DISH NETWORKS: PROTOCOLS, STRATEGIES, ANALYSIS, AND IMPLEMENTATION

LUO TIE

NATIONAL UNIVERSITY OF SINGAPORE

2008

DISH NETWORKS: PROTOCOLS, STRATEGIES, ANALYSIS, AND IMPLEMENTATION

LUO TIE
(*B. Eng (Hons), BUPT*)

A THESIS SUBMITTED
FOR THE DEGREE OF DOCTOR OF PHILOSOPHY DEPARTMENT
OF ELECTRICAL & COMPUTER ENGINEERING
NATIONAL UNIVERSITY OF SINGAPORE
2008

Acknowledgments

I would like to begin by thanking Dr. Vikram Srinivasan, my main supervisor, for the past four years. He has been a wonderful advisor and mentor simultaneously, providing me with a rich source of ideas and endless spiritual support. His insightful thoughts and enthusiasm for research amazes and inspires me. I thank him for the countless hours he has spent on reading and critiquing my papers and talks; he always provides very timely feedback to my research proposals and paper drafts. His assistance during my time at NUS has been invaluable — my life has been enriched professionally, intellectually, and personally.

I would also like to thank Dr. Mehul Motani, my co-supervisor. His guidance has been a precious wealth to me and has greatly enhanced and strengthened the work. His sharp vision for the future is amazing and it is a wonderful learning experience to work with him throughout my research program. He has provided me priceless and inspiring instruction which greatly helps me mature in profession. The four-year experience of being with him is really memorable.

While at CNDS lab, I had the privilege of interacting with bright and talented people. They have taught me much more than I expected, and their advice, feedback, and friendship have made my PhD experience both more educational and fun. I am grateful to them: Ai Xin, Anirudh, Buddhika, Ghasem, Han Mingding, Hu Zhengqing, Jia Jinxi, Ng Hai-Heng, Rob Hoes, Wang Wei, Yap Korkiong, Yeow Waileong, and Zhao Qun.

Finally, I am indebted to my parents for everything that they have given to me. They have been my constant support that always uplifts my spirit when I was frustrated, stressed, and depressed. They have been suffering from serious health problems but, even though I had not been able to stay with them for over a year due to my research work, they always tell me to focus on my work and not worry about them. They are my eternal source of love and it would be impossible for me to complete my work without their support.

Table of Contents

Summary

Cooperative communication has been intensively studied in various contexts for decades, where it has usually been exploited as a data relaying mechanism. However, the wireless channel allows for much richer interaction among nodes. In this thesis, we introduce *Distributed Information SHaring* (DISH) as a new approach for wireless network protocol design. The basic idea of DISH is to allow neighboring nodes to cooperatively share control information with nodes who need it to aid in their decisions making. DISH is a distributed flavor of *control-plane cooperation* which augments the conventional understanding of cooperation at the data plane.

DISH can be applied to many contexts and embodies new paradigms of protocol design. In this thesis, we apply DISH to multi-channel networks to design new MAC protocols. First, we propose a DISH-based protocol called CAM-MAC. Besides its cooperative nature which distinguishes it from other protocols, an important advantage is that it uses a single transceiver and is fully asynchronous. Our extensive simulations show that CAM-MAC boosts throughput significantly and outperforms three recent and representative multi-channel protocols, MMAC, SSCH, and AMCP. Second, we present a theoretical treatment of DISH by evaluating the availability of cooperation, captured by a new metric p_{co}, in a multi-hop network. Our analysis accurately characterizes the behavior of p_{co} with respect to underlying network parameters. Then we investigate the correlation between p_{co} and network performance and obtain several meaningful findings. In particular, we find a near-linear relationship between p_{co} and typical network performance indicators such as throughput and delay. We also demonstrate how to apply the analytical results to a practical channel bandwidth allocation problem. Third, we explore energy issues in DISH and propose two energy-efficient strategies, *in-situ energy conscious cooperation* and *altruistic cooperation*. Our comparative study shows that altruistic cooperation is extremely simple (with zero runtime overhead and no protocol redesign) yet very effective. In comparison to several other protocols, it achieves the highest throughput and the lowest power consumption simultaneously and more than doubles cost efficiency. In-situ energy conscious cooperation, on the other hand, is an appropriate choice only under certain conditions. Fourth, we present our hardware implementation of CAM-MAC (and its several flavors) and the altruistic cooperation strategy. To the best of our knowledge, these prototypes are the first full implementation of asynchronous multi-channel MAC protocols for ad hoc networks. The experimental data confirm the validity of CAM-MAC, altruistic cooperation, and the idea of DISH.

Based on our study, we contend that DISH is a viable idea and merits due consideration in future cooperative communication networks.

List of Tables

List of Figures

List of Symbols

b	the transmission time of a control message
b_0	$b_0 = e_0/c_0$
c_0	the unit cost of a node
$\overrightarrow{D}$	the vector of Euclidean distances for all flows (source-destination pairs)
e_0	the initial energy of a node
$\overrightarrow{F}$	the vector of end-to-end throughput for all flows (source-destination pairs)
G_{max}	the maximum system gain (the ratio between the throughput of a multi-channel system and that of a single-channel system)
L	the size of a data packet
l_c	the size of a control packet
m	the number of available channels
m_{bot}	the maximum number of data channels that can be simultaneously used
n	node density. In a multi-hop network, it is the average number of nodes per R^2. In a single-hop network, it is the total number of nodes.
N_a	the total number of altruists
N_p	the total number of peers
P_a^{max}	the maximum power consumption of all the altruists in the network
P_p^{max}	the maximum power consumption of all the peers in the network
p_{co}	the probability for two arbitrary nodes that create an MCC problem to obtain cooperation, i.e., there is at least one cooperative node with respect to these two nodes
p_{cov}	cooperation coverage
p_{co}^{xy}	the probability that at least one cooperative node with respect to x and y exists
$p_{co}^{xy}(v)$	the probability that node v is a cooperative node with respect to x and y
p_{ctrl}	the probability that a node is on the control channel at an arbitrary point in time
p_{oh}	the probability that an arbitrary node successfully overhears a control message
p_{succ}	the probability that a control channel handshake (initiated by an `McRTS`) is successful
R	radio transmission range
S, S_{co}	the aggregate throughput, without and with cooperation
S_{max}	the maximum aggregate throughput
T_{cca}	duration of a CCA period. Let $T_{cca}^{min} = \min(T_{cca})$
T_{ctrl}	duration of a successful control channel handshake
T_{data}, T_d	duration of a successful data channel handshake
$T_{payload}$	payload length in a data packet
T_{sw}	channel switching delay
W	channel bandwidth
w_c	control channel bandwidth
w_d	data channel bandwidth
$\mathcal{A}_m^*$	the optimal bandwidth allocation scheme, $\mathcal{A}_m^* = (m, \sigma^*)$, which achieves the maximum p_{co} for a given m

$\mathcal{C}_v(t)$	node v is on the control channel at time t
$\mathcal{O}(v \leftarrow i)$	node v successfully overhears node i's control message, given that i sends the message
$\mathcal{S}_v(t_1, t_2)$	node v is silent (not transmitting) on the control channel during interval $[t_1, t_2]$
$\mathcal{I}_v(t_1, t_2)$	node v does not introduce interference to the control channel during interval $[t_1, t_2]$, i.e., it is on a data channel or is silent on the control channel
$\Omega_u(t_1, t_2)$	node u, which is on a data channel at t_1, switches to the control channel in $[t_1, t_2]$
$\mathcal{N}_i, \mathcal{N}_{ij}, \mathcal{N}_{v\setminus i}$	$\mathcal{N}_i$ is the set of node i's neighbors, $AN_{ij} = \mathcal{N}_i \cap \mathcal{N}_j$, $\mathcal{N}_{v\setminus i} = \mathcal{N}_v \setminus \mathcal{N}_i \setminus \{i\}$ (v's but not i's neighbors)
$K_{ij}, K_{v\setminus i}$	$K_{ij} = \lvert\mathcal{N}_{ij}\rvert$, $K_{v\setminus i} = \lvert\mathcal{N}_{v\setminus i}\rvert$
s_i	the time when node i starts to send a control message
$\lambda_c, \lambda_{rts}, \lambda_{cts}$	the average rates of a node sending control messages, `McRTS`, and `McCTS`, respectively, *when it is on the control channel*. Clearly, $\lambda_c = \lambda_{rts} + \lambda_{cts}$
λ	the average data packet arrival rate at each node, including retransmissions
η_{max}	the maximum utilization of a data channel
ρ_{alt}	the density of altruists
ρ_{peer}	the density of peers
ξ, ξ_{co}	data channel collision rate, without and with cooperation
δ, δ_{co}	packet delay, without and with cooperation
η_ξ	$\eta_\xi = \xi_{co}/\xi$
η_δ	$\eta_\delta = \delta_{co}/\delta$
η_S	$\eta_S = S/S_{co}$
σ	$\sigma = w_c/w_d$

List of Abbreviations

BMP	bit-meter-price ratio
CCA	clear channel assessment
COTS	commercial off-the-shelf
CSMA	carrier sensing multiple access
CSMA/CA	carrier sensing multiple access with collision avoidance
DCF	distributed coordination function (IEEE 802.11)
DISH	distributed information sharing
LAN	local area network
MAC	medium access control
MCC	multi-channel coordination (problem)
RTS/CTS	ready to send / clear to send
WiFi	wireless fidelity
WSN	wireless sensor network

Chapter 1

Introduction

When Einstein was asked to explain what is wireless, in the days when wireless was very new and mysterious, he is quoted as saying:

"The wireless telegraph is not difficult to understand. The ordinary telegraph is like a very long cat. You pull the tail in New York, and it meows in Los Angeles. The wireless is exactly the same, only without the cat." [1]

This quote implies that a wireless communication system is the same as a wired communication system, except that the wires have been eliminated. This notion was widespread among people designing and building communication networks, especially in the earlier days, and seems quite firmly entrenched to this day.

However, Einstein's statement should not be overrated in wireless networking. There are fundamental differences between the characteristics of wired and wireless systems. The wired environment is basically fixed and unchanging. A wireless environment, on the other hand, is changing with time, usually randomly, is highly unpredictable, and not fully controllable. In particular, radio waves are being *broadcast* when they are propagated in the air, and as a consequence, signal power is spread everywhere and creates significant interference to other radios.

On the flip side of the coin, the broadcasting nature of wireless communication also creates a vast opportunity of *cooperative communication* — every single transmission automatically disseminates its carried information around, and hence nodes in the neighborhood can readily overhear and may response accordingly. This enables a clique, or an entire network, of nodes to collaborate with each other to perform a joint task.

This idea of cooperation has spurred numerous studies since early 1970s from an information-theoretic perspective (e.g., [2–5]) or a protocol-design perspective (e.g., [6–9]). To date, cooperative communication has been intensively studied in various contexts. In all those contexts, to the best of our knowledge, it has been used as a *data relaying* mechanism where intermediate nodes help relay data from source nodes to destination nodes. In fact, the wireless channel allows for much richer interaction among nodes. We believe that there is much more we can explore.

In this thesis, we introduce a notion called DISH as an enrichment of cooperation. As cooperation is a too broad concept that bears different meanings in different contexts, e.g., routing, coding, etc., we confine our topic to the medium access control (MAC) layer in this thesis.

1.1 DISH in a Nutshell

In rethinking over other possible ways to explore cooperative diversity, we made the following observation. Although the ultimate goal of almost all communication processes is to deliver data payload from one entity to another (or others), extra information for control purposes is usually exchanged to ensure a communication process to be conducted in a predefined manner. This control information, although playing an auxiliary role and typically in a small amount, is crucial to successful data delivery which is typically in a large amount. This suggests an alternative way of looking at cooperation from the control

	Distributed	Centralized
Control Plane	DISH	centralized info sharing
Data Plane	distributed relay (e.g., prior cooperative MAC)	relay via central server (e.g., cellular, WLAN, piconet)

Figure 1.1: A conceptual dissection of cooperation at the MAC layer.

perspective, rather than directly dealing with data itself, and this new way may have a leveraging effect.

Eventually, we have two key arguments. First, cooperation can be realized either at the *data plane* or the *control plane*. At the data plane, nodes collaborate by forwarding data packets for other nodes (usually source-destination pairs), where control information could also be sent but the purpose is to facilitate the forwarding process. The forwarding schemes include decode and forward, amplify and forward, compress and forward, distributed antenna arrays, parallel relaying, and multi-hop transmission [10, 11]. At the control plane, nodes collaborate by exclusively exchanging control information in order to provide a source of knowledge for other nodes to acquire from.

Second, cooperation can be conducted either in a centralized manner or a distributed manner. In a centralized environment, a unique server arbitrates communication processes and resource allocation, which is also a type of cooperation since the central server works jointly with other nodes to accomplish a common task. Typical examples are cellular networks (as in each cell), wireless LANs, and piconet (which is based on Bluetooth technology [12]). In a distributed environment, due to the lack of a central server, any node may arbitrate or help, as appropriate, other nodes' communication at different points in time. Cooperation in such a distributed manner can take more advantage of the broadcasting feature of wireless channel and improve system performance as well.

The above arguments lead to a conceptual dissection of cooperation as depicted in Figure 1.1. Sitting at the data plane is the conventional cooperation which deals with relaying data for source-destination pairs. This consists of centralized schemes such as GSM systems, WiFi LANs, and piconets, and distributed schemes such as cooperative relaying [6–9]. Sitting at the control plane is the new type of cooperation that we introduce, which deals with control information exclusively to accomplish communication tasks in a different way. This notion of cooperation augments the conventional understanding of cooperation at the data plane.

In this thesis, we specifically study a distributed flavor of control-plane cooperation, which we call *Distributed Information SHaring* (DISH). The basic idea of DISH is to allow neighboring nodes to exchange control information to compensate for their insufficient knowledge about the environment, and thus aid them in making more informed decisions. We expect that this idea may lead to substantial benefits because the vagaries of the wireless channel and the location-dependent nature often prevent individual nodes from acquiring sufficient knowledge about the communication environment. The consequence is that nodes often have to make sub-optimal decisions and thereby clamps down on system performance. This is especially remarkable in a distributed environment. By introducing DISH which collaboratively furnishes another source of knowledge, nodes can stay more informed and perform better.

Another benefit is that schemes based on DISH should be *lightweight* compared to conventional cooperative mechanisms relaying data packets. Meanwhile, the leveraging effect that control information has over data delivery may lead to considerable performance change.

Finally as a note, CSMA and its immediate extensions such as CSMA/CA [13] and self-learning algorithms [14–17] are not categorized as cooperation in this thesis. In these mechanisms, node regulate their own behaviors (e.g., adjust the way they back off) based on the dynamics of the environment in order to minimize adverse effects on nearby nodes. As such, CSMA and its immediate extensions can be viewed as a kind of "implicit" cooperation since, again, cooperation is a too broad concept. Therefore, to be specific in this thesis, we limit the concept of cooperation to "explicit" cooperation only: nodes must take explicit actions, such as sending packets for the sake of others, in performing a joint communication

task.

1.2 Scope and Purpose of Study

DISH is a general approach to wireless network protocol design. In this thesis, we specifically apply DISH to multi-channel ad hoc networks for MAC protocol design. The motivation is explained below.

As mentioned in the beginning, interference is a fundamental bottleneck to the performance of wireless networking. Tremendous research effort has been dedicated to address this issue. Recently, multi-channel communication was proposed as an appealing solution, which remarkably increases spatial reuse by allowing for multiple concurrent transmissions in the same geographic area.

