CHANNEL SWITCHING CONTROL POLICY FOR WIRELESS MESH NETWORKS

A Dissertation
Submitted to
the Temple University Graduate Board

in Partial Fulfillment
of the Requirements for the Degree of
MASTER OF SCIENCE

by
Xiaoguang Li
Diploma Date (August 2012)

Thesis Approvals:
Jie Wu, Thesis Advisor, Department of Computer and Information Sciences
Committee Member: Chiu C. Tan and Shan Lin, Department of Computer and Information Sciences

ABSTRACT

Author: Xiaoguang Li

Title: Channel Switching Control Policy for Wireless Mesh Networks

Institution: Temple University

Dissertation Advisor: Jie Wu

Degree: Master of Science

Year: 2012

Dynamic channel assignment algorithms allow wireless nodes to switch channels when their traffic loads exceed certain thresholds. These thresholds represent estimations of their throughput capacities. Unfortunately, the threshold estimation may not be accurate due to *co-channel interference* (CCI) and *adjacent-channel interference* (ACI), especially with high traffic loads in dense networks. When the link capacity is over-estimated, these channel assignment algorithms are not effective. This is because channel switch is not triggered even with overloaded data traffic and the link quality decreases significantly as the channel is overloaded. When the link capacity is under-estimated, the link is under utilized. Moreover, when link traffic load increases from time to time, channel switch occurs frequently. Such frequent channel switches increase latency and degrade throughput, and can even cause network wide channel oscillations. In this paper, we propose a novel threshold-based control system, called *balanced control system* (BCS). The proposed threshold-based control policy consist of deciding, according to the real time traffic load and interference, *whether to switch to another channel*, *which channel should be switched to* and *how to perform the switch*. Our control model is based on a fuzzy logic control. The threshold which assists to make the channel switch decisions, could be deduced dynamically according to the real-time traffic of each node. We also design a novel dynamic channel assignment scheme, which is used for the selection of the new channel. The channel switch scheduler is provided to perform channel-switch processing for sender and receiver over enhanced routing protocols. We implement our system in NS2, and the simulation results show that with our proposed system, the performance improves by 12.3%-72.8% in throughput and reduces 23.2%-52.3% in latency.

ACKNOWLEDGEMENTS

I am grateful to my advisor Dr. Jie Wu for the continuous support of my study and research, for his patience, motivation, enthusiasm, and immense knowledge, for generously giving me invaluable directions and suggestions for my paper revisions and submissions. I would like to thank Dr. Justin Shi for providing invaluable supports and help to my research work.

Special thanks to my committee, Dr. Shan Lin and Dr. C. Chiu Tan for providing indispensable feedback, guidance and helpful suggestions. Thanks must be given to Dr. Xiaojiang Du for their support and help. Thanks also go to my colleagues in the department of Computer and Information Science at Temple University.

Finally, I am deeply in debt to my parents, Min Wu and Changfu Li. This work would be impossible without their support and sacrifice.

I thank everyone for their kindness.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 Wireless Mesh Networks

Wireless mesh networks (WMNs) are gaining significant momentum as an inexpensive way to provide last-mile broadband internet access.

WMNs is also defined as two-tier mesh networks, which consists of a backhaul tier (mesh node to mesh node) and an access tier (mesh node to access node). Different from the traditional wireless networks, the mesh node forward data to and from wired entry points (gateways). Clients or access nodes throughout the coverage area then connect to local mesh nodes to receive connectivity back to the wired network.

Recent years, city-wide two-tier mesh networks are becoming attractive for metropolitan areas of all sizes and thereby, reshaping the traditional roles of municipal access

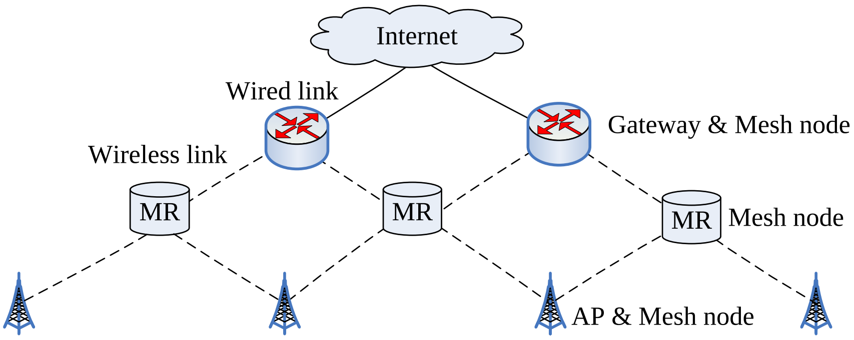


FIGURE 1.1: The architecture of M2WMNs

networks. Many cities have already deployed mesh networks to assist public service and safety personnel, e.g., New Orleans, San Mateo, and Chaska. Other cities, such as Philadelphia, Houston, and San Francisco, plan city-wide two-tier mesh deployments to additionally provide public broadband Internet access.

Almost all the current WMN deployments and proposals adopt the multi-channel multi-interface architecture, where each node is equipped with multiple radios and can use multiple non-overlapping channels. The WMNs with such architecture are generally called the multi-channel multi-interface WMNs, and for simplicity, we define them as M2WMNs. Figure 1.1 illustrates the M2WMNs architecture with an example network, where each node can use four non-overlapping channels, with four interfaces.

1.2 Motivation

Recent studies [1, 2] have shown that equipping each node with multiple interfaces can improve the capacity of WMNs. By equipping interfaces in different channels, a node can communicate with multiple nodes simultaneously. Each channel allows multiple data flow exchanges in both directions, as long as the traffic load does not exceed the link's throughput capacity, i.e., the maximum amount of traffic that the link can carry.

1.2.1 Problem in single channel mesh network

In the initial design of WMNs, the traditional wireless network paradigm is followed, where only one radio (i.e., wireless interface card) is equipped at each node and all nodes share a single channel. In [3], it has been demonstrated that when n identical randomly placed nodes with bandwidth W form a single-channel network, the throughput obtained per node is $\theta(W/n \log n)$. In a single channel multi-hop network, interference can occur not only in the nearby hops but also the nearby flows with a single flow. Figure 1.2 illustrates these two kinds of interference. Link (A to B) have interference with link (B to C). And the flow 1 and flow 2 also interferers with each other.

1.2.2 Static channel assignment and dynamic channel assignment

Many previous research in WMNs usually assume static channel capacity. This simplified assumption does not hold in reality. The throughput capacities in real systems can vary dramatically with time and locations due to fading, shadowing,

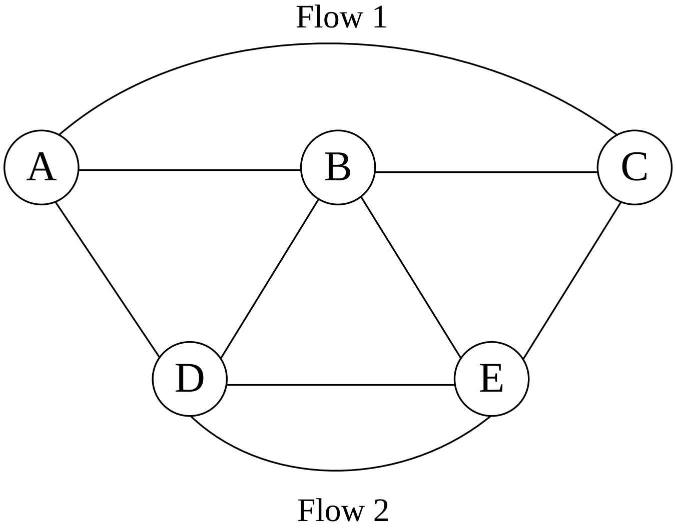


Figure 1.2: The internal interference in single channel mesh network

and interference. As a result, protocols based on static channel capacity may not work well in real systems as channel throughput capacity (or simply link capacity) can be either over-estimated or under-estimated.

Static channel assignment algorithms that switch channels periodically or permanently [4, 5], have been shown to achieve great performance with stable network traffic. However, with dynamic traffic loads, such algorithms are not effective due to the mismatch between dynamic channel throughput capacity and the real-time traffic load. To select a channel based on real-time traffic load, recent studies on dynamic channel assignment algorithms [6–8] can adaptively switch the channel on certain links in a distributed fashion. Accurate estimation of channel throughput capacity is very challenging, as it is notably influenced by both *co-channel interference* (CCI) [9] and *adjacent-channel interference* (ACI) [10], especially when the traffic load is high. When the link capacity is over-estimated, the channel

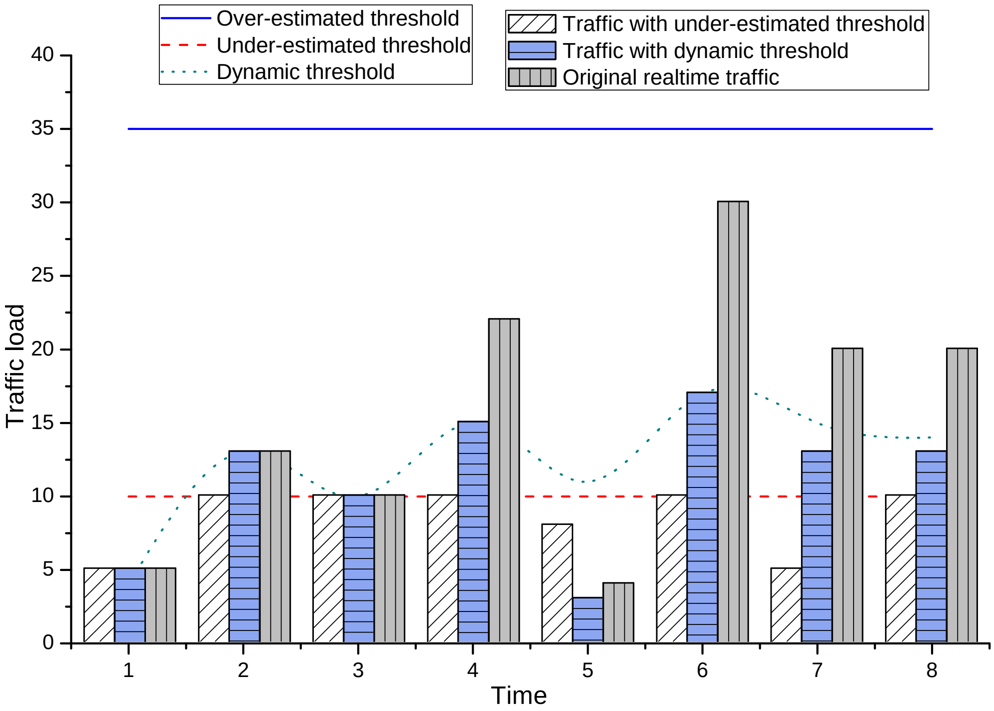


Figure 1.3: Threshold vs traffic load.

saturates and the channel quality degrades before channel switch is triggered. On the other hand, when the link capacity is under-estimated, channel is not fully utilized. Also, when the traffic load experiences temporary increases, existing algorithms tend to switch the channel frequently. Such frequent channel switches degrade the network throughput and increase latency significantly. Moreover, the newly switched links cause interference to other nearby links, introducing link capacity variation on those links and triggering even more channel switches. In the worst case, it can cause network wide channel oscillation.

An intuitive example is shown in Figure 1.4. With an over-estimated threshold, the channel switch is not triggered in all cases, even when the channel saturates and the link quality degrades. While with an under-estimated threshold, the channel capacity is not fully utilized and channel switches occur frequently (at time 2, 4, 6, 7, and 8). If we can choose the link capacity threshold adaptively, the channel is better utilized than using the under-estimated threshold in all the cases. Moreover, the overhead in channel switching is reduced. We note that existing rate adaptation protocols[11, 12] adjust transmission rate based on channel contention.

1.2.3 Ripple effect problem and thrash problem

In dynamic channel assignment strategies, we use a threshold to control the process. Normally, when the current bandwidth of the channel arrives at threshold, it will be switched to the other channel. The new channel will take over the traffic, and improve the performance.

However, interface switching brings new problems, such as switching overheads, ripple effect, and thrash problems. One overhead issue caused by interface switching is switching delay, which is not negligible at present [13]. The ripple effect problem could be described as shown in Figure 1.4.

The ripple effect problems occur when lots of nodes are working in the same channel, as shown in Figure 1.4. Suppose that there are 5 channels in the network. The channel sets $\{1, 2\}$, $\{2, 3\}$, and $\{2, 4\}$ have been assigned to the node A, B, and C respectively. Node A and B construct link (say l_1) with channel 2. Moreover, nodes B and C have the link (say l_2) with channel 2. l_1 and l_2 will share the channel 2. When one of them arrives at the threshold, all three nodes

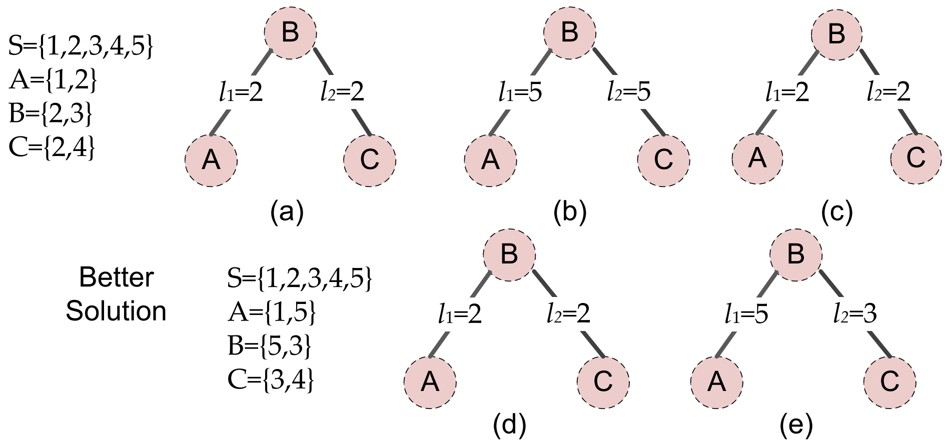


FIGURE 1.4: Ripple effect problem

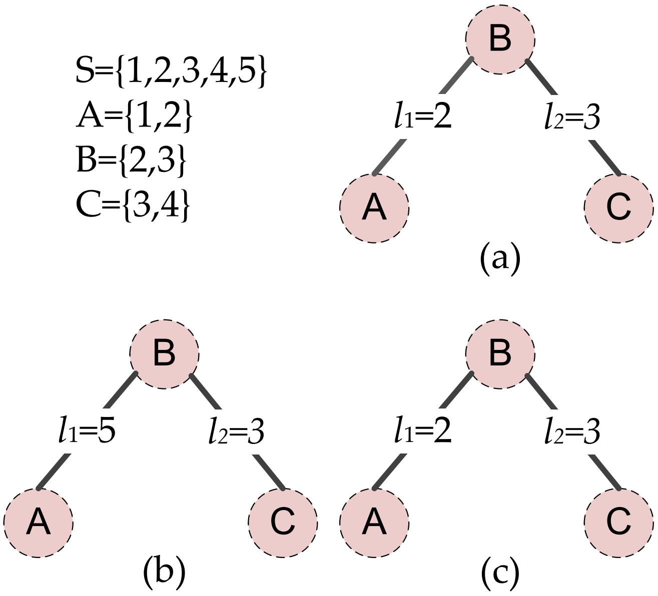


FIGURE 1.5: Thrash problem

need to switch to new channel 5 so as to maintain the connection as shown in Figure 1.4 (b). However, all of them still share the channel. It will be very easy to achieve the threshold, if they still maintain the relationship. As shown in Figure 1.4 (c), they will return to channel 2 when one of the three nodes arrives at the threshold. A better solution as shown in Figure 1.4's (d) and (e), is to calculate the length of the ripple effect and make the two links work with different channels.

