

# 國立交通大學

## 資訊科學與工程研究所

### 碩 士 論 文

在 NCTUns 網路模擬器上支援合作式網路  
之機制並測量其效能



Supporting and Evaluating the Performances of Cooperative  
Networking Schemes on the NCTUns Network Simulator

研 究 生：辜博熙

指 導 教 授：王協源 教授

中 華 民 國 一 百 年 九 月

在 NCTUns 網路模擬器上支援合作式網路之機制並測量其效能

Supporting and Evaluating the Performances of Cooperative  
Networking Schemes on the NCTUns Network Simulator

研 究 生：辜博熙

Student：Po-Hsi Koo

指導教授：王協源

Advisor：Shie-Yuan Wang

國立交通大學  
資訊科學與工程研究所  
碩士論文



Submitted to Institute of Computer Science and Engineering  
College of Computer Science  
National Chiao Tung University

in partial Fulfillment of the Requirements  
for the Degree of  
Master  
in

Computer Science

September 2011

Hsinchu, Taiwan, Republic of China

中華民國一〇一年九月

在 NCTUns 網路模擬器上支援合作式網路之機制並測量其效能

## Supporting and Evaluating the Performances of Cooperative Networking Schemes on the NCTUns Network Simulator

研究生：辜博熙

指導教授：王協源

國立交通大學

資訊科學與工程研究所

### 摘要



合作式網路 (cooperative networking / communication) 機制主要應用於無線網路 (例如 IEEE 802.11a/b/g 網路)。因為無線訊號傳輸時，傳輸有效範圍內的網路設備均可聽見這些傳輸的訊息，因此可以利用這項特性，以合作的方式幫助訊號較弱或距離較遠的網路設備傳輸其資訊。有些合作式網路的設計甚至會利用 IEEE 802.11 網路支援多種傳輸速率的特性，主動尋找鄰近的網路設備，要求幫助傳輸資訊，以達到更快的傳輸速度。本篇論文比較兩種合作式網路機制的設計方式及他們的效能，研究方式為網路模擬，並使用 NCTUns 網路模擬器比較這兩個機制的效能及帶來的優缺點。

關鍵字：IEEE 802.11a, 合作式網路, 媒介存取控制 (medium access control), 協定效能分析, 網路模擬, NCTUns

# **Supporting and Evaluating the Performances of Cooperative Networking Schemes on the NCTUns Network Simulator**

Student: Po-Hsi Koo

Advisor: Shie-Yuan Wang

Institute of Computer Science and Engineering

National Chiao Tung University

## **ABSTRACT**

Exploiting the broadcast nature of wireless networks, the cooperative networking / communication scheme becomes a new paradigm in wireless communication. Some of these designs exploit multi data rate feature in 802.11a/b/g standard to achieve better performance. In this thesis, we compare two cooperative networking designs and their performances. One of them exploits multi data rate feature, while the other does not. We concentrate on simulation research method and use NCTUns network simulator to evaluate the performances of these two designs. Advantages and disadvantages of these designs are also discussed.

Keywords: IEEE 802.11a, cooperative networking, medium access control, protocol analysis, network simulation, NCTUns

# Acknowledgement

在交大的這七年，我所得到的東西很多。如今我即將畢業，我想感謝在這七年之中和我一起度過的人們。

感謝我的老師王協源教授，在老師的指導下，我學到了研究的基礎以及面對人生的挑戰時應有的態度。謝謝老師在很多方面給我的鼓勵。

感謝口試委員林華君教授，吳曉光教授，以及黃仁竑教授，你們的建議讓我知道我的不足之處並能使我的論文內容更加完善。

感謝我的家人，謝謝你們在過去這一年，還能持續支持我的生活與學業，這一年對我十分重要，未來，我也會繼續努力。

謝謝周智良學長，林志哲學長，昱銘，士緯，和照展。我們在實驗室共同努力過，並分享彼此的喜怒哀樂，希望你們未來在工作上一帆風順。

謝謝育緯和楷文，我從你們身上學到了我所缺少的事物。

最後感謝我所認識的實驗室，社團與大學的學弟妹，以及為數眾多的文學家，藝術家，作曲家，導演，小說家，漫畫家，在我失意沮喪之時，你們都曾無形地鼓勵了我。希望你們都能找到人生中不朽的事物。

Carpe diem. Seize the day. Make your lives extraordinary.

# Contents

摘要 .....	i
ABSTRACT .....	ii
Acknowledgement.....	iii
Contents.....	iv
List of Figures .....	v
List of Tables .....	vii
Chapter 1 Introduction .....	1
Chapter 2 Related Work .....	4
Chapter 3 Research Method .....	7
3.1 NCTUns network simulator .....	7
3.2 Design of CoopMAC and CMAC .....	9
3.2.1 CoopMAC .....	9
3.2.2 CMAC .....	15
3.3 Implementation of Cooperative Networking Schemes on NCTUns Network Simulator .....	17
3.3.1 Implementation of CoopMAC on NCTUns .....	20
3.3.2 Implementation of CMAC on NCTUns.....	24
Chapter 4 Simulation Results and Performance Evaluation.....	26
4.1 Simulation Results of CoopMAC Scheme.....	26
4.2 Simulation Results of CMAC Scheme.....	33
4.3 Comparison between These Schemes .....	38
Chapter 5 Discussion .....	45
Chapter 6 Future Work.....	46
Chapter 7 Conclusion.....	48
Bibliography.....	49

# List of Figures

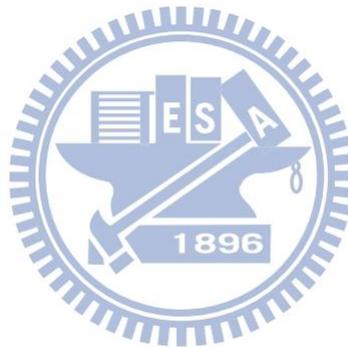
Fig. 3.1 Architecture of NCTUns .....	8
Fig. 3.2 Format of CoopTable .....	9
Fig. 3.3 Frame formats in CoopMAC .....	10
Fig. 3.4 Transmission Flow Charts of CoopMAC.....	13
Fig. 3.5 Concept of CMAC protocol .....	16
Fig. 3.6 Default protocol stack of 802.11a stations .....	18
Fig. 3.7 Structure of RateTable.....	21
Fig. 3.8 Modification to NCTUns modules (CoopMAC) .....	23
Fig. 3.9 Modification to NCTUns modules (CMAC) .....	25
Fig. 4.1 CoopMAC simulation scenario.....	26
Fig. 4.2 CoopMAC simulation scenario.....	29
Fig. 4.3 Performance of CoopMAC – total throughput (UDP).....	29
Fig. 4.4 Performance of CoopMAC – packet delay (UDP) .....	30
Fig. 4.5 Performance of CoopMAC – percentage of packet collisions (UDP).....	30
Fig. 4.6 Performance of CoopMAC – total throughput (TCP).....	31
Fig. 4.7 Performance of original 802.11a MAC – total throughput .....	32
Fig. 4.8 Performance of CoopMAC – total throughput.....	32
Fig. 4.9 Performance of CMAC – total throughput (UDP).....	33
Fig. 4.10 Performance of CMAC – packet delay (UDP).....	34
Fig. 4.11 Performance of CMAC – percentage of packet collisions (UDP) .....	34
Fig. 4.12 CMAC simulation scenario 1 .....	35
Fig. 4.13 Performance of CMAC – scenario 1 .....	36
Fig. 4.14 CMAC simulation scenario 2.....	37

Fig. 4.15 Performance of CMAC – scenario 2 .....	38
Fig. 4.16 Comparison – total throughput.....	39
Fig. 4.17 Comparison – packet delay .....	40
Fig. 4.18 Comparison – percentage of packet collisions.....	40
Fig. 4.19 Performance of CoopMAC – scenario 1 .....	41
Fig. 4.20 Performance of CMAC – scenario 1 .....	42
Fig. 4.21 Comparison - scenario 1.....	42
Fig. 4.22 Performance of CoopMAC – scenario 2.....	43
Fig. 4.23 Performance of CMAC – scenario 2.....	44
Fig. 4.24 Comparison – scenario 2.....	44



# List of Tables

Table 3.1 Appropriate data rate under different signal strengths.....	21
Table 3.2 Definition of duration for different types of packet in CoopMAC.....	22
Table 3.3 Timeout values in CoopMAC.....	22
Table 4.1 Performance of CoopMAC.....	27



# Chapter 1

## Introduction

The concept of cooperative communication/networking is discussed in [1]-[3]. This kind of techniques allow more than two stations to cooperate with each other in their transmission. The developed cooperative networking schemes are generally divided into three paradigms by researchers: “store and forward”, “amplify and forward”, and “coded cooperation” [1], [2].

The basic idea of the “store and forward” method is to cooperate by relaying packets. Obviously the way of choosing relays is important for this method, and the simplest structure of such method is the two-hop relay. The “amplify and forward” method lets several stations transmit together. The partner stations have the noisy version of the original packet in advance. These stations then amplify the received signal and transmit it together so that the final receiver can combine them and decode the final result. In the last “coded cooperation” method, the technology of channel coding is exploited and makes the cooperation more robust. Using channel coding, redundancy can be added into the coded signals so that even if these data are received with some bit errors, they can still be correctly decoded by the receiver.

Network cooperation can be performed at different layers of OSI model, too [4]. For example, at network layer, wireless ad hoc routing protocols such as AODV [5], DSR [6] can be seen as approaches of network cooperation. [2] These protocols are used in environment where no central access points exist and the source station may not communicate with the destination station directly.

The intermediate stations that help route the packets can be seen as partners in the view of cooperative communication.

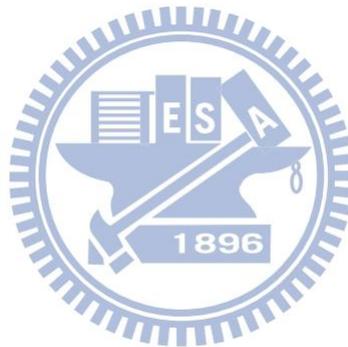
At physical layer, amplify-and-forward method is used to achieve space diversity. Other researches use channel coding and FEC schemes to enhance the performance of physical layer cooperation. MIMO (Multi-Input Multi-Output) systems is also proposed and developed. MIMO technologies are also incorporated into wireless network standard such as Wiman and upcoming 4G systems.

As for data link layer (MAC layer), IEEE 802.11 wireless networks support multi-rate feature at physical layer. For example, 802.11b standard supports four rates including the rate of 1, 2, 5.5, and 11 Mbps, and the later 802.11a standard supports eight rates including the rate of 6, 9, 12, 18, 24, 36, 48, and 54 Mbps. Some researches including [7], [8] suggest exploiting such multi-rate capability. It can be seen as a cross-layer design of MAC layer and physical layer. Information about channel condition is gathered at physical layer, and kept at MAC layer. Later the MAC layer can decide whether it should communicate with other network devices in a cooperative way depending on the gathered information.

We compare two different approaches at MAC layer cooperation, CoopMAC [8] and C-MAC [10]. CoopMAC is a protocol trying to exploit multi-rate feature as far as possible. On the other hand in the C-MAC protocol some network stations just help their neighbors fast-retransmit lost packets. Multi-rate feature is not exploited in this kind of protocols. These two compared protocol must modify the 802.11 protocol as little as possible, even not interfering with network devices without such new feature.

The remaining of the thesis is organized as follows. In chapter 2 we summarize related work about cooperative communication at MAC layer. In

chapter 3 we describe the research method we use to evaluate these different approaches. The results are presented in chapter 4. Further discussion is presented in chapter 5. In chapter 6 other future work is proposed. At last chapter 7 concludes this thesis.



# Chapter 2

## Related Work

In this chapter we summarize cooperative MAC protocols that have been proposed, later studied and validated in various works.

### rDCF [7]

It contains “service advertisement” and “triangular handshake” mechanisms to achieve cooperation. In the scheme, each node in the network periodically advertises a list, containing MAC addresses of stations that can help it transmit packets in the cooperative way. Later when a station has packets to transmit, it will search for relay nodes depending on this list.

### CoopMAC [8]

It is inspired by rDCF and tries to develop a protocol that is backward compatible with the legacy 802.11 system. To achieve this, this protocol does not have the “service advertisement” mechanism as in rDCF. Second it leverages an unused address field in MAC header to fill in information needed by helper stations.

### CD maca [9]

This work slightly modifies the 802.11 protocol to assist the ARQ protocol. When a node detects the start of retransmission and it has heard the original packet in advance, this node will join the retransmission simultaneously.

### CMAC [10]

Similar to CD maca, but in this scheme when a helper station detects the need of retransmission, it tries to retransmit faster than the original source station. If a station overhears a packet and successfully decodes it, it will store the packet in its partner queue. Thus in the future it is able to retransmit the overheard packets.

#### COPE [11]

XOR-operation network coding scheme is used in this work. Each node keeps a packet pool of overheard packets so that they may help each other in the future. In addition each station in the network broadcasts “reception reports” to tell its neighbors which packets it has stored.

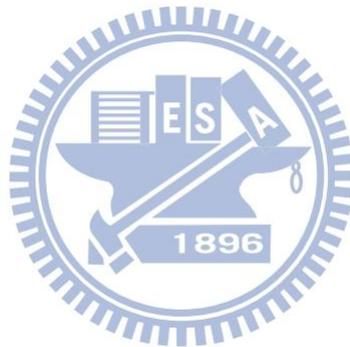
#### HARBINGER [12]

This approach is based on hybrid-ARQ scheme. In other words it uses the “coded cooperation” paradigm mentioned in chapter 1. In conventional hybrid-ARQ schemes retransmitted packets are from the original source station, but in this approach the retransmitted packets can also be supplied by the relay nodes. Thus with the time diversity effect supplied by the conventional hybrid-ARQ, the use of HARBINGER also supplies space diversity effect as well.

#### VMISO [13]

As its name implies, this work aims to create virtual multi-input single-output system. Originally MIMO (multi-input multi-output), SIMO (single-input multi-output) and MISO (multi-input single-output) systems exploit antenna arrays to enhance signal quality. However this work uses the new paradigm of virtual antenna arrays (or cooperative diversity). It combines the abilities of physical layer, MAC layer, and network layer to achieve such diversity. In VMISO appropriately encoded information is transmitted

simultaneously by multiple nodes (thus called “multi-input”) to achieve the advantage brought by antenna arrays. Technologies exploited in this work includes: Alamouti code, time-reversed space-time codes, space-time OFDM, ad-hoc routing protocols such as AODV, DSR, and some modification to 802.11 MAC protocol.



# Chapter 3

## Research Methods

In this thesis we compare cooperative MAC protocols that we are interested in through simulation methods. The simulator we use is NCTUns . We briefly introduce NCTUns network simulator in chapter 3.1. The two approaches we feel interested in will be introduced in chapter 3.2. In chapter 3.3 we describe the modification to NCTUns network simulator in order to support these cooperative MAC protocols and validate them.

### 3.1 NCTUns network simulator

The NCTUns network simulator is an integrated, open-source network simulator. With its unique kernel re-entering simulation methodology, NCTUns allows real-world application program to be run in the simulation network. Real-world network protocol (for example, IP, TCP, and UDP) can also be involved in simulations conducted. On the other hand, data link layer, physical layer, and radio link behavior is simulated by means of protocol modules in NCTUns. These protocol modules are programs that describe protocols (for example, 802.3, 802.11 MAC protocols, or ad hoc routing protocols) or physical properties (for example, the link delay, and fading effect in wireless network). When a researcher wants to know how a specific network will become if these protocols or properties is slightly modified, he can modify these protocol modules and conduct different simulations to validate his thoughts. With the

capability of NCTUns, we are able to implement these cooperative MAC protocols and compare throughputs of application programs achieved by them.

