# 國立交通大學

## 網路工程研究所

## 碩士論文

# IEEE 802.11 無線網狀網路下單一天線多頻道管理之協定設計

# A Channel Management Protocol for Multi-Channel, Single-Radio, 802.11-Based Wireless Mesh Networks

研究生：呂書賢

指導教授：曾煜棋　教授

中華民國 九十七 年六月

# IEEE 802.11 無線網狀網路下單一天線多頻道管理之協定設計
# A Channel Management Protocol for Multi-Channel, Single-Radio, 802.11-Based Wireless Mesh Networks

研究生：呂書賢　　Student：Shu-Hsien Lu

指導教授：曾煜棋　　Advisor：Yu-Chee Tseng

國立交通大學
網路工程研究所
碩士論文

A Thesis
Submitted to Institute of Network Engineering
College of Computer Science
National Chiao Tung University
in partial Fulfillment of the Requirements
for the Degree of
Master
in

Computer Science

June 2008

Hsinchu, Taiwan, Republic of China

中華民國九十七年六月

# 國立交通大學

## 研究所碩士班

## 論文口試委員會審定書

本校　網路工程　研究所　呂書賢　君

所提論文:

IEEE 802.11 無線網狀網路下單一天線多頻道管理之協定設計

合於碩士資格水準、業經本委員會評審認可。

口試委員：賴源煌　曾煜棋

曾建超　張志勇

指導教授：曾煜棋

所　　長：曾建超

中華民國九十七年　六　月　十七　日

**Institute of Network and Engineering**
College of Computer Science
National Chiao Tung University
Hsinchu, Taiwan, R.O.C.

As members of the Final Examination Committee, we certify that we have read the thesis prepared by Shu-Hsien Lu

entitled A Channel Management Protocol for Multi-Channel, Single-Radio, 802.11-based Wireless Mesh Networks

and recommend that it be accepted as fulfilling the thesis requirement for the Degree of Master of Science.

Committee Members:

Chien-Chao Tseng    Chih-Yung Chang

Thesis Advisor:

Director: Chien-Chao Tseng

Date: 2008/06/17

# IEEE 802.11 無線網狀網路下單一天線

# 多頻道管理之協定設計

學生: 呂書賢　　　　指導教授: 曾煜棋 教授

國立交通大學網路工程研究所碩士班

## 摘　　要

無線網狀網路是下一代無線通訊中蓬勃發展的技術之一。在展現更大範圍的網路覆蓋率的同時，無線網狀網路也必須面對多重跳躍網路所需要面對的延伸問題（像是吞吐量、延遲跟封包傳輸率等）。對於這一些問題來說，使用多頻道是一個有效率的解決方法。在本篇論文中，我們提出了一個在單一天線多頻道的無線網狀網路環境下的頻道管理協定。在這個環境下，有兩個主要的問題:頻道配置跟頻道切換排程。目前我們常可以看到的解法都是讓利用變化每個節點的行為來試圖解決這兩個問題，但是，我們的觀察發現在上述網路環境下，有很多的節點其實是不需要考量頻道排程切換的問題。事實上，在有許多節點不做頻道切換的網路環境下，我們還是可以同時保有多頻道協定所帶來的好處（頻道空間再使用），在本篇論文中，我們將網狀網路上的節點分成兩個集合，在第一個集合上的節點都只需要待在某一個固定的頻道上進行封包的傳輸，而第二個集合上的節點會在頻道之間做切換。經由這樣的設計，我們可以降低頻道切換的負擔（如切換延遲跟同步飄移等），除了經由模擬來驗證這多頻道管理協定的效能，我們還將我們的協定實做在真實的無線網狀網路平台上（RTL8186）。

關鍵字:頻道管理協定、IEEE 802.11、IEEE 802.11s、無線網狀網路、無線網路。

# A Channel Management Protocol for Multi-Channel, Single-Radio, 802.11-based Wireless Mesh Networks

Student: Shu-Hsien Lu Advisor: Prof. Yu-Chee Tseng

Department of Computer Science
National Chiao-Tung University

## ABSTRACT

Wireless mesh networks (WMNs) have emerged as a key technology for next-generation broadband wireless access. While benefiting from larger coverage, WMNs also suffer from some scalability problems in terms of throughput, delay, and packet delivery ratio faced by all multi-hop wireless networks. Using multiple channels is a cost-effective solution to these problems. In this paper, we propose a channel management protocol for multi-channel, single-radio 802.11-based wireless mesh networks. In such an environment, two issues should be addressed: *channel assignment* and *channel switch scheduling*. Therefore, most existing multi-channel solutions involve a channel assignment scheme and a scheduling scheme to determine nodes' behaviors at different times. However, we observe that channel switching of some mesh points may not be necessary under this environment. In fact, event without channel switching, the benefit of using multiple channels (spatial reuse) can still be achieved. In this work, mesh points are divided into two sets. Those in the first set will be assigned channels and stay in the same channels to transmit/receive packets. On the other hand, those in the second set will switch to proper channels dynamically. With this design, a lot of channel switching overheads (switching delay, synchronization drift, etc.) can be mitigated. Extensive simulations are conducted to verify the efficiency of the proposed protocol. We also implemented our proposed protocol on the RTL8186 platform.

Keywords: channel management protocol, IEEE 802.11, IEEE 802.11s, wireless mesh network (WMN), wireless network.

# 誌謝

由衷地感謝曾煜棋教授這兩年來不辭辛勞的指導，讓我學到了很多寶貴的知識跟經驗還有為人處事的方法，感謝實驗室的學長姐跟學弟妹，謝謝你們這兩年多來的幫忙跟提攜，讓我得以順利完成學業，感謝爸媽辛苦地工作，讓我在求學的過程中沒有後顧之憂，感謝瑞昱交大研發中心的同仁們，讓我學習了團隊合作的技巧跟職場的經驗，感謝常常跟我一起打球的好朋友們，讓我除了學業之外能有個豐富的生活，最後，特別感謝潘孟鉉學長跟蔡佳宏學長，謝謝你們總是在我心情低落時拉我一把，告訴我正確的人生態度跟處理事情的方法，真的讓我獲益良多。

在此向大家獻上我誠摯地謝意跟祝福

呂 書 賢 謹識於
國立交通大學網路工程研究所碩士班
中華民國九十七年七月

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

Wireless Mesh Networks (WMNs) provide a possibility to offer broadband wireless access for the areas where wired infrastructure is not available or not worthy to deploy [6]. Moreover, due to its self-configuring and self-healing capabilities, a WMN can be deployed and maintained easily. A WMN consists of a collection of mesh points, which cooperatively form a multi-hop backbone network. In this paper, we focus on IEEE 802.11-based WMNs where mesh points communicate with each other by IEEE 802.11 radio interfaces, and end users can connect to mesh points, which will forward traffics on behalf of them. Furthermore, some mesh points can serve as gateways to provide Internet access. An example of a WMN is shown in Fig. 1.1.

While benefiting from larger coverage, WMNs also suffer from some scalability concerns in terms of throughput, delay, and packet delivery ratio faced by other multi-hop wireless networks [11]. There are many approaches to addressing these concerns, such as using directional antennas, adopting transmission power control, assisting by location information, exploiting multiple channels, etc. Observing that most existing wireless interfaces provide multiple channels [14, 15], we look for cost-effective, multiple-channel solutions for WMNs. Routing metrics and routing protocols under multi-channel environments have been studied in [12, 10, 23, 18]. Multi-channel MAC protocols are addressed in [28, 24, 5, 7]. Considering the number of radios per node, these protocols can be further categorized on *single-radio* and *multi-radio* schemes.

Taking practicality and cost effectiveness into consideration, this work proposes a multi-channel, single-radio solution for IEEE 802.11-based WMNs. Because each node has only one radio, it has to switch its operational channel periodically to ensure network connectivity when multiple channels are used. Usually, two issues should be addressed. One is *channel assignment* and the other is *channel switch*

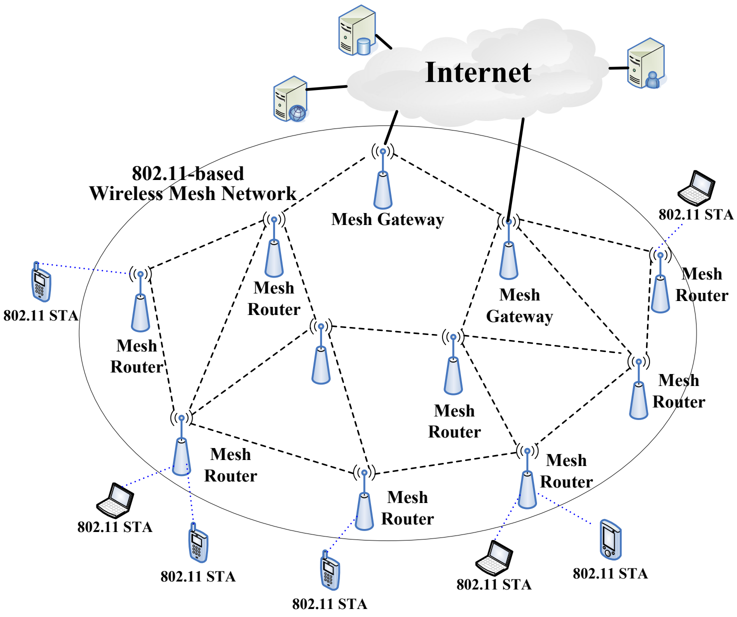


Figure 1.1: An example of a WMN.