The thrash problem may occur during the channel-switch process. As shown in Figure 1.5, the channel sets $\{1, 2\}$, $\{2, 3\}$, and $\{3, 4\}$ have been assigned to the node A, B, and C, respectively. We assume that a heavy traffic occurs in the link l_1, and channel 2 arrives at the threshold. Then, the link l_1 has to switch to another available channel 5 (see Figure 1.5 (b)). If the threshold is still the same and is easy to achieve, another switch will occur. As shown in Figure 1.5 (c), the channel of l_1 returns to channel 2. Then, the switches will continuously occur if the traffic is the same and the threshold is never adjusted.

In addition, node B and C have the link l_2 using channel 3. As shown in Figure 1.5, during the whole process, the l_2 keeps working in channel 3. It could be the situation that channel 3 is actually busy for the traffic, but the threshold is too large to achieve. Meanwhile, there is another channel available to use.

A reasonable threshold can control the network, efficiently. However, it is difficult to decide the exact threshold, because the network is in the thrash all the time. The purpose of the BCS is to dynamically adjust the threshold for all the links in the network.

1.2.4 Routing problems in M2WMNs

First, we must address the problems with current routing protocols in wireless mesh networks. The original routing protocols, such as AODV, DSR, etc. support multiple interfaces. However, this initial design is typically for the multi-home based protocol (SCTP etc.). This is not used for multiple interface wireless mesh networks.

The problems we find can be summarized as follows:

(1) As we can see from Figure 1.1, the “mesh node” here is with multiple interfaces of wireless connections. Moreover, the “mesh portal”is with both wireless and wired connections. All the packets have to be passed through gateways. This is different from the multi-home based protocol, which enables transparent fail-over between redundant network paths. This will lead to more complex problems in routing if we do not modify the existing routing protocols, because the wireless links are equal, but the link to the gateway is not equal.

(2) The other problem is that the ideal design of our system needs fast channel switching to solve the unstable system problem. As we know, after one node, say node i, has sent the “hello”information, the neighbors who received the “hello” information will add node i into the routing table entry. Then, they established the connection. This will remain for future use, until it expires. If node i has changed its channel, but the corresponding neighbor, say node j, doesn’t switch the new one, then, this means they cannot connect to each other after the switch. However, the routing table entry remains in both nodes. When they need the route, they still can find the route according to the routing table entry. However, the packets cannot be forwarded because the actual connection is broken. It will take more

time to establish and find a new route. The time of transferring the packets has to be delayed.

Due to the problems we explained above, we propose modifications to the current routing protocol so as to enable the discovery of channel information from a source to a destination. The proposed enhancement can help the node find the correct route, even if the nodes have switched channels.

1.3 Contribution

Our work focuses on channel switching, which is an orthogonal issue to rate adaptation. Our goal is to dynamically find a channel capacity estimation that fully utilizes link capacity and reduces unnecessary channel switching. In this project, we propose a threshold based control system, called *balanced control system* (BC-S). Unlike existing approaches that use static threshold estimation, our design features a fuzzy control loop to monitor the dynamic traffic load during runtime and adaptively adjust the channel switching threshold. This threshold serves as the bound for traffic load on this channel. The proposed threshold-based control solution consists of deciding, according to the real time traffic load and interference, *whether to switch to another channel*, *which channel should be switched to* and *how to perform the switch*. Our threshold control model is based on a fuzzy logic control. The threshold which assists to make the channel switch decisions could be deduced dynamically according to the real-time traffic of each node. Our control based design allows the dynamic threshold to approximate the runtime capacity accurately, therefore improving the channel utilization and reducing unnecessary channel switches. The contributions of our work are demonstrated as follows:

1. We propose a threshold-based balanced control system (BCS) in which each link in the network finds its own threshold according to the real-time traffic. We also offer a traffic metric model for our BCS. The metric model estimates the traffic load integrated with CCI and ACI problems.

2. We present a dynamic channel assignment scheme for the selection of the new channel. This algorithm fully utilizes variable channel capacities with reduced channel switching overhead. We also provide a channel switch scheduler to perform channel-switch processing for sender and receiver over enhanced routing protocols.

3. We implement our system in NS2. From our simulation results, we demonstrate that our proposed scheme outperforms the current techniques. Although channel switch algorithms for wired networks have been studied and practiced in industry [14]. In wired networks, static channel capacity models are highly accurate. Whereas in wireless networks, fading and interference (ACI and CCI) can cause significant channel throughput variations, resulting in frequent channel switch.

1.4 Organization of the Dissertation

The remainder of the dissertation is organized as shown in Figure 1.6. Chapter 2 introduces the recent related works. This part includes the channel assignment on M2WMNs, the problem for the common default channel, and the study on channel switching problem. Chapter 3 gives the preliminaries of this work. We will cover the several related concepts, such as AODV, Fisheye routing and KN-CA algorithm. We will introduce the network model and several design issues in Chapter 4. We will also offer the assumptions and an overview of our proposed system. In Chapter 5, we will cover the system design. Our system is constructed by traffic metric model, fuzzy control model, dynamic channel assignment, enhanced routing protocols and channel switch scheme. Chapter 6 presents the numeric analysis on the stability. An example and a more detailed discussion are provided. The performance evaluation are presented in Chapter 7. We conducted several simulations on the comparison of routing protocols, different interfaces and channels, and network size. This dissertation concludes in Chapter 8. In addition to the conclusion, we will illustrate the future work.

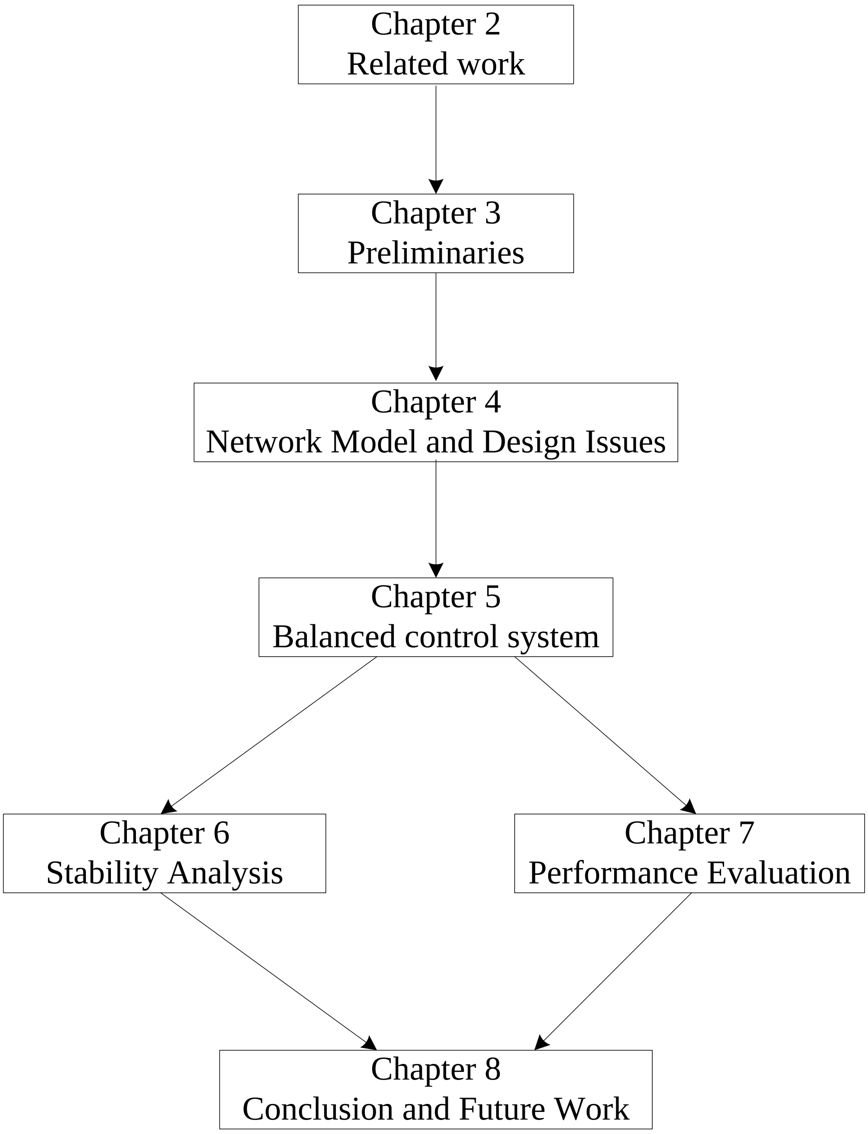


Figure 1.6: The organization of the remainder of the dissertation

CHAPTER 2

RELATED WORK

In this chapter, we will give a overview of related work. We will put it into several categories: channel assignment in multi-channel multi-interface WMNs and control channel design in M2WMNs.

2.1 Channel assignment in WMNs

Extensive studies have been done to utilize multiple channels in WMNs. Some works focus on changing MAC protocols [15, 16]. In [15], a busy tone is used to show the channel reserving information. However, this MAC protocol cannot be applied directly because it is not compatible with commodity hardware. Protocols of [16] seek to use one interface to exploit multiple channels to improve network performance.

Some works need to change MAC protocols [15, 16]. In [15], a busy tone is used to show the channel reserving information. However, those MAC protocols cannot be applied directly because they are not compatible with commodity hardware. Protocols of [16] seek to use one interface to exploit multiple channels.

Yunxia et al. [17] presented a hybrid channel assignment protocol to improve the capacity and flexibility of WMNs. The proposed protocol jointly solves the interface assignment, communication coordination, and dependency problems.

Raniwala et al. [5, 18] devised routing and interface assignment algorithms for mesh networks. The protocols are designed for use in mesh networks, where traffic is directed toward specific gateway nodes. Raniwala's protocol assumes traffic load between all nodes is already known. Moreover, with the load information, interface assignments and route computations are intelligently computed.

So and Vaidya et al. [19] propose an architecture for multi-channel networks that uses a single interface. Each node has a default channel for receiving data. A node with a packet to transmit has to switch to the channel of the receiver before transmitting data. However, the proposal does not consider the effect of channel switching. The packet has to wait for the delay of the transmission.

2.2 Control channel in WMNs

To avoid ripple effect and minimize the switching overheads, most protocols [4, 5, 18] adjust channels after a relatively long period. These protocols do not need a control channel, whereas they need complex coordination technologies in communications.

A common default channel is introduced in [4, 8, 20–22] to handle the network partition caused by the dynamic channel assignment, and to facilitate channel negotiation for data communications. To assign channels to the interfaces, [4] presents a localized greedy heuristic based on the interference cost function defined for pairs of channels.

The authors in [20, 21] consider the mesh networks with main traffic flowing to and from a gateway, which is also in charge of the channel computation. In their channel assignment, to a non-default radio, nodes closer to the gateway and/or bearing higher traffic load receive a better quality channel.

In DCA [22], the default channel is used as a control channel. For each node, one of the radios stays on the control channel to exchange control messages, and other radios dynamically switch to the data channels for transmission. In this case, the utilization of the control channel could be small, even though the data channels can be fully utilized.

2.3 Link capacity estimation

When these models are applied in real systems, the impacts of dynamic environment and interference can cause link capacity estimation to be inaccurate, result in unstable performance

Wu et al. [14] describe the design, implementation and evaluation of WMN system. That system supports both dynamic channel switching and load-balancing/fault-tolerant routing, and successfully runs on low-cost commodity IEEE 802.11-based access points. Azz et al. [23] proposed a new adaptive algorithm, called *Enhance and Explore* (E&E). It maximizes the utility of the network without requiring any explicit characterization of the capacity region. However, the threshold dose not feature the channel switching problem. Kim and Shin [24] proposed an autonomous network reconfiguration system (ARS) that enabled a multi-radio WMN to autonomously recover from local link failures.

Ding et al. [25] proposed a hybrid multi-channel multi-radio wireless mesh networking architecture, where each mesh node has both static and dynamic interfaces. In [26], the authors studied a new framework to mitigate the effects of interference in 802.11b/g mesh networks by fully exploiting the spectrum resource. This method utilized both non-overlapping channels and partially overlapping channel-s. Chieochan and Hossain [27] considered opportunistic listening into the joint problem of channel assignment, network coding, and scheduling in multi-radio WMNs.

In this thesis, our attempt is to study fuzzy control and integrate this model into our system to dynamically find the threshold according to the real-time traffic. The aim of our system is to incorporate expert human knowledge in the control algorithm. In this sense, a fuzzy controller may be viewed as a real-time expert system to balance the unstable network. The only work related with fuzzy logic is discussed in [28]. However, that work is based on the QoS considerations for multimedia transmission.

CHAPTER 3

PRELIMINARIES

In this chapter, we will first introduce the current routing protocols: AODV [29]and FSR [30] used in this work. Then, we will offer a brief review of the channel assignment algorithms.

3.1 AODV and FSR routing protocols

In this section, we will give an overview of current routing protocols: AODV and Fisheye routing. The two routing protocols are used in this work. We will also offer the enhanced designs in the following section.

3.1.1 AODV routing protocols

The Ad hoc On Demand Distance Vector (AODV) routing algorithm [29] is a routing protocol designed for ad hoc mobile networks. AODV is an on demand algorithm, meaning that it builds routes between nodes only as desired by source nodes. It maintains these routes as long as they are needed by the sources. AODV uses sequence numbers to ensure the freshness of routes. It is loop-free, self-starting, and scales to large numbers of mobile nodes.

AODV builds routes using a route request (RREQ)/route reply (RREP) query cycle. When a source node desires a route to a destination for which it does not already have a route, it broadcasts a RREQ packet across the network. Nodes receiving this packet update their information for the source node and set up backwards pointers to the source node in the route tables. A node receiving the RREQ may send a RREP if it is either the destination or if it has a route to the destination with corresponding sequence number greater than or equal to that contained in the RREQ. If this is the case, it unicasts a RREP back to the source. Otherwise, it rebroadcasts the RREQ. Nodes keep track of the RREQ's source IP address and broadcast ID. If they receive a RREQ which they have already processed, they discard the RREQ and do not forward it. As the RREP propagates back to the source, nodes set up forward pointers to the destination. Once the source node receives the RREP, it may begin to forward data packets to the destination. If the source later receives a RREP containing a greater sequence number or contains the same sequence number with a smaller hop count, it may update its routing information for that destination and begin using the better route. As long as the route remains active, it will continue to be maintained. A route is considered active as long as there are data packets periodically traveling from the source to the destination along that path. Once the source stops sending data packets, the links will time out and eventually will be deleted from the intermediate node routing tables. If a link break occurs while the route is active, the node upstream of the break propagates a route error (RERR) message to the source node to inform it of the now unreachable destinations. After receiving the RERR, if the source node still desires the route, it can reinstate route discovery.