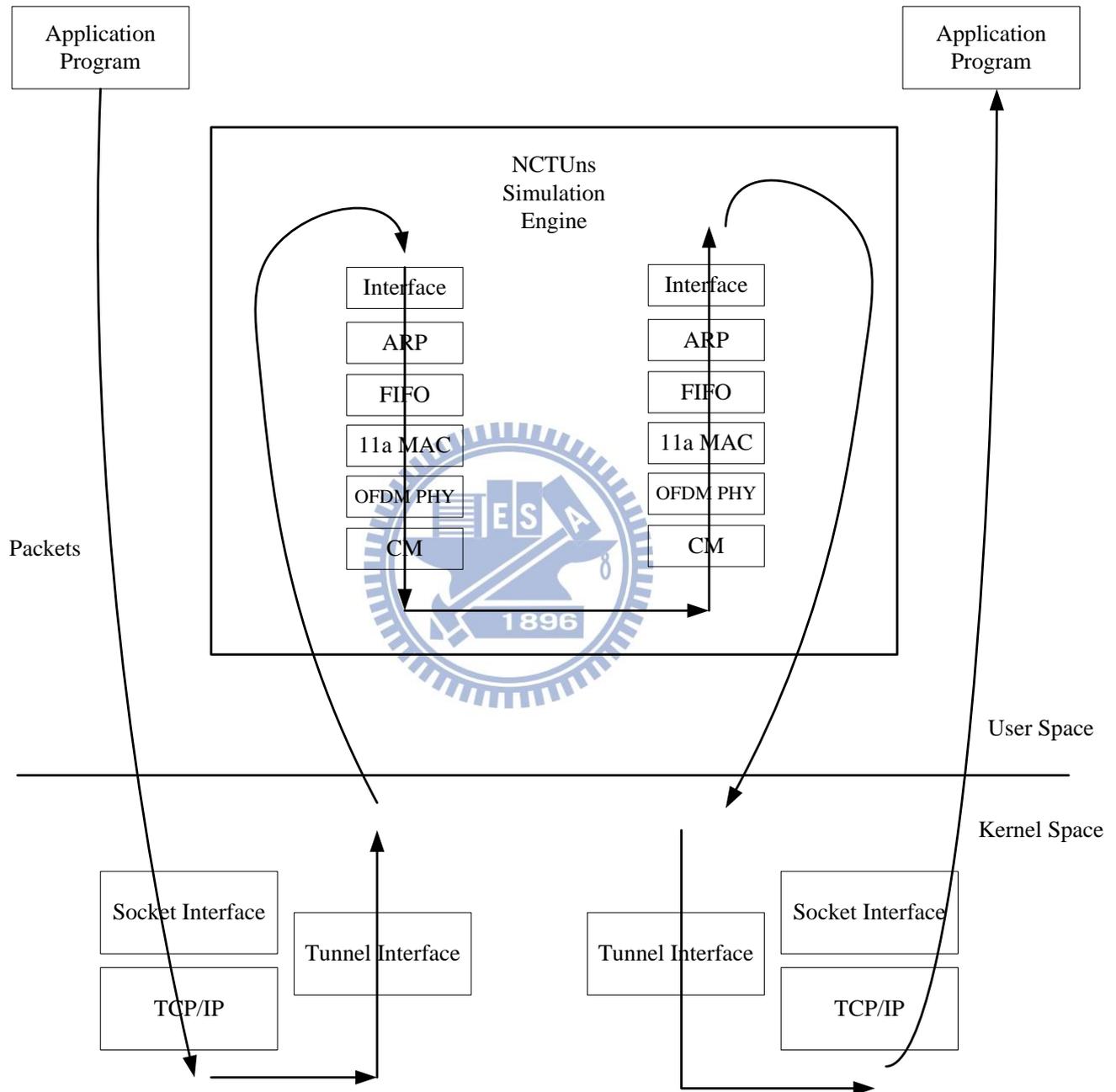


Fig. 3.1 Architecture of NCTUns

## 3.2 Design of CoopMAC and CMAC

### 3.2.1 CoopMAC

CoopMAC is proposed in [8]. The authors design a cooperative 802.11 MAC protocol that is simple and backward compatible with the legacy 802.11 system. This protocol exploits multi-rate capability of the 802.11 system to achieve higher throughput. In CoopMAC protocol every station keeps a table called CoopTable.

ID (48 bits)	Time (8 bits)	$R_{hd}$ (8 bits)	$R_{sh}$ (8 bits)	Number of failures (8 bits)
MAC address of the helper	Time when the last packet is heard from the helper	Transmission rate between the helper and the destination	Transmission rate between the source and the helper	Count of sequential transmission failures

Fig. 3.2 Format of CoopTable

The structure of CoopTable is shown in Fig. 3.2. A station memorizes stations that are helpful for itself in its CoopTable. The stations will decide that another station is helpful depending on rate conditions. The  $R_{sh}$  field in CoopTable represents the transmitting rate between the station and a potential helper. And the  $R_{hd}$  field in CoopTable represents the transmitting rate between the potential helper and another station.  $R_{hd}$  information is obtained from PLCP header in the 802.11 physical layer preamble. So in CoopMAC when a station overhears packets transmitted by other stations, it will receive this packet in spite of its destination (like in promiscuous mode). Then it will gather information from this packet. If

$$\frac{1}{R_{sh}} + \frac{1}{R_{hd}} < \frac{1}{R_{sd}},$$

which means the estimated time needed to transmitted by two-hop relay method is smaller than the time needed by direct transmission, the station will then keep this entry in its CoopTable.

When there are packets to transmit, a station first searches in its CoopTable. If there is a potential helper in CoopTable that may help it transmit packets to the destination, the station will first consider exploiting the helper. But if this does not succeed for several times (the authors suggest 3 times), the station will temporarily discard this entry in its CoopTable and try other potential helpers, or transmit directly.

Frame Control (2 octets)	Duration (2 octets)	Receiver Address (6 octets)	Transmitter Address (6 octets)	CRC (4 octets)
-----------------------------	------------------------	--------------------------------	-----------------------------------	-------------------

(a) RTS frame format

Frame Control (2 octets)	Duration (2 octets)	Receiver Address (6 octets)	Transmitter Address (6 octets)	Helper Address (6 octets)	$R_{sh}$ (1 octet)	$R_{hd}$ (1 octet)	CRC (4 octets)
-----------------------------	------------------------	--------------------------------	-----------------------------------	------------------------------	-----------------------	-----------------------	-------------------

(b) CoopRTS frame format

Frame Control (2 octets)	Duration (2 octets)	Receiver Address (6 octets)	CRC (4 octets)
-----------------------------	------------------------	--------------------------------	-------------------

(c) HTS/CTS frame format

Frame Control (2 octets)	Duration (2 octets)	Address 1 (6 octets)	Address 2 (6 octets)	Address 3 (6 octets)	Sequence Control (2 octets)	Address 4 (6 octets)	Frame Body	CRC (4 octets)
-----------------------------	------------------------	-------------------------	-------------------------	-------------------------	--------------------------------	-------------------------	------------	-------------------

(d) General 802.11 MAC header

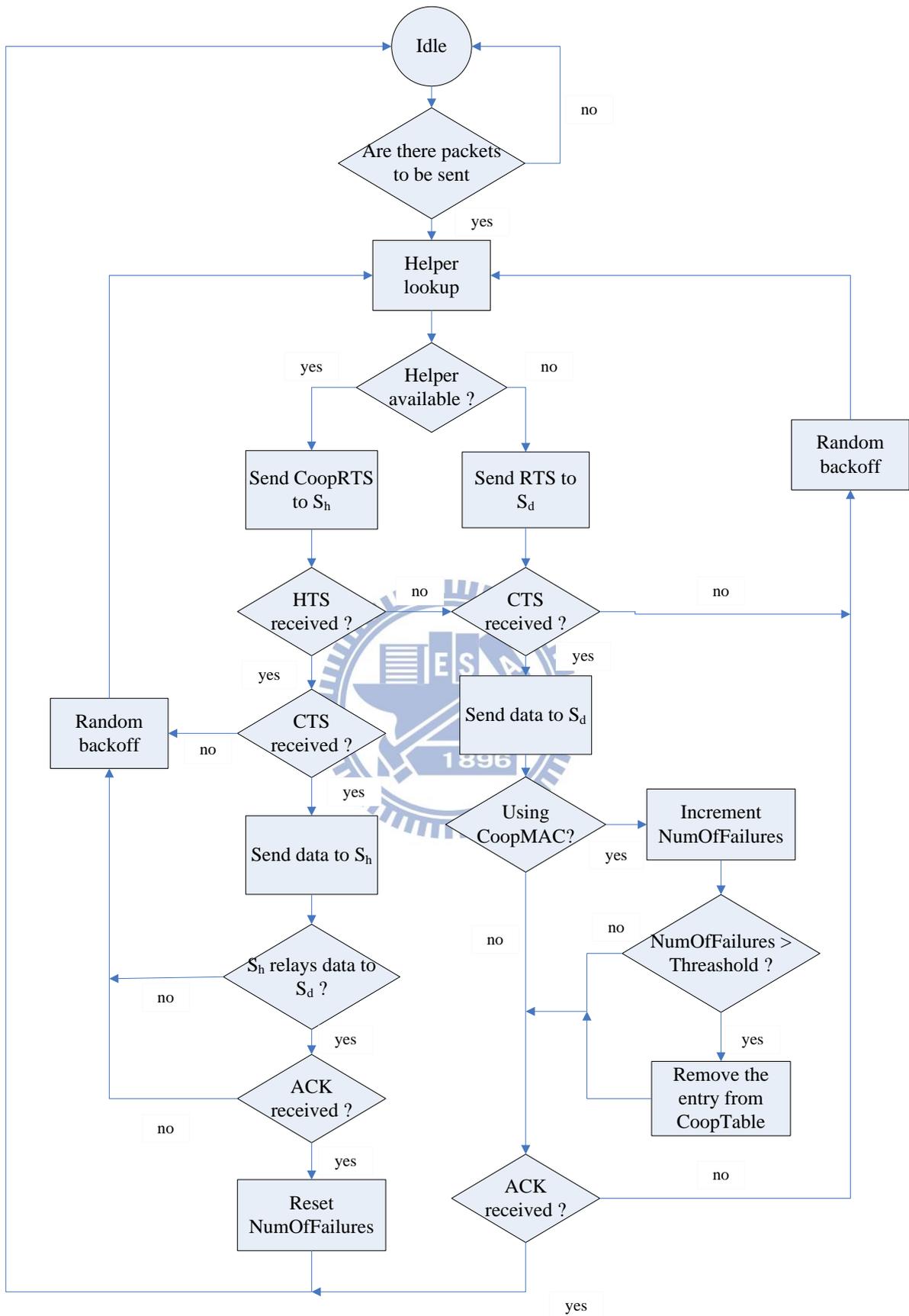
Fig. 3.3 Frame formats in CoopMAC

The frame format used in CoopMAC is shown in Fig. 3.3. CoopMAC defines new HTS packet to achieve cooperation in RTS/CTS mode. The format

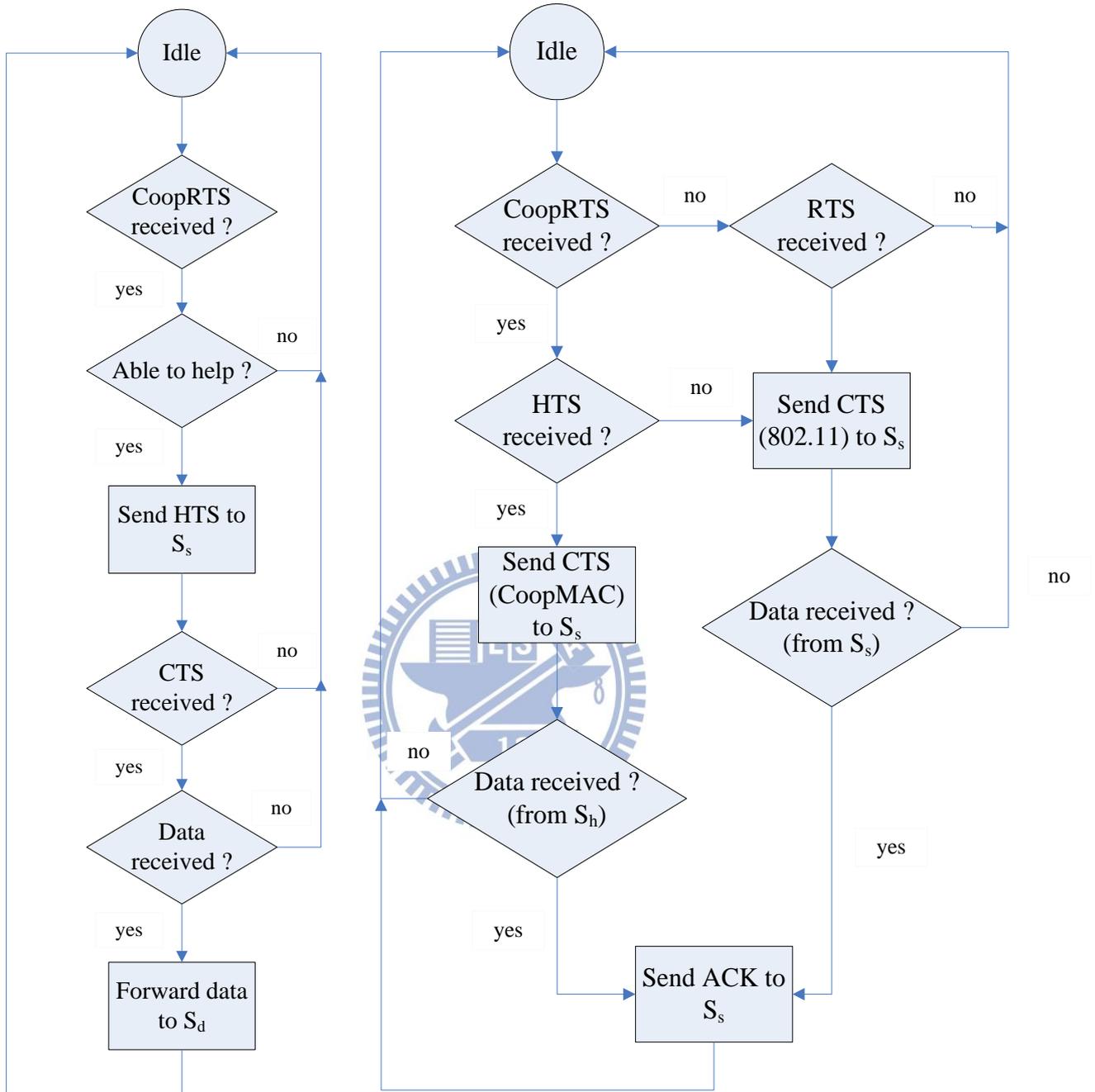
of HTS packets is the same as that of CTS packets for backward compatibility. Another new CoopRTS packet is also included in the protocol, but of different format with the original RTS packet.

On the other hand for base mode (without RTS/CTS), the reserved Address 4 field in 802.11 MAC header is exploited to distinguish between normal data frame and cooperative data frame. The Subtype field of cooperative data frames will be set to a reserved value of 1000. This enables stations to distinguish between packets destined to them and packets that will be relayed by them.





(a) Transmission Flow Chart for Source Stations



(b) Transmission Flow Chart for Helper Stations

(c) Transmission Flow Chart for Destination Stations

Fig. 3.4 Transmission Flow Charts of CoopMAC

The transmission flow charts are shown in Fig. 3.4. In the following we depict CoopMAC protocol in detail.

In the modified RTS/CTS mode, if a source station has packets to transmit, it first looks up in its CoopTable for helpers that may help it relay the packets to the destination. If such helpers exist in its CoopTable, the station sends a CoopRTS packet with the MAC address of the helper station in the Helper Address field. As shown in Fig. 3.3, CoopRTS packets contain more information than normal RTS packets. Now if a legacy 802.11 station receives the CoopRTS packet, it can still understand the Receive Address field and the Transmitter Address field. The different thing is that the packet length is longer than normal RTS packets. Nevertheless the destination station can reply with a CTS packet. When the source station receives only a CTS packet (but not a HTS packet), it assumes the destination station does not use CoopMAC and resume normal 802.11 protocol.

Now if the source station, the helper station, and the destination station all use CoopMAC, after the source station sends a CoopRTS packet, they exchange messages in a different way. When a (helper) station receives CoopRTS packets with its MAC address in the Helper Address field, it decides whether to help the source station. If the helper station decides not to give help, for example because it has its packet to transmit, the helper station just ignores this CoopRTS packet. Otherwise it replies the source station with an HTS (Helper ready To Send) packet after an SIFS time. As mentioned the HTS packets have the same format as CTS packets. After receiving the HTS packet, the destination station sends a CTS packet after another SIFS time to confirm the cooperative transmission.

After the handshaking procedure, the source station transmits data frame to the helper station after an SIFS time. And then the helper station transmits the data frame to the destination station after an SIFS time. Finally the destination

station directly sends an ACK packet to the source station, after an SIFS time.

In the above example, if the helper station decides not to help the cooperative transmission and ignores the CoopRTS packet, the destination station waits for an SIFS time and will not hear any HTS packets. In this situation the destination station send a CTS packet after another SIFS time, and the normal 802.11 protocol is conducted.

In the base mode (direct transmission without RTS/CTS), stations are not able to distinguish between normal data packets and relayed packets. Thus the procedure is slightly modified. If the source station wants to start cooperative transmission in base mode, it first transmits the data frame directly to the helper station. The Address 4 field now contains the MAC address of the destination station, and the frame Subtype value is changed to a reserved value of 1000 (normal data packets use the value of 0000). If a legacy 802.11 station receives the packet, it does not recognize this kind of packet and discards it. Otherwise if the helper station uses CoopMAC and is able to help, it moves the address of the destination station (in Address 4 field) to Address 1 field, recalculates the CRC checksum, and forwards the packet to the destination node. The ACK packet is again sent directly from the destination node to the source node.

### **3.2.2 CMAC**

CMAC is proposed in [10]. Different from CoopMAC, it does not exploit multi-rate capability to achieve higher throughput. Instead it adopts simpler approach. Consequently it modifies the 802.11 protocol little.

The basic concept of CMAC is to let helper stations overhear packets and help retransmit them if these packets are received incorrectly at the destination.

Like in other cooperative protocols, stations must overhear packets so that they can help cooperation in the future.