*scheduling*. In most literatures, these two issues are considered simultaneously, enforcing to hop among a set of channels based on a schedule.

In this paper, we propose a multi-channel, single-radio (MCSR) protocol for tree-based WMNs. We observe that even if some mesh points do not switch their operational channels, the network connectivity and the benefit of using multiple channels can still be achieved. This may significantly reduce a lot of channel-switching overheads. In our protocol, mesh points are divided into two sets. Mesh points in the first set will be assigned fixed channels to transmit/receive packets. On the other hand, mesh points in the second set will switch their operational channels dynamically according to their communication counter parts. We will develop three channel assignment algorithms to allocate channels to the mesh points in the first set. We will also develop two channel switching algorithms to help mesh points in the second set to determine their channel hopping schedule. A special feature of our protocol is that a mesh point needs to perform either channel assignment algorithm or the channel switching algorithm, but not both. In fact, our scheme does enforces the first set of mesh points to interleave with the second set of mesh points in the tree topology. This actually simplifies the design of our channel switching algorithms because whenever the second set of nodes want to transmit/receiver packets, they can switch to channels of the first set without considering their channel switching patterns. By this design, we believe that the side-effect of channel switch (such as channel switch delay, synchronization drift, etc.) can also be significantly mitigated.

The rest of this paper is organized as follows. Related work is discussed in Chap. 2. Chap. 3 presents our protocols. Simulation results are in Chap. 4. Our implementation results are discussed in Chap. 5. Conclusions are drawn in Chap. 6.

# Chapter 2

# Related Works

In wireless networks, a vast amount of researches on multi-channel protocols have been proposed. To begin with, we focus on the multi-channel MAC protocols. Multi-channel MAC protocols have been studied intensively [28, 24, 5, 7, 17, 20, 21, 26, 22, 19, 8, 29]. In [28], a multi-channel MAC protocol called DCA was proposed in which two radios are required. One radio is assigned to a common channel and is used for channel negotiation. The other radio then uses the negotiated channel to transmit data packets. In [22], a multi-channel MAC protocol called PCAM was proposed. The idea of PCAM is similar to DCA but each device equips three transceivers. Two of the transceivers are used for packet transmissions and the third radio is used for transmitting and receiving broadcast messages. PCAM allows all three transceivers to transmit packets in some special cases. In transmitter-based multi-channel MAC protocols [20, 21], the receiver will listen on $N$ channels and the transmitter will choose one best channel for each packet transmission. Because the receiver has to listen on $N$ channels simultaneously, multiple radios are required. In [17], a receiver-based multi-channel MAC protocol was proposed in which the receiver will determine the channel used for packet transmissions. All of the protocols introduced above need more than one radio. This will lead to higher hardware cost.

In [26], a single-radio, multi-channel MAC protocol called RICH-DP was proposed. Nodes in the network have to follow a common channel-hopping sequence. Thus, time synchronization is required. In [24], a single-radio, multi-channel MAC protocol called MMAC was proposed which is designed based on the IEEE 802.11 power-saving mechanism (PSM) used in ad hoc mode[14]. In MMAC, the sender the receiver will negotiate a channel during ATIM (Ad Hoc Traffic Indication Map) window and then switch to the negotiated channel at the end of the ATIM window. Time synchronization is also required. In [7], a single-radio, multi-channel link-layer protocol called SSCH was proposed which does not modify the original IEEE 802.11 MAC protocol. In SSCH, time is divided into slots. Each node will follow a rule to determine the channels used in the slots. Time synchronization is required. In addition, how to support broadcast packets is not addressed. All of the protocols introduced above neither address the routing nor take the network structure into consideration.

Then, we introduce some multi-channel protocols developed for tree-based WMNs. In [9], a MAC protocol called Ripple was proposed for tree-based WMNs. Each mesh point equips with two radios. One radio is for uplink traffic and the other is for downlink traffic. Ripple does not utilize non-overlapping channels well because it only uses two static non-overlapping channels. In [23], a multi-channel WMN architecture called Hyacinth was proposed that combines a tree-based, load-balancing routing protocol and a channel assignment protocol. In Hyacinth, radios on a node are classified into two classes: UP-NICs and DOWN-NICs. UP-NICs of a node are used to communicate with the node's parent and the channels used by the UP-NICs are determined by its parent. On the other hand, DOWN-NICs of a node are used to communicate with the node's children and the channels used by DOWN-NICs are determined by the node itself. In [30], a tree-based channel assignment with sibling-aware routing was proposed for multi-channel WMNs. Each mesh point equips with two radios. One radio is assigned a static channel for receiving packets and the other will switch its operational channel dynamically for transmitting packets. All of the protocols introduced above assume that a node has to equip more than one radio.

In [27], a component-based channel assignment was proposed for single-radio, multi-channel ad hoc networks. However, the proposed protocol is not designed for WMNs where traffic flows are usually generated from mesh points to the gateway (wired networks) or from the gateway to the mesh points. In [25], a cross-layer protocol called JMM was proposed for single-radio, multi-channel WMNs. JMM combines multi-channel link layer with multi-path routing. For the link layer part, a receiver-based channel assignment protocol was proposed in which each mesh point will choose the least-used channel as its receiving channel. When a node $x$ intends to send packets to another node $y$, $x$ has to switch its operation channel to $y$'s receiving channel. Nodes are allowed to determine their transmitting/receiving patterns dynamically, but how to ensure the communication pair can use the same channel is not addressed. For the routing part, a tree-based, multi-path routing protocol was proposed. The benefit of using multiple paths can be achieved only when the traffic is not high so that we can find two low-traffic paths. Besides, using multiple paths may lead to out-of-order arrival of packets at the destination node.

# Chapter 3

# The Proposed Channel Management Protocol

In [25], it has been shown that in a chain topology an ideal multi-channel MAC protocol could achieve an end-to-end throughput as high as $\frac{1}{2}$ of the effective MAC data rate in a single-radio, multi-channel WMN. Fig. 3.1 shows two possible channel switching schedules. In Fig. 3.1(a), each node switches between two channels. Continuous packets sent from A to F can form a perfect pipeline effect separated by two hops. Interestingly, the schedule in Fig. 3.1(b) has a similar effect, but only half of the nodes need to switch channels. The other half can stay in the same channels. With this design, a lot of channel switching overhead (such as switching delay, synchronization drift, etc.) can be mitigated.

Our design will also lead to the following benefits:

- *Improved Channel Reuse*: The very idea of channel reuse is to exploit the most compact spatial reuse pattern of each channel. However, if a node owns a fixed channel without using it (for examples, it may switch to another channel to communicate with other nodes), then that channel could be wasted. Our scheme can effectively avoid this problem because only some nodes have fixed channels. In fact, our scheme has the concept of compactness in mind when choosing those nodes with fixed channels.

- *Mitigated Collision*: Since our scheme enforces nodes with fixed channels to interleave with those without fixed channels, a lot of multi-channel hidden-terminal and missing-receiver problems can be avoided.

- *Simplified Channel Switching*: Since nodes without fixed channels will only

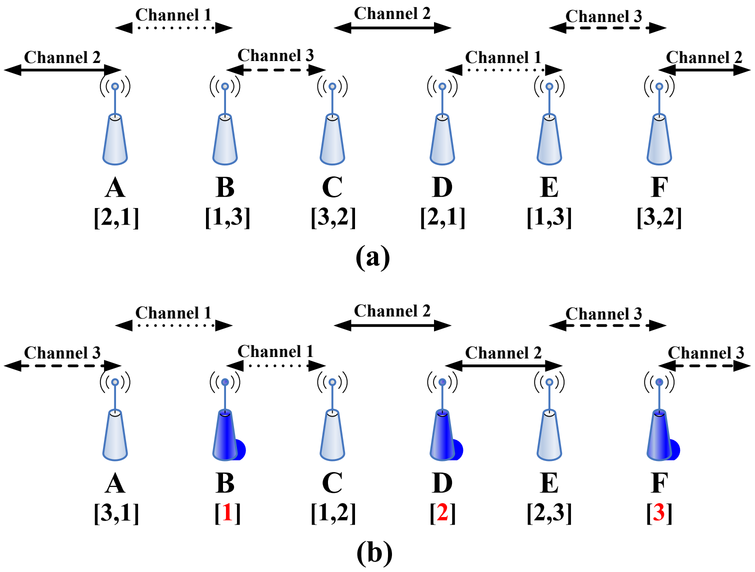


Figure 3.1: Two ideal channel switching schedules in a chain WMN that can achieve the optimal pipeline effect.

communicate with those with fixed channels, the former can always find the latter (at the right channels) whenever they intend to. This will significantly simplify our design of channel switching.