AODV combines the use of destination sequence numbers in DSDV with an on-demand route discovery technique. When node 1's neighbors receive the RREQ message they have two choices; if they know a route to the destination or if they are the destination they can send a RREP message back to node 1, otherwise they will rebroadcast the RREQ to their set of neighbors. The message keeps getting rebroadcast until its lifespan is up. If node 1 does not receive a reply in a set amount of time, it will rebroadcast the request except this time the RREQ message will have a longer lifespan and a new ID number. All of the nodes use the sequence number in the RREQ to insure that they do not rebroadcast a RREQ. In Figure 3.1, node 2 has a route to node 3 and replies to the RREQ by sending out a RREP. Node 4 on the other hand does not have a route to node 3 so it rebroadcasts the RREQ .

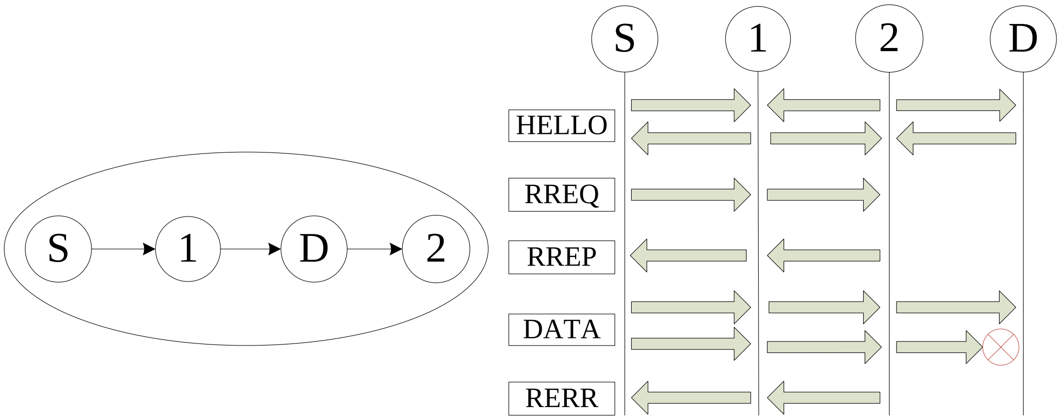


FIGURE 3.1: AODV Routing protocol

3.1.2 FSR routing protocols

FSR routing [30] makes routing decisions using a table-driven routing mechanism similar to link state. FSR introduces the notion of multi-level fisheye scope to reduce routing update overhead in large networks. Nodes exchange link state entries with their neighbors with a frequency which depends on distance to destination. There are three parts of routing process.

(1) **Neighbor discovery**: This method is used for establishing and maintaining neighbor relationships. Neighbors can meet each other simply by transmitting a special packet (a HELLO packet) over the broadcast medium. HELLO packets are periodically broadcasted. The nodes within the transmission range will hear these special packets and record them as neighbors.

(2) **Information Dissemination**: It is responsible for disseminating Link State Packets(LSP), which contain neighbor link information, to other nodes in the network. The main functions are to handle the LSP integrity and updating interval.

In fisheye routing, the scope is defined in terms of the nodes that can be reached in a certain number of hops. The center node has most accurate information about all nodes in the first circle, and becomes less accurate with each outer circle.

(3) **Route Computation**: This function is used to compute routes to each destination using the information of the LSPs. Once the router has a database of LSPs, it computes the routes based on the Djikstra's algorithm which computes all shortest paths from a single vertex. The link metric used for path cost is the hop count.

3.2 Key node channel assignment

Key node channel assignment (KN-CA) was proposed in [31]. By using this strategy, we do not need the prior knowledge of the offered traffic load. In our algorithm, we select the node i which may cause many interferences due to the existing of its neighbors and radios. If the same number of interference radios or the same neighbors have been found in node j, the proposed algorithm recursively calculate the sum of the interference from its neighbors. We present the algorithm in the following:

Algorithm 1: Considering a wireless network with V nodes and L links, there are C frequency channels in the system. Each node $u \in V$ has $I(u)$ network interface cards and its own level list. This list contains the information of the relationship

Algorithm 1 KN-CA

```
Input:
$N$: number of nodes in the network
$N'$: number of nodes which have unassigned interfaces
$list$: the current set of the nodes with most neighbors
$nMax$: the current maximum number of neighbors
$step$: the number of the recursive times
$size(list)$: the number of nodes in the List
$interface(v')$: the number of interfaces for node $v'$
Scheduling:
Let $v = \{v|v \in N\}$;
Initialize subset $N'$; Let $v' = \{v'|v' \in N', N' = N\}$; $level(v') \leftarrow 0$;$step \leftarrow 0$;
while Not all node visited $\{N'\}$ do
  while Size of list changed do
    $size(list) \leftarrow size(N')$;
    if $size(list) > 1$ and the list is different from the previous one then
      Increase the step;
      $n_{current} \leftarrow \min(getNeighbors(v', step))$;
      $n_{current} \leftarrow \min(n_{current}, interface(v'))$;
      if $n_{current} > nMax$ then
        $list$ is set to be $NULL$; $v'$ is added to $list$;
      else if $(ncurrent = nMax)$ then
        $v'$ is added to the list;
      else
        Continue;
  while Not all edge assigned: $v'' = \{v''|v' \in list\}$ do
    Select channels with minimum interference for $v''$;
    Remove $v''$ from $N'$;
```

with other nodes and current node. The node, say i, is the current node. Then we set $level(i) = 0$ for node i, and $level(i) = 1$ for its neighbors. The $level(i)$ here means the hops from the current node i. If node j is one of node i's neighbors, then the level of the node j's neighbors is set to 2. We firstly select a set of nodes which have the most neighbors. If there is only one node then we arrange channels for this node. If there are more than one node, we recursively calculate the neighbors of the node in level 1. Each time, the number of the set will be decreased. At last, we will get a set of nodes which have the most neighbors. The initial assignment typically tries to find the crowded nodes, where more interference occurs due to connections of each other. The edges of the nodes will be visited and the selected channels will be arranged for the edges.

The node, say node i, has been selected to arrange the channels. The minimum interference is calculated by the frequency. As we all know, the more difference between the frequencies can result in the lower internal interference. The node, say node j is in node i's sense range. Then, the channels of node j's are added to the interference channel list. All of the available channels of node i will be calculated with the interference channel list, say C'. The selected channel should satisfy the equation below:

$$diff = \min\{c - c'\} \tag{3.1}$$

where $c' \in C'$ and $c \in C$. The channel assignment is for the basic network setup with the traffic unknown. We only consider the interference involved in this algorithm. However, this is not enough for the real-time network. The ideal channel assignment should be the method according to the real-time network. It is possible to let make the node switch channel from on to another. If the current bandwidth has been occupied too much, i.e. achieved 80%, it is better to switch another channel which is suitable.

Theorem 3.1. *The order of the selected nodes from Algorithm 1 is the order of the nodes with the most interference.*

Proof. In [26], the authors said interference range is equal to twice the communication range. Then, we consider communication range first. Since the two nodes are in the communication range, more interference occurs. (1) In the N nodes network, we first construct list1, which has the most neighbors, and list2 with second most neighbors. The "list1" should be the most important interference factor, and "list2" the second. The rest may be deduced by analogy. One or more nodes should be in list1. (2) Then, we extend the range to the interference range and calculate the interference nodes. This process should include all the nodes in $level(i) = 2$, if node i is the node in list1. "list1-1" should be the sub set of list1, and the size of list1-1 is the same or smaller than list1. Then, the level tree, with order of the interference, can be constructed. Although more than one node is in list1-1, we don't consider extending the range. This is because the node outside the interference will not produce interference. (3) The nodes in list 1-1 have the most interference. We then put them into a new constructed list called "conList", and remove them from the network N.

We continuously repeat (1)-(3) until there are no more nodes in the network N. Then, the order of the con-List is the order of the nodes with the most interference.

□

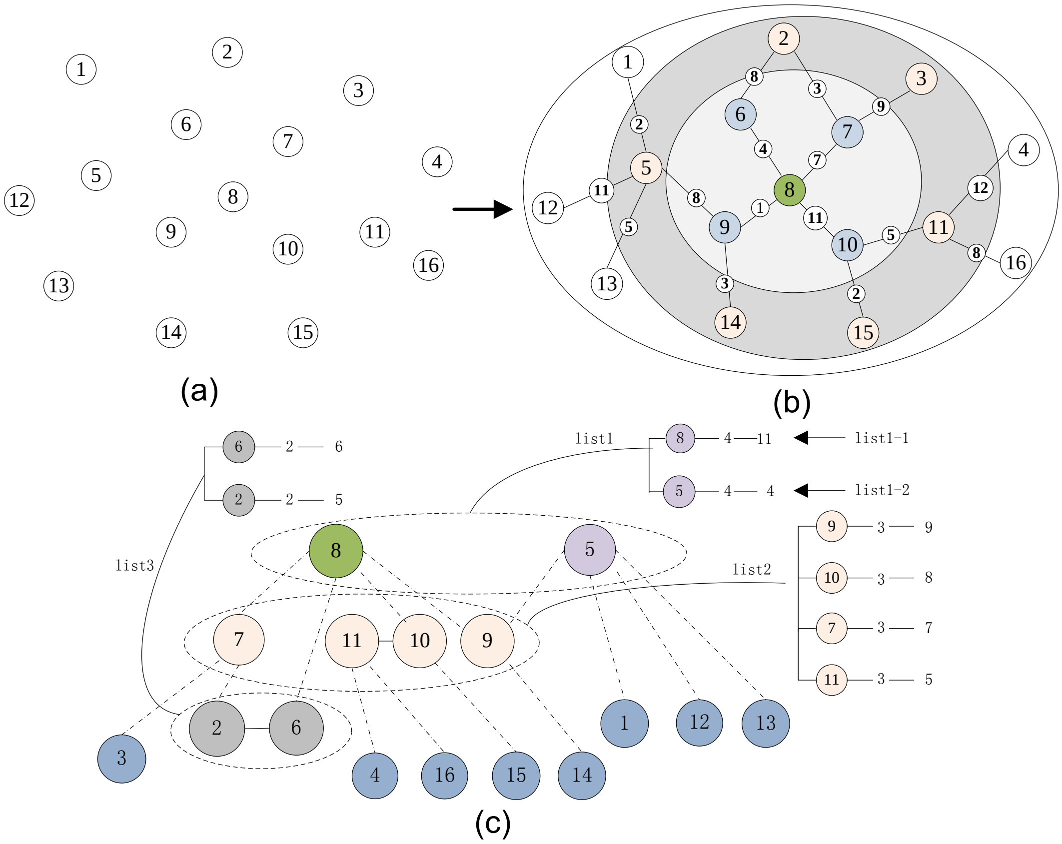


FIGURE 3.2: KN-CA Channel assignment

We can give an example of the KN-CA channel assignment in Figure 3.2 to show how they match the scope of FSR.

We first select a set of nodes (say *list*1), which have the most neighbors. If there is only one node, then we arrange channels for this node. If there are more than one node, we recursively calculate the neighbors of the nodes. Each time, the number of the set will be decreased. At last, we will get a set of nodes, which have the most neighbors. The initial assignment typically tries to find the most density nodes, where more interference occurs due to being connected to each other. The edges of the nodes will be visited, and the selected channels will be arranged for the edges. The detailed algorithm could be found in [31].

Examples: The constructed level tree (see Figure 3.2 (c)) shows the order list for assigned channels of Figure 3.2 (a). As Figure 3.2 shows, node 8 and node 5 constructed $list1$. In $list1$, we also calculate the value of $level(8) = 2$ and $level(6) = 2$. Then, node 8 is the most important factor to be assigned channels since we can see the interference of node 8 in both $list1$ with value 4, and $list1$-1 with value 11. Then, Figure 3.2 (b) is the result of the deployment.

CHAPTER 4

DESIGN ISSUES AND SYSTEM MODEL

In this chapter, we will provide the discussion to propose our model. The assumptions and design issues will be provided. We will also give an overview of our system model.

4.1 Design issues

The characteristics in the above model actually distinguish M2WMNs from other types of wireless networks. With these characteristics, we identify the following key design issues for our system model.

4.1.1 Interference

Interference is the foremost factor that degrades the wireless network performance, so the primary goal of our model is to minimize interference within the M2WMNs by utilizing the multiple radios and multiple channels. To address the interference issue, a model describing the interference effect needs to be assumed. Currently, there are two widely-adopted models: Protocol Model and Physical Model. The Protocol Model is simple, described as follows: (1) each radio has a transmission range and an interference range, with the former less than the latter; and (2) a transmission from radio X to radio Y is successful if Y is in the transmission range of X and not in the interference range of radios other than X that are currently transmitting.

So, our model needs to consider at least the aforementioned factors: the interference model, the number of available channels, the number of radios at each node, and the node deployment. However, the joint consideration of these factors mostly results in NP-hard problems.

4.1.2 Connectivity

This connectivity arises from the communication constraint that, in a multi-channel environment, two neighboring radios must share a common channel to communicate. Due to this constraint, altogether two kinds of discrepancies can occur between the unit disk graph and the network topology: (1) a link between two nodes in the unit disk graph is absent in the network topology if the radios on these two nodes are not assigned a common channel; (2) multiple links exist between two nodes in the network topology if multiple common channels are assigned to the radios on these two nodes.

4.1.3 Stability

The channel switch operation can cause two phenomena that undermine the network stability - ripple effect and channel oscillation, which need to be tackled properly by the channel assignment approaches.

The ripple effect, as described in the former chapter, arises from the communication constraint that two radios in communication must share the same channel. A similar phenomenon to oscillation in routing, channel oscillation means that the channel assignment does not converge and changes back and forth among several choices. This phenomenon usually happens when the channel assignment is based on a dynamic metric. For example, when two nodes discover that a channel is under-utilized according to such a dynamic metric, they may simultaneously switch to this channel and both begin transmission on it, and then switch back because this channel is now overloaded, as indicated by this dynamic metric. Since channel switching involves significant overhead such as switching delay and traffic interruption, frequent channel switching resulted from oscillation will severely impair the network performance.