However, commercial off-the-shelf (COTS) multi-channel wireless network cards have a substantive limitation in that they do not support operating (sending/receiving/listening) on multiple frequencies *simultaneously*. This makes a node miss information disseminated over the channels that it is not tuned to, and results in a *multi-channel coordination* (MCC) problem. This problem is the core problem to multi-channel networking, and has two variants. One is the *deaf terminal problem*, where a node initiates communication with another node that is on a different channel. The other is the *channel conflict problem*, where a node selects a channel to use but the channel is already in use by another node. To solve the MCC problem, significant research work has been carried out toward a desired solution [18–30].

Since the problem stems from the hardware limitation which hinders nodes from acquiring sufficient knowledge, we suspect DISH to be an approach worth trying. This motivated us to apply DISH to multi-channel ad hoc networks for MAC protocol design. More precisely, the context is an ad hoc network where each node is equipped with a single half-duplex transceiver that can dynamically switch between a set of orthogonal frequency channels but can only use one at a time.

The main purpose of our work was twofold:

- To prove the concept of DISH: To prove whether DISH is a feasible and even viable idea, we should actually transform the conceptual idea into a tangible form to see how it works and how well it works. Approaches such as analysis, simulation, and implementation could all be taken.

- To propose a practical multi-channel MAC protocol: multi-channel MAC protocols are promising in boosting spatial reuse, but *practicality* is a key. This means that critical deployment factors, such as complexity, performance and cost, should all be taken into account. This would be challenging but worth doing.

1.3 Contributions

We highlight the following achievements in our study:

- We introduce *distributed information sharing* (DISH), which is a distributed flavor of *control-plane cooperation*, as a new approach to wireless network protocol design. This notion of control-plane cooperation augments the conventional understanding of cooperation, which sits at the *data plane* as a data relaying mechanism.

- We design a multi-channel MAC protocol, CAM-MAC (cooperative asynchronous multi-channel MAC), based on the idea of DISH. Unlike many other peer multi-channel MAC protocols, CAM-MAC uses only a single transceiver and is fully asynchronous. Our extensive performance evaluation demonstrates the significant value of DISH and that CAM-MAC outperforms three recent and representative multi-channel MAC protocols, MMAC [25], SSCH [29] and AMCP [31].

- We provide the first theoretical treatment of DISH by evaluating the availability of cooperation via a new metric p_{co}, the probability of obtaining cooperation. The analysis accurately characterizes the behavior of p_{co} with respect to underlying network parameters, which also yields several meaningful findings. Our investigation, then, reveals a near-linear relationship between p_{co} and

network performance including channel collision rate, packet delay and throughput. This may significantly simplify performance analysis for DISH networks, and suggests p_{co} to be an appropriate performance indicator itself.

- For DISH protocols to achieve energy efficiency, we propose two strategies, *altruistic cooperation* and *in-situ energy conscious cooperation*. Through a detailed comparative study, we find that altruistic cooperation achieves high throughput and low energy consumption simultaneously, and more than doubles cost efficiency. Altruistic cooperation is also extremely simple (with zero runtime overhead and no protocol re-design) and can be generally applied to perhaps all cooperative protocols. In-situ energy conscious cooperation, on the other hand, is appropriate only under certain conditions. The work suggests that altruistic cooperation is the right strategy for DISH.
- We implement DISH protocols (CAM-MAC and its several flavors) and the altruistic cooperation strategy on COTS hardware. To be best of our knowledge, these prototypes are the first full implementation of asynchronous multi-channel MAC protocols for ad hoc networks. We share lessons learned through the implementation process and provide the experimental data. The results confirm the validity of CAM-MAC, altruistic cooperation, and the idea of DISH.
- By applying our analytical results of the availability of cooperation to a practical channel bandwidth allocation problem, we derive optimal allocation schemes and provide general guidelines on bandwidth allocation in DISH networks. In addition, we propose a metric called bit-meter-price ratio (BMP) to evaluate *cost efficiency* of a general network. This metric allows for a fair comparison across different protocols and different networks, and can be used in other studies.

1.4 Dissertation Structure

The rest of this thesis is organized as follows. Chapter 2 gives a background for multi-channel MAC protocol design and reviews related work. Chapter 3 proposes our multi-channel MAC protocol, CAM-MAC, based on the idea of DISH. We elaborate the design of CAM-MAC with specific caveats, and evaluate its performance via extensive simulations. Following that, Chapter 4 develops a theoretical treatment of DISH, by viewing cooperation as a network resource and evaluating the *availability of cooperation*. Apart from that, we go further by investigating the implications and application of our analytical results in great detail, where we make several important observations. In Chapter 5, we explore energy conservation issues with DISH and propose two energy-efficient strategies. We classify MAC protocols in terms of whether they use cooperation and what cooperative strategy they use, into five categories, and define a sample multi-channel MAC protocol for each category. Based on these five protocols, we carry out a comparative study from both theoretical and practical perspectives. The results show that altruistic cooperation is a simple and effective way to explore DISH. In Chapter 6, we present our hardware implementation on COTS devices, where we discuss the implementation process and share lessons learned, and also provide experimental results. Finally, Chapter 7 concludes this thesis and gives future research directions.

Chapter 2

Background and Related Work

In this chapter, as background knowledge, we describe the MCC problem first. Then we review related work to our study, where the reviewed literature covers the following topics: general multi-channel MAC protocols, MAC performance analysis, energy-efficient multi-channel MAC, sleep-wake scheduling algorithms, and multi-channel MAC testbed.

2.1 Multi-Channel Coordination Problem

We first show a rough idea of the MCC problem using an illustration, and then give its formal definition.

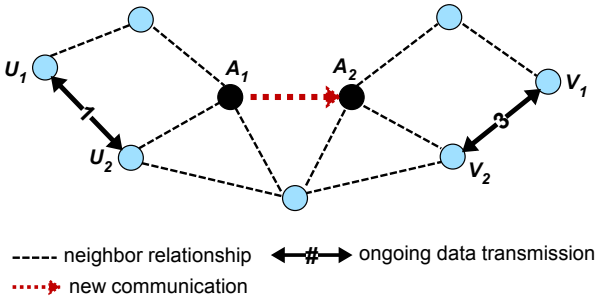


Figure 2.1: An illustration of a multi-channel coordination (MCC) problem.

See Figure 2.1 for a snapshot in a multi-channel ad hoc network. Two node pairs, (U_1, U_2) and (V_1, V_2), are performing data exchanges on channel 1 and 3 respectively, and node A_1 is to initiate a communication with A_2 at this moment. If A_2 is on a channel different from A_1, A_2 will not be able to respond to A_1, which creates a deaf terminal problem. If (A_1, A_2) selects channel 1 or 3 for data exchange, one of the other two pairs, (U_1, U_2) and (V_1, V_2), will be subject to packet collision. This is called a channel conflict problem.

The formal definition is given below.

Definition 2.1 (MCC Problem)**.** A multi-channel coordination (MCC) problem is either a *channel conflict* problem or a *deaf terminal* problem. A channel conflict problem is created when a node, say y, selects a channel to use (transmit or receive packets) but the channel is already in use by a neighboring node, say x. A deaf terminal problem is created when a node, say y, initiates communication to another node, say x, that is however on a different channel. In either case, we say that an MCC problem is created by x and y.

If a protocol employs an one-way data handshake (no acknowledgment packet), a channel conflict problem does not necessarily result in collision. This does not change the above definition though.

2.2 General Multi-Channel MAC Protocols

There are many existing protocols addressing the MCC problem, and can be categorized into multi-radio and single-radio schemes.* The multiple-radio schemes [18–23] use one or more radios at each node to monitor the channel usage when the other radio(s) are engaged in communication. These schemes have to afford the increase of hardware cost, size, and energy consumption. The single-radio schemes typically regulate the node behaviors into well-known time slots [24–26] or employ channel hopping sequences [27–29], such that nodes can have prior knowledge of the channel usage by others. These schemes require time synchronization, which adds considerable complexity [32] and degrades network scalability [33].

2.2.1 Single-Radio Solutions

Control-data window schemes

In these schemes, the time line is divided into successive control and data windows; node negotiate channels in each control window and then exchange data in the subsequent data window on multiple channels concurrently. Since all nodes negotiate channels in the same control window and also on the same (default) channel, all nodes are well informed and thus MCC problems are avoided. However, a common problem is channel under-utilization: all channels other than the default channel remain idle during each control window. Individually, each protocol in this category has its own limitations as reviewed below.

MMAC [25] assumes the IEEE 802.11 power saving mode and divides time into beacon intervals. Each beacon interval is 100ms and consists of a 20ms ATIM window and an 80ms data window. Besides the above under-utilization problem, it has the following drawbacks. (1) It requires time synchronization over the entire network, which is a notoriously hard task involving considerable overhead and complexity [32], and compromises scalability [33]. According to [34], even if clock synchronization is achieved, two partitioned networks may not be able to discover each other if their schedules are not synchronized. (2) Its control window size is fixed, and thus is inflexible to node densities: at low density, the control window has long idle time; at high density, the control window cannot accommodate all negotiations and some nodes have to wait for the next control window. (3) Its data window size is also fixed, and hence has to be set as per the *maximum* data packet size, again leading to inefficiency.

MAP [24] varies the data window size and avoids problem 3, but it still suffers from problem 1 and 2. LCM MAC [35] makes a noticeable improvement by allowing each neighborhood to negotiate the boundaries of control-data windows, by which time synchronization is avoided. However, the negotiated window size can hardly fit for all nodes in the neighborhood, somehow like problem 2 above. In addition, this window negotiation, plus a fine-tuning mechanism it uses, considerably complicates the protocol.

Our proposed protocol, CAM-MAC, is a simple protocol that avoids all the above problems.

Channel hopping schemes

In SSCH [29], each node hops among all channels according to a pseudo-random sequence such that neighboring nodes will have channels overlap in time periodically. Since a transmitter can only communicate to a receiver when they hop onto the same channel, large delay can be incurred; in the worse case, a transmitter has to wait for $m_0 T_{sw} + (m_0 - 1)T_0$ before communicating to its intended receiver, where m_0 is the number of channels, T_{sw} is channel switching delay, and T_0 is a node's sojourn time on each channel (including possible data exchanges). In addition, frequent hopping makes the protocol performance very sensitive to channel switching delay which varies from tens to thousands of microseconds. In CHMA (channel hopping multiple access) [27] and CHAT (CHMA with packet Trains) [28], the entire network adopts a *common* hopping sequence. This does not really solve the large delay problem because each node sojourns on each channel for different periods of time. Moreover, all the channel hopping schemes require clock synchronization.

*In this thesis, we use "radio" and "transceiver" interchangeably.

In CAM-MAC, each node stays on a common channel and only switches channel when a data exchange is established successfully or finished. This avoids switching channel too often and, due to the common channel, does not incur large delay. Besides, again, no clock synchronization is required.

Routing and channel assignment schemes

CBCA [30] combines channel assignment with routing. It proposes to assign each set of intersected flows, called a component, with a single channel, in order to avoid channel switching delay, node synchronization[†] and scheduling overhead at flow-intersecting nodes. CAM-MAC uses a control channel, which automatically avoids the problem of node synchronization and scheduling overhead. Regarding channel switching delay, its effect on network performance is much less than MCC problems: channel switching delay is 40–150μs [36], but a channel conflict can collide *at least* one data packet whose delivery several and even tens of milliseconds.[‡]

In fact, CBCA shifts complexity from the MAC layer to the routing layer. Also, compared to packet, link, and flow based channel assignments, it has the *least* flexibility in exploiting multi-channel diversity: each component, which spans all intersecting flows, can only use one channel. As a consequence, any two nodes in a common component cannot transmit simultaneously unless they are at least three or four hops apart (depending on the interference range). In a single-hop network, since all flows are intersected, a multi-channel network degrades to a single-channel network.

Other schemes

Like CAM-MAC, AMCP [31] uses a single transceiver and is also asynchronous. But on the other hand, unlike CAM-MAC, it uses deferral timers instead of DISH to solve channel conflict problem (recall that this is one variant of an MCC problem), which leads to channel under-utilization. In detail, a node in AMCP attempts to always use its previously used channel unless the channel is occupied by other nodes, in which case it waits until an *avail timer* expires and then randomly selects a free channel. To avoid collision, the avail timer is set to be the duration of a complete data exchange which assumes the maximum data packet size. This conservative approach results in channel under-utilization.

On the contrary, CAM-MAC takes an aggressive approach; a transmitter always attempts to initiate communication unless it is sure that all channels are not available or the receiver is busy. Cooperation would come into play if the attempt creates an MCC problem.

WiFlex [37] is similar to AMCP but it extends AMCP's RTS/CTS exchange to a longer *Review* phase in order to achieve fairness and priority access. We do not compare with WiFlex because it (1) assumes an OFDM-like PHY which allows for transmitting and receiving on multiple channels *simultaneously* and (2) allows for frame aggregation which assembles multiple data packets into one whose size can exceed the 802.11 limit (2312 bytes).

Overall for all the protocols described in Section 2.2.1, not all of them address deaf terminal problem, whereas CAM-MAC solves both deaf terminal and channel conflict problems.

2.2.2 Multi-Radio Solutions

Using multiple radios can easily solve MCC problems by dedicating one radio for monitoring channel usage information. DCA [18] uses two transceivers, one for exchanging control packets and the other for exchanging data packets. The control packets are used to allocate the channels on the data transceiver on demand. The protocol was then extended in [38] to integrate a power control function. [19] proposes a multi-channel CSMA protocol with soft channel reservation. It assumes the number of channels to be equal to the number of transceivers per node, so that all channels can be used simultaneously. This is very expensive. Later they extended their work in [20] where channel selection is based on signal strength. [21] is a protocol similar to DCA in that it also dedicates a transceiver for control purposes, but the difference is that channel selection is done at the receiver end based on signal-to-noise ratio. MUP [22] also uses

[†]The "synchronization" stated in [30] does not mean time synchronization but node synchronization.

[‡]For example, transmitting a 1.5KB packet over an 802.11b 2Mb/s channel takes 6ms transmission time plus handshaking and backoff periods.

two transceivers but it allows both transceivers to exchange control messages and data packets. The two transceivers work independently, each operating over the IEEE 802.11. xRDT [35] extends RDT [39], which uses a (possibly different) quiescent channel for each node to receive packets, by adding a busy-tone radio to each node in order to inform the neighborhood of ongoing data reception, in order to avoid collision and deafness. Kyasanur and Vaidya [40] proposed link-layer protocols for routing in multi-radio multi-channel ad hoc networks. Each node is assigned a fixed interface for receiving packets and multiple switchable interfaces for transmitting packets. This is similar to the idea of quiescent channel but uses more radios to simplify overcoming MCC problems.

Obviously, the key drawback of multi-radio protocols is the increase of device size, cost, and potentially energy consumption.

2.3 MAC Performance Analysis

The performance analysis for single-channel non-cooperative MAC protocols have been intensively studied since 1970s. The cornerstone contribution is attributed to Abramson [41] and Kleinrock and Tobagi [42], who analyzed ALOHA and CSMA protocols, respectively. Later, with the standardization of IEEE 802.11 [43] via the emergence of MACA [44] and MACAW [45], a milestone work which analyzes the performance of 802.11 DCF was conducted by Bianchi [46].

Based on the relevancy to this thesis, the below reviews the analysis for multi-channel MAC protocols or cooperative MAC protocols only.