1896

## 3.1 Network Model

We consider an 802.11-based WMN with one mesh point serving as the gateway. We assume that the WMN is mainly for providing Internet services, so most traffics will go through the gateway. There are multiple non-overlapping channels available. Each mesh point has one wireless radio, except that for performance reason the gateway has multiple radios equal to the number of available channels.

For such a multi-channel, single-radio WMN, we need a channel management protocol to help a mesh point to determine its channel and a channel scheduling protocol to determine its channel-hopping pattern. The main idea of our work is to divide mesh points into two sets $V$ and $V'$. Each mesh point in $V$ will be assigned a *fixed* channel, while each in $V'$ will not have a fixed channel but will switch to a proper *operational* channel channel at proper time.

To determine $V$ and $V'$, we will construct a tree $T$ rooted at the gateway (any such algorithm, such as HWMP[13], can be used here). Let the root be at level 0. Then, mesh points in the odd levels of $T$ will be assigned to $V$, and those in the

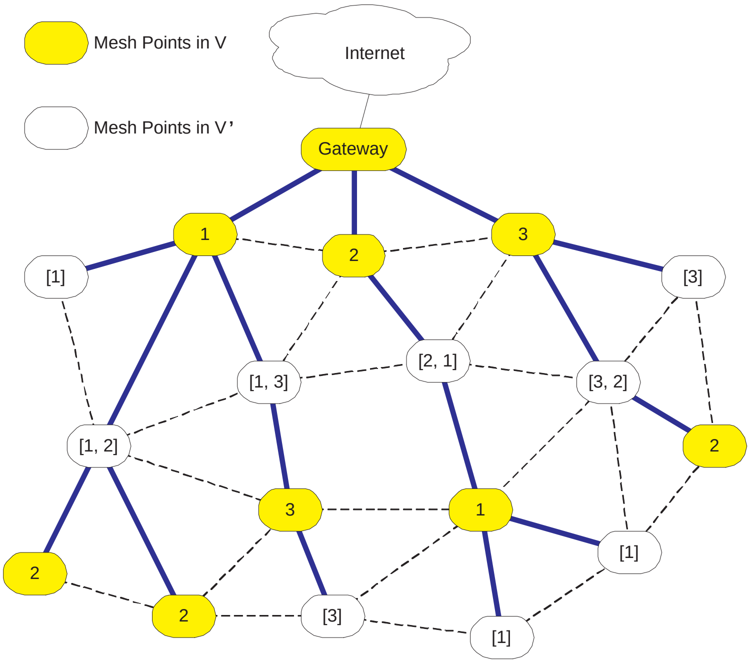


Figure 3.2: An example of our proposed network architecture. Solid lines are tree edges and the dotted lines mean neighboring relations. The numbers labeled on the mesh points in $V$ denote their fixed channels and the numbers labeled on the mesh points in $V'$ are their switchable operational channels.

even levels of $T$ will be assigned to $V'$. An example is shown in Fig. 3.2.

Our scheme works in a TDMA-over-CSMA manner. Time is divided into equal-length frames. Each frame consists of $k$ slots. The first slot is used not only for supporting broadcast but also for sending out the channel switching schedules for nodes in $V'$. So points must switch to the same default channel. For the other k-1 slots, node in $V$ will go to their fixed channels (refer to Chap. 3.2 and nodes in $V'$ will switch to different slots accoring to schedules (refer to Chap. 3.3). Fig. 3.3 shows the frame structure. Inside each slot, node will contend the medium using a typical CSMA protocol (such as IEEE 802.11). We assume that each slot is large enough to accommodate several packet transmissions.

## 3.2 Channel Assignment Protocol for $V$

During general slots, each mesh point $x$ in $V$ will own a fixed channel for transmissions. We propose three strategies for $x$ to choose its fixed channel.

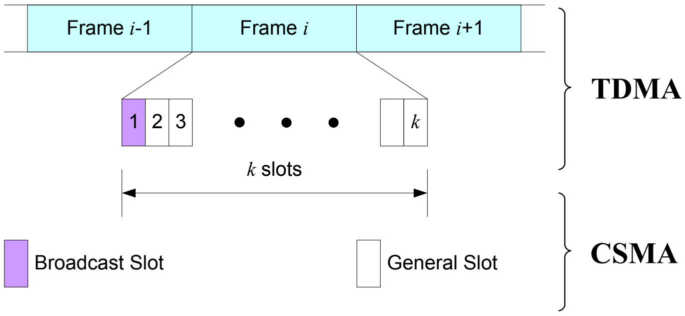


Figure 3.3: The frame structure of our protocol.

- ***Interference-Based Strategy*** : The operation will be triggered starting from the gateway. When $x$ is triggered and gets all of its 2-hop neighbors's permissions, it will choose the least used channel by its 2-hop neighbors as its fixed channel to minimize potential interference. However, $x$ should also avoid using the same channel as its grandparent to prevent collisions at $x$'s parent. Then $x$ will trigger its 2-hop neighbors in $T$ to choose their fixed channels. Fig. 3.4(a) shows a partial network with $y \in V'$. Assuming four available channels, Fig. 3.4(b) shows the assignment result.

- ***Delay-Based Strategy***: The above strategy intends to minimize interference, but it may lead to long delay for $y$ because $y$ has to switch among four operational channels. To relieve this problem, this strategy tries to enforce all children of $y \in V'$ to choose the same fixed channel (refer to Fig. 3.4(c) for an example). Therefore, $y$ only needs to switch between two channels. The operation is also triggered from the gateway. When a node $x \in V$ is triggered, it will try to get all of its 2-hop neighbors' permission to choose a channel. The channel-selecting rules are as follows:

    1. If $x$ is a child of the gateway or none of its siblings has owned a channel, it will choose the least used channel among its 2-hop neighbors except the one used by its grandparent as its fixed channel.
    2. Otherwise, $x$ chooses the same channel used by any of its siblings as its fixed channel.

- ***Hybrid Strategy***: It can be seen that although the delay-based strategy can decrease delay, the corresponding interference among $y$'s childrenmight be raised. In fact, latency and interference are conflicting factors. To balance these concerns we propose a hybrid strategy (refer to the example in

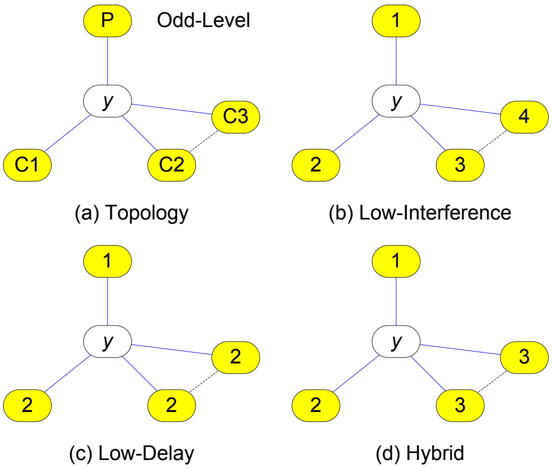
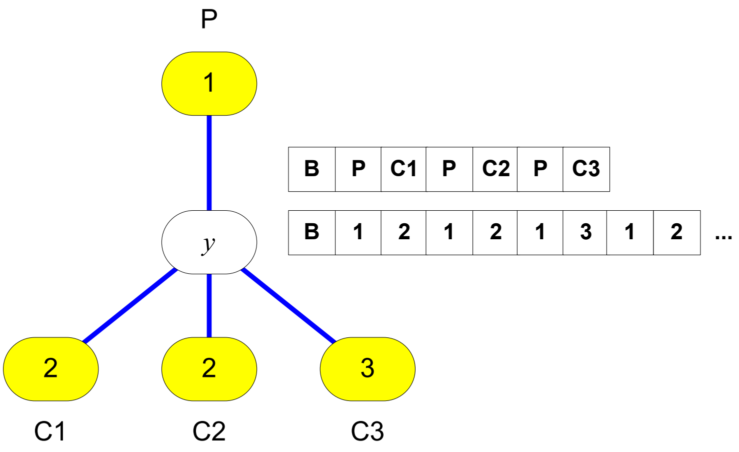


Figure 3.4: Examples of different channel assignment strategies. (a) Network topology. (b) Interference-based strategy. (b) Delay-based strategy. (c) Hybrid Strategy.

Fig. 3.4(d)). The triggering operation is the same. When any $x \in V$ gets the permissions of all its 2-hop neighbors, the following rules are checked.

1. If $x$ is a child of the gateway or none of its siblings has owned a channel, it will choose the least used channel among its 2-hop neighbors except the one used by its grandparent as its fixed channel.
2. Otherwise, if one of $x$'s one-hop siblings has owned a fixed channel, it will choose that channel as its fixed channel.
3. Otherwise, only $x$'s two-hop siblings have owned fixed channels. Then $x$ will choose the least used channel (which is different from its two-hop siblings) among its two-hop siblings as its fixed channel.