4.1.4 Throughput/Latency

Throughput and latency are the two most important measures for network performance. Having a deterministic relationship, they are generally addressed together. To obtain optimal throughput/latency in M2WMNs, the channel assignment approaches need to consider the following two basic strategies. First, reducing interference is the most effective method in achieving the optimality, and it is better to make this method adaptive to the dynamic network traffic. Secondly, links should be treated differently when assigning channels, since different links in M2WMNs impact throughput/latency to different extents. For example, the backbone links carry much more network traffic than the stub links, so they should be given more bandwidth either by assigning more number of concurrent channels or by assigning less interfered channels. In this sense, an M2WMN is analogous to a wired enterprise/campus network consisting of a hierarchy of Ethernet switches, where the ports on an upper-level switch usually have much more bandwidth than the ports on a lower-level switch.

4.1.5 Routing

The network topology of an M2WMN - a basic factor for making the routing decisions - can be changed by the channel switch decisions. Thus, routing is dependent on channel assignment. On the other hand, routing can change the traffic load distribution in the network, which is a primary factor considered by channel assignment to reduce the interference dynamically. So channel assignment is also dependent on routing. With channel assignment and routing mutually dependent, how to combine these two mechanisms to obtain optimal network performance is a very challenging issue.

4.1.6 Fault tolerance

Though the nodes in M2WMNs are stationary, they can fail because of software or hardware problems. Moreover, the wireless links can also fail due to unexpected scenarios such as external interference or temporary obstacles. So it is necessary for a channel assignment approach to support fault tolerance such that the network could operate in a self-healing fashion. Furthermore, though the multiple radios and multiple channels in M2WMNs provide abundant choices for recovering from faults, a selection among these choices has to be made to obtain the optimal results. Therefore, the fault tolerance issue is not an easy task for the channel assignment approaches to address.

4.1.7 Fairness

To obtain the optimal overall network performance, fairness among the nodes is sometimes sacrificed in designing a channel switch approach. But in many scenarios such as the mobile phone backhaul networks, fairness is a necessary property of the network services. A basic fairness criterion for channel assignment is the capability to avoid that the traffic of some nodes only has access to crowded channels shared by many links, while the traffic of other nodes has access to only partly-occupied channels shared by a few links. Though guaranteeing fairness to a certain level introduces additional difficulty to the channel assignment design, it is worth achieving in practice.

4.2 System model

4.2.1 Assumptions

In this subsection, we describe the model formulation for our system in WMNs. Recall that WMNs consist of a set of stationary wireless routers, some of them acting as gateways to the Internet. Specifically, we do not require the presence of special gateway nodes, which could be the source or destination of all traffic in the network. Each neighbor will be associated with a timeout value.

We define our system requirement as follows:

1. Two nodes that can communicate with each other should have at least one common channel.
2. We assume that every node in our system has the channel-switch abilities.

3. The common default channel is required for transferring control messages and is used as a temporary channel for data transfer.
4. Channels refer to different frequency bands. All of the channels are working on the half duplex mode.

4.2.2 Problem statement

Channel assignment algorithms in WMNs select channels for each link in the network in order to optimize network throughput and reduce latency. Recent channel assignment research explores node's ability of dynamically switching channels. In essence, when the total amount of traffic load along this link exceeds the link capacity, nodes can either reduce the traffic load on this link or switch the channel. If the channel capacity is degraded due to CCI and ACI, it is desirable to switch from the current channel to another channel with higher bandwidth. However, channel switching incurs noticeable latency due to synchronization overhead between a pair of nodes, which also decreases the link throughput. Therefore, there is a tradeoff between the benefit of channel switching and its overhead.

In this project, we explore when to perform channel switching under dynamic channel throughput. Previous research on channel switch is usually based on analysis with static channel models. Although these works provide valuable insights on this problem, unfortunately, these assumptions may not be hold in real WMNs:

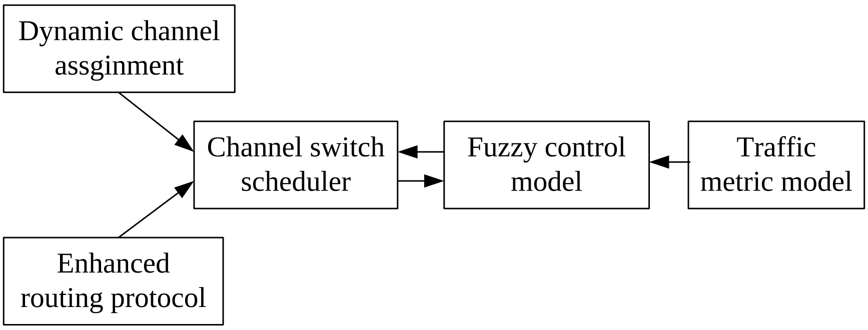


Figure 4.1: System overview.

First, wireless link capacity is sensitive to distance and surrounding environment. In WMNs with fixed topology, even though the distance between any pair of nodes is fixed, the link capacity may vary due to environmental changes. Second, the traffic loads sometimes experience transient increases, which will result in the decrease of the traffic load and interference. These problems are not well modeled in previous studies. Third, the traffic loads affect link capacity, especially in dense networks with high traffic load. When traffic load of a link increases, it can cause channel capacity degrade on itself and other nearby links, even they are assigned with different channels due to interference. Therefore, with the dynamic traffic loads and channel capacity, previous solutions may not work well.

4.2.3 System overview

Our solution to the above problem has four key components: a traffic metric model, a fuzzy control model, a dynamic channel assignment algorithm, and an enhanced routing algorithm. In the fuzzy control model as shown in Figure 4.1, a control loop is designed to monitor the channel capacity and dynamically adjust the threshold.

The threshold serves as a condition for channel switching algorithm. Our control model provides a reasonable fuzzy control mechanism for deciding whether channel switching should be performed and which channel should be switched to.

Based on the proposed model, the dynamic channel assignment algorithm performs channel switch during runtime in a distributed fashion. The conditions for selecting new channels are specified in the fuzzy control model. With the support of channel switch control and channel assignment, enhanced routing algorithms can achieve better performance than previous solutions.

CHAPTER 5

BALANCED CONTROL SYSTEM

Our system is composed of traffic metric model, fuzzy control model, dynamic channel switch scheme, and channel switch scheduler. This chapter will present the details respectively.

5.1 Traffic metric model

There are significant amount works focusing on the model of interference level [32, 33]. In this subsection, we offer an integrated model to estimate the level of traffic load and interference on each link.

Assuming c is the current channel, and i is the current node, if j is the neighbor of node i, then, the traffic load between nodes i and j is the sum of all the outing flow $flow_t$ and incoming flow $flow_r$, as described by Eq. 5.1:

$$bw^c(i) = \sum_{t=1}^{F_t} flow_t + \sum_{r=1}^{F_r} flow_r \tag{5.1}$$

To solve the CCI problem, we use the following Eq. 5.2 to describe the total traffic load on channel c in two hops for the $pair(i, j)$:

$$bwt_{i,j}^c = \sum_{n_i=1}^{N_i} bw^c(n_i) + \sum_{n_j=1, n_j \neq i}^{N_j} bw^c(n_j), \tag{5.2}$$

where N_i and N_j are the number of neighbors of node i and node j in the network, respectively. This equation is used to derive the total traffic load of $pair(i, j)$ and their neighbors, where they work on the current channel c.

The usage of adjacent channels can cause interference. Then, we can obtain the interference ratio γ according to the current channel measurement for ACI:

$$\gamma = 1 - \frac{| c - c_n |}{c_n} \tag{5.3}$$

where c is the current channel, which is used by node i and c_n is the adjacent channel corresponding to other interfaces of node i. $|c - c_n|$ is the distance of interference factor. It is similar to [10]. However, our comparison is based on two channels. The overall metric can be deduced as follows which integrates traffic load and adjacent channel interference factors based on two hops neighborhood

TABLE 5.1: Notations

Notation	Description
$pair(i,j)$	link for nodes i and j
$bw^c(i)$	traffic load of node i on channel c
F_t	number of flows going through $pair(i,j)$
t, r	flow id for transmitting/receiving the packets
$flow_t, flow_r$	traffic load of outgoing flows t or incoming flows r
n_i	neighbor id for node i
$bwt^c_{i,j}$	total traffic load of $pair(i,j)$ on channel c
N_i	number of neighbors of node i
γ	interference ratio for ACI
c, c_n	node's current channel and adjacent channel
$btt^c_{i,j}$	traffic metric of channel c which integrates CCI and ACI

information:

$$btt^c_{i,j} = bwt^c_{i,j} + \sum_{l=1}^{N_i} (\gamma \times bwt^{c_n}_{i,l}) \tag{5.4}$$

Thus, we have offered our traffic metric model. Note that our traffic model is based on the estimation of traffic load and interference level.

5.2 Fuzzy control model

In this subsection, we present our system model. Our control model provides a reasonable fuzzy control mechanism for deciding which channel to be switched to. The goal is to find a per-link threshold for channel switching, An ideal threshold will approximate run-time channel throughput and reduces the unnecessary channel switchings. In our fuzzy control model, if the threshold is set too small which indicates frequent switches, we will increase the threshold value. In contrast, if the threshold is set too large which prevents the system from performing beneficial

Table 5.2: Fuzzy control model table

Notation	Description
x	switching times
$\mu_{i,j}(x)$	threshold for $pair(i,j)$
$\Delta t_{i,j}(x)$	interval time between current and previous switches
$\phi_{i,j}(x)$	control weight of threshold
$t_{i,j}(x)$	the time when the $pair(i,j)$ switches to a new channel
$E(x)$	the value for deviation
$R(x)$	the variance ratio
$channelList$	available channels in the network
$channel(i)(j)$	the channel on the interface j of node i

channel switch; we will lower the threshold value accordingly. Therefore, we need to find the threshold adapting to the link capacity changes.

In this work, we use fuzzy control to perform such an adjustment because it is difficult to accurately estimate the right threshold value a priori. Fuzzy control offers a convenient method for constructing nonlinear controllers via the use of heuristic information [34]. We present our fuzzy control model to adjust the threshold value. Accurate threshold adjustment is important because small errors in the thresholds can induce large channel-switch overheads. However, it is very challenging to directly estimate the exact threshold value since the environment is constantly changing. Thus, we use a fuzzy control model to demonstrate our strategy, which is similar to the automobile "cruise control" example in [13]. In our design, fuzzy interpretations are extended using the fuzzy set theoretic operations [35].

In our fuzzy control model, we use the interval time τ as a timer to monitor the network state. If the interval time between current switch and previous switch happens within τ, we will increase the threshold after this switch. Otherwise, we will decrease the threshold. We use Eq. 5.5 to express this idea. There are two parts in this equation: the first part is that the current interval time is less than τ. When the current bandwidth is larger than constraint bandwidth $\mu_{i,j}(x)$, we need to switch the channel, and also increase the $\mu_{i,j}(x)$; the second part is that the current time is larger than τ. This means there is no switching during this interval. We need to lower the threshold value:

$$\begin{cases} \mu_{i,j}(x) = \mu_{i,j}(x-1) + \phi_{i,j}(x-1), & \Delta t_{i,j}(x) \leq \tau \\ \mu_{i,j}(x) = \mu_{i,j}(x-1) - \phi_{i,j}(x-1), & otherwise \\ \mu_{i,j}(0) = \mu_0, \phi_{i,j}(0) = 0, \end{cases} \tag{5.5}$$

where $\phi_{i,j}(x)$ is the control weight and $\mu_{i,j}(x)$ is the constraint bandwidth for $pair(i,j)$. $\mu_{i,j}(0)$ and $\phi_{i,j}(0)$ are the initial values of the two parameters. The threshold can be improved during τ. As we can see from this equation, the control weight is important for controlling process. That is how much we need to increase or decrease the threshold. Moreover, when the switch should be performed is also an important issue. For this we use the Eq. 5.6 to demonstrate the control weight $\phi_{i,j}(x)$:

$$\phi_{i,j}(x) = \alpha_{i,j}(x) \times E(x) + (1 - \alpha_{i,j}(x)) \times R(x-1) \tag{5.6}$$

where $E(x)$ is the value for deviation, and $R(x)$ is the variance ratio. $\alpha_{i,j}(x)$ is the weight for the balanced formula. $E(x)$ and $R(x)$ have different effects among different switches. Note that $E(x)$ should be much larger than $R(x)$. We use $\alpha_{i,j}(x)$

to adjust the threshold. Sometimes, we want to adjust this threshold quickly, thus, we increase the $\alpha_{i,j}(x)$ for $E(x)$. Other times, we prefer that it changes slowly, thus, we decease the $\alpha_{i,j}(x)$ for $R(x)$. It is shown that $\alpha_{i,j}(x)$ is used to control the two parts. Next, we offer the following equation to obtain $E(x)$, $R(x)$ and the weight $\alpha_{i,j}(x)$.

$$\begin{cases} E(x) = \mu_{i,j}(x) \\ R(x) = \dfrac{\phi_{i,j}(x)}{(t_{i,j}(x) - t_{i,j}(x-1))} \\ \Delta t_{i,j}(x) = t_{i,j}(x) - t_{i,j}(x-1) \\ t_{i,j}(0) = 0, \forall i, j, x \in N, \tau > 1, \end{cases} \tag{5.7}$$

$$\begin{cases} \alpha_{i,j}(x+1) = \mid 1 - \dfrac{t_{i,j}(x) - t_{i,j}(x-1)}{\tau} \mid, \Delta t_{i,j}(x) \leq \tau \\ \alpha_{i,j}(x+1) = 0, otherwise \end{cases} \tag{5.8}$$

where $t_{i,j}(x)$ is the recorded time when the $pair(i,j)$ switches to a new channel. For the weight $\alpha_{i,j}(x+1)$, if the interval time between the current switch and previous switch is larger, the current threshold is close to the estimated threshold. Then, $R(x)$ is larger according to the weight $\alpha_{i,j}(x+1)$. Otherwise, $E(x)$ is larger.

Below, we summarize the whole process. At the start of the controlling process, the deviation $E(x)$ is more important. During this period, the control weight $\phi_{i,j}(x)$ can help the system quickly find the range of the threshold. A decrease in the $\alpha_{i,j}(x)$ will make the variance ratio $R(x)$ become the main factor of the system

Table 5.3: Channel Information of GNOC Scheme

Notation	Description
$neighbor\_Set(i)$	neighbors of the node i
$channelList$	available channels in the network
$channel(i)(l)$	channel id for node i on interface l
$interface(i)$	interface information of node i
$channelState(i)(j)$	channel bandwidth of $pair(i,j)$

and $E(x)$ become less important. Thus, it could adjust the value, and control the estimated threshold in this range.

5.3 A dynamic channel assignment scheme

We design a dynamic channel assignment algorithm (GNOC) to get the next optional channel for our system. The GNOC is used to select new channels for transmission. We will select the channel which is lower than the threshold and with minimum ACI. Table 5.3 offers the notation for the GNOC algorithm.

