel interference. However, one of the major issues in designing efficient channel assignment schemes using POCs is the adjacent channel interference, which is the interference experienced by two neighboring nodes configured on partially overlapping adjacent channels. Similarly, nodes equipped with multiple radios in a typical MRMC-WMNs also suffer from self-interference which means that parallel communication cannot be achieved among multiple radios installed on a node, unless they are configured to orthogonal channels. The first step in developing a mechanism which can take advantage of the partial overlap is to build a model that captures this overlap in a quantitative fashion. Therefore, we proposed the partial overlapped interference graph (POIG) model, which is a weighted undirected interference graph (thus it can model an 802.11b based MRMC-WMN employing partially overlapped channels). The basic idea of our proposed POIG model is to assign weights to the edges of an interference graph in such a way that they represent the amount of interference (adjacent or co-channel) that exists between two radios configured on POCs belonging to neighboring nodes. Our custom designed weight function captures both these interference effects of POC and the corresponding physical distance between mesh nodes operating on those adjacent channels.

After proposing the POIG graph, we designed *i*-POCA; a channel assignment scheme which enables the use of smart ants for assigning orthogonal and partially overlapped channels (non-orthogonal) to radios in order to minimize total network interference and also to fully utilize the available 802.11b/g spectrum in a mesh network. We presented two versions of our *i*-POCA algorithm, i.e., centralized (*i*-POCA-C) and distributed (*i*-POCA-D). We compared our proposed POC-based channel assignment algorithm with other existing partially overlapped channel assignment techniques and found that our scheme performed well in terms of throughput and channel utilization. This was mainly due to the efficient allocation of partially overlapped channels resulting in more parallel communication. We argued that this indirectly validates that our proposed POIG model captures the interferences quite accurately, most of the time.

Future research plans revolve around the design, development and deployment of WMNs. In particular, our objectives are not limited to designing routing and channel assignment schemes for wireless mesh networks, (although this certainly remains a challenge), but we also plan to work on other challenging problems in several existing technologies such as Wi-Fi (based on IEEE802.11), and WiMAX (based on IEEE802.16). Furthermore, we plan on deploying a real WMN testbed on which our proposed solutions can be more thoroughly and realistically tested.

REFERENCES

[1] Ian F. Akyildiz, Xudong Wang, Weilin Wang, "Wireless mesh networks: a survey", Computer Networks and ISDN Systems, v.47 n.4,p.445-487, 15 March 2005.

[2] Duarte, P.B.F., Fadlullah, Z.M, Hashimoto, K, Kato, N., "Partially Overlapped Channel Assignment on Wireless Mesh Network Backbone," in *Proc. IEEE GLOBECOM 2010*, Miami, FL, Dec. 2010.

[3] Perkins and Royer, "Ad hoc On-demand Distance Vector Routing." In IEEE Workshop on Mobile Computing and Systems and Applications, 1999.

[4] Philippe Jacquet, Paul Muhlethaler, Amir Qayyum, Anis Laouiti, Laurent Viennot and Thomas Heide Clausen "Optimized Link-State Routing Protocol", draft-ieft-olsr-04.txt - work in progress, March 2001.

[5] D.S.J. Couto, D. Aguayo, J. Bicket, and R. Morris, "A high-throughput path metric for multi-hop wireless routing," Proc. of the ACM MOBICOM, 2003, vol. 1, pp. 134-146, Sep. 2003.

[6] R. Draves, J. Padhye, and B. Zill, "Routing in Multi-Radio, Multi-Hop Wireless Mesh Networks," Proc. of the ACM MOBICOM, 2004, pp. 114-128.

[7] Y. Yang, J. Wang, and R. Kravets, "Interference-aware load balancing for multihop wireless networks," Tech. Rep. UIUC DCS-R-2005-2665, Dept. of Computer Science, UIUC, 2005.

[8] A. P. Subramanian, M. M. Buddhikot, and S. C. Miller, "Interference aware routing in multi-radio wireless mesh networks," IEEE Workshop on Wireless Mesh Networks (WiMesh), Sept. 2006, pp. 55–63.

[9] M. Dorigo and G. Di Caro, "The ant colony optimization meta-heuristic," in D. Corne, M.

Dorigo, and F. Glover, editors, New Ideas in Optimiziation, pages 11-32. MacGraw Hill, 1999.