It remains to state how a node gets permissions from its 2-hop neighbors to choose channel. Let $x.ID$ and $x.LVL$ be the ID and level of $x$, respectively. For two mesh points $x$ and $y$, we say $x < y$ if $(x.LVL < y.LVL) \vee (x.LVL = y.LVL \wedge x.ID < y.ID)$. A node is triggered when it receives a $CHL\_GRANT$ message. When an odd-level node $x$ receives the $CHL\_GRANT$, it will send a $CHL\_REQ$ to its one-hop and two-hop neighbors. When a mesh point $y$ receivers a $CHL\_REQ$ from $x$, it will send a $CHL\_GRANT$ to $x$ if $y.LVL$ is even. If $y$ does not own a channel yet and $y > x$, $y$ will send a $CHL\_GRANT$ to $x$. Otherwise, $y$ will keep on waiting for others' $CHL_GRANT$s. Once $x$ receives all $CHL_GRANT$s from all its one-hop and two-hop neighbors, it can select its fixed channel based on the earlier strategies. After that, $x$ will send a $CHL\_GRANT$ to all its one-hop and two-hop neighbors.



|  |  |  |  |  |  |  |  |  |  |
|---|---|---|---|---|---|---|---|---|---|
| B | P | C1 | P | C2 | P | C3 |  |  |  |
| B | 1 | 2 | 1 | 2 | 1 | 3 | 1 | 2 | ... |

Figure 3.5: An example of the Ripple strategy. Note that the fixed channels of $C1$ and $C2$ are the same.

Note that when packet loss is possible, a node can proactively ask for $CHL_GRANT$s from its neighbors.

## 3.3 Channel Switching Protocol for $V'$

Next, we discuss how a mesh point $y \in V'$ switches its operational channel in the $k-1$ general slots of a frame. It is clear that if $y$ is a leaf node, it is sufficient to always set its operational channel to its parent's fixed channel. So we only consider a non-leaf $y$ below. We propose two strategies. The first *Ripple* strategy tries to switch $y$'s operational channel regularly to allow packets to be forwarded to or propagated from the gateway. Since this scheme has less flexibility, the second *Dynamic* scheme allows $y$ to switch its operational channel dynamically based on traffic conditions but at the cost of some control packets.

### 3.3.1 Ripple Strategy:

We let $y$ will alternate between its parent's fixed channel and one of its children's fixed channels. (Recall that $y$'s parent and children are in $V$.) Note that $y$ will serve its children in a round-robin fashion. So, $y$ will switch to its parent's channel in the even slots and to its children's channels alternately in the odd slots. An example is shown in Fig. 3.5.

The ripple strategy has two advantages. First, no control packets are required. Second, packets will be transmitted upwards or downwards in a regular ripple man-

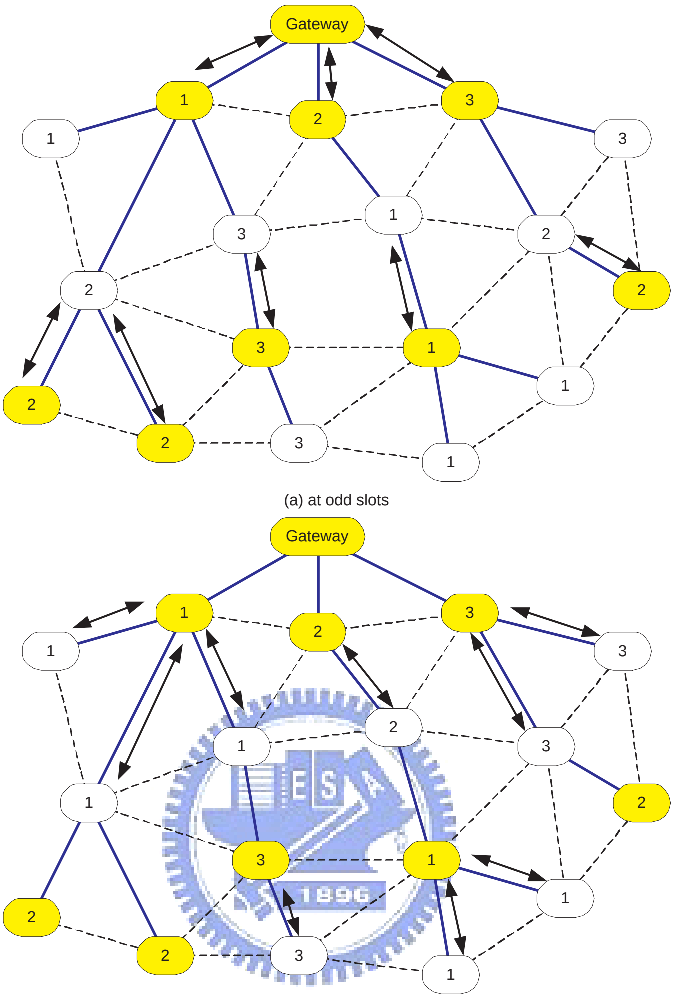


Figure 3.6: An example of ripple effect in the ripple scheme.

ner as shown in Fig. 3.6. This would also relieve the collision among nodes. However, the strategy is less flexible because it cannot adapt to traffic variations.

### 3.3.2 Dynamic Strategy:

In this strategy, traffic conditions will be exploited. In the first slot of each frame, each node $y \in V'$ will announce its channel switching schedule. Let $PC(y)$ be the set of $y$'s parent and children. We define two terms. For each $z \in PC(y)$, let $f(y, z, i)$ be the traffic load between $y$ and $z$ estimated by $y$ in frame $i$ (this can be measured by, say, the number of packets transmitted between them and the backlog at $y$ destinated to $z$). Further, we define $F(y, z, i)$ as the weighted traffic load between $y$ and $z$ estimated by $y$ until the $i$-th frame. such that

- For each $z \in PC(y)$, define $F(y, z, i)$ as the weighted traffic flow between $y$ and $z$ estimated by $y$ until the $i$-th frame. More specifically,

$$F(x, y, i) = \begin{cases} \alpha \times f(y, z, i-1) + (1-\alpha) \times F(y, z, i-1) & , i > 2 \\ f(y, z, 1) & , i = 2 \\ 1 & , i = 1 \end{cases} \tag{3.1}$$

  where $0 \leq \alpha \leq 1$.

Based on $F(y, z, i)$, $y$ calculates its slot allocation as follows. First, it will reserve one slot for each member in $PC(y)$. Here we assume that $(k-1) \geq |PC(y)|$. Then, the remaining slots will be allocated evenly to members in $PC(y)$. More precisely, the number of slots allocated to $z \in PC(y)$ will be

$$\left\lfloor ((k-1) - |PC(y)|) \times \frac{F(y, z, i)}{\sum_{\ell \in PC(y)} F(y, \ell, i)} \right\rfloor + 1. \tag{3.2}$$

Note taht since a floor function is used, some slots will actually be free. We can distribute these slots to $PC(y)$ based on some round-robin scheme. Also, locations of these slots can be permuted with others' slots to ensure fairness.

# Chapter 4

# Performance Evaluation

We have developed a simulator to verify the performance of our channel management protocol by executing comprehensive simulations. We compare our protocol with the single-channel scheme and the modified version of JMM[25] that will be presented later.

Project authorization request (PAR) of IEEE 802.11 TGs [16] said that a target configuration is up to 32 devices participating as AP forwarders in the ESS Mesh. Besides, many existing mesh network solutions such as Cisco Mesh Network[2] and Tropos Mesh Network[4] also follow the idea of small-scale mesh networks. Thus, two small-scale scenarios are simulated. In the first scenario, 25 mesh points are deployed as a $5 \times 5$ grid network. In the second scenario, 30 mesh points are deployed randomly with uniform distribution.

We assume that each node equips only one radio except for the gateway equipping 3 radios and 3 non-overlapping channels are available. Two types flows are simulated: up-link flow and down-link flow. In up-link flow, traffic flows from the mesh points to the gateway. In down-link flow, traffic flows from the gateway to the mesh points.

Unless stated otherwise, 80% down-link and 20% up-link flows are used in our experiments. The impact of channel switching is an important issue. Thus, channel switch delay and time synchronization error are simulated. Besides, each result is achieved by averaging 20 runs. The default values of parameters are shown in Table 4.1.

Table 4.1: The default values of parameters used in the simulation.