Algorithm 2 GNOC

```
Let $tempList \leftarrow cList(i)$;
for $k \leftarrow 0$ to $N_i$ do
    if $channel(j)(k)$ is not in the $tempList$ then
        $channel(j)(k)$ is added into $tempList$;
for $j \leftarrow 0$ to $|channelList|$ do
    for $k \leftarrow 0$ to $|tempList|$ do
        $c \leftarrow k$th channel from $tempList$;
        $diff \leftarrow |channelList|/N_i$;
        $diff \leftarrow \min(diff, |channel(i)(j) - c|)$;
```

In Algorithm 2, the *channelList* is the available channels in WMNs. The current channel, say c, is the current channel used for transmission. We first construct a temporary channel list, say $tempList$, from the available channel list of node i. Then, we get node i's neighbor list. To each neighbor of node i, we add the channels that the neighbors of node i are using to the $tempList$. Then, we obtain the absolute value (abs) according to the channel in the $channelList$ and $tempList$. We set $diff$ to be an extreme value of differences among channels. Then, the final channel is the channel with a minimum $diff$.

The new channel we get is the channel with least ACI. However, we are not sure about the traffic statement of the new channel. Therefore, we also check the current statement of the new channel according to our proposed routing metric. If it does not exceed the threshold $\mu_{i,j}(x)$, then the new channel is the next optional channel. Otherwise, we remove this channel and run Algorithm 1 again until we find it.

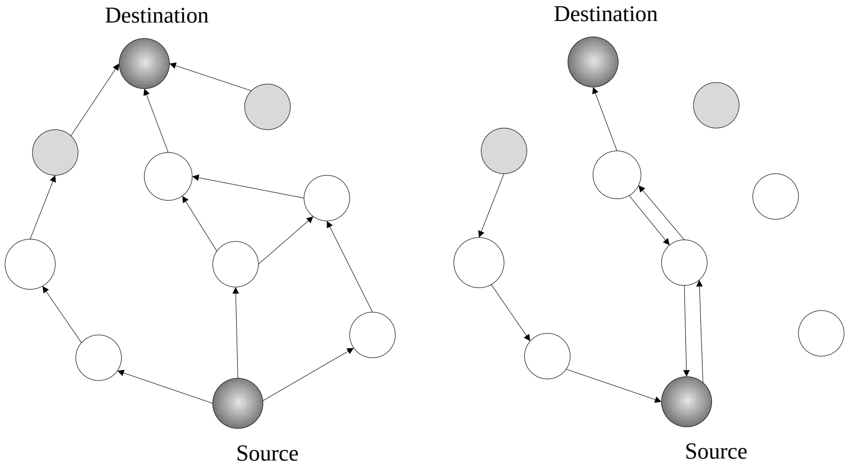
5.4 Enhanced routing protocols in WMNs

First, we demonstrate why the existing routing protocol can not be used in our scheme. The existing routing protocols, such as AODV [29] and FSR [30] support multiple interfaces. However, these designs are typically used for the multi-home based protocol (such as SCTP and DCCP) instead of used for multiple interface wireless mesh networks.

The main problem is that the ideal design of our system needs fast channel switching to solve the unstable system problem. As we all know, after one node, say node i, has sent the "hello" information, the neighbors who received the "hello" information will add node i into the routing table entry. Then, they establish the connection. This will remain for future use, until it expires. If node i has changed its channel, but the corresponding neighbor, say node j, does not switch the new one, then, this means they cannot connect to each other after the switch. However, the routing table entry remains in both nodes. When they need the route, they still can find the route according to the routing table entry. However, the packets cannot be forwarded because the actual connection is broken. It will take more time to establish and find a new route. The time of transferring the packets has to be delayed.

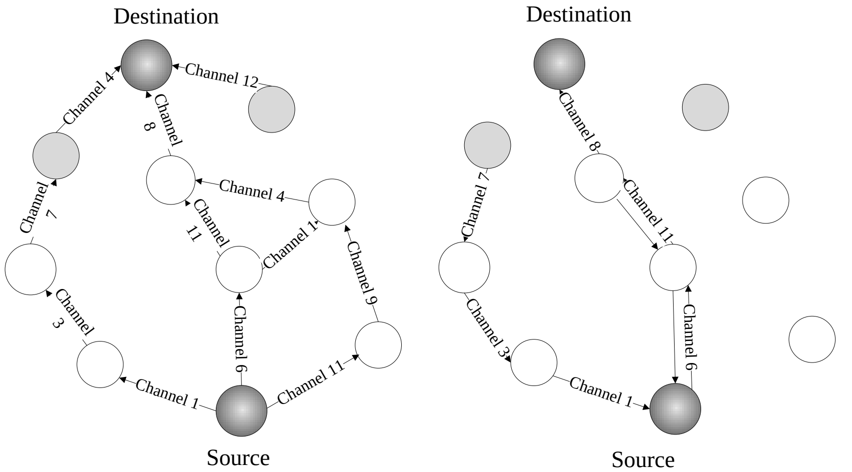
We propose to introduce another field called "channel id" to the routing table entry. This "channel id" can be used among neighboring nodes to coordinate their channel selection processes. The two neighbors along a channel can talk to each other, when they switch to the same channel, specified by "channel id". Interface and channel information can be obtained during initialization. When the node has switched to another channel, and the "channel id" has changed, its routing table should be updated according to the new "channel id" information.

In our model, we propose modifications to the current routing protocols, AODV and FSR, so as to enable the discovery of channel information from a source to a



(a) RREQ Broadcast (b) RREP Propagation and Subsequent Route

Figure 5.1: AODV Routing protocol



(a) RREQ Broadcast (b) RREP Propagation and Subsequent Route

Figure 5.2: AODV Routing protocol

destination. The proposed enhancement can help the node find the correct route, even if the nodes have switched channels.

E-AODV routing: When the source node requests to send the packet to the destination, it will send “hello” information. The broadcast packet RREQ will be flooded to every interface of the node. Channel information is also added into the packets, and the corresponding nodes that receive the notice will update the routing table entry. At that time, the "channel id" is updated, and the nodes will select the correct route to communicate.

E-Fisheye routing: Fisheye Routing is different from AODV Routing, which is routing on-demand. In the fisheye routing scheme, we update the routing table according to the channel table, which will be explained in the next subsection (Channel Switch Scheduler). If the channel table has been changed, the routing table needs to be updated as well. Algorithm 3 is offered to show the detailed process of updating the routing table.

Algorithm 3 Update routing table

```
for $i \leftarrow 0$ to num. of nics on node $x$ do
    for $y \leftarrow 0$ to num. of nics on node $y$ do
        if $(channel(x)(i) = channel(y)(j))$ then
            $channelid \leftarrow channel(x)(i)$
nodes $x$ and $y$ lookup the route, and update routing table with newchannelId;
```

5.5 Channel switch scheduler

In [36], the author proposes the idea of the channel switch being involved in the mesh networks. However, that idea is not flexible and did not consider the detailed routing issues in the channel switch period. In this subsection, we will provide the process of channel switch. The channel and interface information has been maintained by every node in WMNs. Table 5.3 details the information for every node.

Algorithm 4 Executed by sender

```
if $btt^c_{i,j} < \mu_{i,j}(x)$ then
    Forward packet;
else
    if The num. of common channels of $pair(i, j)$ is larger than 1 then
        Update the current routing table entry;
        Go to step 1;
    if Unused interfaces of node $i$ is larger than 0 then
        $newchannel \leftarrow interface(i)$;
        Go to step 1;
    $newchannel \leftarrow$ the next optional channel according to GNOC;
    if $btt^c_{i,j} < btt^{newchannel}_{i,j}$ then
        $newchannel \leftarrow c$;
    Step 1:
    Update $\mu_{i,j}(x)$ according to Eq. 5.5;
    $neighbornodeList \leftarrow yList(x)$;
    Generate $chr\_packet$;
    Update the current routing table entry;
    Send $chr\_packet$ to the nodes in $ylist(x)$;
    End step 1;
    Forward packet using the new routing table;
```

Algorithm 5 Executed by receiver

```
Receive a $chr\_packet$;
if $(nodeId \in neighbornodelist)$ then
    Switch channel and update $\mu_{i,j}(x)$ according to Eq. 5.5;
    Update the current routing table entry;
    Update channel table;
else
    Update channel table;
```

The $neighbor\_Set(i)$ is the set for all the neighbors of the current node, say node i. The number of available channels contains all the channels that can be used for the whole network. Every node in this network can switch channels when it satisfies the threshold $\mu_{i,j}(x)$. Moreover, special conditions need be satisfied:

- Condition 1: There is another common channel for the $pair(i, j)$. If the traffic load of that channel does not exceed threshold $\mu_{i,j}(x)$, then the $pair(i, j)$ will select that channel to communicate.

- Condition 2: If the new channel, taken from Algorithm 2, is already working for another interface of node j, then only node i switches to the new channel.
- Condition 3: There are other nodes that are neighbors of $pair(i, j)$. All of them work on the same old channel. These nodes construct a temporary list called $yList$.

If a new pair is added to the transmission, the traffic load exceeds the threshold of $\mu_{i,j}(x)$ from Eq. 5.5. For example, the $pair(i, j)$ in channel c first checks the condition 1. If it is not satisfied, it will temporally take channel c from Algorithm 2. Then, it will check condition 2 and get the node to switch. If there is an available interface for the node, it will select the available interface.

In this enhanced routing protocols, the channel information is stored on every node's neighbor table. We employ the standard mechanism [4, 20–22] widely used to deal with topology dynamics, such as nodes join the network. If a new node joins the network, it can notify other nodes in the network by broadcasting a hello message via a common control channel. In unicast, the current node will collect the channel state information through the channel information before transmission. The current traffic load can be obtained from Eq. 5.4. Then, we apply the channel switch algorithm, shown in Algorithms 4 (sender) and 5 (receiver). If the current traffic load exceeds the threshold $\mu_{i,j}(x)$, it will switch to another channel. Before that, the node should send the message to every node in the network. A field

Table 5.4: CHR Packet for Channel Switch Scheduler

Notation	Description
$nodeId$	The current node to be switched
$othernodeId$	The other node of the pair to be switched
$oldchannelId$	The old channel of the pair
$newchannelId$	The new channel to be switched for the pair
$neighbornodeList$	The neighbors sharing the same channel

chr_packet (see Table 5.4) is added to a packet to carry the information. If the other node in the $neighbornodeList$ receives the *chr_packet*, and it does not satisfy condition 2, then it is required to switch to the new channel c.

We first apply an initial channel assignment according to the topology generator using KN-CA. The GNOC algorithm 2 is used to select the next optimal channel for this switch. The enhanced routing agent is used for the channel switch scheduler and the proposed BCS is used to make the decision of the switch.

CHAPTER 6

STABILITY ANALYSIS

In this chapter, we will provide the stability analysis. We will first offer the proof of stability. An example will be provided to explain our model. We also include a discussion by providing more insights.

6.1 Existence of threshold

We offer the following theorem to show that our system can finally find the threshold.

Theorem 6.1. *If the control weight can be infinitely close to 0, then our system could finally find the estimated threshold* $\mu_{i,j}(x)$. *Thus, we need to prove the following conclusion:*

$$\lim_{\Delta t \to \tau, x \to \infty} \phi_{i,j}(x) = 0$$

Proof. From Eq.5.6, the problem can be converted to two parts:

$$\lim_{\Delta t \to \tau, x \to \infty} \alpha_{i,j}(x+1) \times E(x) = 0,$$

$$\lim_{\Delta t \to \tau, x \to \infty} (1 - \alpha_{i,j}(x+1)) \times R(x) = 0$$

Here, we know:

$$E(x) < C, \text{such that } C \in \Re$$

Although C could be sufficiently large, C actually is a finite number. Thus, for the first part, we only need to prove:

$$\lim_{\Delta t_{i,j} \to \tau, m \to \infty} \alpha_{i,j}(x+1) = 0$$

From Eq.5.7, we have:

$$\lim_{\Delta t_{i,j} \to \tau, m \to \infty} \alpha_{i,j}(x+1) = \lim_{\Delta t \to \tau} \mid 1 - \frac{\Delta t(x)}{\max(\Delta t_{i,j}(x), \tau)} \mid$$

According to the first part of Eq. 5.5, we know that the interval time $\Delta t_{i,j}(x)$ is increased after the switch, since the threshold is larger and more difficult to meet switching conditions. However, in the second part, the proof is obvious, since $\frac{\Delta t_{i,j}(x)}{\max(\Delta t_{i,j}(x), \tau)} = 1$. Thus, we prove that

$$\lim_{\Delta t_{i,j} \to \tau, x \to \infty} \alpha_{i,j}(x+1) = 0$$

.

Next, we need to verify the following assumption:

$$\lim_{\Delta t \to \tau, x \to \infty} (1 - \alpha_{i,j}(x+1)) \times R(x) = 0$$

Because we have:

$$\lim_{\Delta t \to \tau, x \to \infty} (1 - \alpha_{i,j}(x+1)) \times R(x) \leq \lim_{\Delta t \to \tau, x \to \infty} R(x)$$

$$= \lim_{\Delta t \to \tau, x \to \infty} \frac{\phi_{i,j}(x)}{\Delta t_{i,j}},$$

$\phi_{i,j}(x)$ could be sufficiently large. It is actually bounded by a finite value ϕ_{max}, such that $\phi_{i,j}(x) < \phi_{max}$. According to Cauchy series [37], monotone sequence converges if and only if it is bounded. Since $\Delta t_{i,j}$ is increased during the controlling process, then:

$$\lim_{\Delta t \to \tau, x \to \infty} \frac{\phi_{i,j}(x)}{\Delta t_{i,j}} = 0$$

Thus, this concludes the proof of our solution. □

6.2 Examples

In this section, we will offer an example of our proposed model. We will formalize the theory results according to our analysis. We first change Eq. 5.6 to another

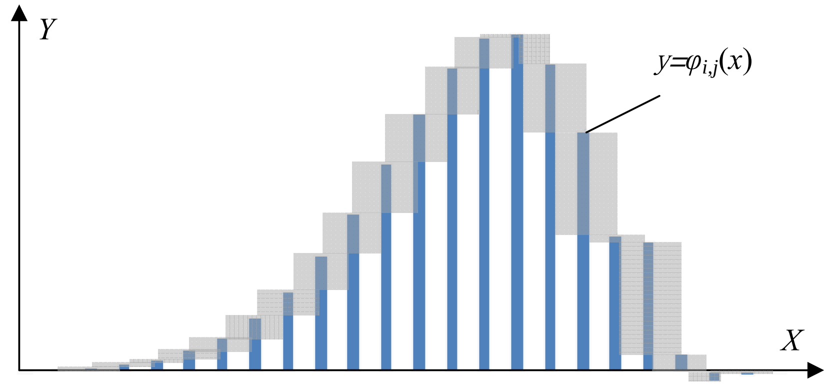


FIGURE 6.1: Formula $\phi_{i,j}(x)$ in the control model

form. We set:

$$\begin{cases} A = \alpha_{i,j}(x+1) \\ B = \dfrac{1-\alpha_{i,j}(x+1)}{t_{i,j}(x)-t_{i,j}(x-1)} \end{cases}$$