If a station receives a packet correctly, and this packet is not destined to itself, it stores this packet in a special queue. This queue is called partner queue in CMAC. The station then expects that an ACK packet should be transmitted in SIFS time, which is defined by the 802.11 protocol. If not, the station thinks this packet is received in error at destination. Now that the station has the correct packet, it performs a fast retransmission to help the source station.

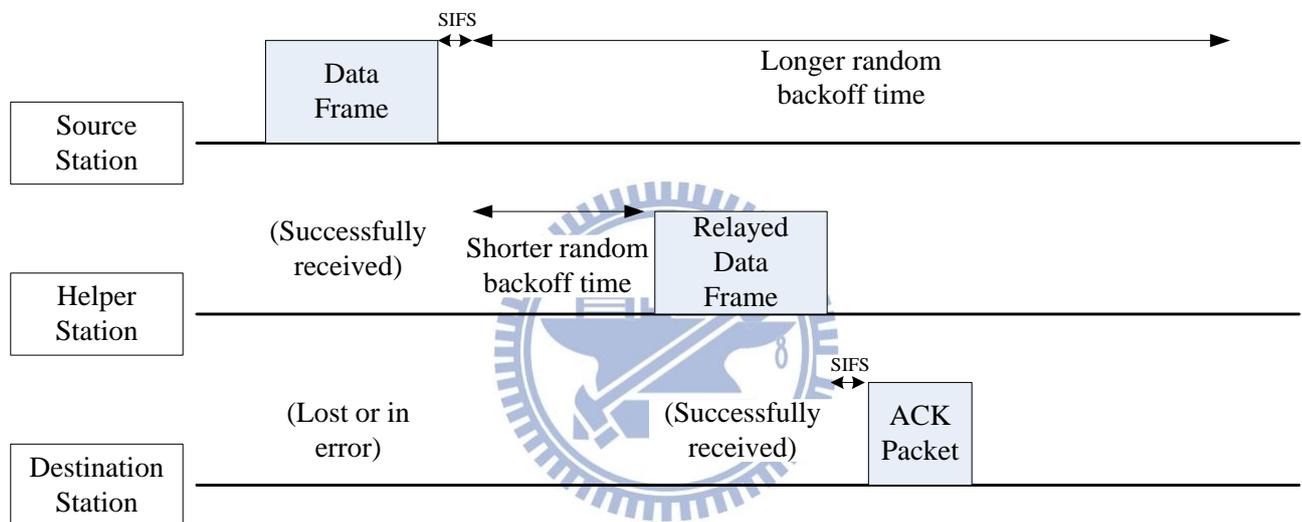


Fig. 3.5 Concept of CMAC protocol

The concept of such retransmission is shown in Fig. 3.5. Originally if a station sends a data packet, and this packet is somehow lost, collided with other stations, or received in error at the destination, the station should perform exponential random backoff. This means its backoff window size grows exponentially if such collision occurs successively. In CMAC helper stations try to fast-retransmit such correctly-received packets. Thus they always use the minimum backoff window size for such cooperative retransmission. For example the minimum backoff window size is 15 slots in 802.11a.

Once the helper station receives an ACK packet from the destination station to the source station, it deletes packets for the destination station in its partner queue. Additionally helper stations only retransmit such packets once. This aims to give other helpers the chance to help the retransmission. Moreover this helps avoid such situations that a helper is in fact far from the destination station, thus giving little help for the retransmission.

### **3.3 Implementation of Cooperative Networking Schemes on NCTUns Network Simulator**

In the thesis we use NCTUns network simulator to compare the two selected cooperative MAC protocols. We conduct simulations in which network stations use 802.11a protocol in ad hoc mode. 802.11a operates in the 5 GHz band. Its maximum data rate is 54Mbit/s. Different from 802.11b, an OFDM-base physical layer is used in 802.11a. On the other hand 802.11b uses FHSS or DSSS in physical layer and operates in the 2.4 GHz band.

In NCTUns the simulation procedure is performed by the Simulation Engine. The Simulation Engine provides a module-based platform for simulation services. In any simulated network devices, we can specify protocol modules you want to simulate, and these selected modules form the protocol stack of this network station. Take 802.11a nodes for example. The default protocol stack of 802.11a nodes in NCTUns includes Interface, ARP, FIFO, Mobile Node, 802.11a MAC, Wireless Tcpdump, 802.11a OFDM PHY, and Channel Model. Packets generated in simulation will then be processed in order by these protocol modules. The whole protocol stack and the processing order are shown in Fig. 3.6.

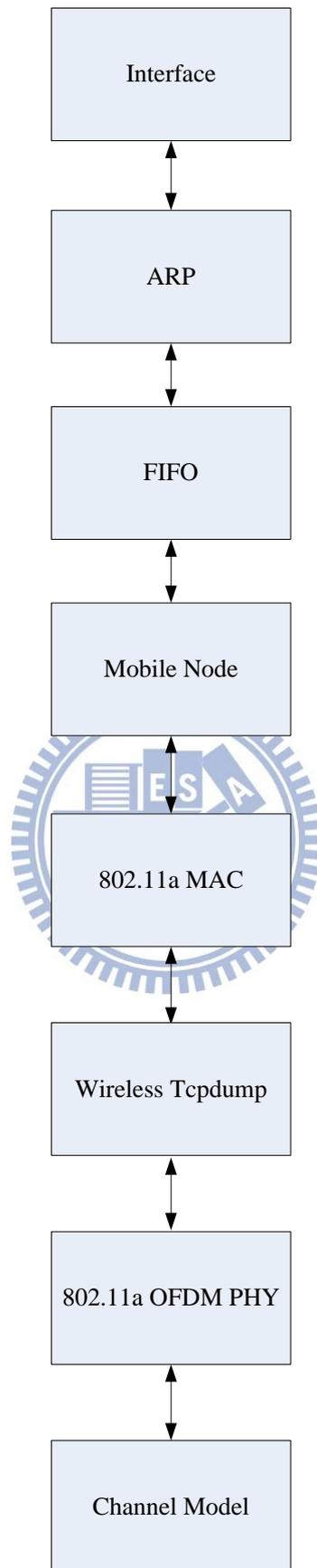


Fig. 3.6 Default protocol stack of 802.11a stations

In the following we summarize the protocols and properties that are simulated in these protocol modules.

**Interface module:**

It is the entry point of packets. When real applications generate packets, the simulation engine reads the packets from kernel tunnel interface. Then these packets are processed first by this module. Likewise after received packets are processed by other protocol modules, they finally come to Interface module, and will be written to kernel tunnel interface and received by real applications.

**ARP module:**

This simulates Address Resolution Protocol. We can specify default entries in the ARP table, or keep it clean and let the protocol update it during simulation.



**FIFO module:**

It simulates the First-In-First-Out queue. Queue length can be changed by user.

**Mobile Node module:**

It simulates management packets and beacon messages.

**802.11a MAC module:**

It simulates Media Access Control mechanisms used by 802.11a DCF (Distributed Coordination Function). These mechanisms include CSMA/CA, random backoff, ARQ, CRC checksum. The compared two cooperative networking schemes both modify the MAC protocol, so our main modification is done on this module.

### **Wireless Tcpcdump module:**

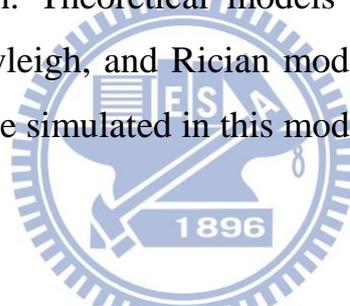
This is for users to capture packets that they have interests in.

### **802.11a OFDM PHY module:**

It simulates 802.11a physical layer. 802.11a uses an OFDM (Orthogonal Frequency-Division Multiplexing) physical layer. The features simulated by the module include rate adaption, channel coding, bit errors in different channel modulations. We slightly modify the module in this research.

### **Channel Model module:**

It simulates propagation channel models that estimate the fading effect of signals during transmission. Theoretical models including free space model, two-ray ground model, Rayleigh, and Rician model can be simulated. Over 30 empirical models can also be simulated in this module.



## **3.3.1 Implementation of CoopMAC on NCTUns**

We modify the 802.11a MAC module and the OFDM PHY module in NCTUns. First of all the CoopTable data structure should be added into this module. The structure of CoopTable is shown in Fig. 3.2. Additionally we need another RateTable to better construct the scenarios we want. This is because in the original 802.11a MAC module we can only specify a static value of data rate in any 802.11a stations, and each rate is used throughout the simulation for each station. This rate does not change for different destinations, and does not change dynamically if the station moves around. In CoopMAC, a source station may transmit packets to a destination in a lower data rate because the distance between them is too long and the bit error rate is too high. In such circumstances

the source station tries to transmit packet to a near helper station in a higher data rate, and lets the helper station relay the packets to the destination station. Thus we must support multi-rate feature during a single simulation.

Destination (48 bits)	Data Rate (8 bits)	Number of failures (8 bits)
MAC address of the destination	Transmission rate between the source and the destination	Count of sequential transmission failures

Fig. 3.7 Structure of RateTable

The structure of RateTable is shown in Fig. 3.7. The user specifies the highest data rate that a station can use. But now during simulation stations may change the data rate based on detected signal strength or based on sequential transmission failures. We set the value to be 10 in our research. So if a station fails to transmit a packet for 10 times, it will consider using a lower data rate.

Data Rate (Mbit/s)	6	9	12	18	24	36	48	54
Signal Strength (dbm)	-82	-81	-79	-77	-74	-70	-66	-65

Table 3.1 Appropriate data rate under different signal strengths

Stations also change their transmission data rate to each destination station based on the signal strengths shown in Table 3.1. These signal strength thresholds are from 802.11a standard [19]. To dynamically change physical layer data rate, we modify some parts in the 802.11a PHY module.

Secondly the CoopMAC protocol described in chapter 3.2.1 is added into the 802.11a MAC module. These includes decisions of using CoopMAC or normal 802.11a MAC, the CoopRTS, HTS message, and the modified base mode in which Address 4 field is exploited.

The Duration field of these new control messages is also slightly different with the original 802.11a MAC protocol. We list them in Table 3.2.

Frame Type	Definition of Duration Field
RTS	$3T_{SIFS} + T_{CTS} + T_{Data} + T_{ACK}$
CoopRTS	$5T_{SIFS} + T_{HTS} + T_{CTS} + 2T_{Data} + T_{ACK}$
HTS	$4T_{SIFS} + T_{CTS} + 2T_{Data} + T_{ACK}$
CTS (802.11a)	$2T_{SIFS} + T_{Data} + T_{ACK}$
CTS (CoopMAC)	$3T_{SIFS} + 2T_{Data} + T_{ACK}$

Table 3.2 Definition of duration for different types of packet in CoopMAC

Timeout setting is also different in CoopMAC. They are listed in Table 3.3.

Timeout Type	Timeout Value
CTS (802.11a)	$T_{SIFS} + T_{CTS}$
HTS or CTS (CoopMAC)	$2T_{SIFS} + T_{CTS}$
CTS after HTS (CoopMAC)	$T_{SIFS} + T_{CTS}$

Table 3.3 Timeout values in CoopMAC

The modification done to the 802.11a MAC module and the OFDM PHY module is summarized in Fig. 3.8.

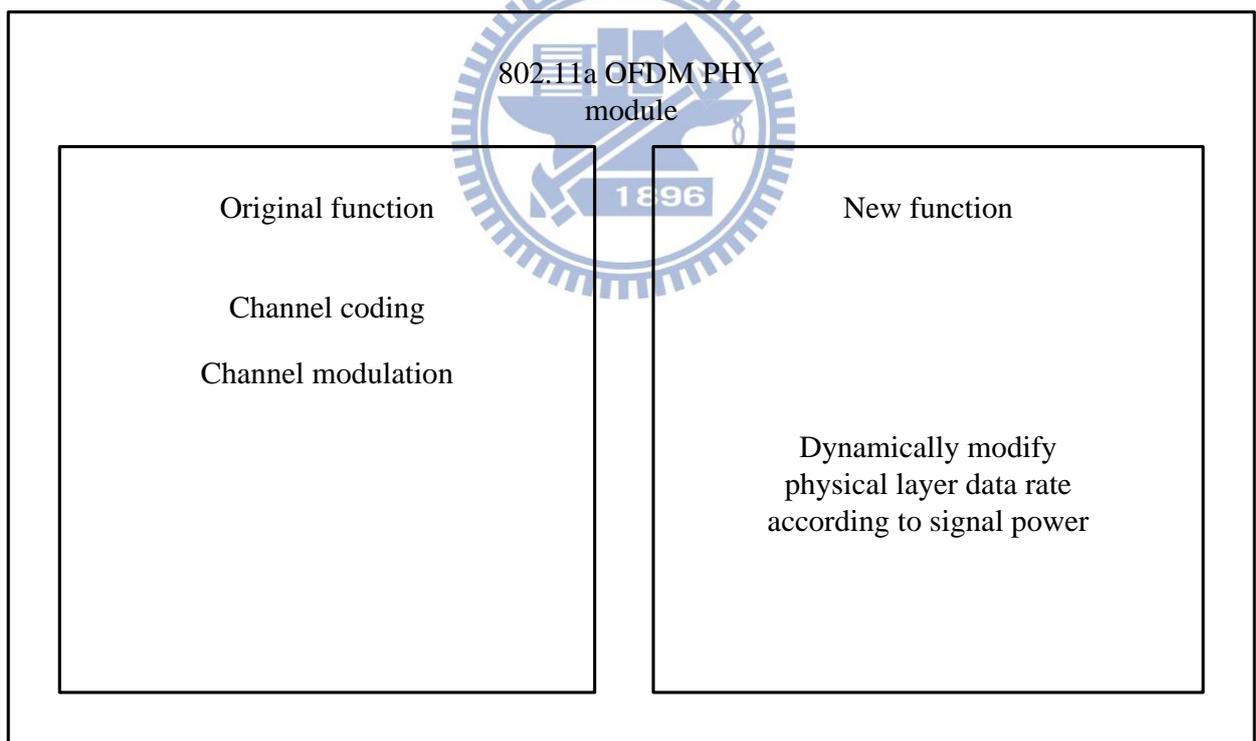
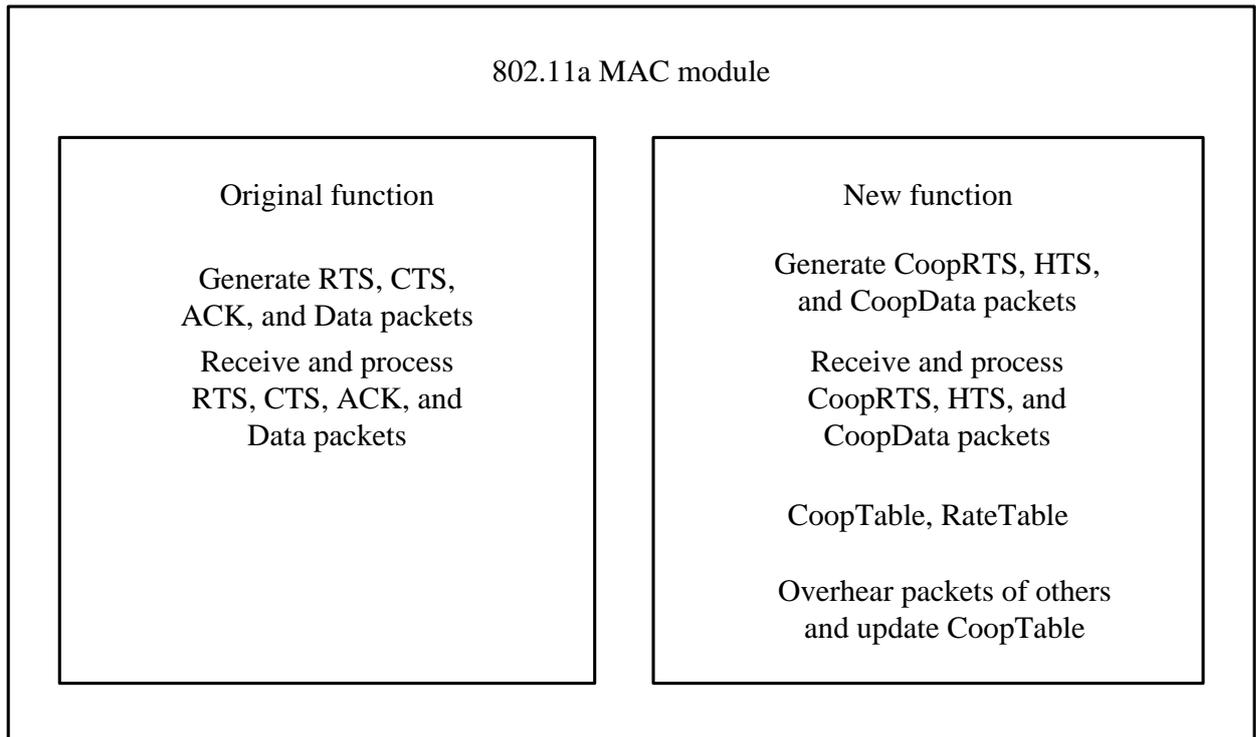


Fig. 3.8 Modification to NCTUns modules (CoopMAC)

### 3.3.2 Implementation of CMAC on NCTUns

Compared with CoopMAC, the design of CMAC is simpler. The modification contains only 802.11a MAC module. Partner queue is added into the 802.11a MAC module. Packets that are overheard without error (but not for the station itself) will be inserted into the queue. Stations now prepare another timer, which is the same as the original ACK timer, but this timer waits for ACK packets for other stations. When ACK packets do not appear as expected, the station will discover this and prepares to help the source station retransmit the packet.