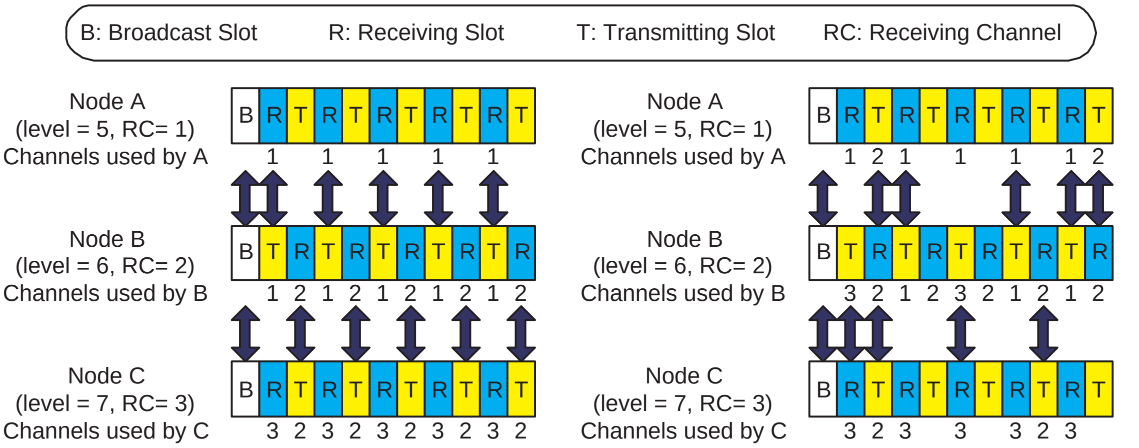
| | |
|---|---|
| *Underlining MAC* | IEEE 802.11b with RTS/CTS |
| *Simulation Area* | 100 units x 100 units |
| *Transmission range* | 25 units ∼ 45 units |
| *Link capacity* | 2 Mbps |
| *Flow data rate* | 300 kbps (for each flow) |
| *Channel Switch Delay* | 12 ms |
| *Slot size* | 400 ms |
| *Data frame size* | 512 bytes |
| *Number of available channels* | 3 |
| *Number of slots in one frame (the broadcast slot is included)* | 11 |
| *Simulation Time* | 200 s |
| *The buffer size* | 200 packets (per node) |

## 4.1 The Operation of Modified JMM

In this section, we give a detailed description on how we modify JMM for making it work with the two proposed channel scheduling schemes. Because we do not address the routing issue in this paper, we only focus on the link layer part of JMM. In JMM, time is divided into slots. There are two types of slots: transmitting slot and receiving slot. JMM adopts a receiver-based channel assignment scheme in which each mesh point will choose the least-used channel as its receiving channel. Thus, in a node's receiving slot, the node will use its receiving channel to transmit/receive packets. On the other hand, in a node's transmitting slot, the node will switch its operational channel to the channel used by its intended receiver.

In JMM, the numbers of transmitting and receiving slots of a node can be adjusted dynamically. However, this may lead to the unmatch problem in which the sender cannot communicate with its intended receiver. Thus, in simulation, we assume that the transmitting and receiving slots of all mesh points are fixed and interleaved. (Note that the odd-level nodes and the even-level nodes will use different transmitting/receiving slots patterns.) Examples of the operation of the modified JMM are shown in Fig. 4.1.

The modified JMM can be combined with the proposed ripple scheduling scheme easily. When a mesh point is in its transmitting slot, it only can communicate with its parent. On the other hand, when the mesh point is in its receiving slot, it only can communicate with its children. By doing so, we can combine the JMM with the



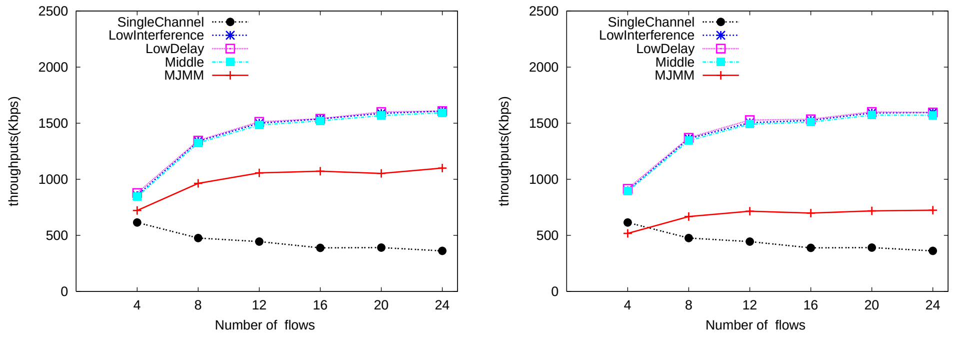
(a) Modified JMM with the ripple scheduling scheme (b) Modified JMM with the dynamic scheduling scheme

Figure 4.1: An example of the operation of the modified JMM. (a) MJMM with the ripple scheduling scheme. (b) MJMM with the dynamic scheduling scheme.

ripple scheduling scheme. An example is shown in Fig. 4.1(a). The modified JMM also can be combined with the proposed dynamic scheduling scheme. We do not have to change the transmitting/receiving patterns of nodes, but we allow a node can communicate with either its parent or its children in its transmitting slot. An example is shown in Fig. 4.1(b). Later, MJMM is referred to the modified JMM.

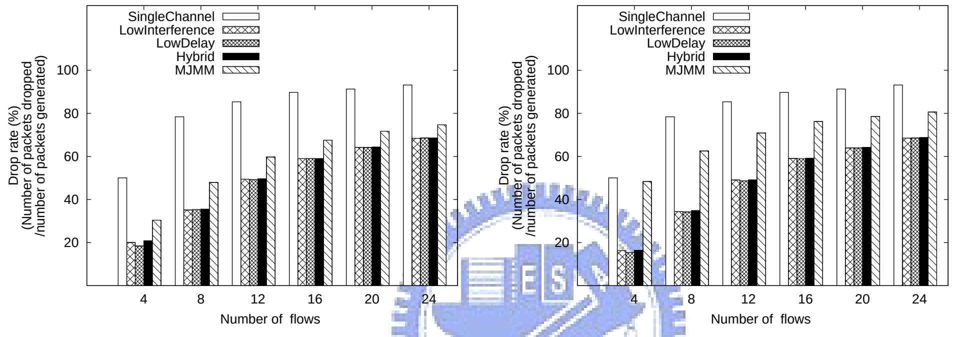
## 4.2 Simulation Results

We first consider a $5 \times 5$ grid WMN where the gateway is located at the center of the network and each mesh point can communicate with up to 4 neighbors. Fig. 4.2 shows the simulation results. We can see that the single channel scheme has the worst performance because it does not benefit from multiple available channels. Our protocol outperforms the MJMM no matter which channel assignment strategy is used. The major reason is that our protocol reduces the channel switch overhead. We allow that several packets can be transmitted in a slot. Thus, when packets arrives at mesh points in $V$, mesh points in $V$ can forward these packets immediately in the same slot. Thus, mesh points do not need to store many packets in their buffers. This can relieve the buffer overflow problem. This can be verified by Fig. 4.3. We can see that our protocol has the best performance in terms of the packet drop rate. (The packet drop rate is defined as the ratio of the number of packets dropped to the number of packet generated.) We can further observe that if there exists one quality of service (QoS) demand of 50% packet drop rate, our protocol can support



(a)Ripple Scheduling (b)Dynamic Scheduling

Figure 4.2: Comparison of throughput performance under a 5 × 5 grid network.



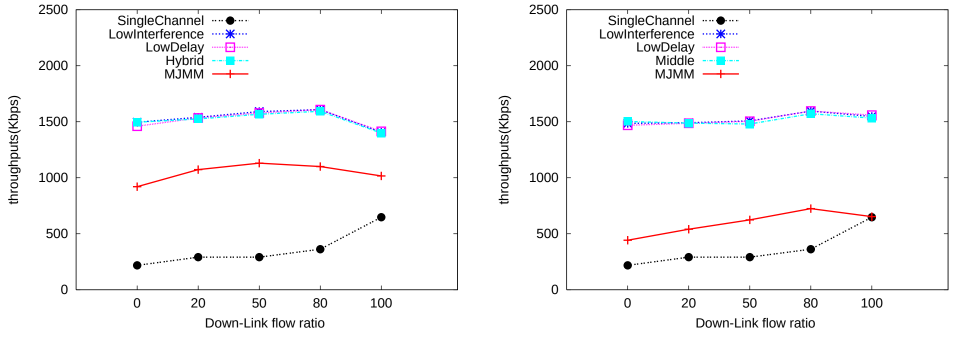
(a)Ripple Scheduling (b)Dynamic Scheduling

Figure 4.3: Comparison of packet drop rates under a 5 × 5 grid network.

up to 12 traffic flows transmission, MJMM can support about 8 traffic flows and single channel can only support 4 traffic flows. Finally, we can see that the channel assignment strategies do not impact on performance deeply. In addition, we can observe that the performances of the ripple scheduling scheme and the dynamic scheduling scheme are similar. The reason is that although the dynamic scheme benefits from its flexibility, nodes may suffer more severe collision in the dynamic scheme.

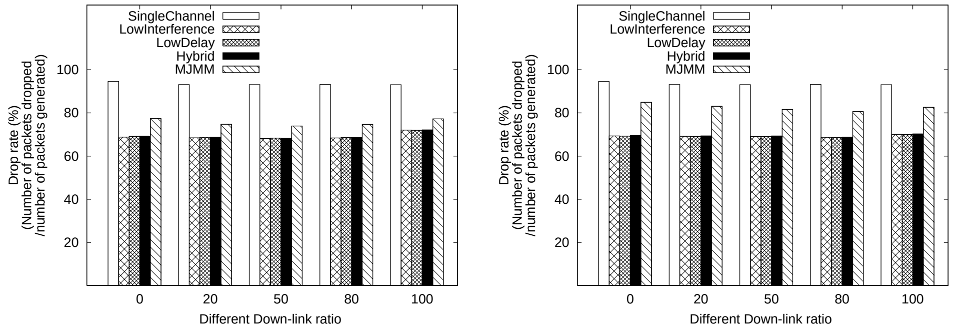
The flow characteristic in WMNs may be different in various applications and therefore we conduct an experiment in which the ratios of down-link flows to up-link flows are varied. Fig. 4.4 and Fig. 4.5 show the results. We can see that our protocol has the best performance in all scenarios.