[10] M. Dorigo, G. Di Caro, and L.M. Gambardella, “Ant algorithms for discrete optimization,” Arfitcial Life, 5(2): 137-172, 1999.

[11] G. Di Caro and M. Dorigo, “AntNet: distributed stigmergetic control for communications networks,” Journal of Artificial Intelligence Research (JAIR), 1998.

[12] R. Schoonderwoerd, O. Holland, J. Bruten, and L. Rothkrantz, “Ants for load balancing in telecommunication networks,” HP Lab., Bristol, U.K., Tech. Rep. HPL-96-35, 1996.

[13] G. Di Caro and M Dorigo, “Two ant colony algorithms for best-effort routing in datagram networks,” Proc. of the 10th IASTED Int., 1998, pp. 541–546.

[14] S. Kamali, J. Opatrny, “POSANT: a position based ant colony routing algorithm for mobile ad-hoc networks,” Elsevier Wireless and Mobile Communications, pp 21–21, 2007.

[15] M. Gunes, U. Sorges, I. Bouazizi, “ARA-the ant-colony based routing algorithm for MANETs,” Proc. International Conference on Parallel Processing, 2002.

[16] G. Di Caro, D. Gambardella, “AntHocNet: an adaptive nature-inspired algorithm for routing in mobile ad hoc networks,” European Transactions on Telecommunications, 2005, vol. 16, no. 5, pp. 443–455, 2005.

[17] X. Wang, F. Li, S. Ishihara, T. Mizuno, A multicast routing algorithm based on mobile multicast agents in ad-hoc networks, Special Issue on Internet Technology, IEICE Transactions on Communications E84-B (8) (2001).

[18] Y.J. Suh, H.S. Shin, D.H. Kwon, An efficient multicast routing protocol in wireless mobile networks, Wireless Networks 7 (2001) 443–453.

[19] R.R. Choudhury, S. Bandyopadhyay, K. Paul, A distributed mechanism for topology discovery in ad hoc wireless networks using mobile agents, in: Proceedings First Annual Workshop on Mobile Ad Hoc Networking Computing, MobiHOC, Boston, MA, USA, August 2000.

[20] R.R. Choudhury, K. Paul, S. Bandyopadhyay, MARP: a multi-agent routing protocol for mobile wireless ad hoc networks, Autonomous Agents and Multi-Agent Systems 8 (2004) 47–68

[21] L. Rosati, M. Berioli, G. Reali, “On ant routing algorithms in ad hoc networks with critical connectivity,” Elsevier Ad Hoc networks, pp 827–859, 2008.

[22] F. Correia, T. Vazao, “Simple ant routing algorithm strategies for a (Multipurpose) MANET model,” Elsevier Ad Hoc networks, pp 810–823, 2010.

[23] Fawaz Bokhari, Gergely Zaruba, “AntMesh: An Efficient Data Forwarding Scheme for Load Balancing in Multi-Radio Infrastructure Mesh Networks” *Proceedings of 4th IEEE International Workshop on Enabling Technologies and Standards for Wireless Mesh Networking (MeshTech’10) held in conjunction with IEEE MASS, San Francisco, California,* November 8-12, 2010.

[24] Fawaz Bokhari, Gergely Zaruba, “AMIRA: Interference-aware Routing Using Ant Colony Optimization in Wireless Mesh Networks” *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC2009)*, *Budapest, Hungary*, April 5-8, 2009.

[25] Fawaz Bokhari, Gergely Zaruba, “On the Use of Smart Ants for Efficient Routing in Wireless Mesh Networks", international Journal of Wireless & Mobile Networks (IJWMN). In Press.

[26] Fawaz Bokhari, “Channel Assignment and Routing in Multiradio Wireless Mesh Networks using Smart Ants” (extended abstract/poster), *Proceedings of 9th IEEE International*

Conference on Pervasive Computing and Communications (PerCom 2011), in Seattle, Washington, March 21-25, 2011

[27] Fawaz Bokhari**,** "Artificial Ants for Efficient Data Forwarding in Wireless Mesh Networks" (extended abstract/poster), *Proceedings of the IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM 2010), Montreal, Canada, June 14-17, 2010*

[28] M. Dorigo & L.M. Gambardella, "Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem," in IEEE Transactions on Evolutionary Computation, 1(1):53-66, 1997

[29] L.M. Gambardella, M. Dorigo, Solving symmetric and asymmetric tsps by ant colonies, in: Proceedings IEEE International Conference on Evolutionary Computation, Nagoya, Japan, May 1996, pp. 622–627.