Another backoff timer is also added to 802.11a MAC module. When a station helps another station retransmit packets, it should also run the backoff procedure. For such cooperative retransmission in CMAC, the backoff window size is always the minimum 15 slots. If a packet is retransmitted once by a station, the packet will be deleted from the partner queue of this station. If other stations retransmit the packet faster than a station, the packet in partner queue of this station is still kept. The backoff timer is reset but backoff window size is still the minimum 15 slots. If later in the future an ACK packet for the destination station appears, all partners deletes packets for the destination node in their partner queue.

The modification done to the 802.11a MAC module is summarized in Fig. 3.9.

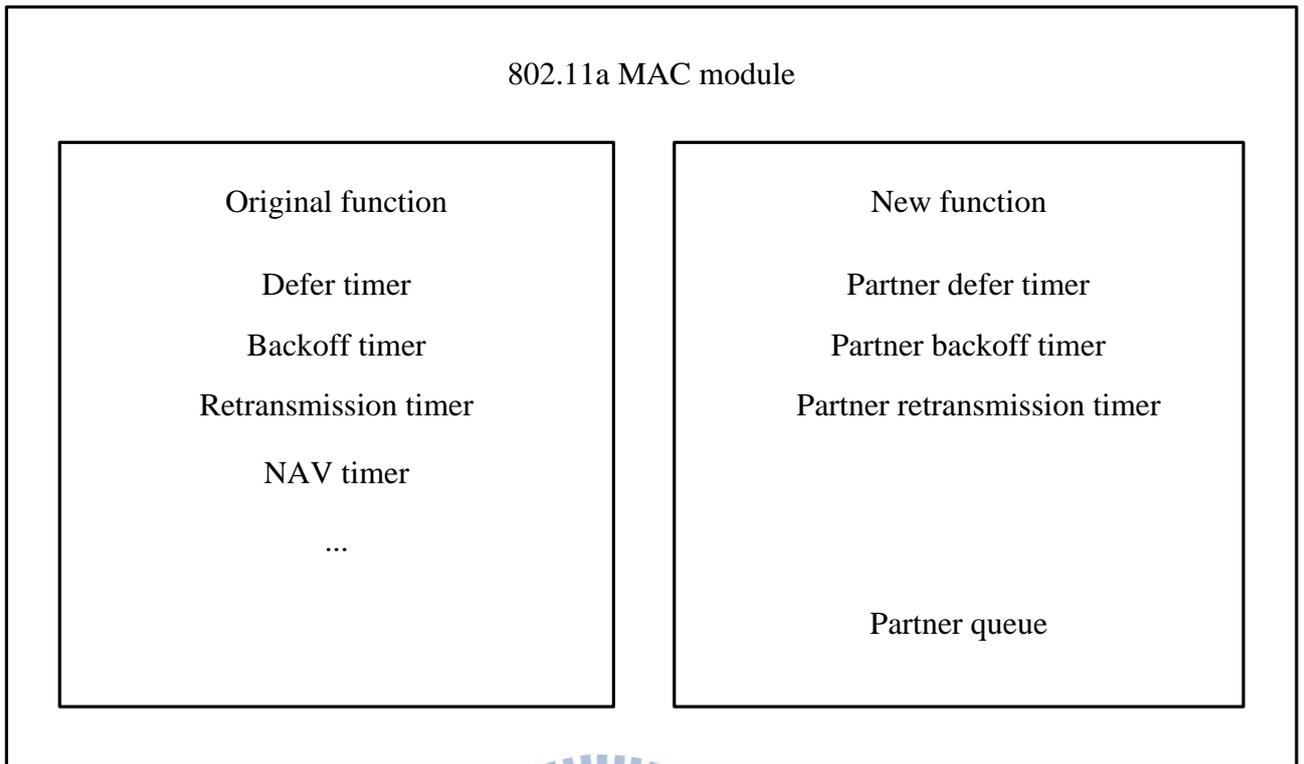
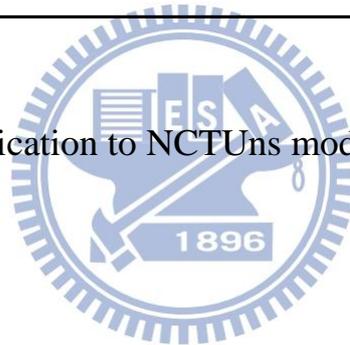


Fig. 3.9 Modification to NCTUns modules (CMAC)



# Chapter 4

## Simulation Results and Performance Evaluation

### 4.1 Simulation Results of CoopMAC Scheme

CoopMAC exploits multi-rate capability to achieve higher throughput. To validate this we first establish a simple simulation scenario, which is shown in Fig. 4.1. In the scenario the source station sends greedy UDP traffic to the destination station. In one case the helper station assists the two stations, in the other case it does not. The result throughput is shown in Table 4.1.

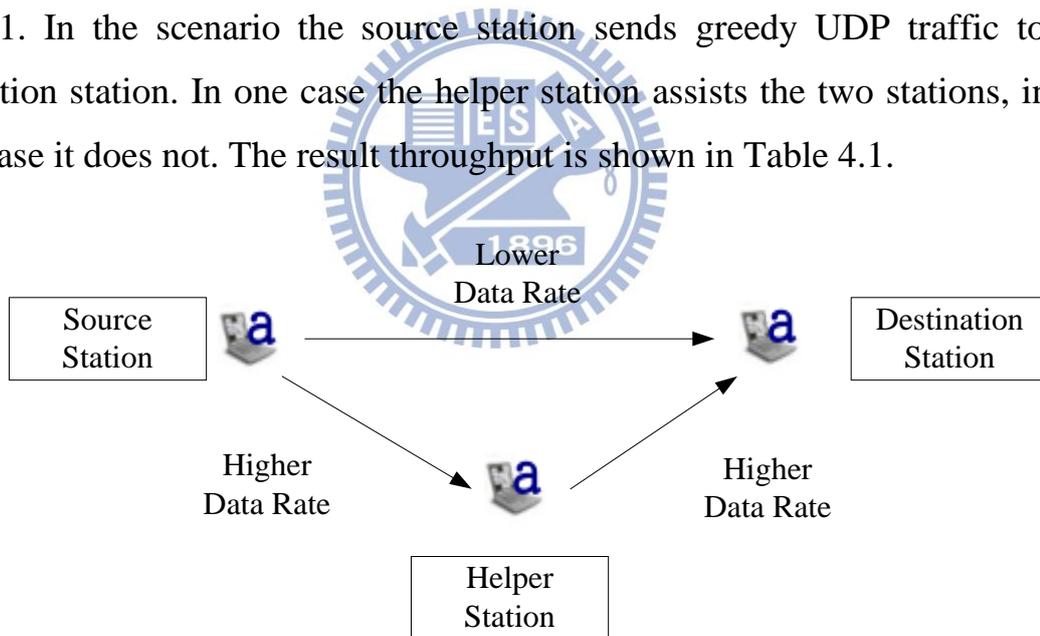


Fig. 4.1 CoopMAC simulation scenario

First we can easily see in Fig. 4.1 that if these stations use a fixed data rate, cooperative transmission results in lower throughputs. This is because all packets transmitted experience two-hop transmission using the same data rate as in direct transmission. That is why the condition  $\frac{1}{R_{sh}} + \frac{1}{R_{hd}} < \frac{1}{R_{sd}}$ , as described in

chapter 3.2.1, is used to decide whether a helper station can be exploited. Take the result in Table 4.1 for example. If the source station can communicate with the destination station using the data rate of 6 Mbps, but the data rate between the source station and the helper station is 18 Mbps. Also the data rate between the helper station and the destination station can also achieve 18 Mbps. Under such circumstances, they can use CoopMAC to achieve a higher throughput, from 5.03 Mbps to 7.00 Mbps.

<b>Physical layer data rate (Mbps)</b>	<b>Greedy UDP throughput (direct transmission) (Mbps)</b>	<b>Greedy UDP throughput (with CoopMAC) (Mbps)</b>
6	5.03	2.60
9	7.91	4.17
12	9.25	4.93
18	12.8	7.00
24	15.9	8.87
36	21.1	12.1
48	25.0	14.8
54	26.6	16.0

Table 4.1 Performance of CoopMAC

To further understand the performance of CoopMAC when there are more stations transmitting packets, we establish scenarios with different number of stations and traffic flows in the same area. An example scenario is shown in Fig. 4.2. In these scenarios we set one UDP traffic flow for every 3 stations. As in the previous scenario, they act as the source station, the helper station, and the

destination station, respectively. Due to the signal fading, the source stations communicate with the destination stations with the data rate of 6 Mbps. Meanwhile the source stations communicate with the helper stations with the data rate of 24 Mbps, and so do the helper stations and the destination stations.

The total throughput of all stations with different number of traffic flows is shown in Fig. 4.3. We can see in the result that due to the broadcast nature of wireless network and that the carrier sense region almost covers all the stations, the total throughput does not increase as the number of traffic flow increases. In both cases of direct transmission and cooperative transmission, the total throughput decreases when the number of traffic flows increases, due to increasing collisions. In the case of direct transmission, when there is only 1 traffic flow, the total throughput is 4.088 Mbit/s. When there are 10 traffic flows, the total throughput is 2.664 Mbit/s, decreased by 34%. On the other hand, in the case of cooperative communication, the throughput is also decreased by about 20%. The percentage of packets that experience collisions is shown in Fig. 4.5 as a reference. In both the case of direct transmission and cooperative communication, the percentage of collided packets over all transmitted packets increases to about 60% as the number of traffic flows becomes large.

Packet delay is also an important performance index in networking. The average packet delay is shown in Fig. 4.4. In the case of cooperative communication packets experience two-hop transmission, but they are transmitted using higher data rates. As a result the average packet delay is shorter in the case of cooperative communication. When there are more traffic flows, these traffic flows share the whole bandwidth. Thus the packet delay grows linearly in both cases.

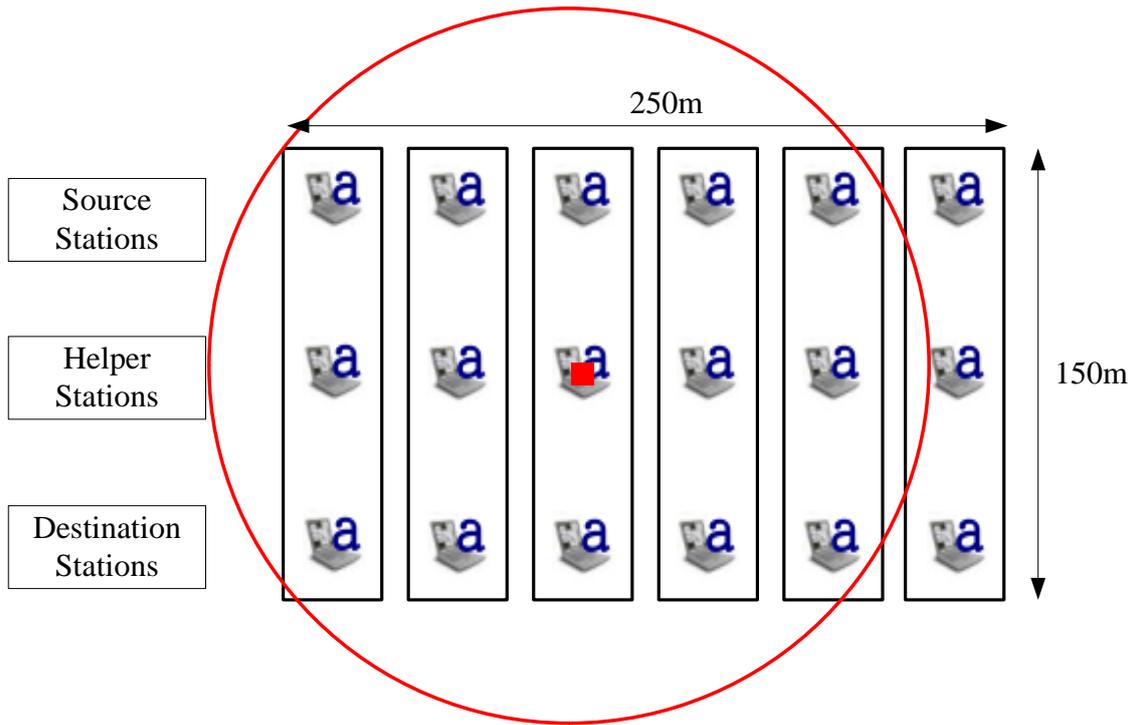


Fig. 4.2 CoopMAC simulation scenario

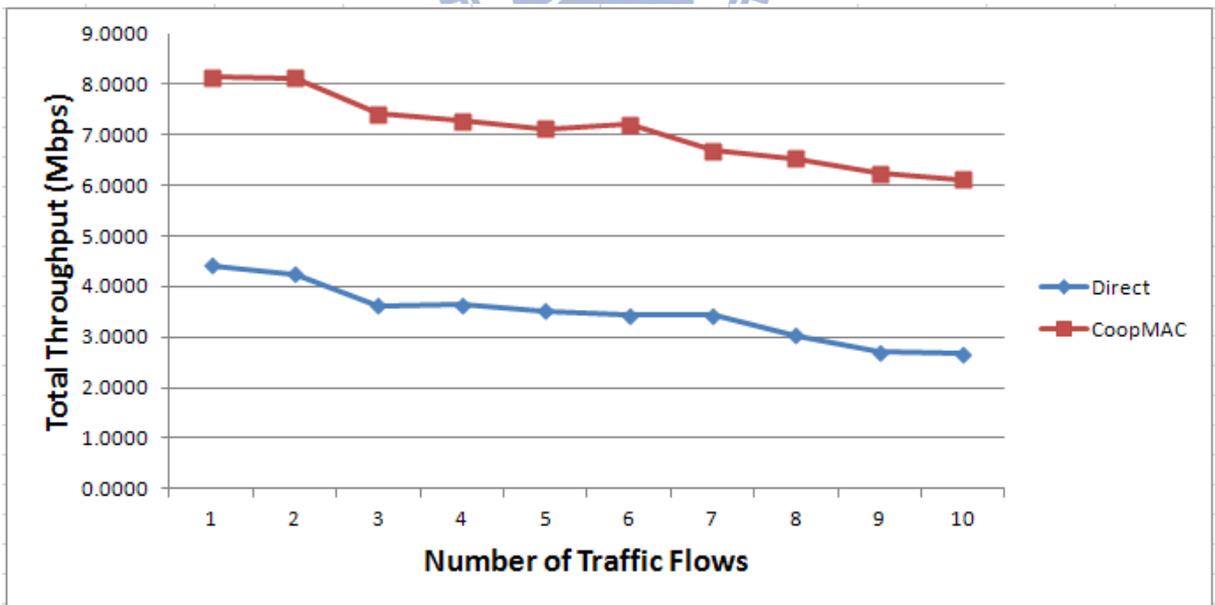


Fig. 4.3 Performance of CoopMAC – total throughput (UDP)

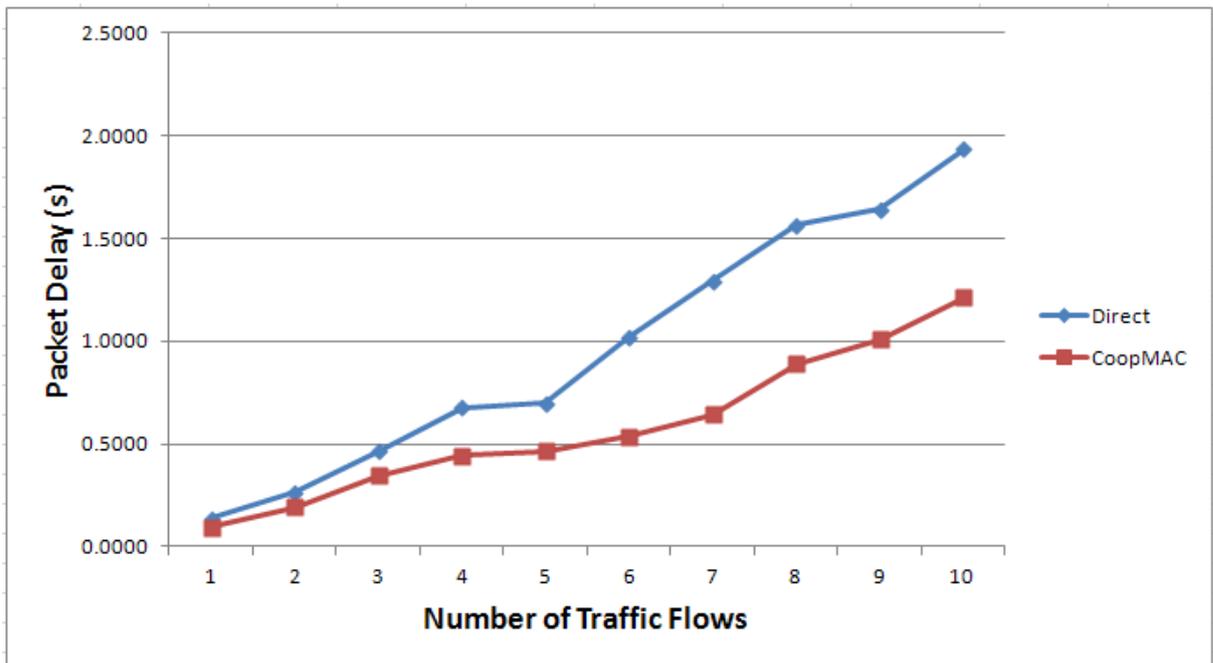


Fig. 4.4 Performance of CoopMAC – packet delay (UDP)