Next, we further investigate the performance of our protocol under random deployment. Three scenarios are conducted. In the 10-node scenario, 10 mesh points are deployed randomly with uniform distribution and the transmission range of mesh points is 45 units. In the 20-node scenario, 20 mesh points are deployed randomly



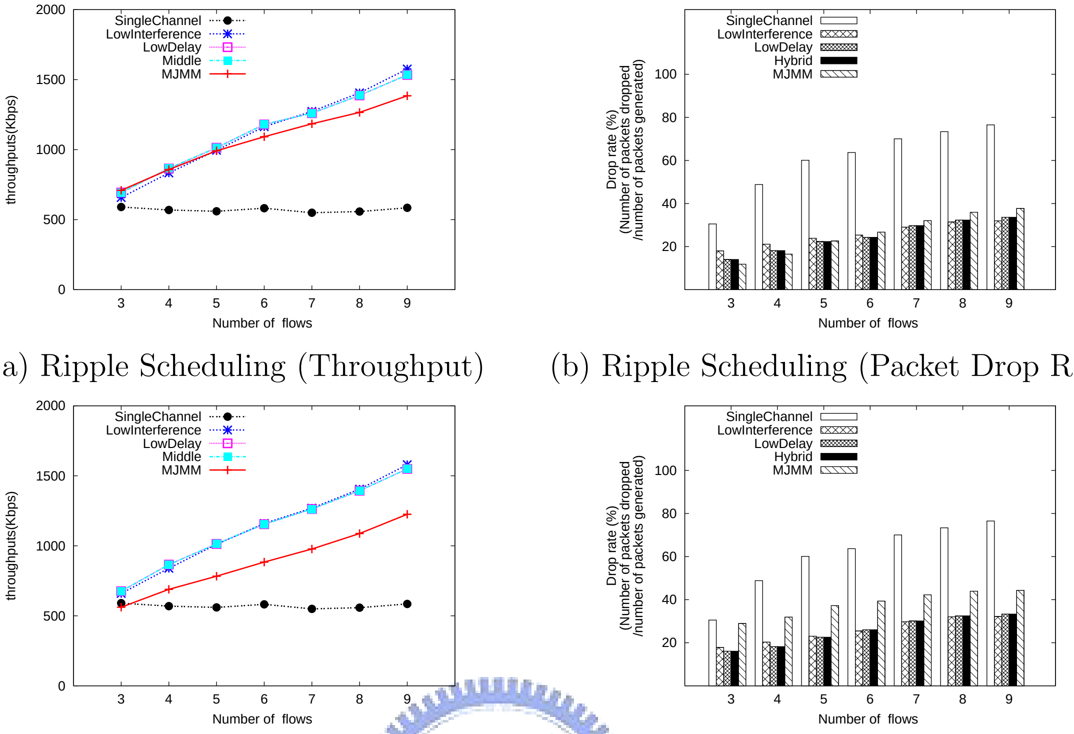
(a)Ripple Scheduling (b)Dynamic Scheduling

Figure 4.4: Comparison of throughput performance under a 5 × 5 grid network.



(a)Ripple Scheduling (b)Dynamic Scheduling

Figure 4.5: Comparison of packet drop rates under a 5 × 5 grid network.

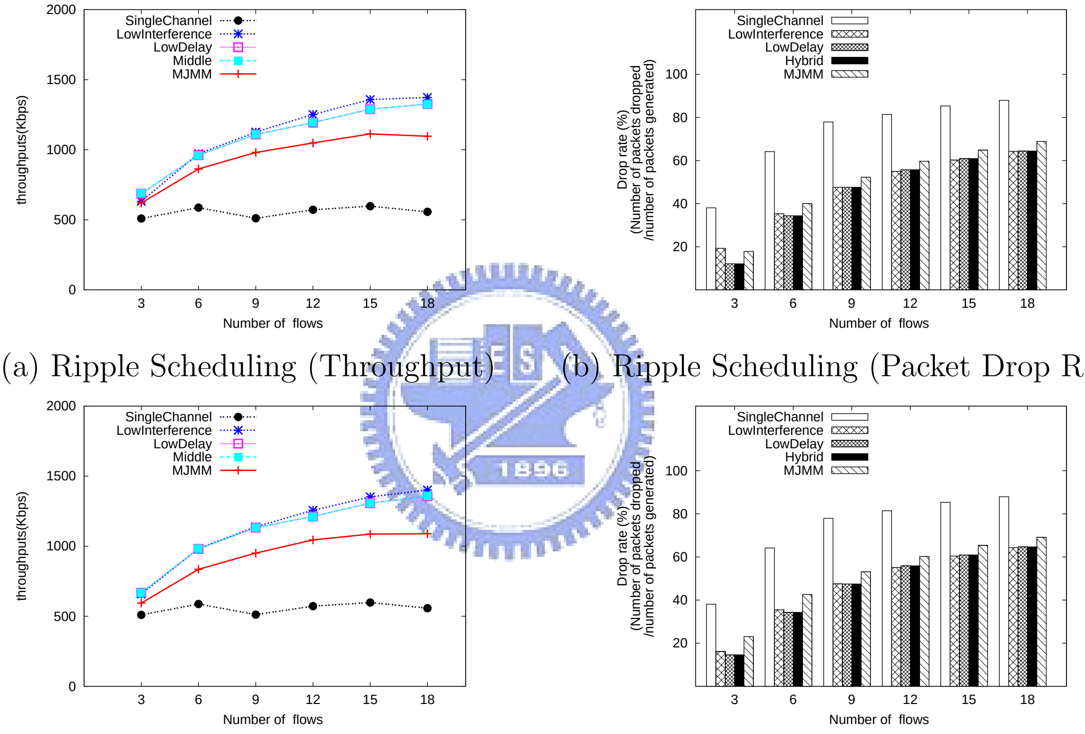


(a) Ripple Scheduling (Throughput) (b) Ripple Scheduling (Packet Drop Rate)

(c) Dynamic Scheduling (Throughput) (d) Dynamic Scheduling (Packet Drop Rate)

Figure 4.6: Simulation results of the 10-node scenario.

with uniform distribution and the transmission range of mesh points is 40 units. In the 30-node scenario, 30 mesh points are deployed randomly with uniform distribution and the transmission range of mesh points is 35 units. The results are shown in Fig. 4.6, Fig. 4.7, and Fig. 4.8. Our protocol is still the best one because our protocol can benefit from low channel switch overhead.

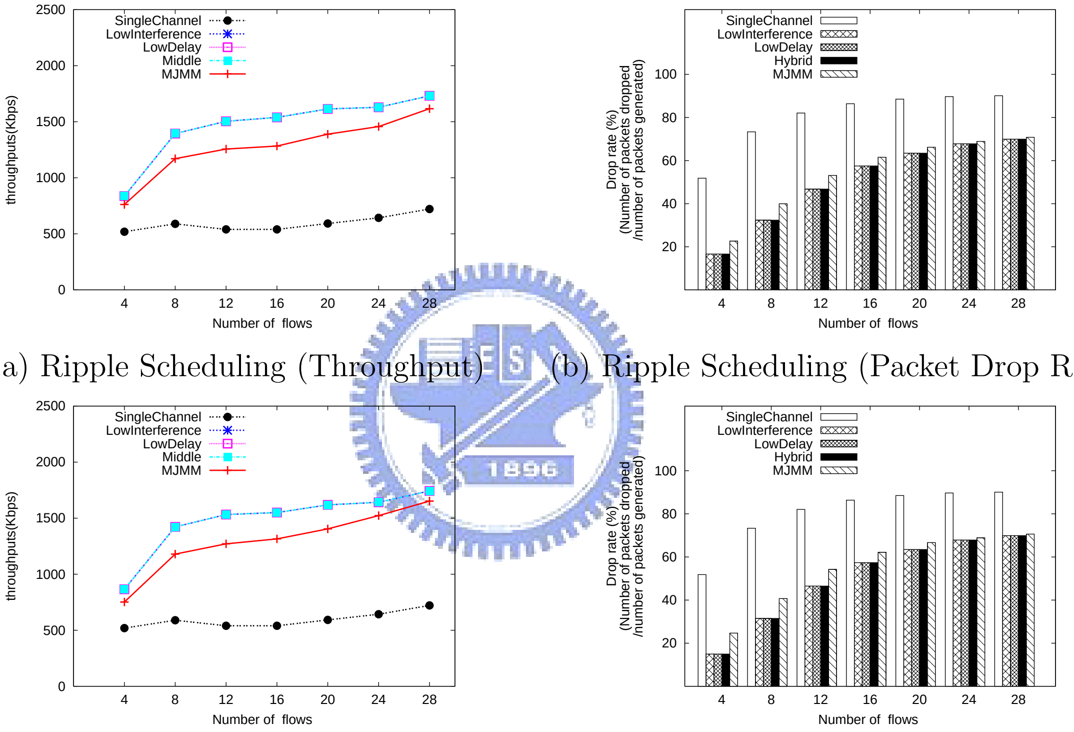


(a) Ripple Scheduling (Throughput) (b) Ripple Scheduling (Packet Drop Rate)

(c) Dynamic Scheduling (Throughput) (d) Dynamic Scheduling (Packet Drop Rate)

Figure 4.7: Simulation results of the 20-node scenario.