Then, Eq. 5.6 can be deduced:

$$\phi_{i,j}(x+1) = A \cdot (\mu_{i,j}(0) + \sum_{k=0}^{x-1} \phi_{i,j}(k)) + B \cdot \phi_{i,j}(x) \tag{6.1}$$

Figure 6.1 demonstrates theoretical results of formula $\phi_{i,j}(x)$ (see Eq. 6.1). Because other values could be deduced after each switch, we only need to consider A and B of Eq. 6.1, because $\Delta t_{i,j}(x)$ of B is the unknown result measured from the real-time record. We set the value of $\Delta t_{i,j}(x)$ from 1 to 20. The increase is 1 each time. These values could be different each time, but they must be incremental. The reason is that each time the threshold increases, it becomes more difficult to meet the channel switch condition. Without the loss of generality, we set $\tau = 20$

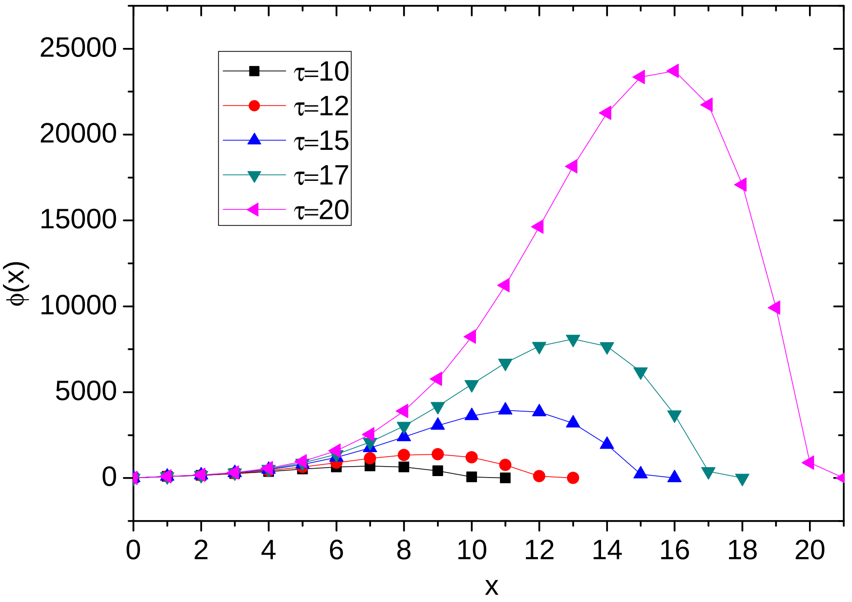


FIGURE 6.2: Comparison of different τ

and $\mu_{i,j}(0) = 100$. From Figure 6.1, we can see that $\phi_{i,j}(x)$ first increases and then decreases as time goes on. This confirms the validity of our design. The purpose of the control system is to dynamically find the threshold and make the system stable. To achieve this goal, we should continuously reduce the selected area of the constraint $\mu_{i,j}(x)$ until we find the exact value $\mu_{i,j}(x)$. In other words, we first reduce the area as quickly as possible, and then adjust it. The increased process of $\phi_{i,j}(x)$ is to find the smallest area. In addition, the decreased process is to adjust the value in the small selected area. That is the reason we need the control weight $\phi_{i,j}(x)$ to be first increasing and then decreasing.

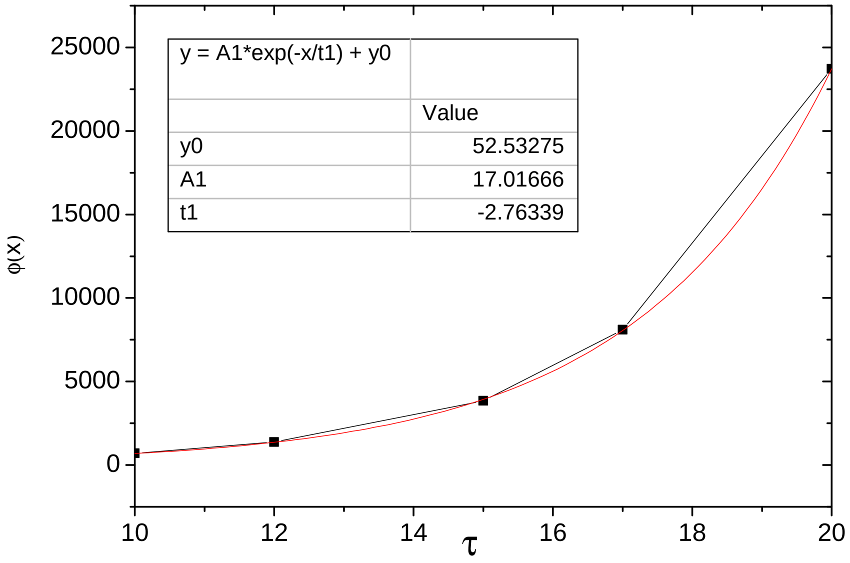


FIGURE 6.3: Comparison with Fitting function

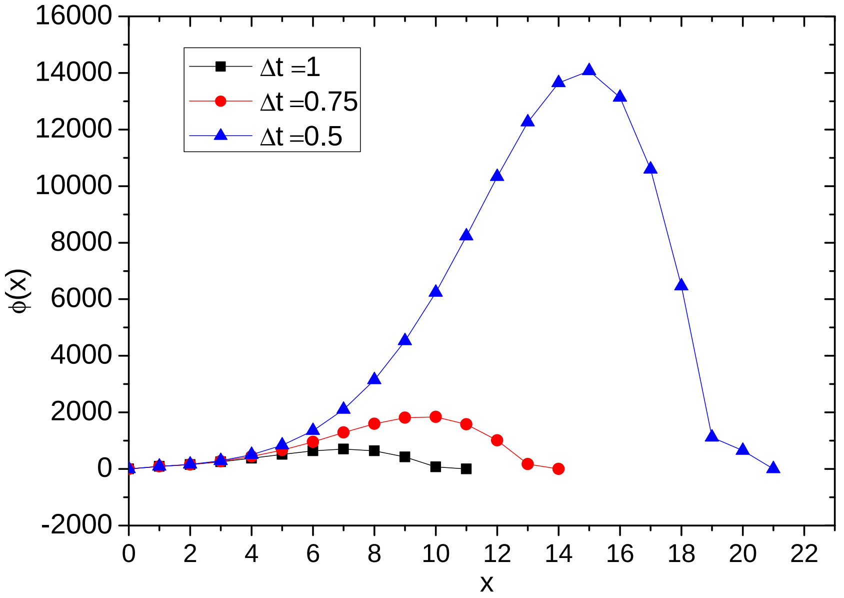


FIGURE 6.4: Comparison of different Δt

6.3 Discussion

According to our analysis, we know that our method can dynamically adjust the threshold value according the traffic demand. If the amount of traffic increases dramatically beyond the threshold, the threshold will be adjusted higher more aggressively, and vice versa. The node might also be switched to use the channel which has less traffic load. More specifically, when τ become larger, the increase of $\phi(x)$ scales in an exponential way. We present this analytical result in Figure 6.2 by varying the parameter τ from 10 to 20. The larger τ with the same interval will incur higher ϕ. This will result in larger threshold value. In this figure, the largest values for points on each curves vary from 699.67 to 23710. We plotted these largest values in Figure 6.3. From this figure, we can see that the threshold will increase in an exponential way as τ value increases. We also provide an analysis subject to changes of $\Delta t_{i,j}(x)$ in Figure 6.4. When $\Delta t_{i,j}(x)$ is set to increase as 0.5, 0.75 and 1 separately, the increase of $\phi(x)$ is exponential. This indicates that if $\Delta t_{i,j}(x)$ is smaller, the traffic become more dynamic and the increase is larger.

CHAPTER 7

PERFORMANCE EVALUATION

In this chapter, we evaluate the performance of our proposed system through simulation. We implemented our solution: channel switch control and dynamic channel assignment algorithm in NS-2 simulator.

7.1 Simulation Setup

As NS-2 provides rich physical layer models, the NS-2 based simulation have been widely used in research studies. We test the enhanced routing algorithms: AODV and FSR on top of our solutions, which are default routing protocols according to 802.11s [38]. AODV [29] is a reactive routing protocol while FSR [30] is a proactive routing protocol. FSR controls its update overhead using a policy of non-uniform frequency for update. The inner scope nodes are updated more frequently (and hence have more accurate information) than the outer scope nodes. Our solutions can also work with other routing protocols in WMNs.

We select the two-ray ground reflection model. The transmission range is 22 meters, so two nodes that are 22 meters apart can communicate with each other. The listening range is 44 meters, so nodes that are within 44 meters can cause interference to each other. We adopt KN-CA in our evaluation. There are 12 channels in the 802.11b network.

We evaluate our BCS with two enhanced routing protocols: AODV and FSR. With the 802.11b environment, the actual maximum throughput B_{th}, with MSDU size of 200 bytes, is 1.21 Mbps. The range of $\mu_{i,j}(x)$ is from 0.49 Mbps to 3.848 Mbps according to interfaces per node. We select 491,510, 891,510, and 1,291,510 bytesps (bytes per second) as the initial values. This traffic profile is fixed for all the simulations. There are 25 nodes in this simulation, and each node has up to five interfaces. Four of these interfaces can be switched for data transmission and one interface is fixed as the default control channel. Besides the default control channel, we test two interfaces (2-nics) and three interfaces (3-nics) for data transmission in Section 7.2. In the rest of evaluations, we test four interfaces (4-nics) for data transmission.

In the simulation of WMNs, there are several available extensions [5] for M2WMNs. We extend the existing work with switching abilities using NS2. Extensively simulation results demonstrate that our algorithm outperforms existing solutions without control [31]. Overall, our solution improves existing solutions by 12.3%-72.8% in throughput and reduces 23.2%-52.3% in latency.

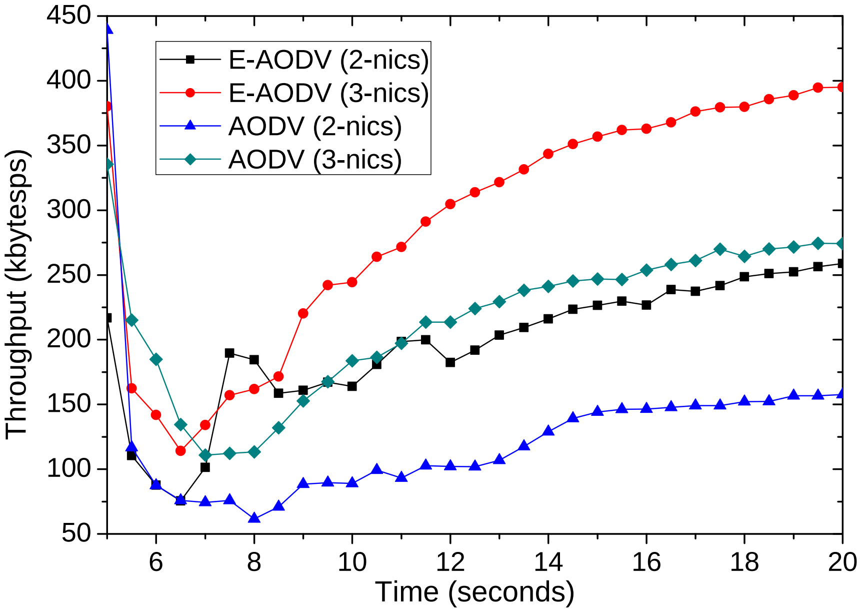


Figure 7.1: Throughput comparison with AODV and E-AODV

7.2 Performance Evaluation of Routing Protocols

We consider different number of interfaces as our evaluation study. Due the space limitation, we focus on comparison between enhanced-AODV (E-AODV) and AODV. The comparison result between enhanced-FSR (E-FSR) and FSR has the same trend. We use an 802.11b network environment. We adopt an interference based static channel assignment (KN-CA) and AODV routing in a simulation study. The evaluation consists of the three interfaces (3-nics) and 2 interfaces (2-nics). Figure 7.1 gives the comparison results of the proposed AODV routing and enhanced AODV routing (E-AODV).

As Figure 7.1 shows, the throughput of our enhanced AODV routing is higher than

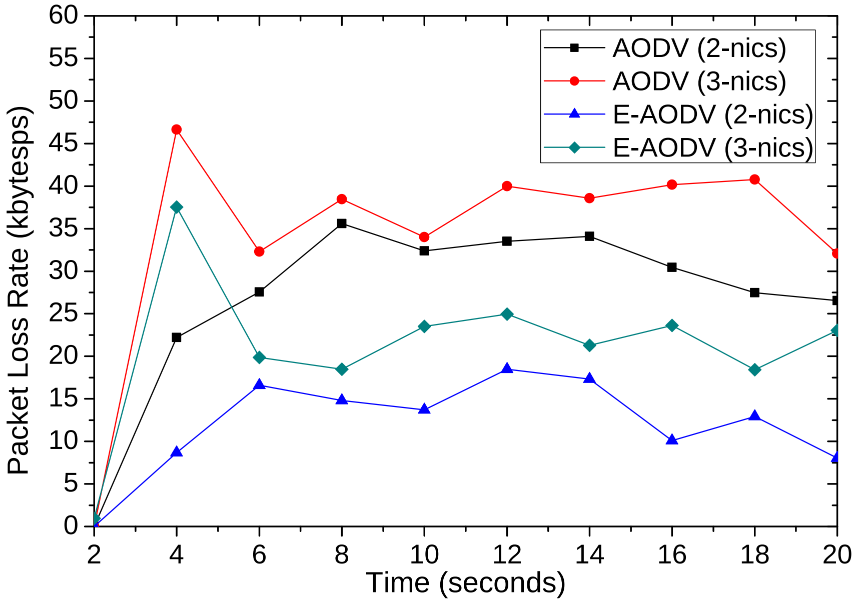


FIGURE 7.2: Packet delay comparison with AODV and E-AODV

the original AODV. The 3-nics AODV is better than the 2-nics AODV. The reason is because with enhanced AODV, the selected shortest route is calculated by the common channel. All the pairs of the route are working in the same channel. With 30 heavy traffics, the 3-nics AODV can partake in three different interfaces. The throughput can also be improved.

We can also see that our E-AODV achieves higher reliability than AODV, both in 2-nics and 3-nics cases in regards to the packet loss rate comparison (see Figure 7.2). From the above simulation results, we observe that our enhanced scheme for routing protocols performs better in multiple-channel multiple-interface environments. Then, we evaluate our balanced controlling system in the following sections. We will evaluate the BCS with both E-AODV and E-FSR routing protocols, separately.