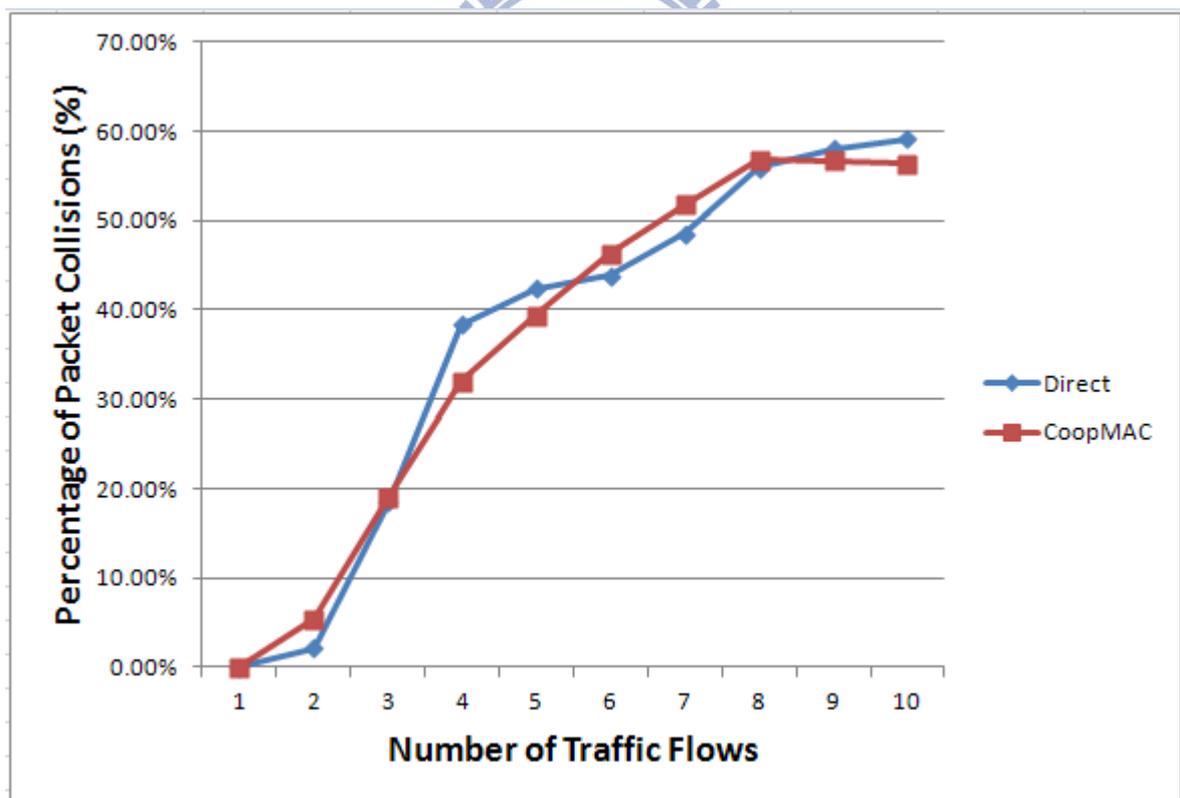


Fig. 4.5 Performance of CoopMAC – percentage of packet collisions (UDP)

The performance of TCP traffic is also evaluated. The total TCP throughput with different number of traffic flows is shown in Fig. 4.6. The throughputs of TCP and UDP traffic are compared in Fig. 4.7 and Fig. 4.8. Compared with UDP traffic, TCP traffic has only half throughputs in both cases of direct transmission and CoopMAC. This is because the congestion control of TCP would reduce its transmission speed when collisions occur, while UDP does not.

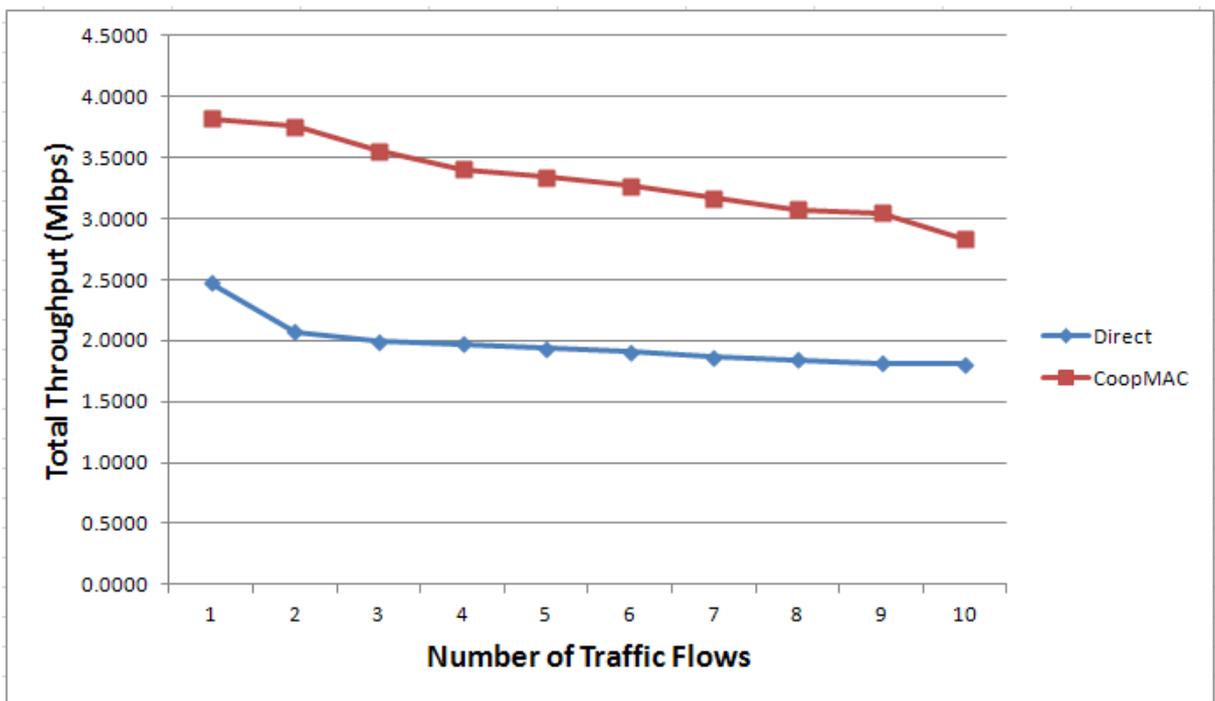


Fig. 4.6 Performance of CoopMAC – total throughput (TCP)

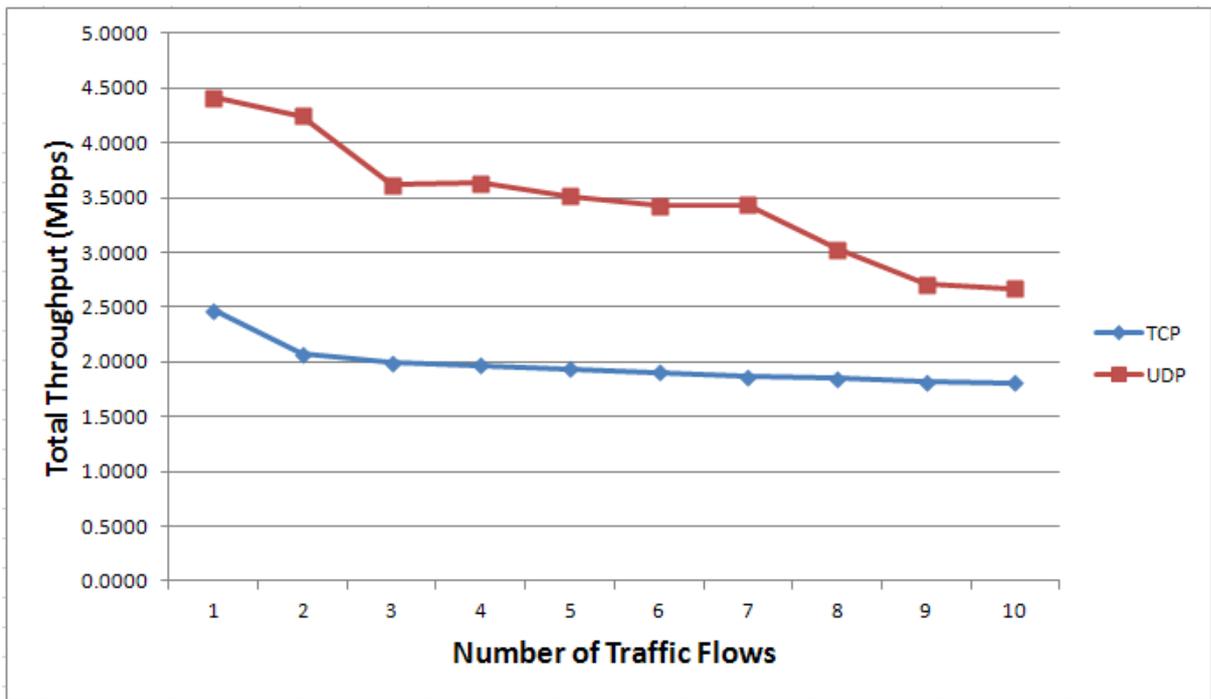


Fig. 4.7 Performance of original 802.11a MAC – total throughput

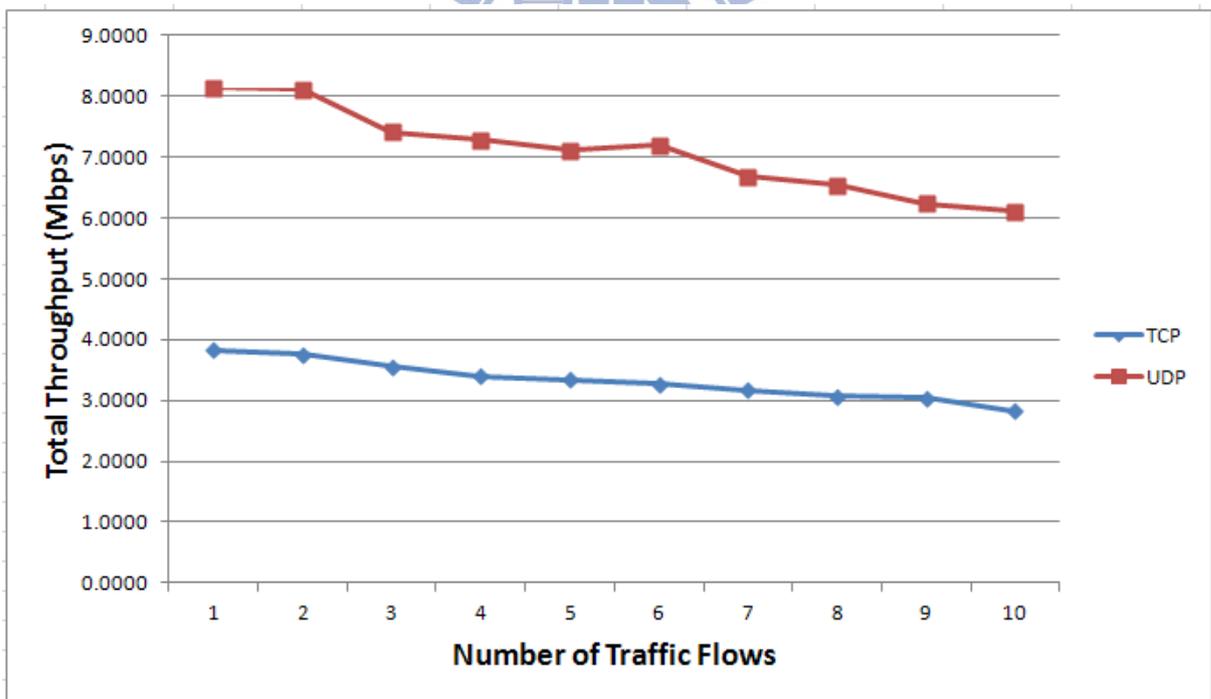


Fig. 4.8 Performance of CoopMAC – total throughput

## 4.2 Simulation Results of CMAC Scheme

Different from CoopMAC, CMAC does not aim to exploit multi-rate capability to improve throughputs. Alternatively helper stations help the source stations fast-retransmit packets in CMAC. We use the same scenarios and settings as in chapter 4.1 to evaluate the performance achieved by CMAC protocol.

The total throughput of all stations with different number of traffic flows is shown in Fig. 4.9. In average the CMAC protocol has a 1.89% improvement in throughput compared with the original protocol. The average packet delay is shown in Fig. 4.10. In average the packet delay is 0.016 second less if CMAC is used. The percentage of packets that experience collisions is shown in Fig. 4.11. If CMAC is used, this percentage is 0.33% higher than the original protocol.

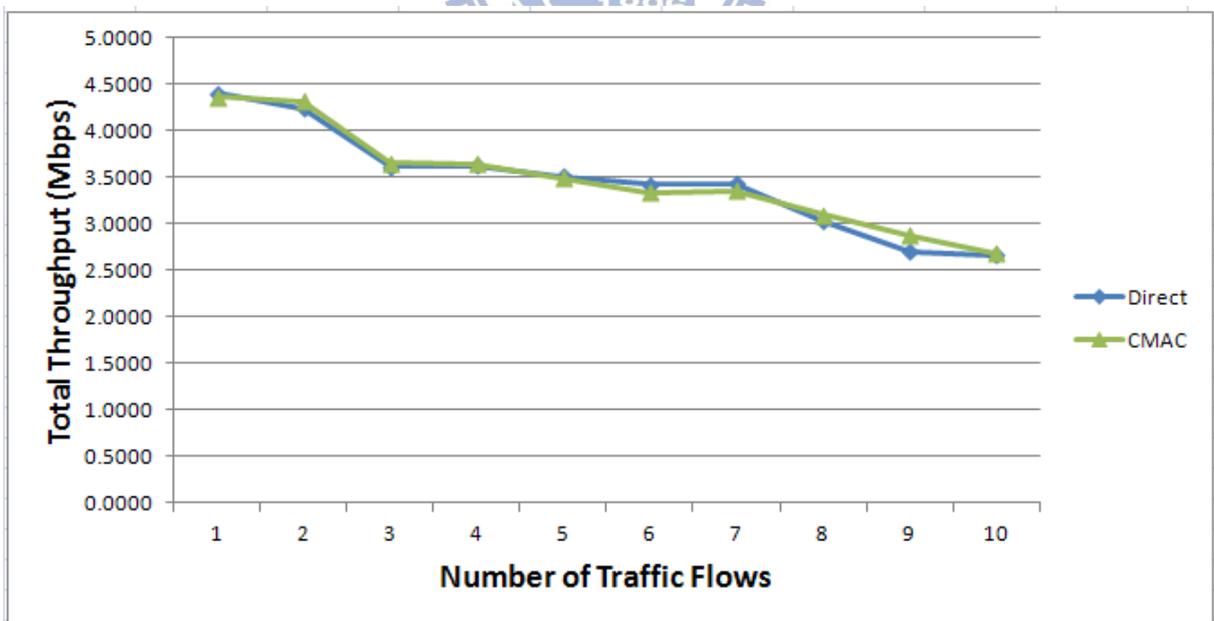


Fig. 4.9 Performance of CMAC – total throughput (UDP)