1896



(a) Ripple Scheduling (Throughput)
(b) Ripple Scheduling (Packet Drop Rate)

(c) Dynamic Scheduling (Throughput)
(d) Dynamic Scheduling (Packet Drop Rate)

Figure 4.8: Simulation results of the 30-node scenario.

# Chapter 5

# Implementation

We have implemented our channel management protocol on the Realtek RTL8186 [1] platform. The RTL8186 is a chipset solution for IEEE 802.11s draft 1.0 (Note that IEEE 802.11 Working Group has set up the Mesh Network Task Grop(TGs) to define the standard of 802.11-based wireless mesh network). The 802.11s-supporting module is developed by Realtek-NCTU Joint Research Center[3] and this mesh network product will be commercialized in the future. We directly implemented our channel management protocol as a extension of RTL8186 wireless driver. Our protocol has software timer, channel packet queue and management frames logic and all mesh points support all of these functions. In the rest of this Chapter we describe the details of our implementation.

- ***Software Timer***:With the design of frame structure mentioned in Chap. 3.1, each mesh point has to know whether it is at the broadcast slot or not and the mesh points in set $V'$ have to know the exact slot among $k-1$ general slots (from slot 2 to slot k in one frame) for their channel switching schedule. We register a software timer in the RTL8186 operating system for the above requirements. The register operations are completed at the initial steps of RTL8186 and these steps are shown at Fig. 5.3. Based on the fact that the measured channel switching delay is about 8$\sim$12ms, to register a 400ms software timer for one slot becomes reasonable because the channel switching delay may affect the network throughput seriously if the slot interval is too short.

- ***Channel Packet Queue***:In single-radio and multi-channel environment, each mesh point in set $V'$ executes one channel switching operation for every 400ms. Among this 400ms slot interval, the relayed packets in these set

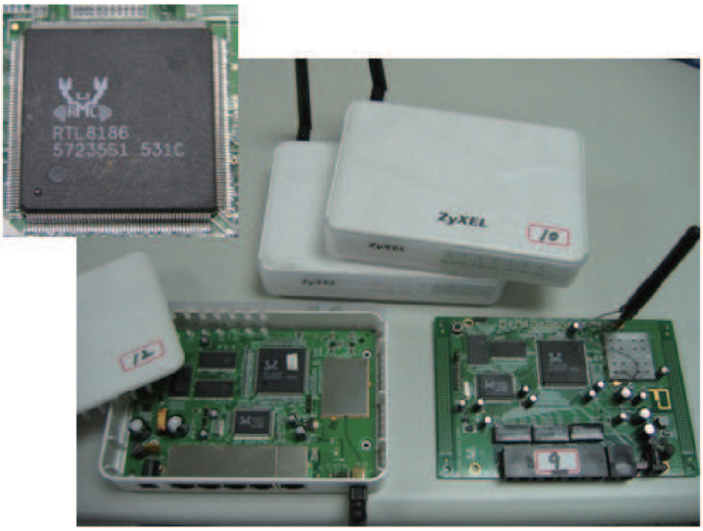


Figure 5.1: Realtek RTL8186 platform.

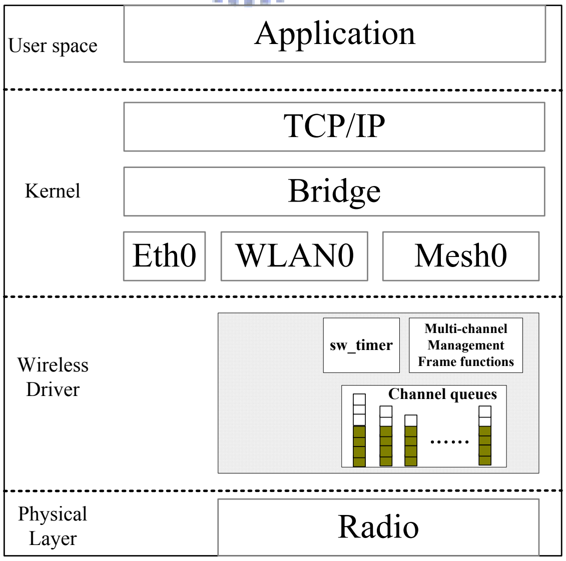
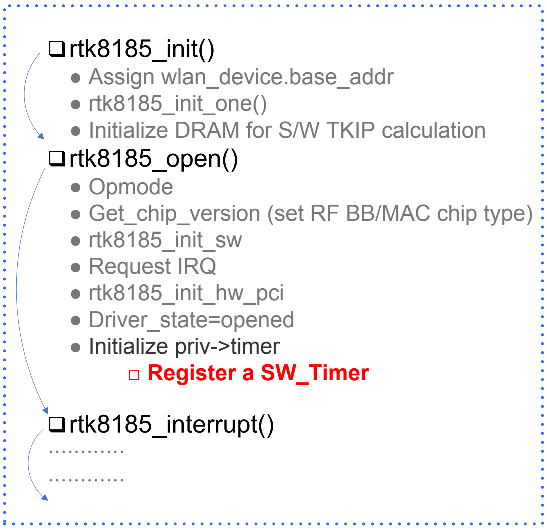


Figure 5.2: System architecture.



❑rtk8185_init()

- Assign wlan_device.base_addr
- rtk8185_init_one()
- Initialize DRAM for S/W TKIP calculation

❑rtk8185_open()

- Opmode
- Get_chip_version (set RF BB/MAC chip type)
- rtk8185_init_sw
- Request IRQ
- rtk8185_init_hw_pci
- Driver_state=opened
- Initialize priv->timer
  - □ **Register a SW_Timer**

❑rtk8185_interrupt()

............

............

Figure 5.3: Initial steps of RTL8186.

$V'$ mesh points are buffered at their channel packet queues if the receivers of these packets are at different channels. Similarly, the mesh points in set $V$ also buffer the packets destinated to one mesh point in set $V'$ that is at different channel. In short, packets can be delivered smoothly only when the transmitter and receiver are at the same channel.

- ***Management Frames***:We have designed and used some management frames to meet many requirements (such as one mesh point wants to get its 2-hop neighbors' permission) in our channel management protocol. Functions for processing these management frames are also directly implemented in RTL8186 wireless driver. To follow the frame design mentioned in 802.11s draft 1.0, we use the Category and Action fields to help wireless driver to identify our channel management frame. The Category field is set to 5 (representing mesh management) and the Action field is set to 255 (representing vendor specific mesh management). Beside the above two fields, we add a new ID field to recognize our CHL_GRANT, CLH_REQ and broadcast frame (for switching schedule). The frame format of these three management frame is shown at Table 5.1.

- ***Performance Test***: With above implementation, we simply evaluate the performance of our channel management protocol in a chain topology. The environment of our test is shown at Fig. 5.4(a). Four RTL8186 boards (A, B, C,

Table 5.1: Management frame formats used in our channel management protocol.

| | Category | Action | ID |
|---|---|---|---|
| *CHL_GRANT* | 5 | 255 | 1 |
| *CHL_REQ* | 5 | 255 | 2 |
| *Broadcast Switching Schedule* | 5 | 255 | 3 |

D) form a chain mesh network and board A is the mesh gateway. When finishing our channel management protocol, board A, B and D own a fixed channel since they are belong to set $V$ and board C arranges its channel switching schedule between channel 1 and 11. We attach each RTL8186 board with one notebook (through wired network) and these 4 notebook (NB1∼NB4) play the roles of traffic generators or destinations. More specifically, NB2, NB3 and NB4 generate data traffic with constant bit rate and these packets are all destinated to NB1. Fig. 5.4(b) presents this network throughput experiment. In this experiment, the used traffic type is UDP (User Datagram Protocol), the experiment time is set to 300 seconds and the link capacity is 2 Mbps. The network throughput evaluation result is shown at Fig. 5.5 and this experiment proves that our protocol outperforms 1.5 times than single channel mesh network when the gateway only equips one radio in both of the schemes.