Han et al. [47] studied a multi-channel MAC protocol that adopts ALOHA on the control channel to reserve data channels. They compared the throughput performance in a fixed-total-bandwidth scenario and a fixed-channel-bandwidth scenario (as called therein), in order to see whether a multi-channel scheme is more advantageous than a single-channel scheme. A queueing-theoretic approach was taken to calculate the throughput. The study, however, has some noteworthy limitations. First, only a *single-hop* scenario was considered. Second, each node was assumed to be able to communicate on a control channel and a data channel simultaneously. This essentially requires *two transceivers* per node, and consequently leads to collision-free data channels, which oversimplifies the problem. Third, a unique virtual queue was assumed to store the packets arriving at *all* nodes for the ease of *centralized transmission scheduling*, and the precise status of the queue was assumed to be known to the entire network. This assumption is impractical and eventually results in a throughput upper bound. Fourth, the access to the control channel adopts the ALOHA algorithm, rather than the more practical and sophisticated mechanism of CSMA/CA.

Our performance analysis, as will be described in Chapter 4, was conducted in multi-hop contexts which directly applies to single-hop contexts as a special case. Second, we assume a single radio per node only, which is usually more practical. Third, our model is purely distributed and there is no central coordination. Fourth, our control channel uses CSMA/CA instead of ALOHA.

CoopMAC [9] is a cooperative MAC protocol which exploits *data relaying* as many other protocols do, such as [6–8]. Specifically, it replaces a single low-rate transmission with two high-rate transmissions by using a relay node, in order to achieve higher throughput. The paper provides a protocol analysis which involves computing the probability that a relay node is available. In our work, we define and analyze a new metric called p_{co}, which is essentially the probability that a cooperative node is available (to an MCC problem). These two seemingly similar probabilities, in fact, are very different. The probability defined in [9] is a "static" metric, meaning that it is purely determined by nodes' (static) locations — whether there is a node in a certain region (the intersection radio range of the source and the destination). This can be easily calculated via simple geometric analysis (as the paper assumes uniformly distributed nodes). On the other hand, p_{co} is a "dynamic" metric meaning that, not only by static node locations, it is also determined by (dynamic) node activities, e.g, some certain events must happen at specific points in time. Another main difference between the protocol analysis in [9] and our work is that the former problem context is a *wireless LAN* using *a single channel*, whereas our context is a multi-hop network using multiple channels. Finally and most importantly, [9] and our work deal with different cooperations, one at the data plane and the other at the control plane.

Kyasanur and Vaidya [48] derived the lower and upper bounds on the capacity of static multi-channel

ad hoc networks, where the number of interfaces is smaller than the number of channels. They presented asymptotic results, of which a main one is that a single radio may suffice (not cause capacity loss) for random networks with up to $O(\log n)$ channels. However, as [48] studies the scaling law, it only readily applies to the scenarios with an infinite number of nodes. Also, the region where the number of channels scaling as $O(\log n)$ is not of immediate practical interest (since it is too large). Finally, the capacity-optimal lower bound construction is based on some assumptions that may not be satisfied in practice, such as unconstrained transmission range control, centralized route assignment and scheduling. Kodialam and Nandagopal [49] derived upper and lower bounds on achievable throughput for multi-radio multi-channel *mesh* networks. The analysis is also asymptotic, similar to [48].

Besides, SRMA [50] also provides a performance analysis under a single-hop setting, but this protocol is not a multi-channel MAC protocol in essence, because it uses only one data channel (and two control channels).

At the bottom line, the significance of our performance analysis is the unique problem context: a multi-channel DISH MAC protocol in a multi-hop network.

2.4 Energy-Efficient Multi-Channel MAC Protocols

This is a relatively new topic and a few proposals only appeared recently. In ad hoc networks, PSM-MMAC [51] allows nodes to select awake or doze state based on the estimated number of active links, queue lengths and channel conditions. TMMAC [26] uses the 802.11 ATIM window not only for channel negotiation (like MMAC [25]) but also for *time negotiation*, which enables nodes to sleep in negotiated time slots.

MMSN [52] was proposed for wireless sensor networks (WSNs). Energy saving, however, is not its specific design concern; its energy conservation comes as a natural consequence of using multiple channels (as interference is reduced), and when there are only a few channels, MMSN consumes more energy than single-channel CSMA. Another multi-channel sensor MAC was proposed in [53] and shown to have better energy efficiency than MMSN. However, its results are based on two impractical assumptions: (i) all cluster heads (the paper assumes a cluster-based network) can *directly* communicate with each other, and (ii) there are many sink nodes and hence there is no single-sink bottleneck. These two protocols both require time synchronization. On the other hand, CMAC [54] is asynchronous, but each node needs to be assigned a channel not overlapping with any other node within the 2-hop range. This means that, for example, for a network with node density $10/r^2$, as will be used in our study, 126 channels are needed.

Our work described in Chapter 5 differs from existing work in the following aspects: (i) instead of proposing a protocol, we propose *strategies* generally applicable to a class of protocols, (ii) we focus on *cooperative* protocols, (iii) we do not require multiple radios as in PSM-MMAC and CMAC, nor time synchronization as in TMMAC, MMSN and [53], and (iv) our proposal applies to both single-hop and multi-hop networks, unlike PSM-MMAC which supports WLAN only.

2.5 Sleep-Wake Scheduling

Tseng et al. [34] proposed three wake-up patterns for ad hoc networks. They support multi-hop communication and do not require synchronization. They trade off between power-saving capability and neighbor discovery time in different manners and can be chosen according to application requirements. There are lots of proposals for WSNs, and most of them can be adapted to non-mobile ad hoc networks as sensor devices are more resource-constrained. In S-MAC [55], nodes *negotiate* sleep-wake schedules such that they wake up simultaneously. This is imposed on each neighborhood but nodes on the border of two adjacent neighborhoods will use two schedules to maintain connectivity. Therefore, network-wide synchronization is not required. T-MAC [56] improves S-MAC by shortening the awake period when the channel is idle. In each awake period, a node listens to the channel for a short time and return to sleep immediately if there is no incoming data. B-MAC [57] uses low power listening and long preamble transmission. Nodes have independent schedules, and hence do not need synchronization. To send a data packet, a node transmits a preamble slightly longer than the sleep period of the receiver, in order

for the receiver to detect the transmission. X-MAC [58] improves B-MAC by embedding receiver address information into the preamble and strobing the preamble, and hence non-target receivers can return to sleep earlier.

In our work described in Chapter 5, we assume an ideal sleep-wake scheduling, in which a node sleeps whenever idle and can automatically wake up upon a communication request (from a transmitter). This avoids coupling the results to a specific real algorithm, and is valid since this study is a comparative study and the same scheduling will be applied to all the power-saving protocols under comparison.

2.6 Multi-Channel MAC Testbed

There are a few hardware implementations of multi-channel MAC protocols. Chereddi et al. [59] reported a testbed of a hybrid multi-channel protocol proposed in [40] based on a channel abstraction module. The protocol requires two interfaces at each node, one tuned to a "fixed" channel for receiving packets and the other can switch channels for transmitting. The protocol was implemented on Linux with Atheros chipset, and experiments were conducted on 4 nodes in a single-hop network. McMAC [60] uses only one radio, and a *simplified version* was implemented on Telos [61] as a proof of concept. There are no experimental data reported on typical performance metrics such as throughput, delay, or energy consumption; the only shown performance is how long it takes to synchronize sender-receiver pairs onto common channels. Y-MAC [62] is another single-radio multi-channel MAC but is proposed for WSNs. It is TDMA based and specifically deals with busty traffic in dense WSNs. It classifies every time slot as a send or a receive slot, and divides each slot into a contention window and a send/receive window. The protocol was implemented in RETOS operating system [63] on TmoteSky motes [64], and the experiments indicated a low duty cycle and delivery latency. However, throughput was not measured.

All the above protocols require time synchronization ([40] needs loose synchronization). Recently, So et al. [32] showed that synchronization in multi-channel networks is difficult and incurs significant overhead. They also implemented a multi-channel time-synchronizing protocol which periodically exchanges beacon packets. But because data handshakes are not implemented (see Section 7.1 therein), the work does not measure typical performance metrics.

Most recently, there appeared two implementations of *asynchronous* multi-channel MAC protocols for WSNs. One is TMCP [65], designed for *data collection* applications (the traffic considered was many-to-one CBR streams) and for networks with only a small number of channels. A network is partitioned into multiple subtrees and each subtree is allocated different channels. The protocol was implemented on MicaZ motes and the performance was evaluated in terms of packet delivery ratio, which reflects throughput in some sense but has left out energy consumption. Le et al. [66] also built a multi-channel MAC testbed using MicaZ motes. They evaluated performance in terms of the number of received messages, and did not consider energy issues. Similarly, the protocol was also designed for data collection and aggregation applications in WSNs, and the random traffic pattern which is typical in ad hoc networks will lead to poor performance (see Section 6 therein).

Our hardware implementation, as will be addressed in Chapter 6, differs from prior work in the following aspects: (i) the protocol is designed for ad hoc network, using a single radio and fully asynchronous, (ii) it can evaluate typical performance metrics such as throughput and energy consumption, and (iii) it is a full (as opposed to simplified) implementation of the original protocol.

Chapter 3

A DISH Protocol: CAM-MAC

3.1 Introduction

In a multi-channel network, DISH allows neighboring nodes who identify an MCC problem to notify transmitter-receiver pairs of *channel conflicts* and *deaf terminals* to prevent collisions and retransmissions. Figure 3.1a gives an illustration based on the example used in Figure 2.1. Two node pairs, (U_1, U_2) and (V_1, V_2), are performing data exchanges on channel 1 and 3 respectively, and node A_1 is to initiate a communication with A_2 at this moment. If A_2 is on a channel different from A_1, a deaf terminal problem is created. If (A_1, A_2) selects channel 1 or 3 for data exchange, a channel conflict problem is created. In either case, the neighboring nodes C, D, or E may have relevant channel usage information (see Figure 3.1b) and could share with (A_1, A_2) to solve the MCC problem.

Based on the idea of DISH, we design a single-radio cooperative asynchronous multi-channel MAC protocol called CAM-MAC for ad hoc networks. We evaluate CAM-MAC from both theoretical and practical perspectives, where we choose a specific set of protocol parameters for illustration and evaluation purposes: (i) we show that its throughput upper bound is 91% of the system bandwidth and its average throughput approaches this upper bound with a mere gap of 4%, (ii) we show that it can saturate 15 channels at maximum and 14.2 channels on average, which indicates that, although CAM-MAC uses a control channel, it does not realistically suffer from control channel bottleneck, (iii) to investigate the *value of cooperation*,* we compare CAM-MAC with its non-cooperative version called UNCOOP, and observe a throughput ratio of 2.81 and 1.70 between them in single-hop and multi-hop networks, respectively, and (iv) we compare CAM-MAC with three recent and representative multi-channel MAC protocols, MMAC [25], SSCH [29], and AMCP [31], and the results show that CAM-MAC substantially outperforms all of them.

In the rest of this chapter, we first identify new challenges to designing a cooperative protocol in Section 3.2, and then we present the protocol details in Section 3.3 together with mathematical analysis. Following that, Section 3.4 provides simulation results in various scenarios. We discuss relevant issues in Section 3.5 and, finally, give concluding remarks in Section 3.6.

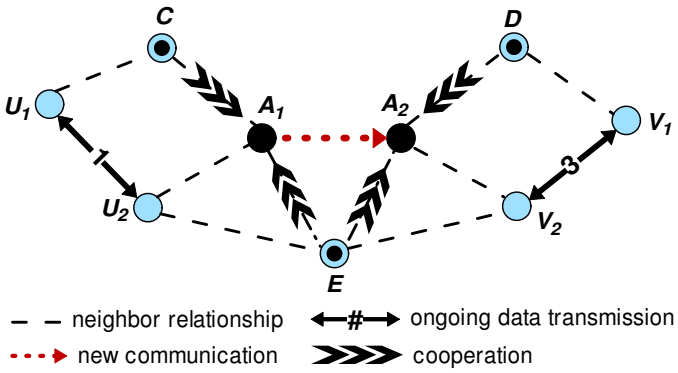
3.2 Caveats to Cooperative Protocol Design

We identify three major issues in designing a cooperative MAC protocol, which will adversely affect protocol performance unless properly addressed.

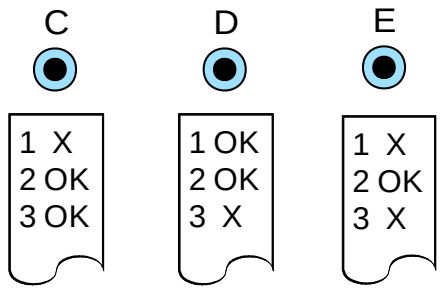
3.2.1 Control Channel Bottleneck

Using a dedicated control channel can facilitate the design of a cooperative protocol, because a control channel provides a unique rendezvous for nodes to disseminate, gather and share information. However,

*As long as there is no ambiguity, we use "cooperation" and "DISH" interchangeably in the sequel of this thesis.



(a) A scenario where an MCC problem is created (based on Figure 2.1).



(b) By consolidating the knowledge at nodes C and D, or acquiring knowledge from node E, it shows that the conflict-free channel is channel 2.

Figure 3.1: An illustration where DISH could help solve an MCC problem.

this design scheme may come with a drawback: when a large number of channels and nodes are present, the single control channel which is used to set up communications can be highly congested and become a performance bottleneck. In this section, we define a metric to analyze this bottleneck problem, and derive a condition by which this problem can be avoided.

Without loss of generality, suppose a complete communication process comprises a *control channel handshake* preceded by a random CCA (clear channel assessment) period, and a immediately followed *data channel handshake*. We use the following notation:

- T_{ctrl}: duration of a successful control channel handshake.
- T_{data}: duration of a successful data channel handshake.
- T_{cca}: duration of a CCA period. Let $T_{cca}^{min} = \min(T_{cca})$.
- T_{sw}: channel switching delay.

Consider a network where all nodes are within communication range of each other. We define a metric, m_{bot}, to be the maximum number of data channels that can be simultaneously used for a given protocol (with the above parameters). When a bottleneck problem happens (Figure 3.2), m_{bot} data channels are simultaneously in use, and there are still free data channels and backlogged nodes on the control channel. However, no more than m_{bot} data channels can be used, because at the time t when a subsequent $((m_{bot}+1)$th) data channel usage happens, at least one data channel will become free (indicated by <1> in Figure 3.2). Therefore, m_{bot} reflects the "capacity" for a multi-channel system.

By noticing in Figure 3.2 that T_{data} is bounded by the duration of m_{bot} successive control channel handshakes, each of which lasts a period of at least $T_{cca}^{min} + T_{ctrl}$, m_{bot} is given by

$$m_{bot} = \left\lceil \frac{T_{data}}{T_{cca}^{min} + T_{ctrl}} \right\rceil. \tag{3.1}$$

Note that T_{sw} does not affect m_{bot} (a data channel is actually free during T_{sw} and can be used by other nodes).