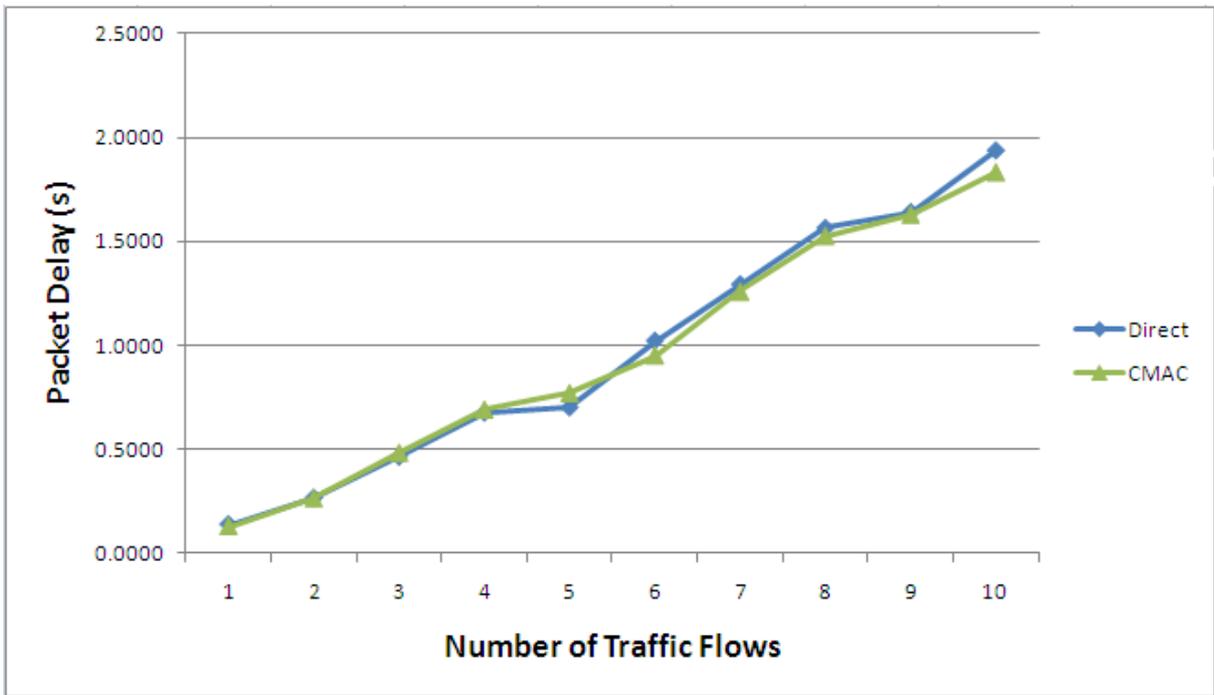


Fig. 4.10 Performance of CMAC – packet delay (UDP)

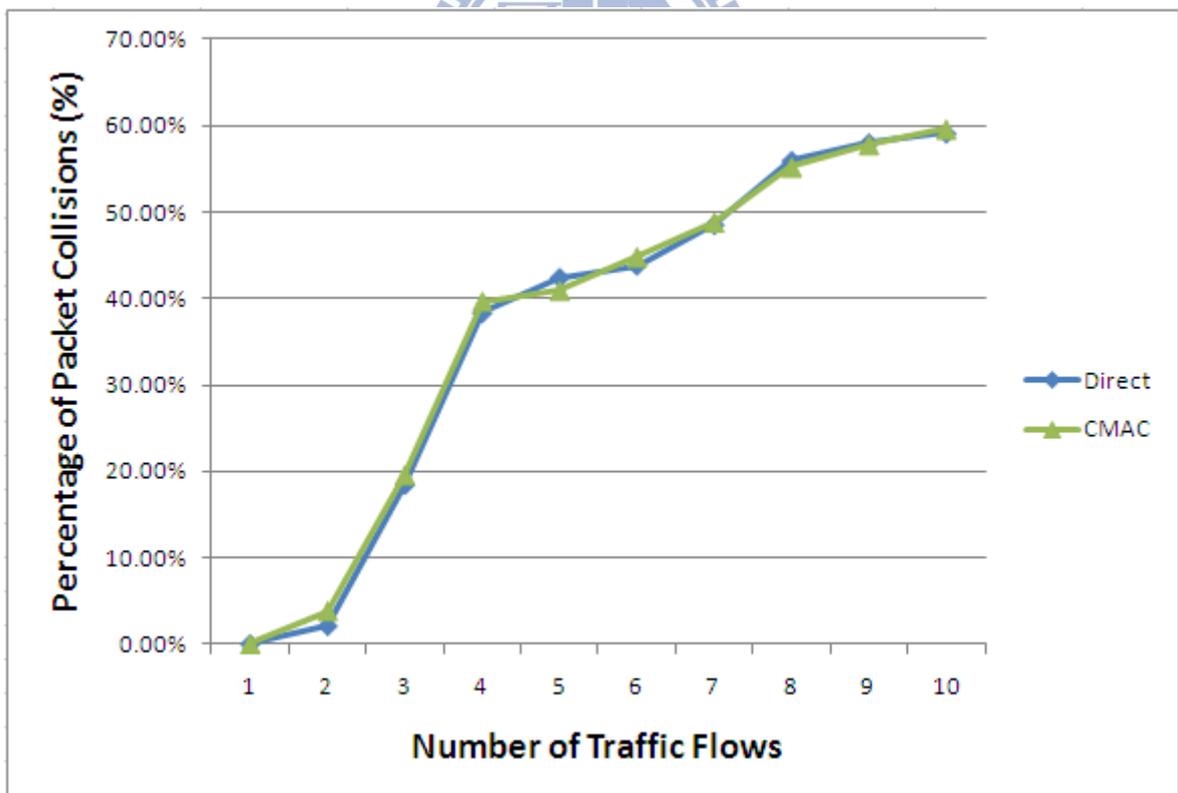


Fig. 4.11 Performance of CMAC – percentage of packet collisions (UDP)

We establish two more scenarios with moving stations to evaluate the performance of CMAC protocol. The first scenario we use to evaluate CMAC is shown in Fig. 4.12. In the scenario the source station first moves toward the destination station. The helper station is between the source station and the destination station. The helper station will try to help retransmit packets if the packets are lost or in error.

After the source station reaches the destination station, it keeps moving away from the destination station. The helper station now also becomes far away from the source station and less possible to receive correct packets, so it tends to be less helpful to the source station during the last-half period.

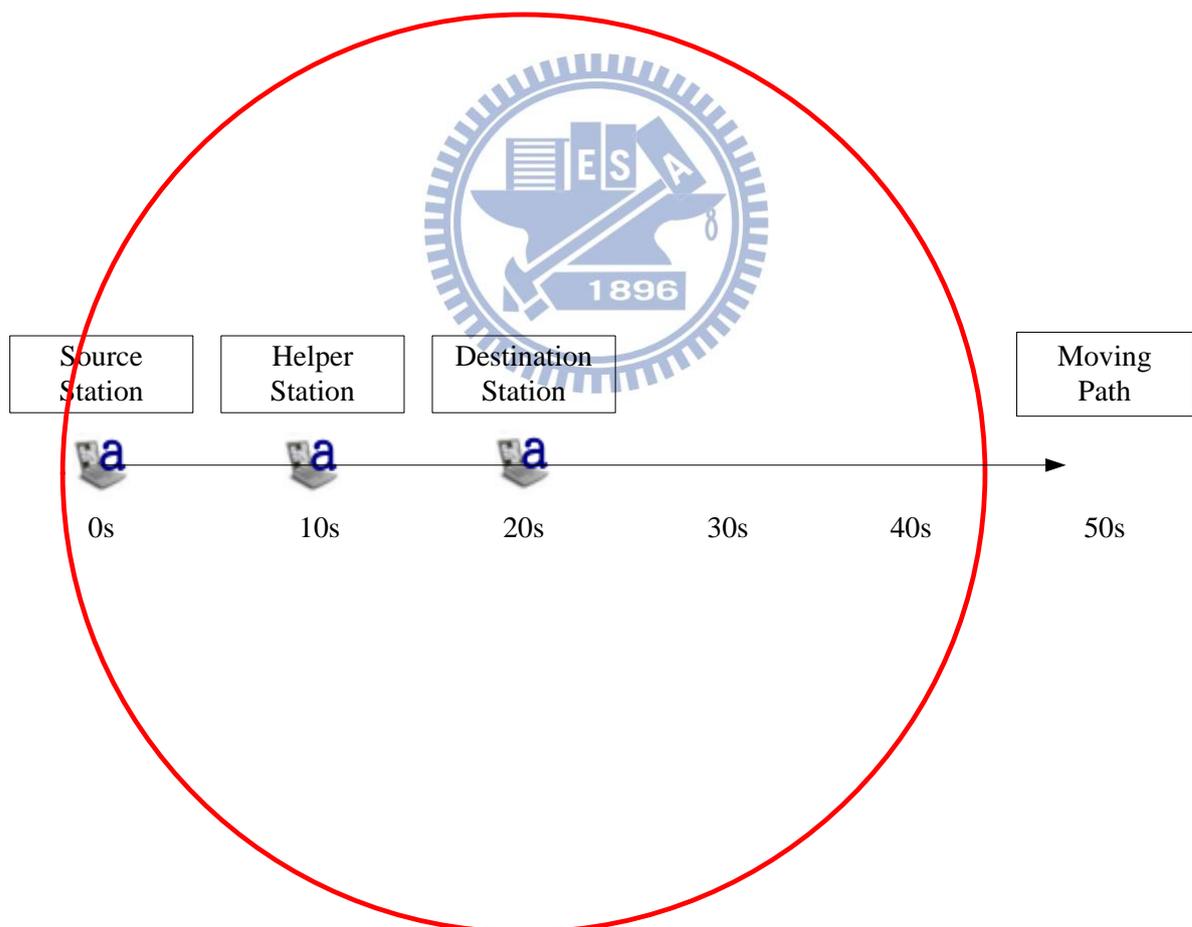


Fig. 4.12 CMAC simulation scenario 1

The result is shown in Fig. 4.13. We can see that CMAC performs a little better than the original MAC protocol only in the period when the source station moves away from the destination station and the helper station (40<sup>th</sup> second to 60<sup>th</sup> second). This is because in this period the BER (bit error rate) between the source station and the destination station is high. Thus the helper station fast-retransmits some packets that are not correctly received at the destination station.

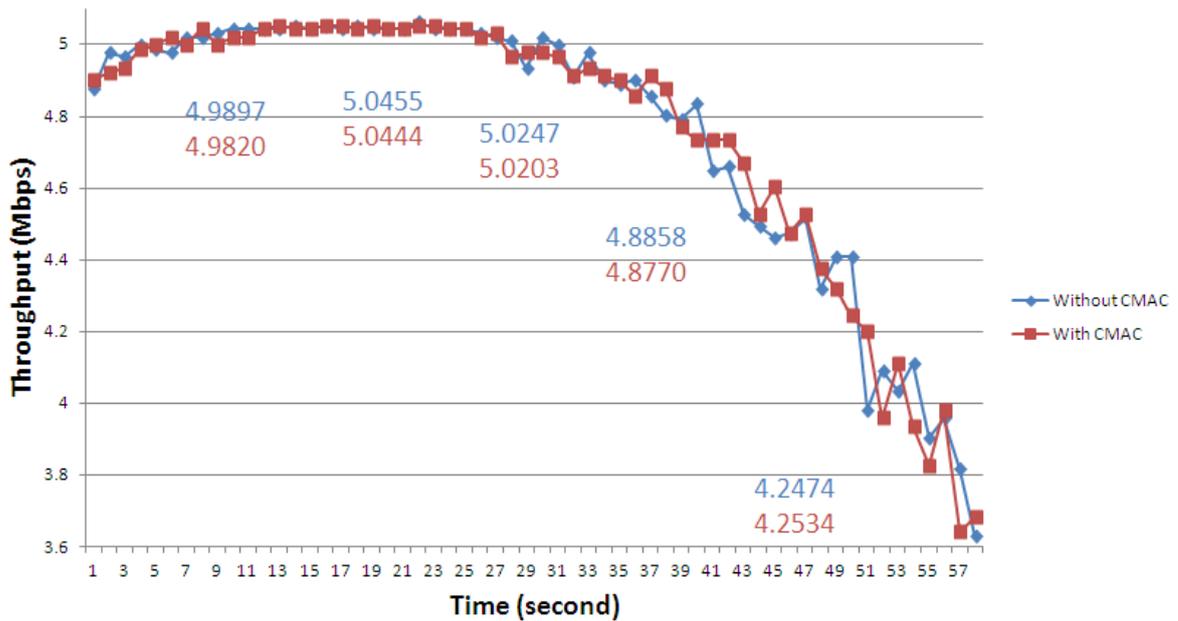


Fig. 4.13 Performance of CMAC – scenario 1

The second scenario is shown in Fig. 4.14. In this scenario the source station moves around the destination station. In the period of 0<sup>th</sup> second to 15<sup>th</sup> second, 2 helper stations help the source station retransmit packets. In the period of 15<sup>th</sup> second to 30<sup>th</sup> second, another helper station joins. And in the period of 30<sup>th</sup> second to 45<sup>th</sup> second, only 1 helper station remains to help. At last in the period

of 45<sup>th</sup> second to 60<sup>th</sup> second, no helpers can help the source station. The throughput under these different situations is shown in Fig. 4.15. We can see in the result that when there are two helper stations nearby, the throughput is a little better.

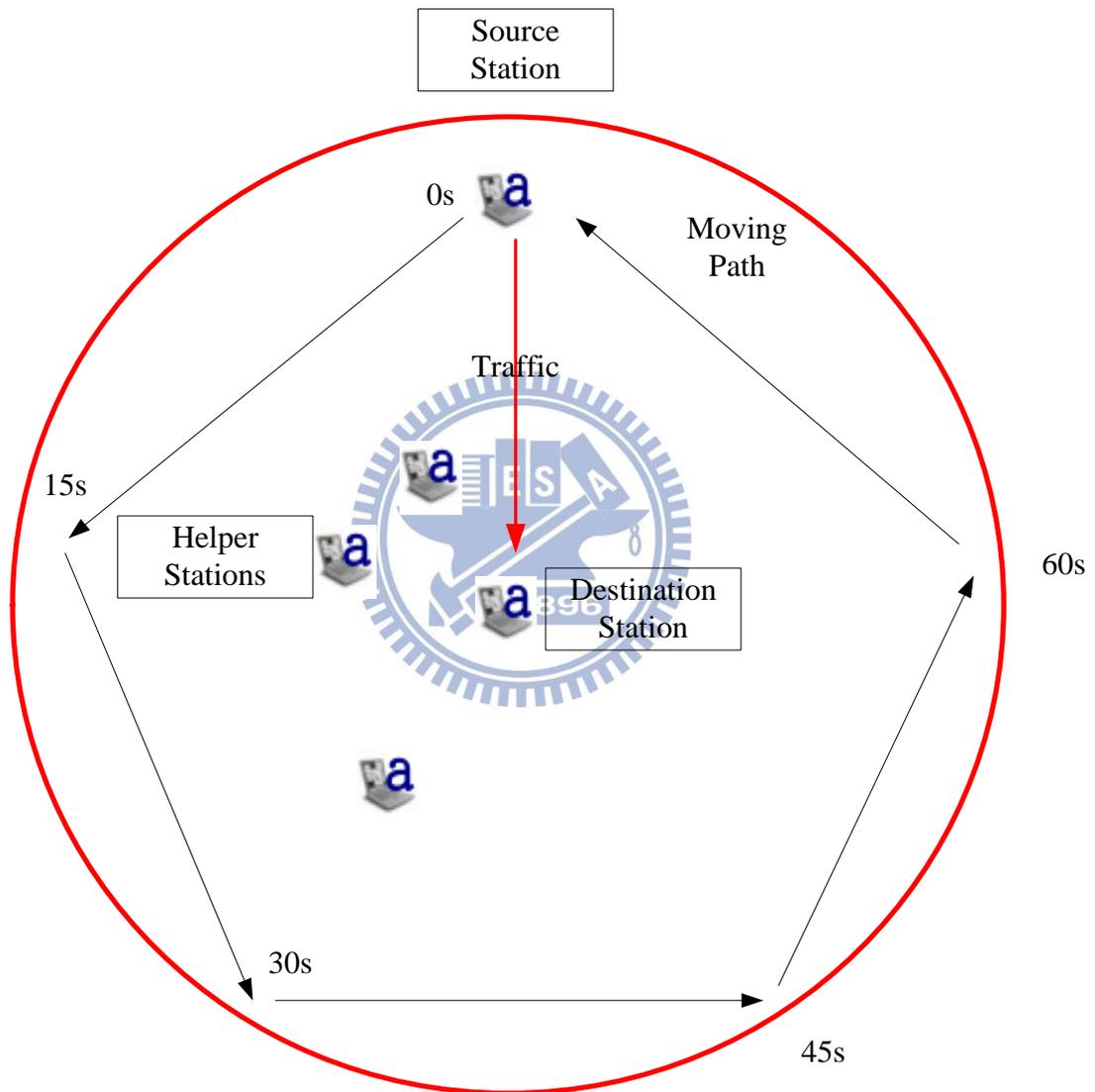


Fig. 4.14 CMAC simulation scenario 2

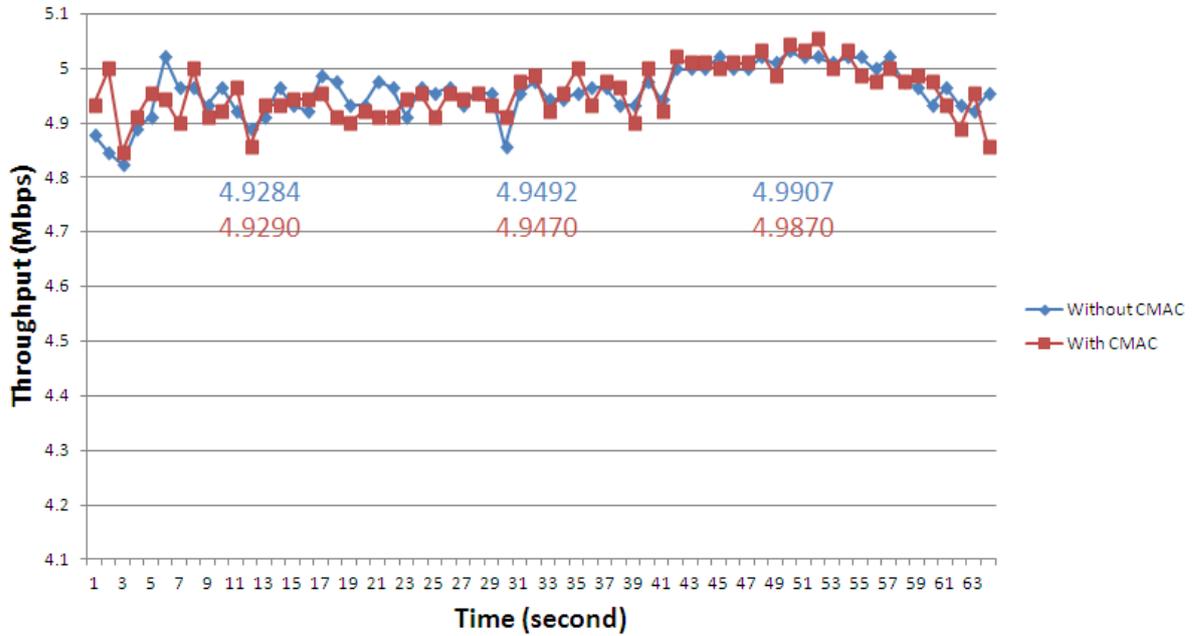


Fig. 4.15 Performance of CMAC – scenario 2

### 4.3 Comparison between These Schemes

In the following we compare the results of the two selected cooperative networking schemes. First we compare the performance of these schemes in the scenario shown in Fig. 4.2. Performance indexes we consider include throughput, packet delay, and percentage of collided packets. The compared results are shown through Fig. 4.16 to Fig. 4.18.