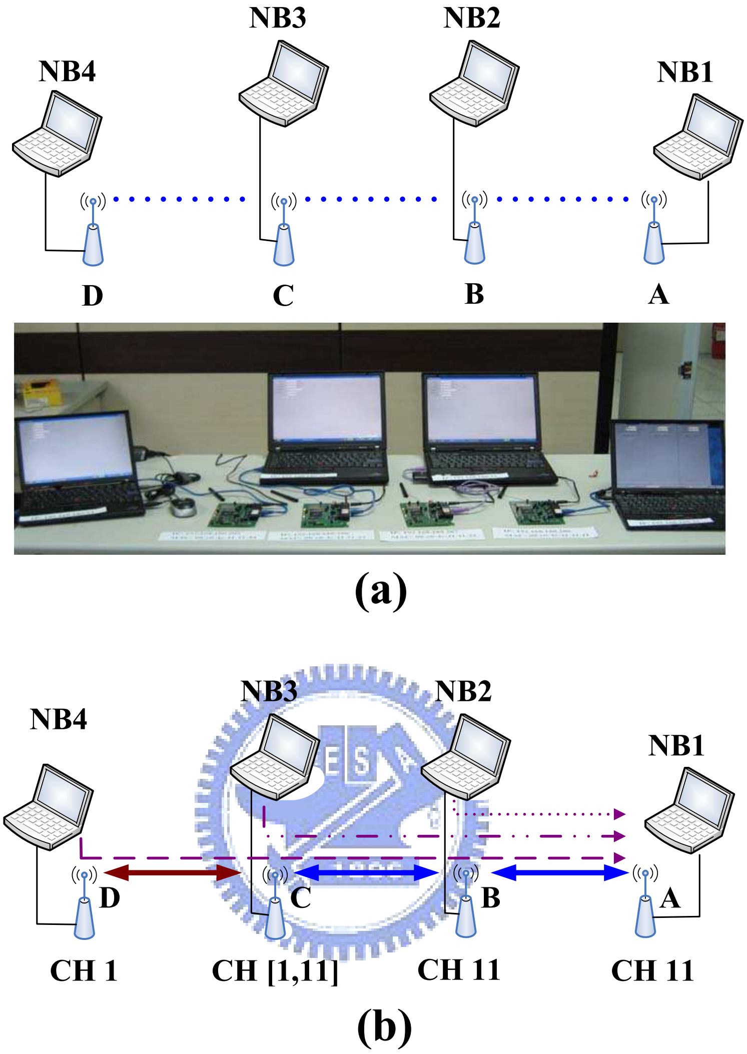


Figure 5.4: Performance test in a realistic 802.11s mesh network.

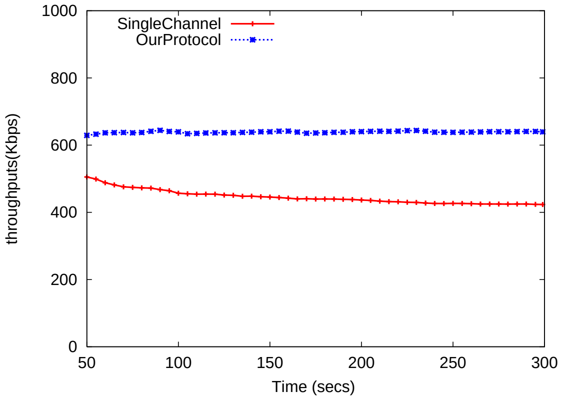


Figure 5.5: Network throughput in our experiment.

# Chapter 6

# Conclusions

This paper is a research combined with theory and implementation. We propose a new channel management protocol to mitigate channel switching overhead and increase network throughput. Our protocol is a distributed and hierarchical approach to deploy a multi-channel environment in tree-based wireless mesh networks. Under simulation, it proves that the network throughput in our protocol can outperform 1∼2 times than the modified JMM and 2∼4 times than single channel schemes. Beside verification by simulation, we also implement our protocol in IEEE 802.11s wireless mesh networks with real world platform (Realtek RTL8186). We conduct an experiment to evaluate the performance of our protocol and this experiment also proves that our protocol can be realized and perform well.

# Bibliography

[1] Realtek Semiconductor Corp., http://www.realtek.com.tw/.

[2] Cisco Mesh networks. http://www.region43.org/docs/800mhz/Cisco200602.pdf.

[3] Realtek-NCTU Joint Research Center. http://realtek.cs.nctu.edu.tw/.

[4] Tropos Mesh networks. http://www.connect802.com/tropos.htm.

[5] A. Adya, P. Bahl, J. Padhye, A. Wolman, and L. Zhou. A Multi-Radio Unification Protocol for IEEE 802.11 Wireless Networks. In *International Conference on Broadband Networks (BROADNETS)*, pages 344–354, October 2004.

[6] I. F. Akyildiz, X. Wang, and W. Wang. Wireless mesh networks: a survey. *Computer Networks*, 47(4):445–487, March 2005.

[7] P. Bahl, R. Chandra, and J. Dunagan. SSCH: Slotted Seeded Channel Hopping for Capacity Improvement in IEEE 802.11 Ad-hoc Wireless Networks. In *Proceedings of the ACM International Conference on Mobile Computing and Networking(MobiCom)*, September 2004.

[8] J. Chen and Y.-D. Chen. AMNP: Ad Hoc Multichannel Negotiation Protocol for Multihop Mobile Wireless Networks. In *Proceedings of the IEEE Interneational Conference on Communications (ICC)*, June 2004.

[9] R.-G. Cheng, C.-Y. Wang, and L.-H. Liao. Ripple: A distributed medium access protocol for multi-hop wireless mesh networks. In *IEEE VTC2006-Spring*, May 2006.

[10] R. Draves, J. Padhye, and B. Zill. Routing in multi-radio, multi-hop wireless mesh networks. In *Proceedings of the ACM International Conference on Mobile Computing and Networking(MobiCom)*, September 2004.

[11] M. Gerla, R. Bagrodia, L. Zhang, K. Tang, and L. Wang. TCP over Wireless Multihop Protocols: Simulation and Experiments. In *Proceedings of the IEEE International Conference on Communications (ICC)*, June 1999.

[12] M. X. Gong and S. F. Midkiff. Distributed Channel Assignment Protocols: A Cross-Layer Approach. In *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC)*, March 2005.

[13] IEEE. Draft amendment: ESS mesh networking. *IEEE P802.11s/D1.07*, September 2007.

[14] IEEE 802.11 Working Group. Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications, 1997.

[15] IEEE 802.11a Working Group. Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications - Amendment 1: High-speed Physical Layer in the 5 GHz band, 1999.

[16] D. B. J. Hauser and S. Conner. Draft PAR for IEEE 802.11 ESS Mesh (R3). *Technical Report IEEE 802.11-03/759r22, IEEE*, Novermber 2003.

[17] N. Jain, S. Das., and A. Nasipuri. A Multichannel CSMA MAC Protocol with Receiver-Based Channel Selection for Multihop Wireless Networks. In *IEEE International Conference on Computer Communications and Networks (IC3N)*, October 2001.

[18] P. Kyasanur and N. H. Vaidya. Routing and interface assignment in multi-channel multi-interface wireless networks. In *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC)*, March 2005.

[19] S.-C. Lo and C.-W. Tseng. A Novel Multi-Channel MAC Protocol for Wireless Ad Hoc Networks. In *Vehicular Technology Conference, 2007. VTC2007-Spring.*, pages 46–50, April 2007.

[20] A. Nasipuri and S. R. Das. Multichannel CSMA with Signal Power-Based Channel Selection for Multihop Wireless Networks. In *IEEE Vehicular Technology Conference (VTC)*, September 2000.

[21] A. Nasipuri, J. Zhuang, and S. Das. A Multichannel CSMA MAC Protocol for Multihop Wireless Networks. In *Proc. of IEEE Wireless Communications and Networking Conference (WCNC)*, September 1999.

[22] J. S. Pathmasuntharam, A. Das, and A. K. Gupta. Primary channel assignment based MAC (PCAM) - a multi-channel MAC protocol for multi-hop wireless networks. In *Wireless Communications and Networking Conference (WCNC)*, March 2004.

[23] A. Raniwala and T. cker Chiueh. Architecture and Algorithms for an IEEE 802.11-Based Multi-Channel Wireless Mesh Network. In *Conference on Computer Communications (INFOCOM)*, March 2005.

[24] J. So and N. H. Vaidya. Multi-channel MAC for Ad Hoc Networks: Handling Multi-channel Hidden Terminals Using a Single Transceiver. In *Proceedings of the ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, pages 222–233, May 2004.

[25] W.-H. Tam and Y.-C. Tseng. Joint Multi-Channel Link Layer and Multi-Path Routing Design for Wireless Mesh Networks. In *Conference on Computer Communications (INFOCOM)*, May 2007.

[26] A. Tzamaloukas and J. Garcia-Luna-Aceves. A Receiver-Initiated Collision-Avoidance Protocol for Multi-Channel Networks. In *Proceedings of IEEE INFOCOM*, 2001.

[27] R. Vedantham, S. Kakumanu, S. Lakshmanan, and R. Sivakumar. Component Based Channel Assignment in Single Radio, Multi-channel Ad Hoc Networks. In *ACM International Conference on Mobile Computing and Networking (MOBICOM)*, pages 378–389, May 2006.

[28] S.-L. Wu, C.-Y. Lin, Y.-C. Tseng, and J.-P. Sheu. A New Multi-Channel MAC Protocol with On-Demand Channel Assignment for Multi-Hop Mobile Ad Hoc Networks. In *International Symposium on Parallel Architectures, Algorithms, and Networks (I-SPAN)*, pages 232–237, December 2000.

[29] Q. Xue and A. Ganz. Temporal Topologies in Multi-channel Multihop Wireless Access Networks. In *International Conference on Broadband Networks (BROADNETS)*, October 2005.

[30] B. Zhang, M. N. Shirazi, and K. Hattori. A Tree-Based Channel Assignment and Sibling-Aware Routing Protocol for Multi-Channel Wireless Mesh Networks. In *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Science*, July 2007.