Suppose the objective is to design a protocol capable of saturating $m_{bot}^{\star}$ data channels, then $m_{bot} \geq m_{bot}^{\star}$ must be satisfied, which is equivalent to

$$\frac{T_{data}}{T_{cca}^{min} + T_{ctrl}} > m_{bot}^{\star} - 1.$$

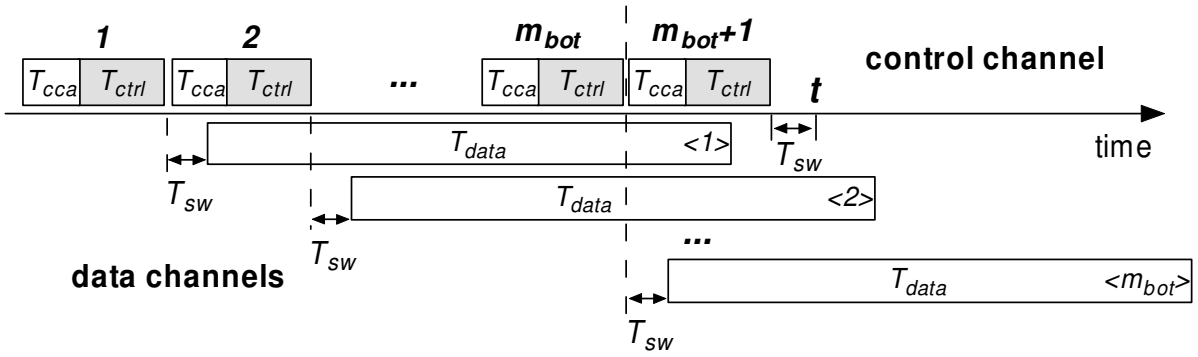


Figure 3.2: Illustration of control channel bottleneck: no more than m_{bot} data channels can be simultaneously in use.

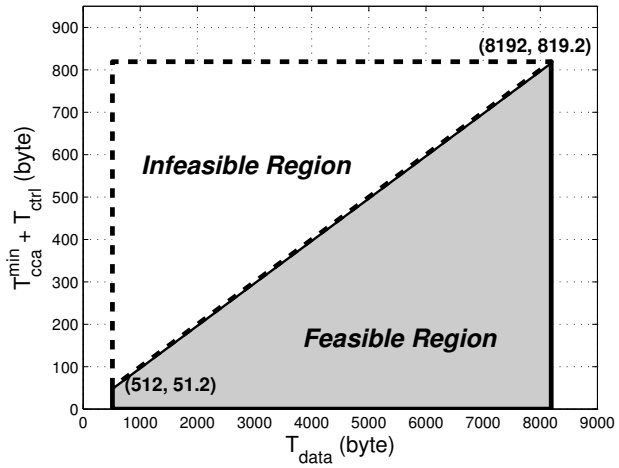


Figure 3.3: The feasible region for choosing design variables for a multi-channel MAC protocol based on IEEE 802.11a. We use byte as the unit of duration (a duration τ is converted into bytes via $\tau C/8$, where $C = 54$Mb/s is channel capacity), and suppose $T_{data} \in [512, 8192]$ bytes. The shaded area gives the feasible values of $T_{cca}^{min} + T_{ctrl}$ to saturate all the 12 channels.

Note that the equality sign can be removed because the r.h.s is an integer. The above resolves to

$$T_{cca}^{min} + T_{ctrl} < \frac{T_{data}}{m_{bot}^{\star} - 1}. \tag{3.2}$$

This is the condition for the design of a multi-channel protocol to avoid control channel bottleneck. Note that T_{cca}^{min} and T_{ctrl} are variables subject to design while T_{data} and $m_{bot}^{\star}$ are given parameters (although T_{data} involves variables such as the size of ACK, it is dominated by payload size which is typically a given parameter).

As an example, suppose we are designing a multi-channel protocol based on IEEE 802.11a, and we wish to saturate all the twelve non-overlapping channels that 802.11a supports. This leads to $m_{bot}^{\star} = 11$ as there is a control channel. Therefore, we need to satisfy $T_{cca}^{min} + T_{ctrl} < T_{data}/10$ according to (3.2), which determines a feasible region for choosing protocol design variables, as plotted in Figure 3.3.

The condition given by (3.2) is necessary and not sufficient, but a protocol satisfying this condition can practically avoid the bottleneck problem with high probability. This is based on our observations in simulations, whose details will be provided in Section 3.3.2 where we revisit this issue. On the other hand, we point out that the bottleneck problem is not necessarily catastrophic even if a protocol insignificantly

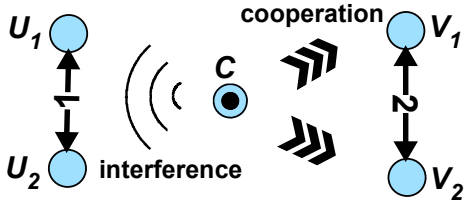


Figure 3.4: Cooperation interference. (U_1, U_2) and (V_1, V_2) are not within interference range of each other, but if (V_1, V_2) creates an MCC problem and node C sends a cooperative message, it may interfere with (U_1, U_2) setting up communication.

violate the condition. This is because the control channel bottleneck problem requires at least $m_{bot} + 1$ transmit-receive pairs in a single-broadcast region *and* that each pair carries heavy traffic, which is not often the case.

3.2.2 Cooperation Coordination

An MCC problem can be identified by multiple neighboring nodes and hence their simultaneous response of sending cooperative messages will result in collision. This creates an issue of *cooperation coordination*. One solution is to make neighbors sequentially respond via a priority-based or slot-based mechanism, thereby ensuring all cooperative messages to be transmitted without collision. However, this is very inefficient because (i) there can be many wasted (idle) intervals because not all neighbors may identify the problem, and (ii) cooperative messages pertaining to the same MCC problem carry redundant information and hence receiving all of them is not necessary. Another solution is to let each neighbor send such messages probabilistically, in order to reduce the chance of collision. However, an optimal probability (optimal in the sense of minimizing the chance of collision) is hard to determine, and such a scheme can result in no response which essentially removes cooperation. Therefore, a simple yet effective coordination mechanism is needed.

3.2.3 Cooperation Interference

This issue means that the cooperative messages sent by neighbors for a transmit-receive pair can unconsciously cause interference to another (nearby) transmit-receive pair, as illustrated in Figure 3.4, This is a new type of interference created by the introduction of cooperation, and our simulations found that it frequently happens and considerably intensifies channel contention. As such, a mechanism needs to be devised to address this deleterious side-effect.

3.3 Protocol Design and Analysis

Our assumption is that each node is equipped with a single half-duplex transceiver that can dynamically switch between a set of orthogonal frequency channels but can only use one at a time. Such transceivers are widely available in the market.

We do not assume specific channel selection strategies; CAM-MAC runs on top of any such strategy. For quantitative performance evaluation, we will consider two strategies in simulations and experiments: (1) RAND selection, where a node randomly selects one from a list of channels that it deems free based on its knowledge, (2) MRU selection, where a node always selects its *most recently used* (MRU) data channel unless it finds the channel to be occupied by other nodes, in which case RAND selection strategy is used.

We do not assume equal channel bandwidth or channel conditions such as noise levels; these can be taken into account by channel selection strategies (e.g., choose the channel with the highest SNR) which are not in our assumptions.

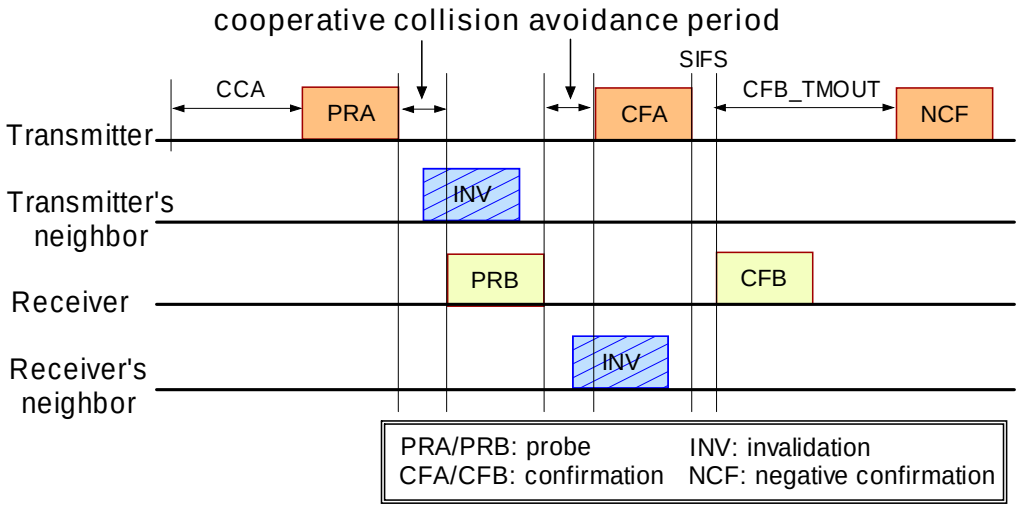


Figure 3.5: The CAM-MAC control channel handshake.

We also do not assume any (regular) radio propagation patterns, nor assume any relationship between communication ranges and interference ranges. Intuitively, none of the nodes is *responsible* for providing cooperation; a node cooperates if it *can* (it is idle and overhears a handshake that creates an MCC problem), and simply does not cooperate otherwise. Actually, there often exists at least one neighboring node that can cooperate, and even in the worse where no one can cooperate, the protocol still proceeds (as a traditional non-cooperative protocol).

3.3.1 Protocol Design

One channel is designated as the control channel and the other channels are designated as data channels. A transmitter and a receiver perform a handshake on the control channel to set up communication and then switches to their chosen data channel to perform a DATA/ACK handshake, after which they switch back to the control channel. The control channel handshake is depicted in Fig. 3.5. A transmitter sends a PRA and its receiver responds with a PRB, like IEEE 802.11 RTS/CTS for channel reservation. Meanwhile, this PRA/PRB also probes the neighborhood inquiring whether an MCC problem is created (in the case of a deaf terminal problem, it is probed by PRA only). Upon the reception of the PRA or PRB, each neighbor performs a check and, if identifying an MCC problem, sends an INV message to invalidate the handshake (the receiver can also send INV after receiving PRA, since it is also one of the transmitter's neighbors). If no INV is sent and the transmitter correctly receives PRB, it sends a CFA to confirm the validity of PRA to all its neighbors (including the receiver), and the receiver will send a CFB to confirm the validity of the PRB if it correctly receives CFA. This marks the end of a control channel handshake. If any INV is sent, the handshake will not proceed and the transmitter will backoff. The NCF is merely used by the transmitter to inform its neighbors that the PRA and CFA are invalid when it fails to receive CFB (the receiver gets INV after sending PRB).

The cooperative collision avoidance period is for mitigating INV collision caused by multiple neighbors sending INVs simultaneously. It is a simple CSMA-based mechanism where each neighbor schedules to send INV at a random point in this period and continues sensing the channel. Once the node that schedules at the earliest time starts to send, others in its vicinity cancel sending their INVs (a receiver can also cancel its PRB).

One concern could be that the four-way handshake plus INV is expensive. In fact, the overhead is not really significant and the pay-off is rewarding, because (i) CFA/CFB are very small packets (shown shortly), (ii) INV terminates a handshake right after PRA or PRB and the rest of the handshake will not occur, and (iii) a mere INV can avoid collision between at least two data packets.

A possible set of frame formats are shown in Figure 3.6. Both PRA+CFA and PRB+CFB carry the

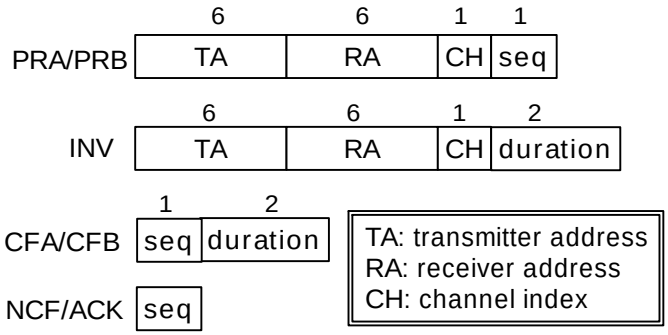


Figure 3.6: A possible set of frame formats.

TA	RA	CH	until
A_1	A_2	1	11:30:52
B_1	B_2	3	11:30:56

Figure 3.7: Channel usage table.

channel usage information of a communication *being established*, and an INV carries the channel usage information of an *established* communication that is to be collided (in the case of a channel conflict problem) or engages the receiver (in the case of a deaf terminal problem). A node may overhear this channel usage information and will cache it in the node's *channel usage table*, shown in Figure 3.7. Note that the `until` column does not imply clock synchronization: it is calculated by adding the `duration` in a received CFA/CFB/INV message to the node's *own* clock. Similarly, when sending INV, a node does a reverse conversion from `until` to `duration` using a substraction. Also note that this table is by caching overheard information while not by sensing data channels. This is because sensing data channels often obtains different channel status at the transmitter and the receiver, and resolving this discrepancy adds protocol complexity. In addition, this may lead to more channel switchings and radio mode (TX/RX/IDLE) changes and thus incurs longer delay.

3.3.2 Caveats Revisited

Now we explain how we address the caveats stated in Section 3.2 in the design of CAM-MAC.

Control channel bottleneck

Recall the metric, m_{bot}, which is the maximum number of data channels that can be simultaneously used. Now we can calculate that $m_{bot} = 14$ ($\lceil 13.92 \rceil$) according to Eq. (3.1) based on the example parameters (Figure 3.6) for CAM-MAC, where $T_{ctrl} = 113.75$byte, $T_{cca}^{min} = 37.25$byte, $T_{payload} = 2048$byte, $T_{data} = 2101.5$byte. This means that the protocol can theoretically saturate up to fifteen channels (including the control channel), which sufficiently exceeds the number supported by current standards (three and twelve channels by IEEE 802.11b/g and 802.11a, respectively).

Since the analysis only provides the maximum value, we also evaluate average performance via simulations. We configure 30 source-destination pairs where source nodes are always backlogged in order to simulate heavy traffic scenarios. We vary the number of data channels from 1 to 30. The results are summarized in Figure 3.8, where CAMMAC-RAND and CAMMAC-MRU are CAM-MAC using RAND and MRU channel selection strategies, respectively. We see that 12.5 and 13.2 data channels (hence 13.5 and 14.2 channels) can be saturated by these two versions of CAM-MAC, respectively. They are close to the theoretical maximum (15 channels) and also exceed what current standards support. Therefore, we can conclude that CAM-MAC does not realistically suffer from the control channel bottleneck.

Cooperation coordination

Recall that this issue is to coordinate multiple neighbors to send cooperative messages as efficiently as possible. We address this using the cooperative collision avoidance period described in Section 3.3.1. It ensures that only one node will send out a cooperative message (INV) in each single-broadcast region, assuming that propagation delay is negligible. We found via simulations that this can reduce 70–85% collisions between cooperative messages.

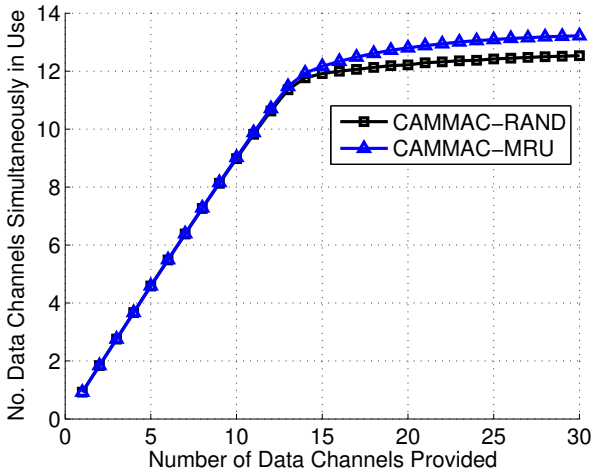


Figure 3.8: The number of data channels that CAM-MAC can saturate.