[30] I. A. Wagner, M. Lindenbaum, A. M. Bruckstein, "ANTS: Agents, Networks, Trees, and Subgraphs," Future Generation Comp. Systems journal, North Holland June 2000

[31] M. Dorigo and St¨utzle, "The ant colony optimization meta-heuristic: algorithms, applications, and advances," F. Glover, editors, metaheuristic handbook, 2001

[32] B. Baran and R. Sosa, "A new approach for AntNet routing," Proceedings of the 9th Int. Conf. Computer Communications Networks, Las Vegas, NV, 2000.

[33] I. Katzela and M. Naghshineh, "Channel assignment schemes for cellular mobile telecommunication systems," IEEE Personal Communications, vol. 3, pp. 10–31, 1996.

[34] IEEE Std 802.11-2007, "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," LAN/MAN Standards Committee, New York, NY, USA, pp. C1–

1184, June 2007. [Online]. Available: http://dx.doi.org/10.1109/IEEESTD.2007.373646

[35] A. Mishra, V. Shrivastava, S. Banerjee, and W. Arbaugh, “Partially overlapped channels not considered harmful,” in *ACM SIGMetrics/Performance*, 2006, pp. 63–74.

[36] Yong Cui, Wei Li, Xiuzhen Cheng, "Partially overlapping channel assignment based on “node orthogonality” for 802.11 wireless networks," in *Proc. IEEE INFOCOM 2011*, Shanghai, China, Apr. 2011.

[37] K. Shih, C. Chang, D. Deng, and H. Chen, "Improving Channel Utilization by Exploiting Partially OverlappingChannels in Wireless Ad Hoc Networks," in *Proc. IEEE GLOBECOM 2010*, *pp. 1-5,* Miami, FL, Dec. 2010.

[38] Y. Liu, R. Venkatesan and C. Li, “Channel assignment exploiting partially overlapping channels for wireless mesh networks,” in *Proc. of IEEE GLOBECOM'09*, pp. 1-5, Honolulu, USA, Nov. 2009.

[39] Yuting Liu, R. Venkatesan, and Cheng Li. "Load-Aware Channel Assignment Exploiting Partially Overlapping Channels for Wireless Mesh Networks". In *Proc. IEEE GLOBECOM10, Florida, USA*, Dec 2010.

[40] V. Bukkapatanam, A. Franklin, and C. Murthy, “Using partially overlapped channels for end-to-end flow allocation and channel assignment in wireless mesh networks,” in *Proc. IEEE ICC ’09.*, Dresden, Germany, June 2009.

[41] D. Wang, P. Lv, Y. Chen and M. Xu, “POCAM: Partially overlapped channel assignment in multi-radio multi-channel wireless mesh networks,” in *Proc. IEEE ISCIT ’11.*, Hangzhou, China, October 2011.

[42] Anand Prabhu Subramanian, Himanshu Gupta, Samir R. Das, "Minimum Interference Channel Assignment in Multi-Radio Wireless Mesh Networks", IEEE Transactions on Mobile Computing,Vol. 7,No. 11, 2008

[43] A. Rad, Vincent W.S. Wong, ”Partially overlapped channel assignmentfor Multi-Channel wireless mesh network,” *in Proc. ICC*, 2007, pp. 3770-2775.

[44] M. Hoque, X. Hong, and F. Afroz, “Multiple radio channel assignment utilizing partially overlapped channels,” in *Proc. IEEE GLOBECOM 2009*, Honolulu, HI, Nov. 2009.

[45] K. N. Ramachandran, E. M. Belding, K. C. Almeroth, and M. M. Buddhikot. "Interference-aware channel assignment inmulti-radio wireless mesh networks". In *Infocom,* 2006.

[46] Christopher L. Barrett, Gabriel Istrate, V. S. Anil Kumar, Madhav V. Marathe, Shripad Thite, Sunil Thulasidasan, "Strong Edge Coloring for Channel Assignment in Wireless Radio Networks", in *Proc IEEE PerCom Workshops*, 2006.

[47] Das, A.K.; Vijayakumar, R.; Roy, S., ”WLC30-4: Static Channel Assignment in Multi-radio Multi-Channel 802.11 Wireless Mesh Networks: Issues, Metrics and Algorithms,” in *Proc IEEE GLOBECOM*, Nov. 27 - Dec. 1 2006.