The throughput achieved by CoopMAC is the highest because CoopMAC exploits the multi-rate capability of wireless network. CoopMAC also ensures the benefit of cooperation using the condition  $\frac{1}{R_{sh}} + \frac{1}{R_{hd}} < \frac{1}{R_{sd}}$ , as mentioned in chapter 3.2.1. If the helper station is really idle to help the source station and the two-hop transmission succeeds, a higher throughput should be achieved. On the other hand CMAC only performs a little better than the original MAC protocol

(1.89% improvement in average) since in CMAC the helper stations does not use higher data rate. Instead the helper stations only save the backoff time and fast-retransmit correctly-received packets. Thus the performance improvement achieved by CMAC is little compared with CoopMAC.

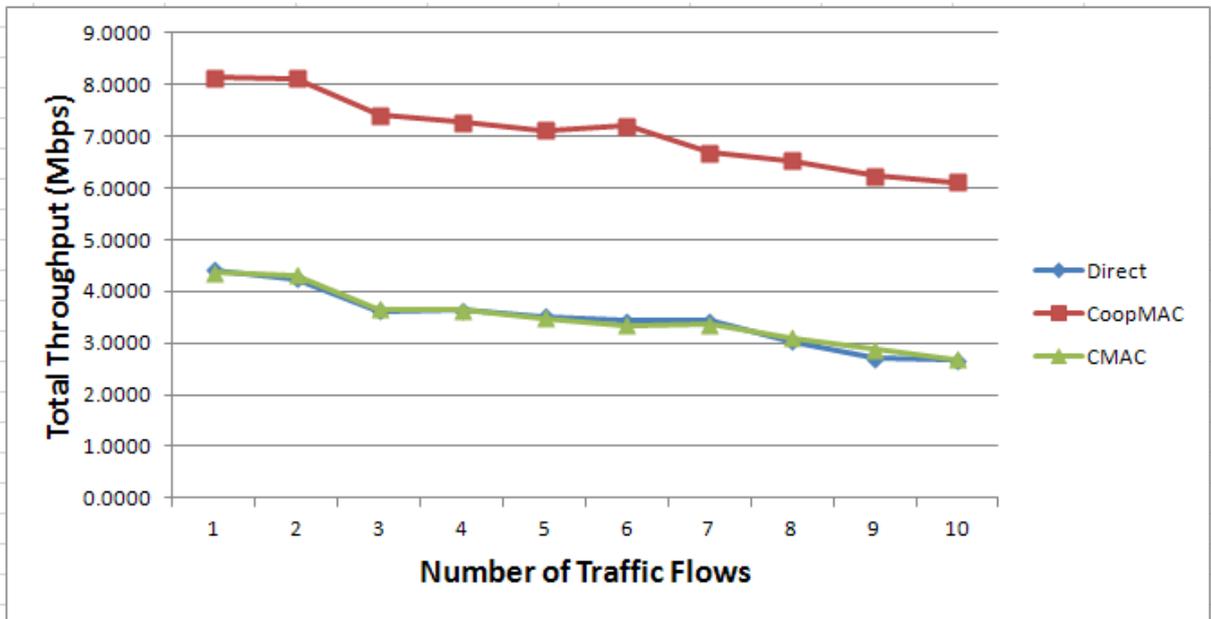


Fig. 4.16 Comparison – total throughput

CoopMAC also results in shorter packet delay. This also comes from the higher data rate used by the two-hop transmission. As for the case of CMAC, the packet delay is almost the same as the original MAC protocol.

Three protocols have similar amount of packet collisions. When the number of traffic flows grows to 10, the percentage of packet collisions achieves 55~60% in all three protocol evaluated. This explains the throughput decrease that occurs as there are more traffic flows.

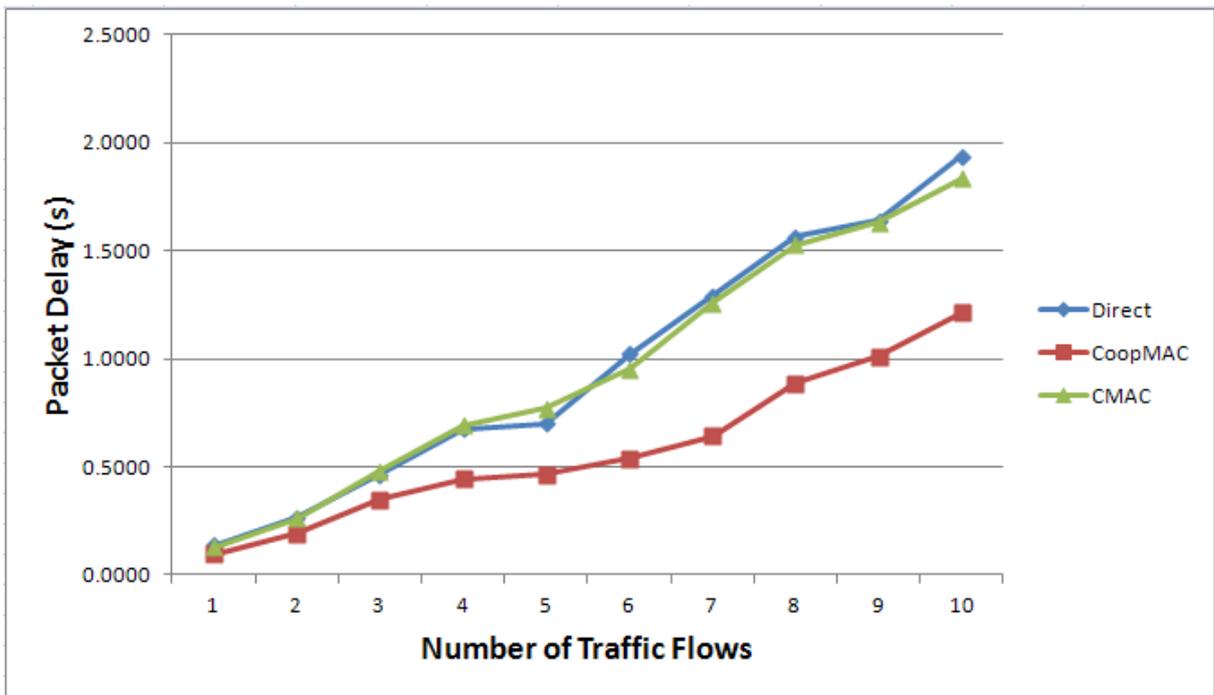


Fig. 4.17 Comparison – packet delay

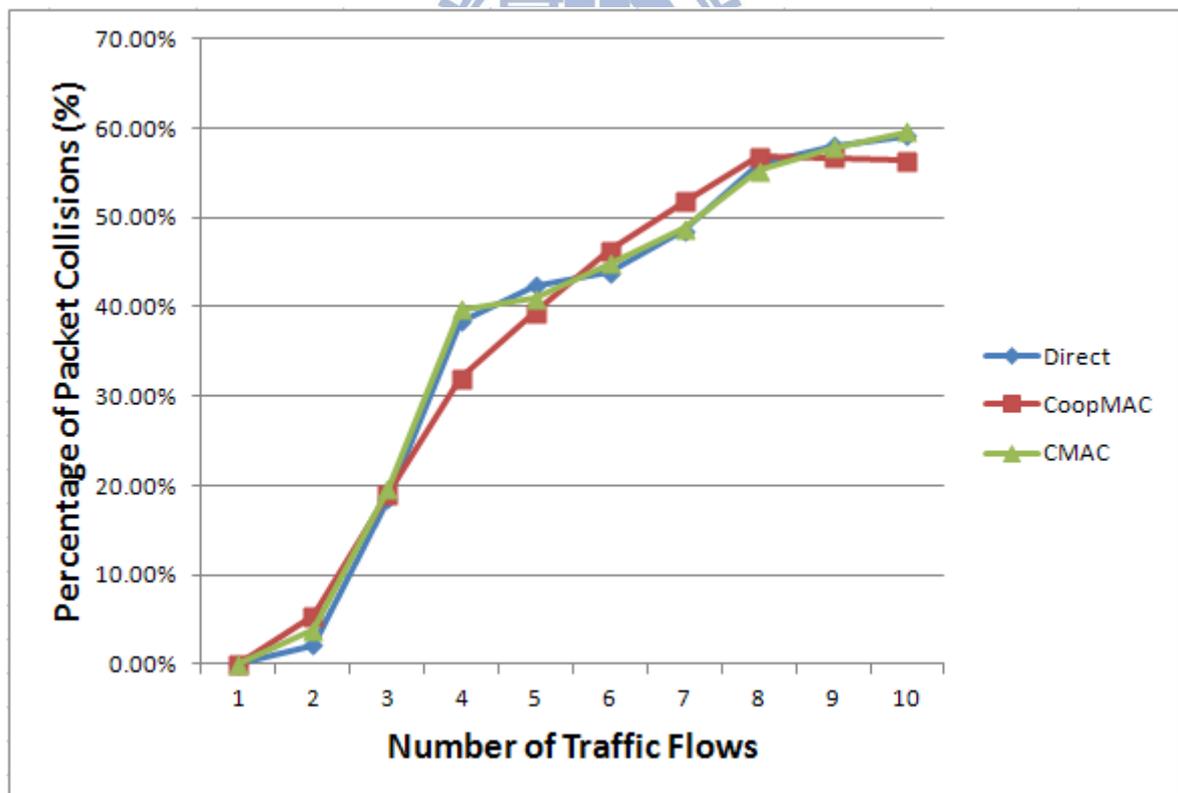


Fig. 4.18 Comparison – percentage of packet collisions

Additionally we compare the performance in scenarios with moving stations. The scenarios used are the two mentioned in chapter 4.2 (Fig. 4.12 and Fig. 4.14). In the first scenario in which the source station moves along a line, we can observe that CoopMAC also experiences signal fading like CMAC and has lower throughput when the source station goes far away from the destination station. But even under such signal fading, CoopMAC still achieve higher throughput. The compared results are shown in Fig. 4.19, 4.20, and 4.21. Note that CoopMAC resumes to direct transmission due to too many transmission failures at about 45<sup>th</sup> second.

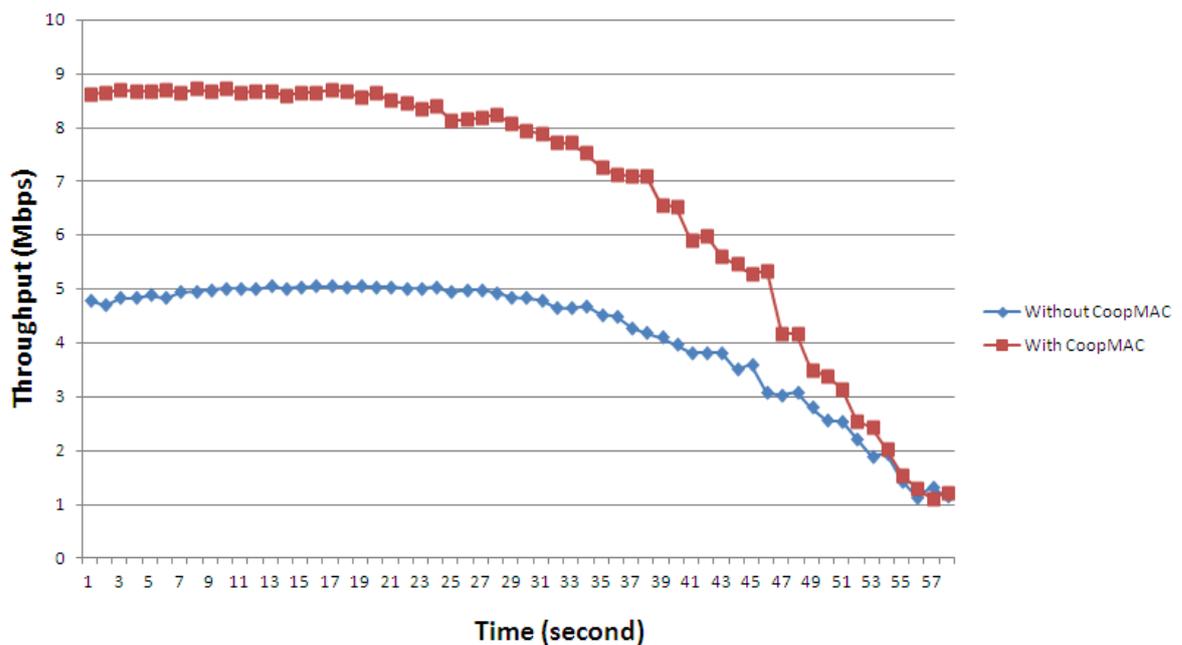


Fig. 4.19 Performance of CoopMAC – scenario 1

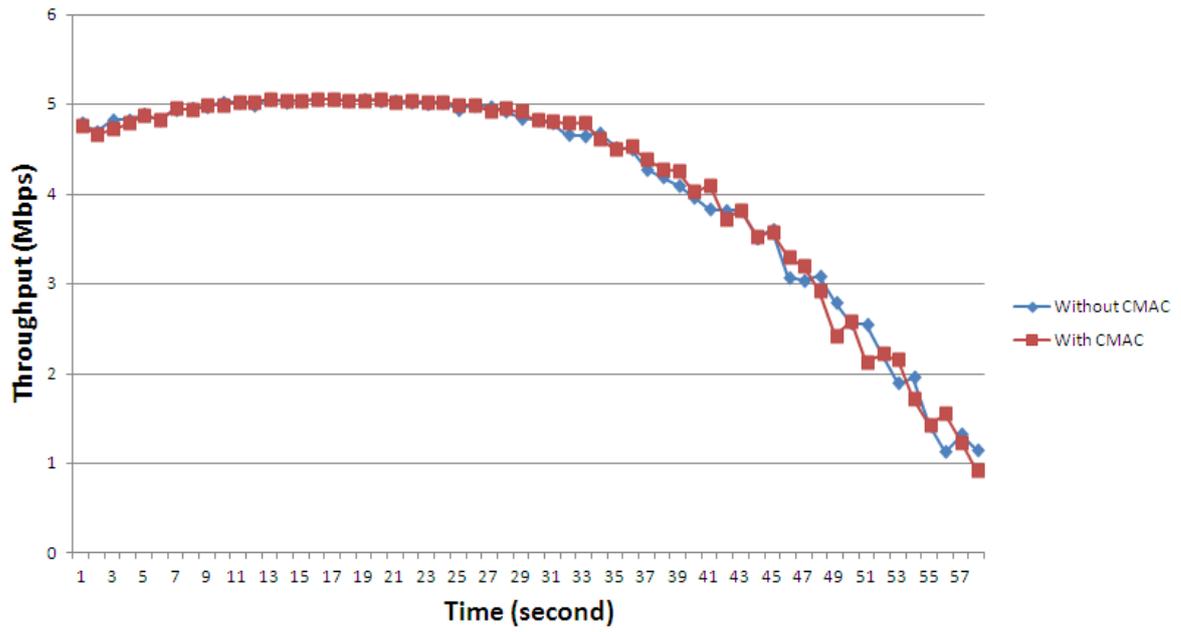


Fig. 4.20 Performance of CMAC – scenario 1

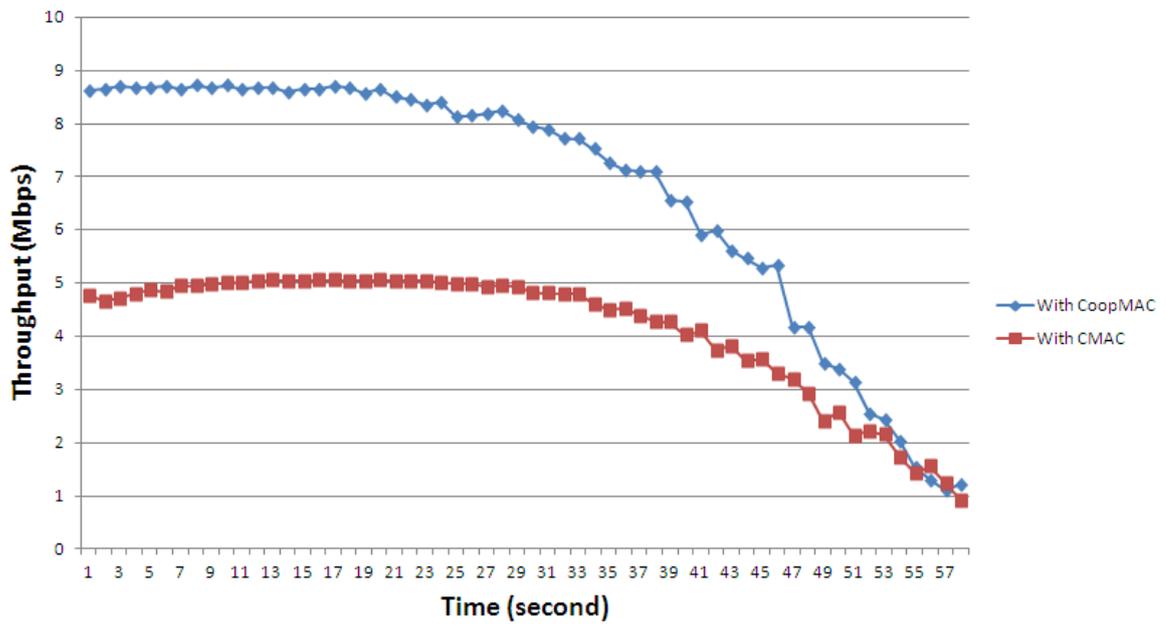


Fig. 4.21 Comparison - scenario 1

In the second scenario, when the source station moves around, the number of potential helpers differs because of its position. For CMAC, the performance is better when there are more potential helpers nearby. CoopMAC exploits two-hop transmission with higher data rate. But CoopMAC can only exploit one such helper station at a time. Thus the throughput does not change with the number of potential helper stations. The compared results are shown in Fig. 4.22, 4.23, and 4.24.