In case that collisions still happen (due to propagation delay or because not all cooperative nodes can hear each other), it is not a serious problem because CAM-MAC makes such collisions *meaningful* by using *negative feedback* only. That is, since INV always means invalidation, a collision resulting from INV still conveys that the handshake should not proceed. Actually, using negative feedback is a logical design. First, a node expects a *binary feedback* since it selects one channel, instead of selecting a list of channels which needs multi-bit feedback indicating busy/free channel*s*. Second, sending a positive feedback can be misleading because ensuring no MCC problem requires full information while a cooperative node cannot guarantee to have.

Cooperation interference

Recall that this issue is that cooperative messages may cause interference to nearby transmitter-receiver pairs. We address this using *loyal periods*, which borrows the idea of IEEE 802.11 NAV. A node enters a loyal period when it hears a PRA (from a transmitter) or PRB (from a receiver) and does not identify an MCC problem with this handshake H_0. During this loyal period, the node *always keeps silent* (becomes "loyal" to H_0) even if it (i) identifies an MCC problem with *another* control channel handshake H_1 or (ii) receives another PRA addressed to it (itself being an intended receiver). It exits this loyal period after H_0 ends successfully or is invalidated by cooperation. Note that the rules (i) and (ii) are reasonable because, although rule (i) disallows the "loyal" node to cooperate, there most probably exist other nodes that can cooperate (with H_1), and for rule (ii), the loyal node should be able to respond to a *subsequent* PRA (retry) from the same transmitter since the loyal period will expire shortly. This mechanism of loyal period effectively mitigates cooperation interference. We observed via simulations a throughput improvement of 5–30% in various scenarios.

3.3.3 Protocol Analysis

We analyze the throughput upper bound for CAM-MAC in single-hop networks. This serves two purposes: (i) it tells whether the upper bound can approach total channel capacity, and (ii) the upper bound can be used to compare against the actual throughput obtained via simulations to see how close this upper bound can be approached.

In this analysis, for achieving the maximum throughput, MCC problems are eliminated (nodes can always choose conflict-free channels and receivers are always available) and protocol overhead is kept at the minimum level. Part of notation is from Section 3.2.1.

- *Unsaturated Network*

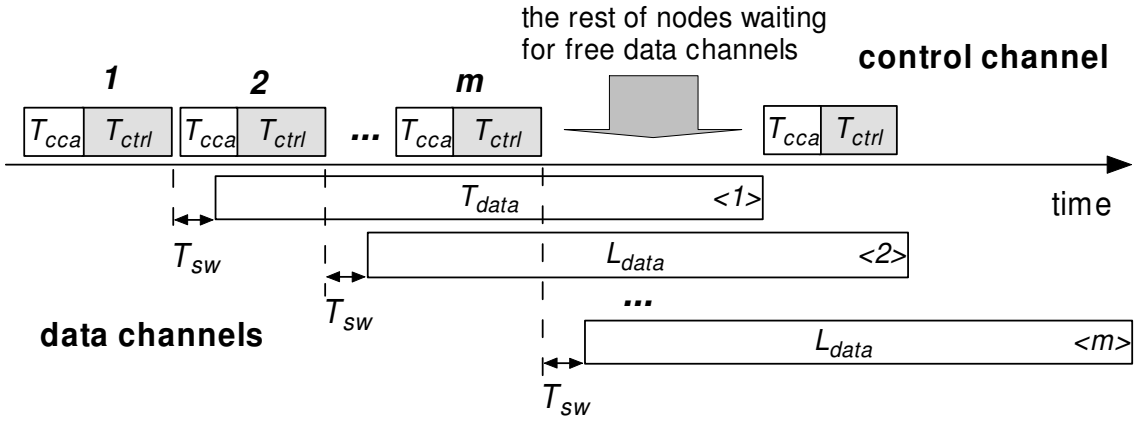


Figure 3.9: Case 1. $m \leq m_{bot}$. The bottleneck is at data channels and thus some nodes have to wait for free data channels. A node starts a control channel handshake only if there is at least one free data channel.

A network is unsaturated (stable) if all input traffic gets through within finite delay. The throughput upper bound is simply given by

$$S_{max} = \sum_i \lambda_i,$$

where λ_i is the offered load of node i. Then by assuming a homogeneous traffic pattern in which $\lambda_i = \lambda$, $\forall i$, the above reduces to

$$S_{max} = n_f \lambda \tag{3.3}$$

where n_f is the number of source-destination pairs (i.e., flows).

• *Saturated Network*

The arrival traffic exceeds the network capacity and the queues of nodes build up infinitely. Denote the number of data channels by m.

1) $m \leq m_{bot}$ (bottleneck at data channels)

If $n_f > m$, all m data channels can be simultaneously in use by m node pairs, and the rest of nodes have to wait on the control channel until some data channel becomes free. Figure 3.9 depicts this scenario, where we can see a period of $T_{cca} + T_{ctrl} + T_{sw} + T_{data}$ will appear periodically. Hence, the *maximum utilization* of a data channel is

$$\eta_{max} = \frac{T_{payload}}{T_{cca}^{min} + T_{ctrl} + T_{sw} + T_{data}}, \tag{3.4}$$

where $T_{payload}$ is the transmission time of the payload in a data packet. So,

$$S_{max} = \eta_{max} m C, \tag{3.5}$$

where C is the capacity of a data channel.

If $n_f \leq m$, S_{max} is achieved by assigning each source-destination pair with a dedicated data channel. In this case, η_{max} remains the same as (3.4), and

$$S_{max} = \eta_{max} n_f C. \tag{3.6}$$

2) $m > m_{bot}$ (bottleneck at the control channel)

If $n_f > m_{bot}$, the control channel becomes bottleneck (the reader can refer back to Figure 3.2). Since data channels are more than what can be saturated ($m > m_{bot}$), the best case is that each control channel

handshake leads to a successful transmission of $T_{payload}$, which translates to a *maximum system gain* of

$$G_{max} = \frac{T_{payload}}{T_{cca}^{min} + T_{ctrl}}, \tag{3.7}$$

and

$$S_{max} = G_{max}C. \tag{3.8}$$

If $n_f \leq m_{bot}$, then S_{max} is achieved again by the dedicated channel assignment as in (3.6), i.e.,

$$S_{max} = \eta_{max} n_f C.$$

Finally, the necessary and sufficient conditions for a network to be unsaturated, are derived from the above:

$$\begin{aligned} &\lambda < \eta_{max}C, && \text{if } n_f \leq \min(m, m_{bot}), \\ &\lambda < \eta_{max}mC/n_f, && \text{if } n_f > m, \text{ and } m \leq m_{bot}, \\ &\lambda < GC/n_f, && \text{if } n_f > m_{bot}, \text{ and } m > m_{bot}. \end{aligned}$$

• *Remark*

First we compute two key quantities, the maximum channel utilization η_{max} and the maximum system gain G_{max}, by substituting the example protocol parameters (in Section 3.3.2, and $T_{sw} = 0$ to compute the maximum) into (3.4) and (3.7). We get

$$\eta_{max} = 91\% \quad \textit{and} \quad G_{max} = 13.56.$$

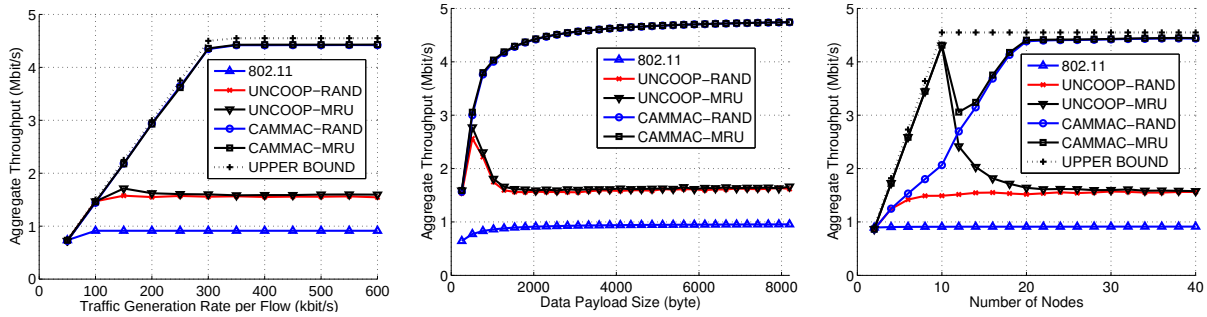
Then we revisit the two purposes mentioned in the beginning of this subsection. (i) Compared against total channel capacity, CAM-MAC can theoretically achieve an utilization of 91%. (ii) In the comparison between the upper bound and simulation results, there are two cases: in the case of control channel bottleneck, the upper bound is $13.56C$ and the throughput of CAM-MAC is $13.2C$ (Figure 3.8), indicating a ratio of 97%; in the case of no control channel bottleneck, the upper bound is $0.91mC$ or $0.91n_fC$ ((3.5) and (3.6)), and the throughput of CAM-MAC achieves 96% of the upper bound, as will be shown in Section 3.4.1.

3.4 Performance Evaluation

We evaluate and compare five protocols, namely IEEE 802.11, CAMMAC-RAND, CAMMAC-MRU, UNCOOP-RAND and UNCOOP-MRU, using a discrete-event simulator which we developed on Fedora Core 5 with a Linux kernel of version 2.6.9. In these five protocols, IEEE 802.11 is used as a baseline in comparison, X-RAND and X-MRU are two versions of protocol X using RAND and MRU channel selection strategies, respectively. The protocol UNCOOP is identical to CAM-MAC except that the cooperation element is removed, i.e., neighboring nodes do not participate in communication by sending INV messages. This comparison will enable us to investigate the value of cooperation.

We use three performance metrics: (i) aggregate (end-to-end) throughput, (ii) data channel conflict rate, defined as the packet collisions on data channels per second over all nodes, and (iii) packet delivery ratio, defined as the number of data packets successfully received by destinations normalized by the number of data packets sent by sources. The packet delivery ratio takes into account deaf terminal problems which can lead to packet drops.

Nodes are uniformly distributed in a plane area. The transmission range is 250m and the interference range is 500m. Capture threshold is 6dB. Each node has a single data packet queue (instead of *per-neighbor queues*, such as used by [29, 67], which bypass head-of-line (HOL) blocking and will yield higher throughput and lower delay). In single-hop scenarios, the terrain is 100m×100m and nodes form disjoint source-destination pairs (i.e., flows). In multi-hop scenarios, the terrain is 1500m×1500m and n nodes



(a) Impact of traffic load. (b) Impact of data payload size. (c) Impact of the number of nodes.

Figure 3.10: Single-hop simulation results.

form n non-disjoint flows randomly (each node is the source of one flow and the destination of another flow). Shortest path routing is used.

There is one control channel and five data channels with bandwidth 1Mbit/s each. PHY and other MAC layer parameters, i.e., PLCP, SIFS, and retry limit, are the same as in IEEE 802.11 [43]. Each source generates data packets with 2KB payload according to a Poisson point process. The cooperative collision avoidance period is $35\mu s$. In the comparison of CAM-MAC and UNCOOP, we ignore channel switching delay as both protocols use the same handshake. However in comparison to the other protocols, namely MMAC, SSCH, and AMCP, we use the parameters that they respectively use, including channel switching delay.

We terminate each simulation when a total of 100,000 data packets are sent over the network, and all results are averaged over 15 randomly generated networks.

3.4.1 Single-hop Networks

Impact of traffic load (Figure 3.10a)

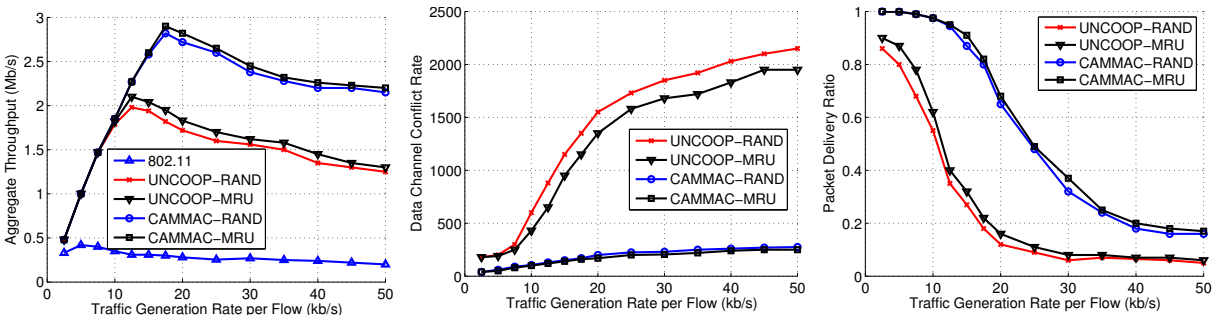
There are 30 nodes (i.e., 15 flows) and each source node generates traffic from 50–600kb/s. We see that the throughput of UNCOOP quickly saturates at 1.6Mb/s while that of CAM-MAC keeps increasing until saturation at 4.5Mb/s. This indicates a remarkable ratio of 2.81. CAM-MAC also approaches the throughput upper bound with a gap of merely 4%. Another important observation is that there is almost no difference between the MRU and the RAND version of either CAM-MAC or UNCOOP. We explain this in discussing the impact of the number of nodes (Figure 3.10c).

Impact of data payload size (Figure 3.10b)

There are 30 nodes while source nodes are always backlogged, and data payload size is varied from 256–8192byte. Interestingly, the throughput of UNCOOP is not monotonic; it first ascends, then descends, and finally levels off. This results from three counteracting factors: (i) a larger payload size offsets protocol overhead more effectively and thus lead toward higher throughput, (ii) a longer data packet is more susceptible to channel conflicts, i.e., it is more likely to be collided, and (iii) longer data packets keep nodes on data channels longer and hence fewer nodes will be possible to initiate new communication on the control channel, which reduces the possibility of channel conflicts. On the other hand, in CAM-MAC, cooperation resolves many channel conflicts and hence weakens factor (ii) and (iii). Therefore, factor (i) stands out and the throughput of CAM-MAC continuously increases.

Impact of the number of nodes (Figure 3.10c)

Unlike the previous two sets of simulations (varying traffic and payload size), this set of simulations shows an evident difference between the RAND and the MRU version: when the number of nodes is not more than 10 (i.e., ≤ 5 flows), the throughput of X-MRU is much higher than that of X-RAND and



(a) Aggregate throughput. (b) Data channel conflicts. (c) Packet delivery ratio.

Figure 3.11: Impact of traffic load in multi-hop networks. Node density is $10/r^2$.

approaches the upper bound. This is because MRU strategy in effect assigns each flow with a *dedicated* data channel (recall that there are five data channels). When there are 12 nodes, MRU channels are frequently occupied by non-owner nodes since the number of transmitter-receiver pairs is one more than the number of data channels. This degrades MRU strategy close to RAND strategy. After that, as the number of nodes increases, MRU channels are more frequently occupied but additional nodes also boost the availability of cooperation. This explains why the throughput of UNCOOP-MRU continues to decrease while that of CAMMAC-MRU gradually recovers. Finally, at saturation, MRU and RAND versions converge, like in the previous two sets of simulations. This is because, at a large number of nodes, MRU channels are deprived very frequently and thus MRU strategy degrades to RAND strategy in effect. On the other hand, we see that only cooperation makes big difference: there is a large gap between CAM-MAC and UNCOOP, and CAM-MAC again achieves a throughput of 2.81 times that of UNCOOP, meanwhile approaching the upper bound with a factor of 96%.