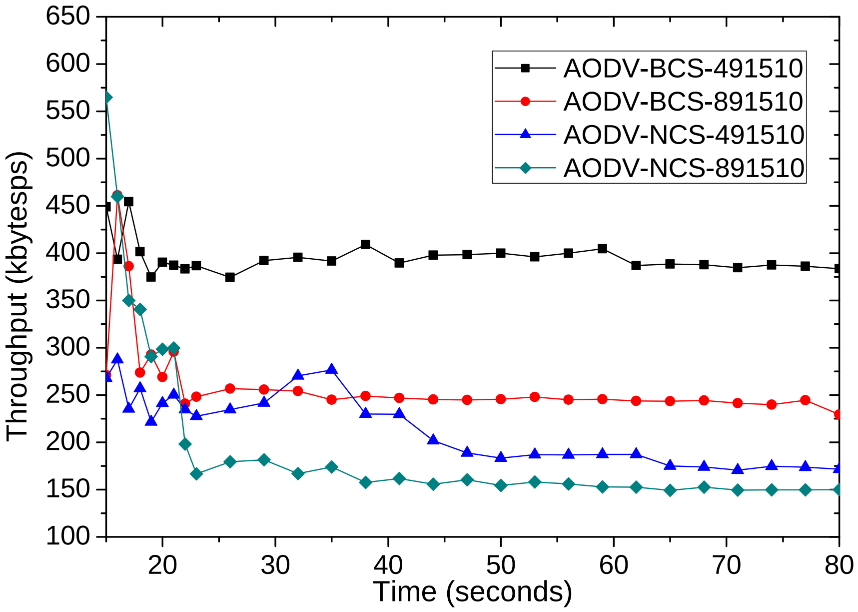


Figure 7.3: Throughput comparison between BCS and NCS using E-AODV

7.3 Performance Evaluation with Enhanced-AODV

In our solution, if the traffic load is high, the constraint $\mu_{i,j}(x)$ should be updated for that link. This link also needs a new channel that has less interference.

However, it is not easy to determine the exact value $\mu_{i,j}(x)$. So, BCS will try to get $\mu_{i,j}(x)$ as quickly as possible, according to Eq. 5.5. The simulation time is 80 seconds, and 30 heavy traffics are added separately, with the interval 0.4 seconds. Figure 7.3 shows the comparison results of the BCS, and without the channel switch control (NCS). The value behind the name in the figures is the initial value of $\mu_{i,j}(0)$.

It is obvious from the throughput comparison that with the same traffic profile, when the system uses the balanced control strategy, the network performance

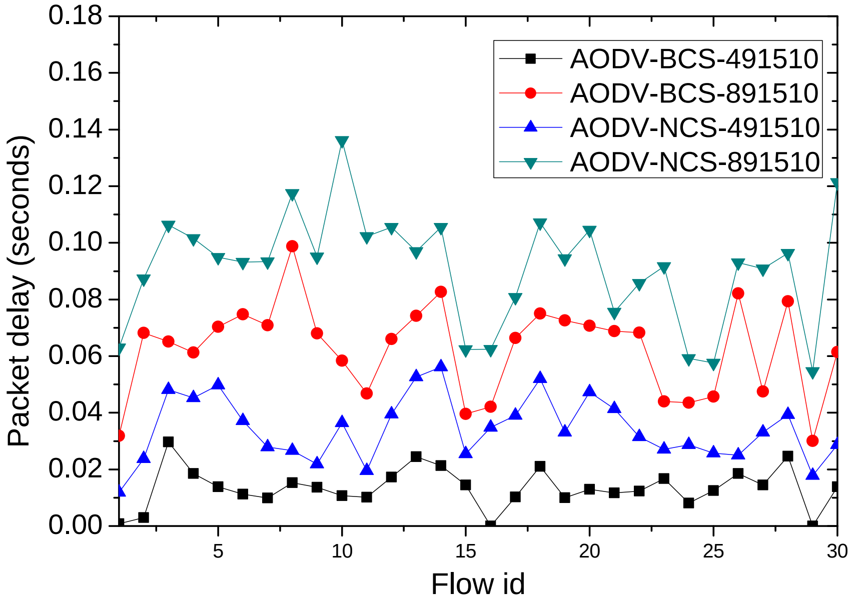


FIGURE 7.4: Packet delay comparison between BCS and NCS using E-AODV

is stable, and also better than the system without the BCS by 40%-70%. With different initial $\mu_{i,j}(0)$, the value of 491,510 bytesps is better than 891,510 bytesps. The reason is that the smaller $\mu_{i,j}(x)$ causes more switches, and the larger value $\mu_{i,j}(x)$ is difficult to achieve. This smaller $\mu_{i,j}(0)$ causes the default channel to take over some of the traffics.

We also give the comparison of packet delay (see Figure 7.4). Our BCS is also more efficient than the system without the BCS. The reason for this is that the network is involved in the control system, preventing the invalid channel switching. This is desirable because these invalid channel switches decrease the network performance. We also investigate the other parameters, such as packet delay and packet loss rate. With a BCS and the parameter 491,510 bytesps, the system achieves the best performance using the same traffic profile.

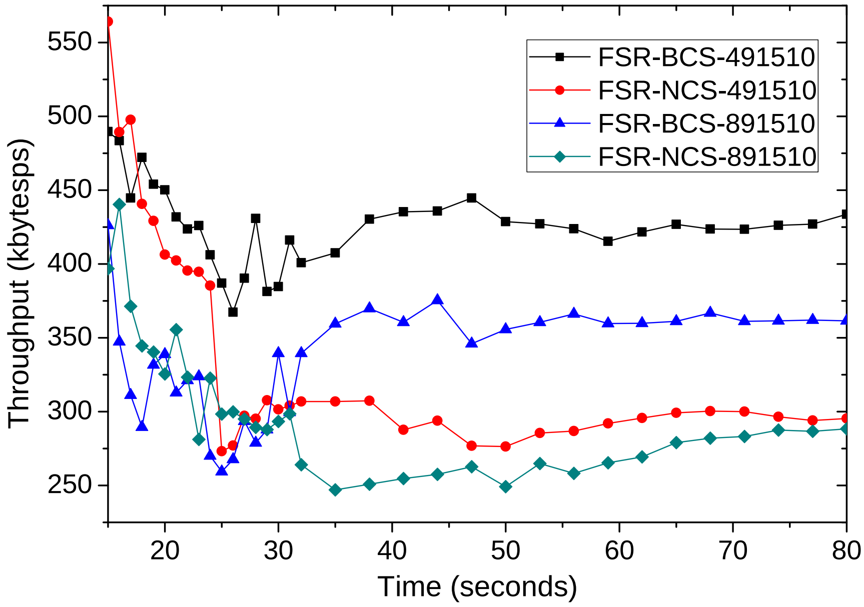


Figure 7.5: Throughput comparison between BCS and NCS using E-FSR

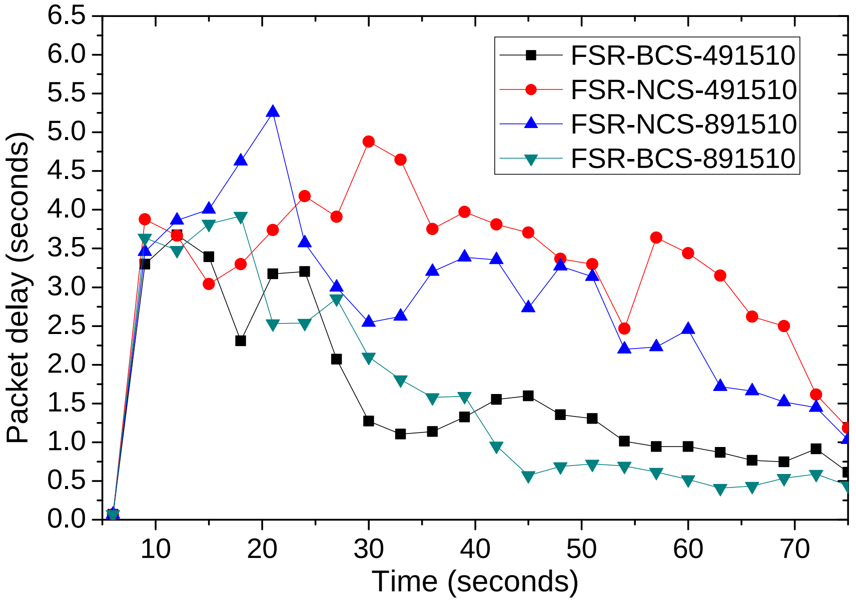


Figure 7.6: Packet delay comparison between BCS and NCS using E-FSR

7.4 Performance Evaluation with Enhanced-FSR

As Figure 7.5 demonstrates, our control system is also better when compared to NCS after 15 seconds. The performance of $\mu_{i,j}(0)$ is better than $\mu_{i,j}(0) = 891, 510$.

The reason is that 491,510 bytesps is the value that can be easily achieved, thus, some of the traffic is divided to the default channel.

Figure 7.6 gives the comparison results of the packet delay. We can see that the packet delay increased before 15 seconds. The performance is similar among the four situations. This is because the traffics are added as time goes on. Therefore, the packet delay increases. We can also see that when all the traffics are stable after 15 seconds, the BCS will balance the traffics. It is clear that after 25 seconds, the performance is better when it is used with the BCS. Also, the system of $\mu_{i,j}(0) = 491,510$ is outperforming $\mu_{i,j}(0) = 891,510$, both with the BCS and without the BCS.

7.5 Performance Comparison with Different Enhanced Routing Protocols

This part will demonstrate the comparison between enhanced AODV and FSR routing protocols with the parameter $\mu_{i,j}(0) = 1,291,510$. We have shown the results of $\mu_{i,j}(0) = 491,510$ and $891,510$. We can roughly see the throughput results above in Figures 7.3 and 7.5. The performance of the FSR protocol is better. To further verify the result, we select another parameter $\mu_{i,j}(0) = 1,291,510$ and compare its performance to the one without channel switch control.

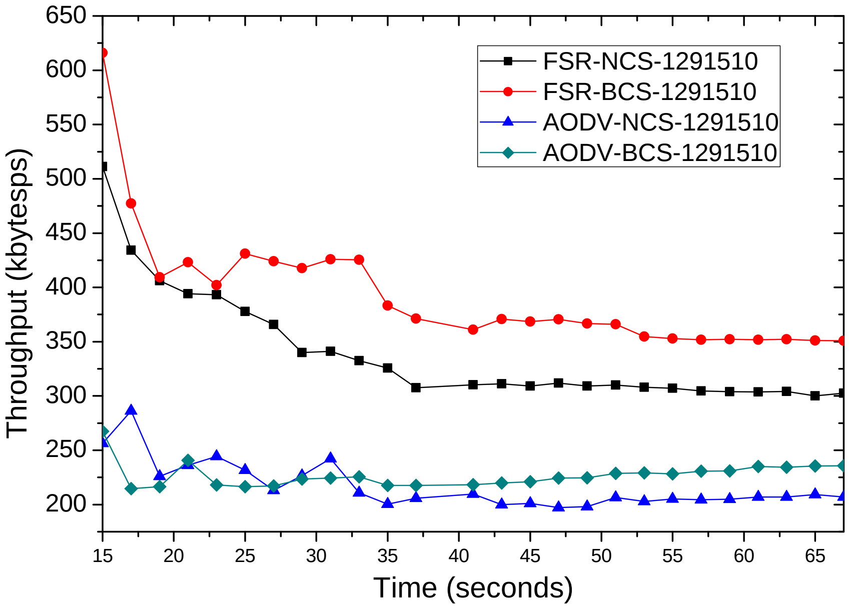


FIGURE 7.7: Throughput comparison between E-AODV and E-FSR

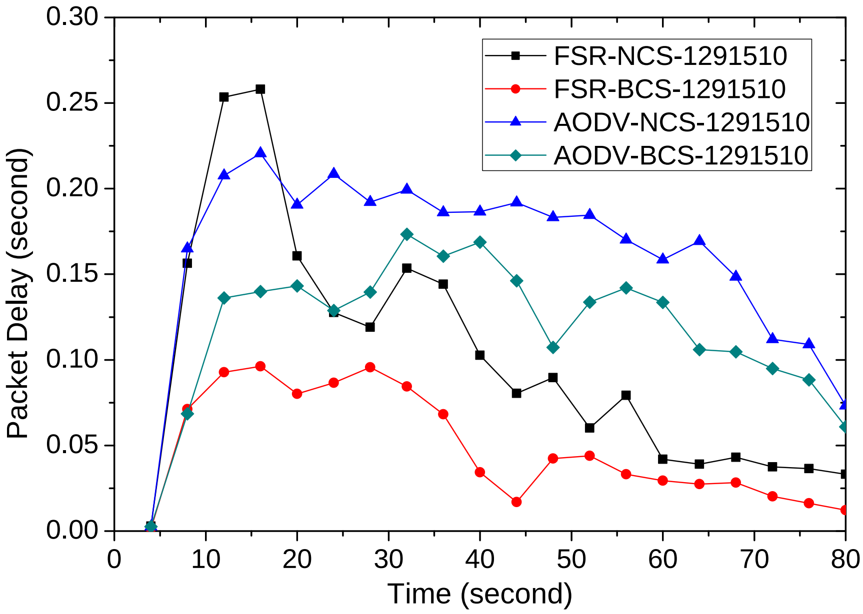


FIGURE 7.8: Packet delay comparison between E-AODV and E-FSR

Figure 7.7 shows the throughput comparison with different routing protocols: AODV and FSR. The performance of the FSR routing protocol is better than AODV routing protocol.

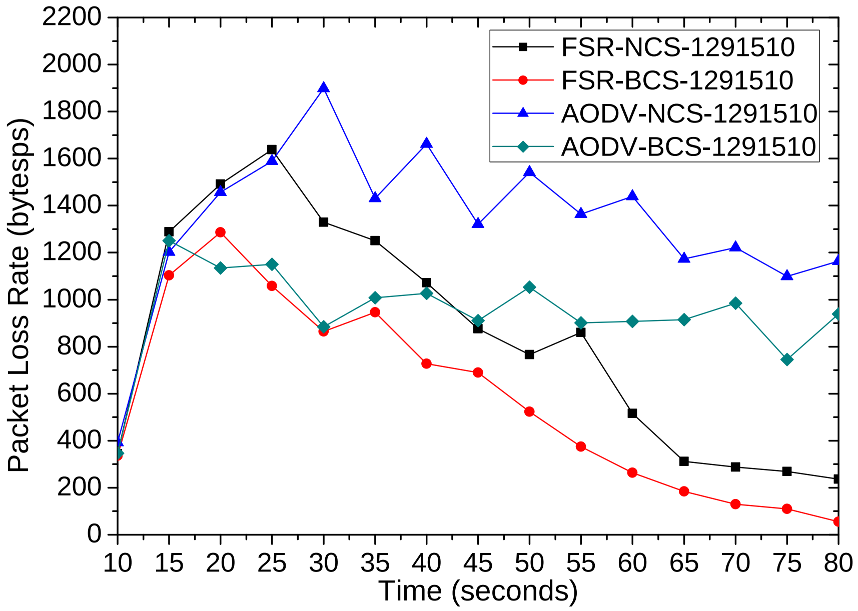


Figure 7.9: Packet loss rate comparison between E-AODV and E-FSR

We also present the packet delay and packet loss rate comparison in Figure 7.8. Since our solution continuously adjusts the channel according to the bandwidth and interference, the packet delay has also been decreased. The packet delay is increased before 15 seconds, and decreased thereafter. The reason for the situation is that we add the traffic, flow by flow, with 0.4 second intervals. After 15 seconds, 30 flows are stable in the network system, and no more traffic will be added in. But, our solution still adjusts the traffic until no more bandwidth exceeds $\mu_{i,j}(x)$. With the same condition, the channel switch control can quickly find the right parameter, making it more efficient. Also, the packet delay is lower with the FSR routing protocol.