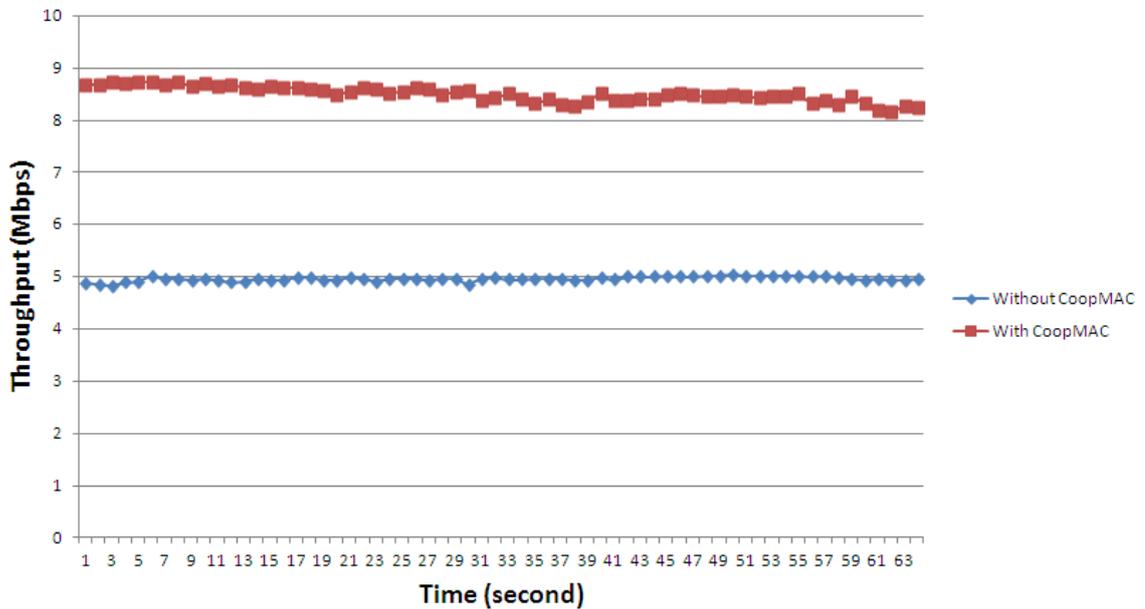


Fig. 4.22 Performance of CoopMAC – scenario 2

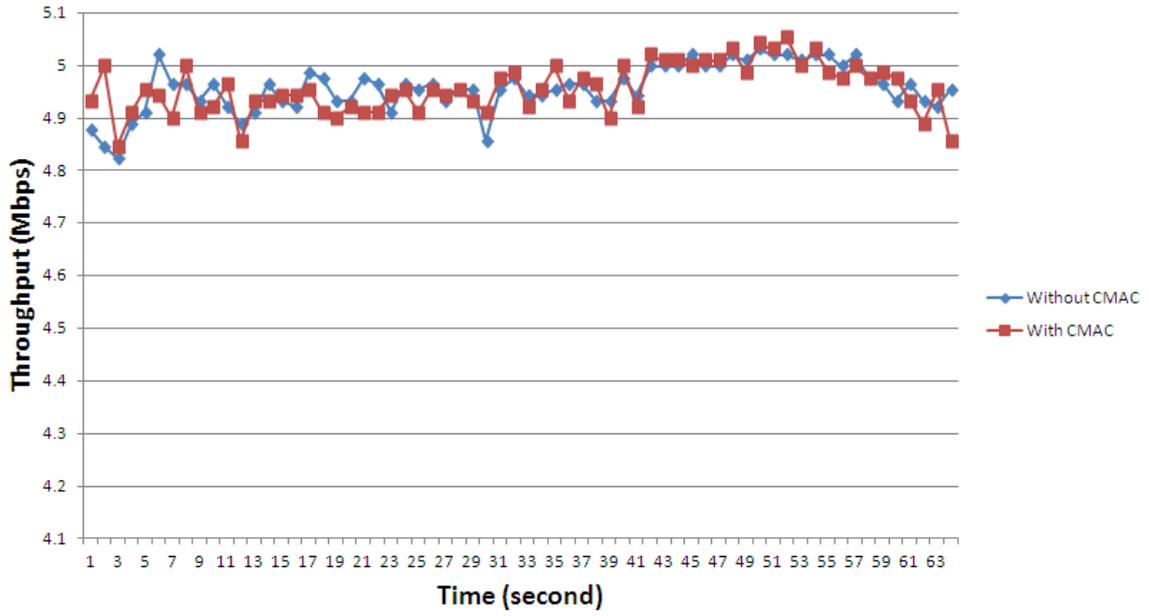


Fig. 4.23 Performance of CMAC – scenario 2

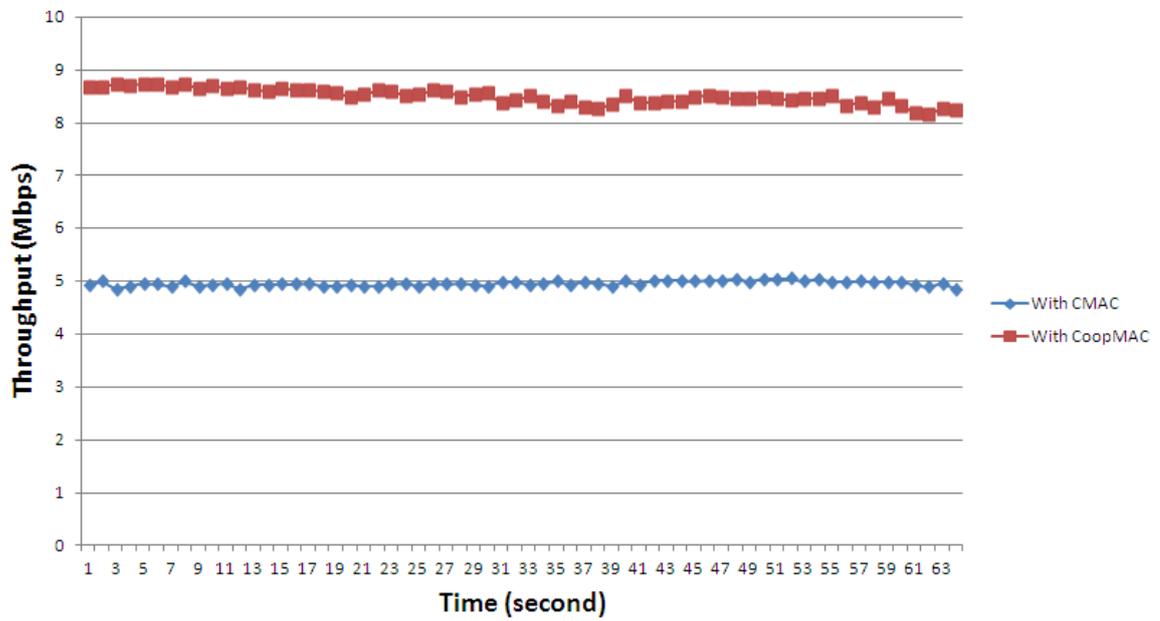


Fig. 4.24 Comparison – scenario 2

# Chapter 5

## Discussion

Among many developed cooperative communication schemes, we choose CoopMAC and CMAC, implement their capabilities on NCTUns network simulator, and evaluate their performances. The performance indexes we evaluate include UDP/TCP traffic throughput, packet delay, and packet collision percentage. We also consider two kinds of scenarios. The first one is the static scenario. In the scenario stations do not move, and we evaluate the performances of the two cooperative networking schemes with different number of traffic flows. The second one is the dynamic scenario. Stations in this kind of scenarios can move and thus some factors, such as signal strength and BER, change as the stations move around.

In general CoopMAC performs better than CMAC, as shown in chapter 4.3. But in CoopMAC to achieve cooperation, the helper station must be idle during the period of such cooperation. So if a helper station keeps helping to transmit packets, it cannot transmit its own packets. On the other hand in CMAC, the fast-retransmissions do not happen all the time (Of course when the BER is high, retransmissions happen more often). Thus even if the helper station helps retransmit packets when needed, it can still transmit its own packets afterward. But CMAC improves the throughput little. This is because it only saves the backoff time that is originally wasted by the source station. Meanwhile CMAC does not exploit higher data rate.

# Chapter 6

## Future Work

The following issues can be further studied in the future.

- Performance of other cooperative networking schemes

Other cooperative networking schemes mentioned in chapter 2 are quite similar to either CoopMAC or CMAC. But their performances are still worth studying.

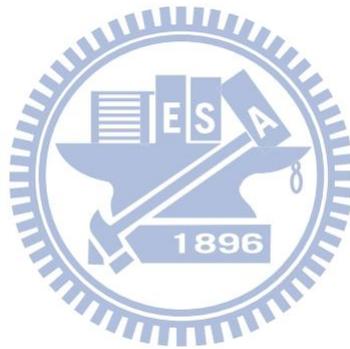
- When is appropriate to use cooperative networking schemes?

The simulations conducted in this thesis are generally simple. For CoopMAC, if there are many helper stations around the source station but they are often busy, the throughput improvement then becomes limited. For CMAC, if there are too many helper stations around, they may compete to retransmit packets and result in more collisions. The two schemes can both be further improved to handle such scenarios.

- Combine the capabilities of CoopMAC and CMAC

CMAC does not exploit the multi-rate capability. Though this makes it simple, CMAC has little performance improvement. If CMAC retransmits packets using higher data rate as CoopMAC does, the performance may become better. Also in this thesis we do not examine the performance of CMAC under higher data rates. In our study CMAC has only 1.89% improvement in throughput, as mentioned in chapter 4.2.

By conducting simple simulation with 3 stations, we can find out that such throughput improvement can reach 8~12% if higher data rates are used. Such improvement can be further studied.



# Chapter 7

## Conclusion

In the paradigm of cooperative communication /networking, except for the source station and the destination station, other helper station(s) are also involved in the communication. Thus in general researchers also see wireless ad hoc routing protocols such as AODV, DSR as schemes of cooperative communication, like in [2]. In this thesis we concentrate on MAC layer designs. Unlike ad hoc routing protocols, which may have lower throughput due to multi-hop transmission, these MAC layer schemes only exploit two-hop transmission and manage to achieve higher throughput.

We choose two such schemes and evaluate their performances. CoopMAC exploits multi data rate feature of the 802.11 network, while CMAC only help fast-retransmit packets. By conducting simulations using NCTUns, we can see that CoopMAC can achieve higher throughputs. But the higher throughput relies on idle helper stations. On the other hand the throughput improvement of CMAC is very little, but even if the helper station has its packets to transmit, it can occasionally help the retransmission of others without disturbing its own packet transmission.

It is possible to combine the advantages of these two schemes to achieve better performance when the helper stations also have their own packets to transmit.

# Bibliography

- [1] Aria Nosratinia, Todd E. Hunter, and Ahmadreza Hedayat, "Cooperative Communication in Wireless Networks," IEEE Communication Magazines, Vol. 42, Issue 10, October 2004.
- [2] Gerhard Kramer, Ivana Maric and Roy D. Yates, "Cooperative Communications," Foundations and Trends in Networking, Vol. 1, Issue 3, August 2006.
- [3] Dzmitry Kliazovich, Fabrizio Granelli, Nelson L. S. da Fonseca, Cooperative Inter-node and Inter-layer Optimization of Network Protocols
- [4] ISO/IEC 7498-1:1994, Information technology -- Open Systems Interconnection -- Basic Reference Model: The Basic Model
- [5] C. Perkins, E. Belding-Royer, and S. Das, "Ad hoc On-Demand Distance Vector (AODV) Routing", July 2003.
- [6] D. Johnson, Y. Hu, and D. Maltz, "The Dynamic Source Routing Protocol (DSR) for Mobile Ad Hoc Networks for IPv4", February 2007.
- [7] Hao Zhu and Guohong Cao, "rDCF: A Relay-Enabled Medium Access Control Protocol for Wireless Ad Hoc Networks," IEEE transactions on Mobile Computing, Vol. 5, No. 9, September 2006.
- [8] Pei Liu, Zhifeng Tao, Sathya Narayanan, Thanasis Korakis, and Shivendra S. Panwar, "CoopMAC: A cooperative MAC for wireless LANs," IEEE Journal on Selected Areas in Communications, Vol. 25, No. 2, February 2007.
- [9] Xin Wang and Chenyang Yang, "A MAC Protocol Supporting Cooperative

Diversity for Distributed Wireless Ad Hoc Networks,” IEEE 16<sup>th</sup> International Symposium on Personal, Indoor, and Mobile Radio Communicaitons, September 2005.

[10] Sai ShanKar N, Chun-Ting Chou, Monisha Ghosh, “Cooperative Communication MAC (CMAC) – A New MAC protocol for Next Generation Wireless LANs,” International Conference on Wireless Networks, Communications and Mobile Computing, June 2005.

[11] Dina Katabi, Sachin Katti, Wenjun Hu, Hariharan Rahul, and Muriel Medard, “On Practical Network Coding for Wireless Environments,” International Zurich Seminar on Communications, February 2006.

[12] Bin Zhao, Student Member, IEEE, and Matthew C. Valenti, Member, IEEE, “Practical Relay Networks: A Generalization of Hybrid-ARQ,” IEEE Journal on Selected Areas in Communications, Vol. 23, No. 1, January 2005.

[13] Gentian Jakllari, Srikanth V. Krishnamurthy, Michalis Faloutsos, Prashant V. Krishnamurthy, and Ozgur Ercetin, “A Framework for Distributed Spatio-Temporal Communications in Mobile Ad hoc Networks,” IEEE 25<sup>th</sup> International Conference on Computer Communications (INFOCOM 2006), April 2006.

[14] Shie-Yuan Wang, Chih-Liang Chou, Chih-Che Lin, and Chih-Hua Huang, “The Protocol Developer Manual for the NCTUns 6.0 Network Simulator and Emulator”.

[15] Shie-Yuan Wang, Chih-Liang Chou, and Chih-Che Lin, “The GUI User Manual for the NCTUns 6.0 Network Simulator and Emulator”.

[16] Sachin Katti, Hariharan Rahul, Wenjun Hu, Dina Katabi, Muriel Medard, Jon Crowcroft, “XORs in the Air: Practical Wireless Network Coding,”

Proceedings of the 2006 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications, September 2006.

[17] Thomas M. Cover, fellow, IEEE, and Abbas A. El Gamal, member, IEEE, “Capacity Theorems for the Relay Channel,” IEEE Transactions on Information Theory, Vol. 25, Issue 5, September 1979.

[18] IEEE Std 802.11-1999, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications.

[19] IEEE Std 802.11a, Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications: Higher-speed Physical Layer Extension in the 5 GHz Band, Supplement to Part 11.

[20] IEEE Std 802.11k/D1.0, Part 11, Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications, Amendment 7: Radio Resource Measurement.