3.4.2 Multi-hop Networks

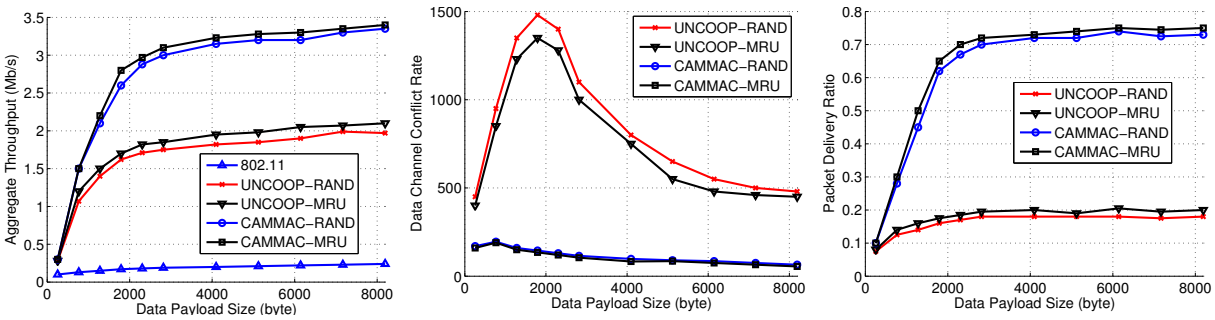
Impact of traffic load (Figure 3.11)

We randomly place 360 nodes in the network, which translates to a node density of $10/r^2$ where r is the transmission range. Traffic generation rate varies from 2.5–50kb/s per flow. The throughput results are shown in Figure 3.11a, where we see that the four multi-channel MAC protocols achieve much higher throughput than 802.11 thanks to the higher spatial utilization. The other large difference is between CAM-MAC and UNCOOP; for example at the traffic generation rate of 50kb/s, the throughput ratio between CAM-MAC and UNCOOP is 1.70. The results of channel conflict rate are summarized in Figure 3.11b, where we see remarkable gap between CAM-MAC and UNCOOP. This similarly happens to the results of packet delivery ratio shown in Figure 3.11c.

A noticeable phenomenon is that the difference between CAM-MAC and UNCOOP in channel conflict rates is often much larger than that in throughput. This is because throughput does not relate to channel conflict rates *linearly*: a cooperative protocol has less data channel usages than an uncooperative protocol, because many conflicting channel usages are prevented by cooperation.

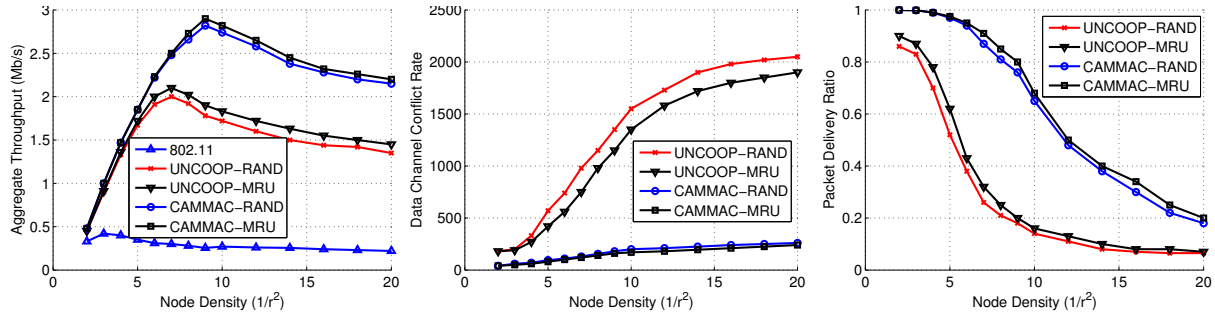
Impact of data payload size (Figure 3.12)

There are 360 nodes and the traffic load is 20kb/s. Data payload size varies from 256–8192byte. Interestingly, for all the protocols, although throughput (Figure 3.12a) and packet delivery ratio (Figure 3.12c) monotonically increase, the channel conflict rate (Figure 3.12b) exhibits a bell shape. Actually, this is accounted for by the two contradicting factors (ii) and (iii) described in the discussion of Figure 3.10b (Section 3.4.1).



(a) Aggregate throughput. (b) Data channel conflicts. (c) Packet delivery ratio.

Figure 3.12: Impact of data payload size in multi-hop networks. Node density is $10/r^2$, and traffic generation rate is 20kb/s.



(a) Aggregate throughput. (b) Data channel conflicts. (c) Packet delivery ratio.

Figure 3.13: Impact of node density in multi-hop networks. Traffic generation rate is 20kb/s.

Impact of node density (Figure 3.13)

We vary node density from 2–$20/r^2$ and fix traffic load at 20kb/s. The curves are similar to Figure 3.11 (impact of traffic load). This is simply because increasing node density and increasing traffic load have the same consequence of (linearly) increasing the *aggregate* traffic load for the network.

Summary: Our single-hop and multi-hop simulations manifestly demonstrate the value of cooperation: cooperation effectively mitigates MCC problems and substantially enhances system performance.

3.4.3 Comparison with MMAC, SSCH, and AMCP

We compare CAM-MAC with three recent multi-channel MAC protocols, MMAC [25], SSCH [29] and AMCP [31].† They all use a single transceiver, but MMAC and SSCH require clock synchronization while AMCP does not. In the comparison, CAM-MAC adopts MRU strategy, and uses the same setup as MMAC, SSCH, and AMCP use respectively, for the purpose of comparing with their reported results under the same settings.

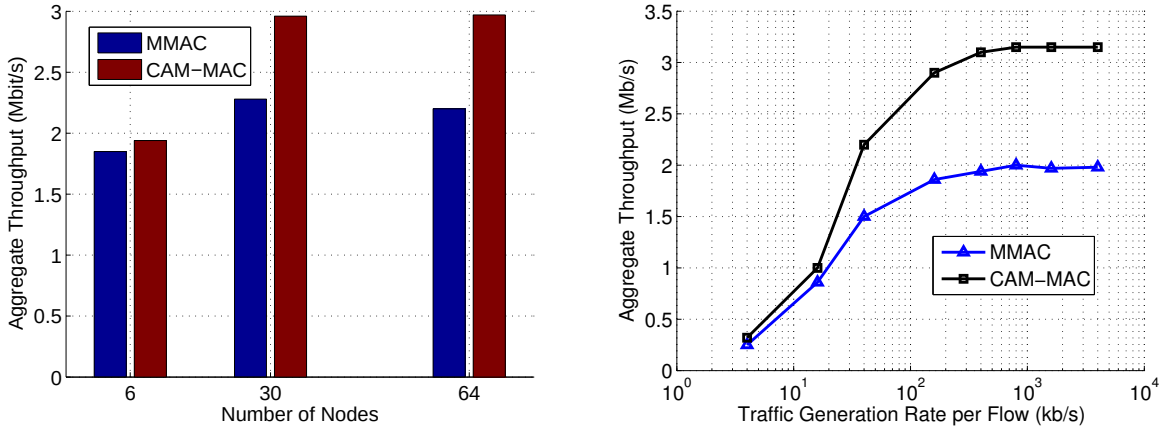
Comparison with MMAC

The parameters are shown in Table 3.1 (CAM-MAC uses only two and three data channels in single- and multi-hop networks, respectively). The first set of simulations are conducted in a wireless LAN, where nodes are configured as disjoint and fully-loaded flows, the same as in MMAC. The results are

†DCA [18] is an early representative protocol and uses multiple transceivers. It has been shown in [25] to be outperformed by MMAC.

Table 3.1: Parameters for Comparison with MMAC

channel	WLAN		multi-hop	
capacity	no. channels	packet size	no. channels	packet size
2Mb/s	3	512byte	4	1024byte



(a) Wireless LAN.

(b) Multi-hop networks.

Figure 3.14: Comparison with MMAC.

presented in Figure 3.14a, where we see that CAM-MAC achieves a throughput of 1.05, 1.30, and 1.35 times that of MMAC at 6, 30 and 64 nodes, respectively. The second set of simulations are conducted in a multi-hop network. Also the same as in MMAC, 100 nodes are randomly placed in a 500m×500m area, and 40 sources and 40 destinations are randomly chosen. The results are shown in Figure 3.14b. We see that, at saturation, CAM-MAC achieves 1.57 times the throughput of MMAC.

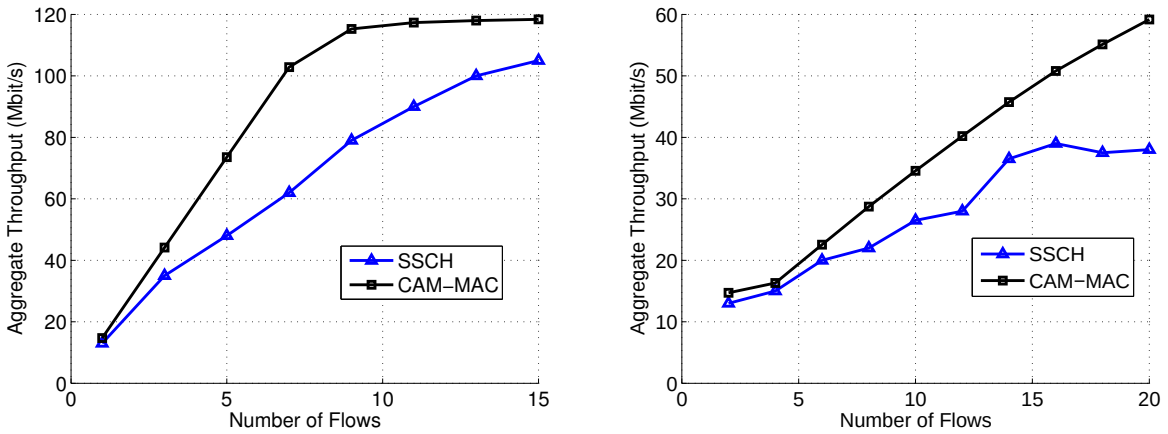
Comparison with SSCH

The parameters are shown in Table 3.2 (CAM-MAC uses 12 data channels). As in SSCH, a disjoint-flow configuration and a non-disjoint-flow configuration are used, where the latter configuration means randomly selecting source-destination pairs (flows) on a packet-by-packet basis. In both configurations, all flows are always backlogged.

Note that the simulation parameters (Table 3.2) are favorable to SSCH but unfavorable to CAM-MAC. In SSCH, since nodes hop among channels following their *respective* sequences, a transmitter often has to wait for its intended receiver to hop onto the same channel before starting communication. Therefore, SSCH prefers short data packets and channel switching delay to reduce this waiting time (the reader may refer back to Section 2.2). On the other hand, CAM-MAC favors long data packets to offset control packet overhead, and its performance does not depend on channel switching delay as significantly as channel hopping schemes such as SSCH. Finally, SSCH uses *per-neighbor queues* which bypass the HOL blocking while CAM-MAC uses only a single queue.

Table 3.2: Parameters for Comparison with SSCH

no. channels	channel capacity	packet size	channel switching delay
13	54 Mb/s	512byte	$80\mu s$



(a) Disjoint flows.

(b) Non-disjoint flows.

Figure 3.15: Comparison with SSCH.

Table 3.3: Parameters for Comparison with AMCP

channel capacity	packet size	channel switching delay
2 Mb/s	1000byte	$224\mu s$

Nevertheless, from the results shown in Figure 3.15, we see that CAM-MAC still outperforms SSCH with a factor of up to 1.50, in both disjoint and non-disjoint flow cases.

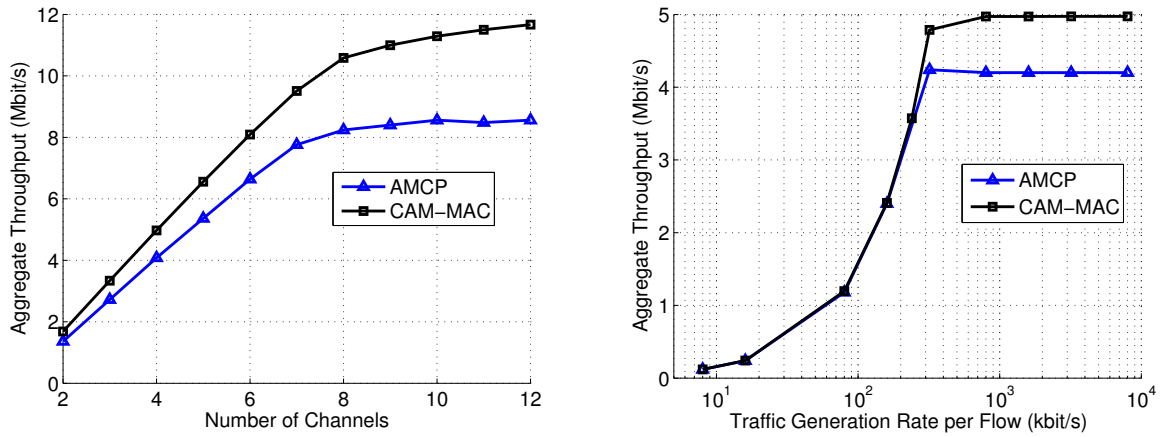
Comparison with AMCP

The parameters are shown in Table 3.3. There are 30 nodes forming 15 disjoint flows in a single-hop network. In the first set of simulations, the flows are always backlogged and the number of channels varies from 2 to 12. The results are shown in Figure 3.16a. We see that CAM-MAC achieves a throughput of 11.86Mb/s while AMCP achieves 8.5Mb/s when there are 12 channels, which indicates a ratio of 1.40. Furthermore, AMCP saturates at 10 channels whereas CAM-MAC still exhibits a growing trend beyond 12 channels. Note that this is consistent with our analysis in Section 3.3.3. To see how, substitute the parameters (in Table 3.3 and Section 3.3.2) into (3.1) and obtain $m_{bot} = 7$ ($\lceil 6.98 \rceil$). This means $m > m_{bot}$ and $n_f > m_{bot}$ ($n_f = 15$), which directs us to use (3.7) and (3.8) and accordingly obtain $S_{max} = 13.24$Mb/s ($G_{max} = 6.62$). Comparing this upper bound S_{max} with the throughput that CAM-MAC achieves at 12 channels (11.86Mb/s) shows that CAM-MAC still has space for throughput growth (recall that, in our simulation results in Section 3.4.1, CAM-MAC approaches the upper bound very closely).

In the second set of simulations, there are four channels and the traffic generation rate varies from 8kb/s to 8Mb/s. The results are shown in Figure 3.16b. Both CAM-MAC and AMCP have equal throughput at light traffic load, but apparent difference appears at medium to high load, and finally CAM-MAC saturates at 5Mb/s while AMCP saturates at 4.2Mb/s.

Furthermore, recalling the channel under-utilization of AMCP as mentioned in Section 2.2, we can expect larger difference if *variable* data packet sizes are used.

Summary: Our extensive comparison with representative multi-channel MAC protocols demonstrates the high productivity of CAM-MAC.



(a) Throughput versus number of channels.

(b) Throughput versus traffic load. Four channels.

Figure 3.16: Comparison with AMCP.

3.5 Discussion

3.5.1 Availability of Cooperation

An important issue related to CAM-MAC is how likely a node can obtain cooperation. This is addressed in Chapter 4 where we proposed a metric p_{co}, which is defined as the probability of obtaining cooperation when an MCC problem is created, to characterize the availability of cooperation. We analyzed this metric in multi-hop multi-channel networks, and the results show that it is high (> 0.7) in most cases. While cooperation is not always available, it does not mean that, on the average, a percentage of $1 - p_{co}$ handshakes will suffer from MCC problems; the percentage is in fact much lower. This is because p_{co}, by its definition, only counts those handshakes that create MCC problems (and not those that will succeed in data transmission without cooperation). Therefore, the probability that an arbitrary handshake will suffer from an MCC problem is lower than $1-p_{co}$. Combined with the high level of p_{co}, this helps explain why CAM-MAC can significantly outperform UNCOOP even if cooperation is not always available.