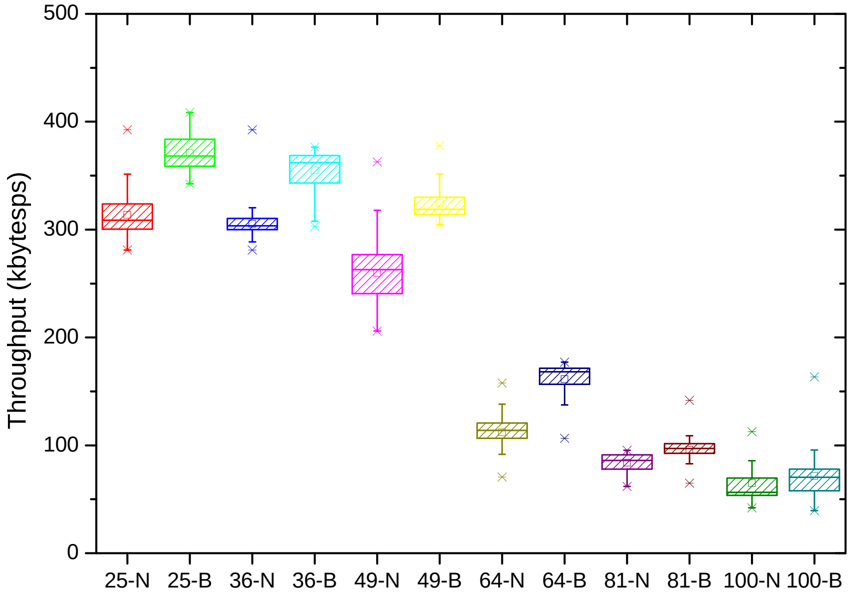


FIGURE 7.10: Throughput comparison with different network sizes

7.6 Performance Comparison with Different Network Size

In this subsection, we offer the simulation results of different network size. We use the parameter $\mu_{i,j}(0) = 891,510$. Since we still want to maintain the connectivity, the grid topology was adopted into the work. We use 5×5 to 10×10 as the network size. All of the experiments are conducted with the same traffic profile.

Figure 7.10 shows the simulation results regarding to different network size, where the label with N and B stand for NCS and BCS, respectively. The results are collected after 240 tests for all the simulations. As the traffic load are the same, the number of packets for each node is smaller. Thus, the throughput is lower. Using our BCS, the throughput can be improved by 10.5%- 34.2%. Similar results

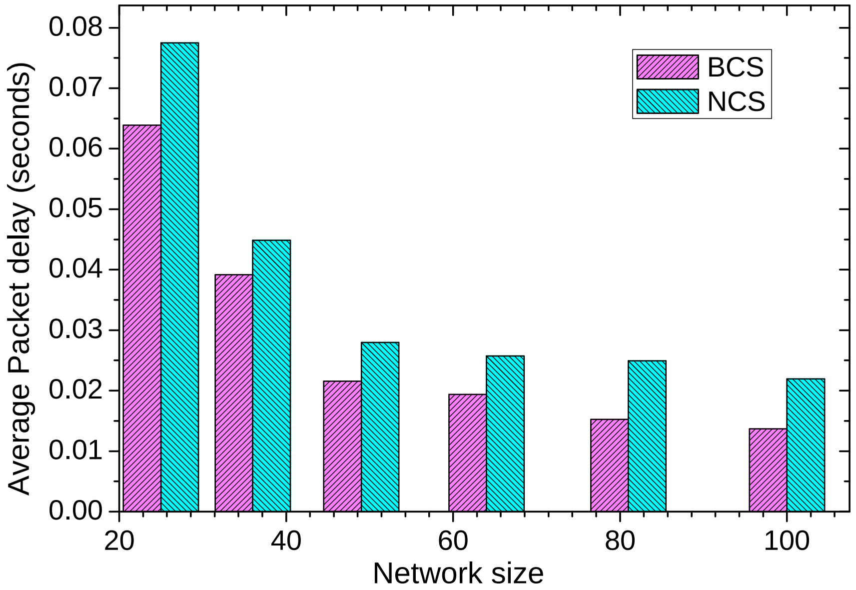


Figure 7.11: Packet delay comparison with different network sizes

Table 7.1: Confidential interval for throughput(kbps)

Node size	Low (NCS)	High (NCS)	Low (BCS)	High (BCS)
25	272	355	342	400
36	278	332	307	381
49	203	316	294	355
64	83	141	129	193
81	68	100	73	119
100	28	102	37	106

of average packet delay are also presented in Figure 7.11. The packets in larger network size have lower latency due to the less traffic load. With BCS, the packet delay can decrease by 14.6% -32.9%. Table 7.1 shows the confidential interval of throughput within 95%.

The simulation results are summarized as follows:

1. Both E-AODV and E-FSR have superior performance than regular AODV and FSR, respectively.

2. Our proposed balanced control system makes the system more efficient than the normal system in a dynamic environment. The simulation results show that throughput, packet loss rate and packet delay are all better than the system without control.

3. The initial parameter is usually difficult to decide. However, in our simulation study, the system performs better when working with a lower parameter $\mu_{i,j}(0)$.

4. Considering different routing protocols working with our channel switch control, the FSR routing protocol performs better than the AODV routing protocol.

CHAPTER 8

CONCLUSION AND FUTURE WORK

8.1 Conclusion

In this thesis, we proposed a novel threshold-based channel switching system, called balanced channel control system. In our design, the threshold could be dynamically deduced according to the real-time traffic and corresponding throughput of each node. Our threshold control model is based on a fuzzy logic control. We also designed a novel dynamic channel assignment scheme, which is used for the selection of new channel. To perform channel switch between a pair of neighboring nodes, we designed a channel switch scheduler. The channel switch scheduler is used to perform channel-switch processing for sender and receiver over enhanced routing protocols. We evaluated this system in NS2, and the simulation results showed that our BCS improves the throughput by 12.3%-72.8% and reduces the latency by 23.2%-52.3% over existing solutions. Our work is not confined with channel switch over a single link, it can be extended to ensure stability over a local region, and hence, the network as a whole. It is well known that interference in reality is very complicated.

8.2 Future work

We plan to conduct real system experiments with our BCS. The proposed system in this dissertation is based on the NS-2 simulation. Current product such as open-mesh and cognitive mesh, creates ultra low-cost, zero config, plug and play in wireless mesh networks which spread an Internet connection throughout everywhere. To manage these mesh points, the cloud controller has been adopted. Therefore, the current mesh network is also called cloud mesh. This is centralized solution. Based on the current techniques, the future work can be summarized as follows:

(1) Cloud computing refers to the delivery of computing and storage capacity as a service to a heterogeneous community. Lots of research works have been proposed in cloud computing. However, from the point view of wireless network, the integration of the two techniques are more complicated. Hybrid cloud solution is needed of cloud computing and cloud mesh.

(2) In current cloud mesh, we have two types of cloud: mesh private cloud and mesh enterprise cloud. The services of storage (in-home/out-home) and mesh connections can be more complicated.

Detailed study of this extension will be our future work.

BIBLIOGRAPHY

[1] P. Kyasanur and N. H. Vaidya, "Capacity of multi-channel wireless networks: impact of number of channels and interfaces," in *Proc. of ACM Mobicom*, 2005.

[2] J. Li, C. Blake, D. S. De C., H. I. Lee, and R. Morris, "Capacity of ad hoc wireless networks," in *Proc. of ACM Mobicom*, 2001.

[3] "Wireless lan medium access control (MAC) and physical layer (PHY) specifications IEEE 802.11 standard," *1999 Edition.*

[4] K. N. Ramachandran, E. M. Belding, K. C. Almeroth, and M. M. Buddhikot, "Interference-aware channel assignment in multi-radio wireless mesh networks," in *Proc. of IEEE Infocom*, 2006.

[5] A. Raniwala, K. Gopalan, and T.-C. Chiueh, "Centralized channel assignment and routing algorithms for multi-channel wireless mesh networks," *ACM SIGMOBILE MC2R*, vol. 8, 2004.

[6] P. Bahl, R. Chandra, and J. Dunagan, "Ssch: Slotted seeded channel hopping for capacity improvement in IEEE 802.11 ad hoc wireless networks," in *Proc. of ACM Mobicom*, 2004.

[7] J. Shi, T. Salonidis, and E. W. Knightly, "Starvation mitigation through multi-channel coordination in CSMA multi-hop wireless networks," in *Proc. of ACM Mobihoc*, 2006.

[8] J. So and N. H. Vaidya, "Multi-channel MAC for ad hoc networks: Handling multi-channel hidden terminals using a single transceiver," in *Proc. of ACM Mobihoc*, 2004.

[9] J. Tang, G. Xue, and W. Zhang, "Interference-aware topology control and qos routing in multi-channel wireless mesh networks," in *Proc. of ACM Mobihoc*, 2000.

[10] V. Raman and N. Vaidya, "Adjacent channel interference reduction in multi-channel wireless networks using intelligent channel allocatio," *Technical Report*, 2009.

[11] S. H. Y. Wong, H. Yang, S. Lu, and V. Bharghavan, "Robust rate adaptation for 802.11 wireless networks," in *Proc. of ACM Mobicom*, 2006.

[12] I. Pefkianakis, Y. Hu, S. H. Y. Wong, H. Yang, and S. Lu, "Mimo rate adaptation in 802.11n wireless networks.," in *Proc. of ACM Mobicom*, 2010.

[13] J. Zhu and S. Roy, "802.11 mesh networks with two-radio access points," in *Proc. of IEEE ICC*, 2005.

[14] G. Wu, S. Singh, and T. Chiueh, "Implementation of dynamic channel switching on IEEE 802.11-based wireless mesh networks," in *Proc. of the 4th Annual International Conference on Wireless Internet*, 2008.

[15] R. Maheshwari, H. Gupta, and S. Das, "Multichannel MAC protocols for wireless networks," in *Proc. of IEEE SECON*, 2006.

[16] N. Jain, S. Das, and A. Nasipuri, "A multichannel CSMA MAC protocol with receiver based channel selection for multihop wireless networks," in *Proc. ICCCN*, 2001.

[17] Y. Feng, M. Li, and M.-Y. Wu, "Improving capacity and flexibility of wireless mesh networks by interface switching," in *Proc. of IEEE ICC*, 2008.

[18] A. Raniwala and T.-C. Chiueh, "Architecture and algorithms for an IEEE 802.11 based multi-channel wireless mesh network," in *Proc. of IEEE Infocom*, 2005.

[19] J. So and N. Vaidya, "A routing protocol for utilizing multiple channels in multi-hop wireless networks with a single transceiver," *Technical Report, University of Illinois at Urbana-Champaign*, 2004.

[20] B.-J. Ko, V. Misra, J. Padhye, and D. Rubenstein, "Distributed channel assignment in multi-radio 802.11 mesh networks," in *Proc. of IEEE WCNC*, 2007.

[21] H. Skalli, S. Das, L. Lenzini, and M. Conti, "Traffic and interference aware channel assignment for multi-radio wirelesss mesh networks," *Technical Report, Institutions Markets Technologies*, 2006.

[22] S.-L. Wu, C.-Y. Lin, Y.-C. Tseng, and J.-L. Sheu, "A new multi-channel MAC protocol with on-demand channel assignment for multi-hop mobile ad hoc networks," in *Proc. of the International Symposium on Parallel Architectures, Algorithms and Networks*, 2000.

[23] A. Aziz, J. Herzen, R. Merz, S. Shneer, and P. Thiran, "Enhance & explore: an adaptive algorithm to maximize the utility of wireless networks," in *Proc. of the 17th annual international conference on Mobile computing and networking*, 2011.

[24] K. Kim and K. Shin, "Self-reconfigurable wireless mesh networks," *IEEE/ACM Transactions on Networking*, 2011.

[25] Y. Ding, K. Pongaliur, and L. Xiao, "Channel allocation and routing in hybrid multi-channel multi-radio wireless mesh networks," *IEEE Transactions on Mobile Computing*, 2011.

[26] Y. Ding, Y. Huang, G. Zeng, and L. Xiao, "Using partially overlapping channels to improve throughput in wireless mesh networks," *IEEE Transactions on Mobile Computing*, 2011.

[27] S. Chieochan and E. Hossain, "Channel assignment for throughput optimization in multi-channel multi-radio wireless mesh networks using network coding," *IEEE Transactions on Mobile Computing*, 2011.

[28] C.-F. Kuo, H.-W. Tseng, and A.-C. Pang, "Fuzzy-based cross-layer transmission scheme with QoS considerations for wireless mesh networks," in *Proc. of IWCMC*, 2009.

[29] C. Perkins, E. Belding-Royer, and S. Das, "Ad hoc on-demand distance vector (AODV) routing," *Request For Comments (RFC)*, 2003.

[30] G. Pei, M. Gerla, and T. Chen, "Fisheye state routing in mobile ad hoc networks," in *Proc. of IEEE ICDCS Workshop on Wireless Networks and Mobile Computing*, 2000.

[31] X. Li and C. Xu, "Joint channel assignment and routing in real time wireless mesh network," in *Proc. of IEEE WCNC*, 2009.

[32] J. Padhye, S. Agarwal, V. Padmanabhan, L. Qiu, A. Rao, and B. Zill, "Estimation of link interference in static multi-hop wireless networks," in *Proc. of ACM SIGCOMM conference on Internet Measurement*, 2005.

[33] A. Subramanian, H. Gupta, and S. Das, "Minimum interference channel assignment in multi-radio wireless mesh networks," in *Proc. of IEEE SECON*, 2007.

[34] K. M. Passino and S. Yurkovich, "Fuzzy control," *Addison Wesley Longman*, 1998.

[35] K. Lee, "First course on fuzzy theory and applications," *Advances in Soft Computing*, 2005.

[36] P. Kyasanur and N. Vaidya, "Routing and interface assignment in multi-channel multi-interface wireless networks," in *Proc. of IEEE WCNC*, 2005.

[37] H. Matsumura, "Commutative algebra," *Addison Wesley*, 1998.

[38] "Wireless medium access control (MAC) and physical layer (PHY) specifications: ESS mesh networking," *IEEE P802.11s/D1.00*, 2006.