[48] Y. Ding, Y. Huang, G. Zeng, and L. Xiao, “Channel assignment with partially overlapping channels in wireless mesh networks,” in *WICON*, 2008.

[49] “IEEE Std 802.11 Wireless LAN Medium Access Control (MAC) and Phusical Layer (PHY) Specifications,” 1999.

[50] P. Gopalakrishnan, P Spasojevic, L. Greenstein, I.A. Seskar, “Method for predicting the throughput characteristics of rate-adaptive wireless LANs,” VTC2004-Fall. 2004, pp. 4528-4532

[51] Fall and Varadhan, “NS notes and documentation,” in The VINT Project, UC berkely, LBL, USC/ISI, and Xerox PARC, 1997.

[52] A. Raniwala and T.-C. Chiueh, “Architecture and algorithms for an IEEE 802.11-based multi-channel wireless mesh network,” Proc. of the IEEE INFOCOM, 2005, pp. 2223-34.

[53] Johnson DB, Maltz DA. *Mobile Computing,* Chapter Dynamic Source Routing in Ad Hoc Wireless Networks. Kluwer: Norwell, MA, 1996; 153-181

[54] Y. Yang, J. Wang, and R. Kravets, "Designing routing metrics for mesh networks," Proc. of IEEE Workshop on Wireless Mesh Networks (WiMesh), June 2005.

[55] Y. Shi, Y. T. Hou, J. Liu, and S. Kompella, "How to correctly use the protocol interference model for multi-hop wireless networks," Proceedings of MobiHoc, USA, May 2009.

[56] Y. Yang and R. Kravets, "Contention-Aware Admission Control for Ad Hoc Networks," IEEE Trans. Mobile Computing, vol. 4, no. 1, pp. 363-377, July/Aug. 2005.

[57] AntNet implementation for wired networks, http://antalgorithm.googlecode.com/files/antnet.tar.gz

[58] Mishra A, Rozner E, Banerjee S, Arbaugh W (2005) Using partially overlapped channels in wireless meshes. In: Wimesh, Santa Clara, 26 September 2005.

[59] A. Mishra, E. Rozner, S. Banerjee, and W. Arbaugh, "Exploiting partially overlapping channels in wireless networks: Turning a peril into an advantage," in *ACM SIGCOMM*, 2005, pp. 29–29.

[60] Freifunk firmware, http://ff-firmware.sourceforge.net/

[61] T.S. Rappaport, *Wireless Commnications: Principle and Practice, Second Edition,* Prentice Hall, 2002

[62] B. A. Tesman. *Set T-coloring.* Congrussus Numerantium, 77:229-242, 1990

[63] Y. Liu, R. Venkatesan and C. Li, "Channel assignment exploiting partially overlapping channels for wireless mesh networks," in *Proc. ofIEEE GLOBECOM'09*, pp. 1-5, Honolulu, USA, Nov. 2009.

[64] Fawaz Bokhari, Gergely Zaruba, "Partially Overlapping Channel Assignments in Wireless

Mesh Networks ", chapter abstract accepted in the book: *Wireless Mesh Networks* edited by Dr. Andrey Krendzel - ISBN 979-953-307-833-4.

[65] *Submitted*. Fawaz Bokhari, Gergely Zaruba, "*i*-POCA: Interference-aware Partially Overlapping Channel Assignment in 802.11-based Meshes" *Proceedings of the IEEE Globecom 2012 to be held in Anaheim, California,* December 3-7, 2012.

BIOGRAPHICAL INFORMATION

Fawaz Saleem Bokhari received his Bachelor degree in Computer Science from University of Central Punjab (UCP), Lahore, Pakistan in 2001. He worked in a private software firm for some time and also taught in various institutes in Pakistan for more than 2 years. He then joined LUMS, Lahore for Masters in Computer Science and graduated in 2007. During his studies in LUMS, he applied for Fulbright Scholarship and was awarded the scholarship in 2007 for PhD studies in US at The University of Texas at Arlington (UTA), where he received his PhD degree in Computer Science in 2012. His research interests involve broadband wireless networking, network and MAC layer designs, routing and channel assignment schemes, wireless multimedia communications, 4G wireless technologies (LTE and WiMAX), network monitoring and deployment. He also had several short term appointments of teaching assistantship for graduate and under graduate levels.