3.5.2 Two-hop neighbor discovery

CAM-MAC needs two-hop neighbor information for cooperation. To visualize this, see Figure 3.1a again, where node C can only cooperate if it knows that node U_2 is adjacent to node A_1, though U_2 is not adjacent to C itself. One simple way to acquire this information is to make use of the `Hello` messages traditionally used in (one-hop) neighbor discovery or other broadcast messages used in routing etc. Specifically, each node simply piggybacks its neighbors' IDs as well as its own ID when sending `Hello` messages, and nodes that receive this message can easily learn the two-hop neighbor information. This process does not noticeably incur overhead and complexity because it occurs in a very low frequency or only in the initialization phase, and can reuse the existing one-hop neighbor discovery process.

3.5.3 Impact of mobility

Mobility is a major factor attributing to network dynamics and affecting the reliability of (one-hop and two-hop) neighbor information. One simple way of adapting CAM-MAC to a mobile environment is to accordingly increase the frequency of updating neighbor information. We conducted multi-hop simulations using random waypoint model [68], with the same setup as in Section 3.4.2. We do the same three sets of simulations (varying traffic, payload size, and node density), except that each node moves at a speed uniformly distributed in (0, 10]m/s and toward a randomly chosen target point for each

movement. Each node independently updates neighbor information every 8 seconds. Our results showed only a marginal (3–8%) performance degradation in comparison to the static scenarios in Figure 3.11, Figure 3.12 and Figure 3.13. Actually, these results are not surprising because (i) in essence, cooperation is not a *compulsory* coordination mechanism but an *auxiliary* helping mechanism, which means that communication can proceed without cooperation, and (ii) *one cooperative node is enough* to prevent MCC problems and thus mobility can rarely make all neighboring nodes fail to cooperate. Consequently, cooperation is robust to low mobility levels which is the case in most application scenarios.

3.5.4 Energy consumption

Energy consumption is a critical issue for battery-powered devices commonly used in ad hoc networks. To save energy and prolong network lifetime, nodes should be kept in sleep mode as long as possible. However, they also need to stay active to gather channel usage information for both self-use (select free channels and receivers) and for cooperation purposes. This dilemma presents a challenge to multi-channel MAC protocol design especially in a cooperative environment. In this chapter, we focus on throughput improvement and do not specifically consider energy. This is also for a fair comparison with other state-of-the-art protocols, as those protocols do not consider energy either. Nevertheless, we recognize that energy conservation is an important issue and have addressed it in Chapter 5.

3.5.5 Multi-channel sensor networks

Wireless sensor networks were initially motivated by low data rate applications, but new applications demanding higher throughput and/or lower delay quickly emerged after a few years, such as those in wireless multimedia sensor networks [69]. Current sensor platforms, however, only provide very limited bandwidth, e.g., 19.2kb/s on MICA2 [70], 250kb/s on MICAz [70] and Telos [61]. On the other hand, some RF transceivers such as CC2420 used by MICAz and Telos provide multiple frequency channels. Therefore, we believe that multi-channel sensor MAC protocols are both needed and feasible. Two such protocols, MMSN [52] and CMAC [54] recently appeared. However, MMSN is highly complicated and requires tight clock synchronization, and CMAC requires a large number of channels to be operable. In our future work, we would like to investigate these issues and particularly consider cooperative sensor protocols.

3.6 Summary

DISH enables transmitter-receiver pairs to exploit the knowledge at individual idle neighbors to make more informed decisions in communication. In this chapter, we apply DISH to multi-channel MAC protocol design, and propose a cooperative multi-channel MAC protocol called CAM-MAC, where idle neighbors share control information with transmitter-receiver pairs to overcome MCC problems. This protocol uses a single transceiver and, unlike many other protocols, is fully asynchronous.

The simple idea of DISH turns out to be very effective. In the comparison of CAM-MAC with and without DISH, we observe remarkable performance difference. In the comparison with three recent and representative multi-channel MAC protocols, MMAC, SSCH and AMCP, CAM-MAC significantly outperforms all of them.

In a sense, DISH enables nodes to *store* channel usage information at their neighbors, and *retrieve* this information when it is needed. We also highlight that this is not a compulsory *coordination* mechanism; a network does not *rely on* cooperation and still operates when cooperation is not available. Ultimately, we believe that DISH merits due consideration in the future design of wireless network protocols.

Chapter 4

Performance Analysis, Implications, and Application

4.1 Introduction

In Chapter 3, we propose a cooperative multi-channel MAC protocol based on the idea of DISH, and demonstrated performance enhancement via simulations. However, what is lacking is a theoretical understanding of this new notion of cooperation. The benefit of DISH is that it can remove the need of using multiple transceivers [18, 19, 21–23] and time synchronization [24–29] in designing multi-channel MAC protocols. This motivates us to understand DISH from a theoretical perspective.

In this chapter, we view cooperation as a network resource and evaluate the *availability of cooperation* via a metric, p_{co}, the probability of obtaining cooperation (a more precise definition will be given in Def. 4.3). First, we analytically evaluate this metric in the context of multi-channel multi-hop wireless networks with randomly distributed nodes. Second, we verify the analysis via simulations and the results show that our analysis accurately characterizes the behavior of p_{co} as a function of underlying network parameters. This step also yields important insights into DISH with respect to network dynamics. Third, we investigate the correlation between p_{co} and network performance in terms of collision rate, packet delay, and throughput. The results indicate a near-linear relationship, which may significantly simplify performance analysis for cooperative networks and suggests that p_{co} be used as an appropriate performance indicator itself. In this work, we utilize, as appropriate, three different DISH contexts — model-based DISH, ideal DISH, and real DISH — to explore p_{co}. Finally, we exploit p_{co} as an instrument and apply our analysis to solving a channel bandwidth allocation problem, where we derive optimal schemes and provide general guidelines on bandwidth allocation for DISH networks.

4.1.1 Summary of Findings

Our aim in this chapter is to understand DISH and the availability of cooperation (p_{co}) from an analytical perspective. We provide an analysis which discloses *what* underlying factors and *how* these factors affect cooperation, and guides us in provisioning a DISH network toward the maximal performance.

Throughput analysis for multi-hop networks is difficult (and still an open problem in general), and it gets even more complicated in a multi-channel context with DISH. Our approach in this paper is to first look at p_{co} and then correlate it with network performance. We found a simple relationship between p_{co} and typical performance metrics.

The specific findings of this study are:

1. The availability of cooperation is high ($p_{co} > 0.7$) in typical cases, which suggests that DISH is feasible to use in multi-channel MAC protocols.

2. The performance degradation due to an increase in node density can be alleviated due to the simultaneously increased availability of cooperation.

3. The metric p_{co} increases with packet size for a given *bit* arrival rate, but decreases with packet size for a given *packet* arrival rate.

4. Node density and traffic load have opposite effects on p_{co} but node density is the dominating factor. This implies an improved scalability for DISH networks as p_{co} increases with node density.

5. p_{co} is strongly correlated to network performance and has a near-linear relationship with metrics such as throughput and delay. This may significantly simplify performance analysis for cooperative networks, and suggests that p_{co} be used as an appropriate performance indicator itself.

6. p_{co} is concave and not monotonic with respect to the ratio between control channel bandwidth and data channel bandwidth; there exists one and only one maximum p_{co} for each given set of parameters.

7. The optimal bandwidth ratio between the control and a data channel increases with node density but decreases with traffic load. This tells us that, for example, in a sparser network with heavier traffic, the control channel should be allocated less bandwidth for larger p_{co}.

8. In most cases (the number of channels is not too small), to boost the availability of cooperation, the control channel should be allocated more bandwidth than each data channel, rather than using *smaller* frequency band for a control channel or the usual *equal* bandwidth partition as suggested by many other studies.

In the rest of this chapter, Section 4.2 defines the system model and Section 4.3 presents our analysis. The analysis is then verified in Section 4.4 where we also provide an detailed investigation of p_{co} in different contexts of DISH. Next, we present an application of our analysis in Section 4.5. Finally, Section 4.6 gives concluding remarks.

4.2 System Model

We consider a static and connected ad hoc network in which each node is equipped with a single half-duplex transceiver that can dynamically switch between a set of orthogonal frequency channels but can only use one at a time. One channel is designated as a control channel and the others are designated as data channels. Nodes are placed in a plane area according to a two-dimensional Poisson point process.

We consider a class of multi-channel MAC protocols with their common framework described below. A transmitter-receiver pair uses an `McRTS`/`McCTS` handshake on the control channel to set up communication (like 802.11 RTS/CTS) for their subsequent `DATA`/`ACK` handshake on a data channel. To elaborate, a transmitter sends an `McRTS` on the control channel using CSMA/CA, i.e., it sends `McRTS` after sensing the control channel to be idle for a random period (addressed below) of time. The intended receiver, after successfully receiving `McRTS`, will send an `McCTS` and then switch to a data channel (the `McRTS` informs the receiver of the data channel). After successfully receiving the `McCTS`, the transmitter will also switch to its selected data channel, and otherwise it will backoff on the control channel for a random period (addressed below) of time. Hence it is possible that only the receiver switches to the data channel. After switching to a data channel, the transmitter will send a `DATA` and the receiver will respond with a `ACK` upon successful reception. Then both of them switch back to the control channel.

In the above we have mentioned two random periods of time. They are assumed to be designed such that idle intervals on the control channel are well randomized. Specifically, when a node is on the control channel, it sends control messages (an aggregated stream of `McRTS` and `McCTS`) according to a Poisson process with an unknown average rate (will be determined in Section 4.3). The validity of this assumption will be verified via simulations.

Note that we use `McRTS`, `McCTS`, `DATA` and `ACK` to refer to different packets (frames) without assuming specific frame formats. Since, logically, they must make a protocol functional, we assume that `McRTS` carries channel usage information (e.g., "who will use which channel for how long") and, for simplicity, `McCTS` is the same as `McRTS`.

We assume that, after switching to a data channel, a node will stay on that channel for a period of T_d, where T_d is the duration of a successful data channel handshake. This is also valid for a failed data channel handshake. To elaborate, let J_t and J_r be the period of staying on a data channel for a transmitter and a receiver, respectively, and let T_{pkt}, T_{pkt}^{tmo} and T_{pkt}^{det} be the transmission time of a *pkt* (`DATA` or `ACK`), the timeout interval for receiving a *pkt*, and the interval for detecting *pkt* collision, respectively. Hence clearly, $T_d = T_{data} + T_{ack}$. For a failed data channel handshake, there are three possible cases: (i) `DATA` is collided, in which case $J_t = T_{data} + T_{ack}^{tmo}$ and $J_r = T_{data}^{det}$, (ii) `ACK` is collided, in which case $J_t = T_{data} + T_{ack}^{det}$ and $J_r = T_d$, or (iii) only the receiver switches to a data channel (`McCTS` fails to reach the transmitter), in which case $J_r = T_{data}^{tmo}$. In all above cases, $J_t \approx J_r \approx T_d = T_{data} + T_{ack}$, because $T_{data}^{tmo} \approx T_{data} \gg T_{ack}^{tmo}$, and $T_{pkt} \approx T_{pkt}^{det}$ (collision detection is typically by checking CRC at the footer of a packet).

We ignore channel switching delay as it will not fundamentally change our results if it is negligible compared to T_d (the delay is $80\mu s$ [29] while T_d is more than $6ms$ for a 1.5KB data packet on a 2Mb/s channel). We also ignore SIFS and propagation delay for the same reason, provided that they are smaller than the transmission time of a control message.

We assume a uniform traffic pattern — all nodes have the equal data packet arrival rate, and for each data packet to send, a node chooses a receiver equally likely among its neighbors. We also assume a stable network — all data packets can be delivered to destinations within finite delay. In addition, packet reception fails if and only if packets collide with each other (i.e., no capture effect), transmission and interference ranges are equal, and neighboring nodes do not start sending control messages simultaneously (there is no time synchronization).

We do not assume a specific channel selection strategy; how a node selects data channels will affect how often conflicting channels are selected, but will not affect p_{co}. This is because, intuitively, we only care about the availability of cooperation (p_{co}) when a multi-channel coordination problem (a precise definition is given in Def. 4.1), which includes channel conflicting problem, *has been* created.

We do not assume a concrete DISH mechanism, i.e., nodes do not physically react upon a multi-channel coordination problem, because analyzing the availability of cooperation does not require the use of this resource. In fact, assuming one of the (numerous possible) DISH mechanisms will lose generality. Nevertheless, we will show in Section 4.4 that, when an ideal or a real DISH mechanism is used, the results do not fundamentally change. This could be an overall effect from contradicting factors which will be explained therein.

The following lists all parameters that are assumed known:

- n: node density. In a multi-hop network, it is the average number of nodes per R^2 where R is the transmission range. In a single-hop network, it is the total number of nodes.
- λ: the average data packet arrival rate at each node, including retransmissions.
- T_d: the duration of a data channel handshake.
- b: the transmission time of a control message. $b \ll T_d$.

4.3 Analysis

4.3.1 Problem Formulation and Analysis Outline

We first formally define p_{co}, which depends on two concepts called the *MCC problem* and the *cooperative node*.

Definition 4.1 (MCC Problem)**.** A multi-channel coordination (MCC) problem is either a *channel conflict* problem or a *deaf terminal* problem. A channel conflict problem is created when a node, say y, selects a channel to use (transmit or receive packets) but the channel is already in use by a neighboring node, say x. A deaf terminal problem is created when a node, say y, initiates communication to another node, say x, that is however on a different channel. In either case, we say that an MCC problem is created by x and y.

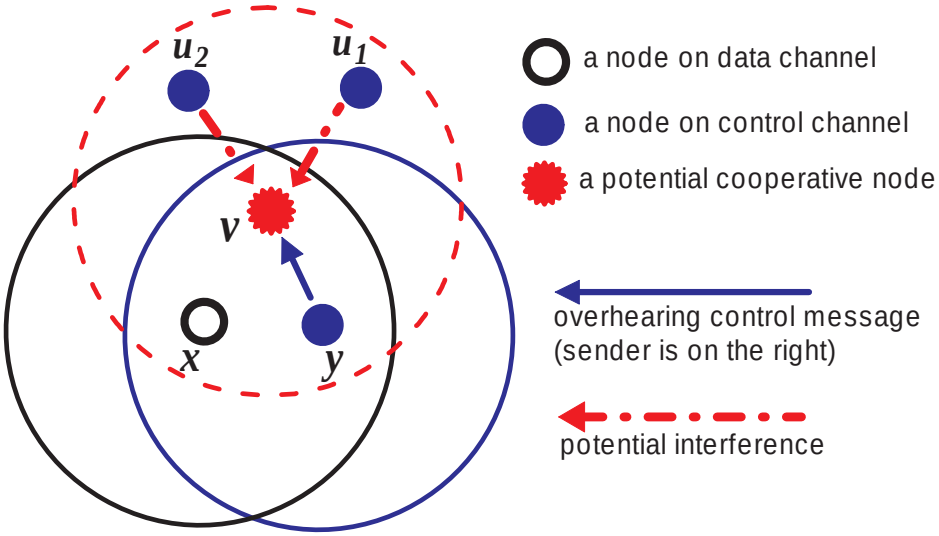


Figure 4.1: Illustration of an MCC problem and a cooperative node. Node x is performing a data channel handshake on CH_x, and y has just sent a control message during a control channel handshake. If this control message is an `McRTS` addressed to x, then a *deaf terminal* problem is created. If this control message indicates that y selects CH_x (recall that a control message carries channel usage information), a *channel conflict* problem is created. In either case, if a third node v identifies this problem (by overhearing x's and y's control messages successively), it is a cooperative node.

In a protocol that transmits `DATA` without requiring `ACK`, a channel conflict problem does not necessarily indicate an impending data collision. We do not consider such a protocol.

Definition 4.2 (Cooperative Node)**.** A node that identifies an MCC problem created by two other nodes, say x and y, is called a cooperative node with respect to x and y.

Figure 4.1 gives a visualization of the above two concepts in our system model.

Definition 4.3 (p_{co})**.** p_{co} is the probability for two arbitrary nodes that create an MCC problem to obtain cooperation, i.e., there is at least one cooperative node with respect to these two nodes.

Note that if there are multiple cooperative nodes and they are allowed by a DISH mechanism to send cooperative messages concurrently, a collision can result at node y. However, this collision *still indicates an MCC problem* and thus cooperation is still obtained. CAM-MAC [71] also implements this.

We distinguish the receiving of control messages. A transmitter receiving `McCTS` from its intended receiver is referred to as *intentional receiving*, and the other cases of receiving are referred to as *overhearing*, i.e., any node receiving `McRTS` (hence an intended receiver may also be a cooperative node) or any node other than the intended transmitter receiving `McCTS`.

Our notation is listed in Table 4.1. Overall, we will determine p_{co} by following the order of $p_{co}^{xy}(v) \rightarrow p_{co}^{xy} \rightarrow p_{co}$.

Consider $p_{co}^{xy}(v)$ first. Figure 4.1 illustrates that node v is cooperative if and only if it successfully overhears x's and y's control messages successively. Hence $\forall v \in \mathcal{N}_{xy}$,

$$\begin{aligned} p_{co}^{xy}(v) &= \Pr[\mathcal{O}(v \leftarrow x), \mathcal{O}(v \leftarrow y)] \\ &= \Pr[\mathcal{O}(v \leftarrow x)] \cdot \Pr[\mathcal{O}(v \leftarrow y) | \mathcal{O}(v \leftarrow x)]. \end{aligned} \tag{4.1}$$

Consider $\mathcal{O}(v \leftarrow i)$. For v to successfully overhear i's control message which is being sent during interval $[s_i, s_i + b]$, v must be silent on the control channel and not be interfered, i.e.,

$$\begin{aligned} \Pr[\mathcal{O}(v \leftarrow i)] = \Pr[\mathcal{S}_v(s_i, s_i + b), \bigcap_{u \in \mathcal{N}_v \setminus \{i\}} \mathcal{I}_u(s_i, s_i + b)], \\ \forall v \in \mathcal{N}_i. \end{aligned} \tag{4.2}$$

Table 4.1: Notation

Probabilities	p_{co}^{xy}	the probability that at least one cooperative node with respect to x and y exists
	$p_{co}^{xy}(v)$	the probability that node v is a cooperative node with respect to x and y
	p_{ctrl}	the probability that a node is on the control channel at an arbitrary point in time
	p_{succ}	the probability that a control channel handshake (initiated by an `McRTS`) is successful
	p_{oh}	the probability that an arbitrary node successfully overhears a control message
Events	$\mathcal{C}_v(t)$	node v is on the control channel at time t
	$\mathcal{O}(v \leftarrow i)$	node v successfully overhears node i's control message, given that i sends the message
	$\mathcal{S}_v(t_1, t_2)$	node v is silent (not transmitting) on the control channel during interval $[t_1, t_2]$
	$\mathcal{I}_v(t_1, t_2)$	node v does not introduce interference to the control channel during interval $[t_1, t_2]$, i.e., it is on a data channel or is silent on the control channel.
	$\Omega_u(t_1, t_2)$	node u, which is on a data channel at t_1, switches to the control channel in $[t_1, t_2]$
Others	$\mathcal{N}_i, \mathcal{N}_{ij}, \mathcal{N}_{v\setminus i}$	$\mathcal{N}_i$ is the set of node i's neighbors, $\mathcal{N}_{ij} = \mathcal{N}_i \cap \mathcal{N}_j$, $\mathcal{N}_{v\setminus i} = \mathcal{N}_v \setminus \mathcal{N}_i \setminus \{i\}$ (v's but not i's neighbors)
	$K_{ij}, K_{v\setminus i}$	$K_{ij} = \lvert\mathcal{N}_{ij}\rvert$, $K_{v\setminus i} = \lvert\mathcal{N}_{v\setminus i}\rvert$
	s_i	the time when node i starts to send a control message
	$\lambda_c, \lambda_{rts}, \lambda_{cts}$	the average rates of a node sending control messages, `McRTS`, and `McCTS`, respectively, *when it is on the control channel*. Clearly, $\lambda_c = \lambda_{rts} + \lambda_{cts}$.

Now we outline our analysis as below.

- Section 4.3.2: solves (4.2).
- Section 4.3.3: solves (4.1) and the target metric p_{co}.
- Section 4.3.4: case study in single-hop networks.

4.3.2 Solving Equation 4.2

Lemma 4.1. *If node u is on a data channel at t_1, then the probability that u does not introduce interference to the control channel during $[t_1, t_2]$, where $t_2 - t_1 = \Delta t < T_d$, is given by*

$$\Pr[\mathcal{I}_u(t_1, t_2) | \overline{\mathcal{C}_u(t_1)}] = 1 - \frac{\Delta t}{T_d} + \frac{1 - e^{-\lambda_c \Delta t}}{\lambda_c T_d}.$$

Proof. By the total probability theorem,

$$l.h.s. = \Pr[\Omega_u(t_1, t_2)] \times \Pr[\mathcal{I}_u(t_1, t_2) | \Omega_u(t_1, t_2)] + \Pr[\overline{\Omega_u(t_1, t_2)}] \times 1.$$

Let t_{sw} be the time when node u switches to the control channel (see Figure 4.2). It is uniformly distributed in $[t_1, t_1 + T_d]$ because the time when u started its data channel handshake is unknown, and hence

$$\Pr[\Omega_u(t_1, t_2)] = \frac{\Delta t}{T_d}. \tag{4.3}$$

Since control channel traffic is Poisson with rate λ_c,

$$\Pr[\mathcal{I}_u(t_1, t_2) | \Omega_u(t_1, t_2)] = \Pr[\mathcal{S}_u(t_{sw}, t_2) | \Omega_u(t_1, t_2)] = \mathbb{E}[e^{-\lambda_c \tau}]$$

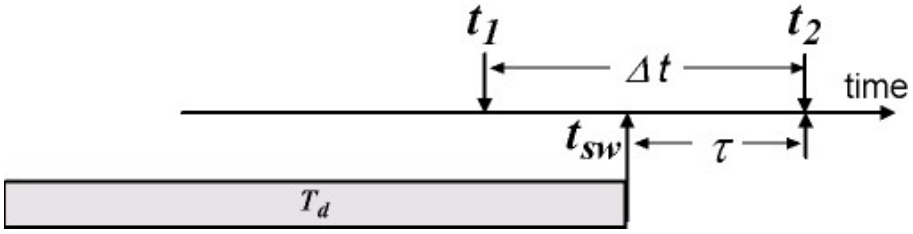


Figure 4.2: A node switches to the control channel after data channel handshaking.

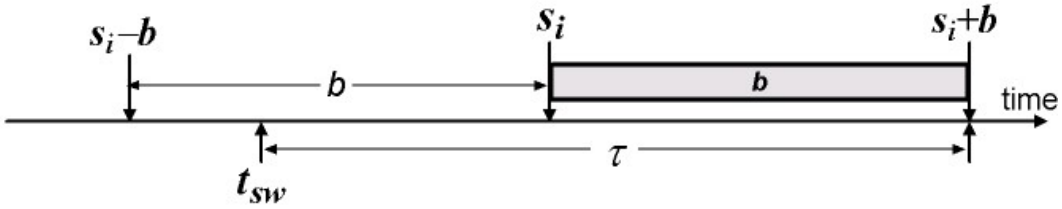


Figure 4.3: The vulnerable period of v is $[s_i - b, s_i + b]$, in which node $u \in \mathcal{N}_{v\setminus i}$ should not start transmission on the control channel.

where $\tau = t_2 - t_{sw}$ is uniformly distributed in $[0, \Delta t]$ by the same argument leading to (4.3). Hence

$$\mathbb{E}[e^{-\lambda_c \tau}] = \int_0^{\Delta t} e^{-\lambda_c \tau_0} \frac{1}{\Delta t} d\tau_0 = \frac{1 - e^{-\lambda_c \Delta t}}{\lambda_c \Delta t},$$

and then by substitution the lemma is proven. □

Proposition 4.2. *If node v is overhearing a control message from node i during $[s_i, s_i + b]$, then the probability that a node $u \in \mathcal{N}_v$ does not interfere with v is given by*

$$\Pr[\mathcal{I}_u(s_i, s_i + b)] = \begin{cases} 1, & u \in \mathcal{N}_{vi}; \\ p_{ni\text{-}oh}, & u \in \mathcal{N}_{v\setminus i}. \end{cases}$$

where

$$p_{ni\text{-}oh} = \; p_{ctrl} \cdot e^{-2\lambda_c b} + (1 - p_{ctrl}) \cdot (1 - \frac{2b}{T_d} + \frac{1 - e^{-2\lambda_c b}}{\lambda_c T_d}).$$

Proof. In the case of $u \in \mathcal{N}_{vi}$, no matter u is on the control channel at s_i, or is on a data channel at s_i but switches to the control channel before $s_i + b$, it will sense a busy control channel (due to CSMA) and thus keep silent.

In the case of $u \in \mathcal{N}_{v\setminus i}$, see Figure 4.3. Note that the vulnerable period of v is $[s_i - b, s_i + b]$ instead of $[s_i, s_i + b]$, because a transmission started within $[s_i - b, s_i]$ will end within $[s_i, s_i + b]$. Therefore, by the total probability theorem,

$$\begin{aligned} p_{ni\text{-}oh} =& \Pr[\mathcal{C}_u(s_i - b)] \cdot \Pr[\mathcal{I}_u(s_i - b, s_i + b) | \mathcal{C}_u(s_i - b)] \\ &+ \Pr[\overline{\mathcal{C}_u(s_i - b)}] \cdot \Pr[\mathcal{I}_u(s_i - b, s_i + b) | \overline{\mathcal{C}_u(s_i - b)}] \\ =& \; p_{ctrl} \cdot e^{-2\lambda_c b} + (1 - p_{ctrl}) \cdot (1 - \frac{2b}{T_d} + \frac{1 - e^{-2\lambda_c b}}{\lambda_c T_d}), \end{aligned}$$

where $\Pr[\mathcal{I}_u(s_i - b, s_i + b) | \overline{\mathcal{C}_u(s_i - b)}]$ is solved by Lemma 4.1. □

Thus (4.2) can be reduced to

$$\Pr[\mathcal{O}(v \leftarrow i)] \approx p_{ctrl} \; p_{ni\text{-}oh}^{K_{v\setminus i}}. \tag{4.4}$$

Proof. Based on the proof for the case $u \in \mathcal{N}_{vi}$ in Proposition 4.2, it is easy to show that $\mathcal{S}_v(s_i, s_i+b) \Leftrightarrow \mathcal{C}_v(s_i)$. Hence

$$\Pr[\mathcal{S}_v(s_i, s_i+b)] = \Pr[\mathcal{C}_v(s_i)] = p_{ctrl}.$$

Treating events $\mathcal{S}_v(s_i, s_i+b)$ (node v is silent on the control channel) and $\mathcal{I}_u(s_i, s_i+b)$ (node u does not interfere the control channel) being independent of each other, as an approximation, we have

$$\Pr[\mathcal{O}(v \leftarrow i)] \approx p_{ctrl} \prod_{u \in \mathcal{N}_{v\setminus i}} p_{ni\text{-}oh} = p_{ctrl}\, p_{ni\text{-}oh}^{K_{v\setminus i}}.$$

□

The above contains two unknown variables, p_{ctrl} and λ_c, and the following solves for them.

For p_{ctrl}, consider a sufficiently long period T_0. On the one hand, the number of arrival data packets at each node is λT_0. On the other hand, each node spends a total time of $(1-p_{ctrl})T_0$ on data channels, a factor η of which is used for sending arrival data packets. Since the network is stable (incoming traffic is equal to outgoing traffic), we establish a balanced equation:

$$\lambda T_0 T_d = \eta\,(1-p_{ctrl})T_0.$$

To determine η, noticing that a node switches to data channels either as a transmitter (with an average rate of λ) or as a receiver (with an average rate of λ_{cts}), we have $\eta = \lambda/(\lambda+\lambda_{cts})$. Substituting this into the above yields

$$p_{ctrl} = 1 - (\lambda + \lambda_{cts})T_d. \tag{4.5}$$

For λ_c (and λ_{cts}), we need a proposition and two lemmas.

Proposition 4.3. *If node i (transmitter) is intentionally receiving `McCTS` from node j (receiver) during $[s_j, s_j+b]$, then the probability that a node $u \in \mathcal{N}_i$ does not interfere with i is given by*

$$\Pr[\mathcal{I}_u(s_j, s_j+b)] = \begin{cases} 1, & u \in \mathcal{N}_{ij}; \\ p_{ni\text{-}cts}, & u \in \mathcal{N}_{i\setminus j}. \end{cases}$$

where $p_{ni\text{-}cts} =$

$$(1-p_{ctrl})[1 - \frac{b}{T_d}(1 + \frac{b}{T_d} - \frac{1-e^{-\lambda_c b}}{\lambda_c T_d} - e^{-\lambda_c b})] + p_{ctrl}.$$

Proof. The case of $u \in \mathcal{N}_{ij}$ follows the same line as the proof for Proposition 4.2. In the case of $u \in \mathcal{N}_{i\setminus j}$, the only difference from Proposition 4.2 is that now we are implicitly given the fact that i was transmitting `McRTS` during $[s_j-b, s_j]$. This excludes i's any neighbor u interfering in $[s_j-b, s_j]$. Therefore i's vulnerable period is $[s_j, s_j+b]$ instead of $[s_j-b, s_j+b]$ as compared to Proposition 4.2. So

$$\begin{aligned} p_{ni\text{-}cts} =& \Pr[\mathcal{C}_u(s_j-b)] \cdot \Pr[\mathcal{I}_u(s_j, s_j+b) | \mathcal{C}_u(s_j-b)] \\ &+ \Pr[\overline{\mathcal{C}_u(s_j-b)}] \cdot \Pr[\mathcal{I}_u(s_j, s_j+b) | \overline{\mathcal{C}_u(s_j-b)}]. \end{aligned}$$

Note that we condition on $\mathcal{C}_u(s_j-b)$ instead of $\mathcal{C}_u(s_j)$, because s_j is not an *arbitrary* time due to i's `McRTS` transmission during $[s_j-b, s_j]$, which leads to $\Pr[\mathcal{C}_u(s_j)] \neq p_{ctrl}$.

First, $\Pr[\mathcal{I}_u(s_j, s_j+b) | \mathcal{C}_u(s_j-b)] = 1$. This is because, as $\mathcal{C}_u(s_j-b) \Leftrightarrow \mathcal{S}_u(s_j-b, s_j)$ which is easy to show, u will successfully overhear i's `McRTS`, and hence will keep silent in the next period of b to avoid interfering with i receiving `McCTS`.

Next consider $\Pr[\mathcal{I}_u(s_j, s_j+b) | \overline{\mathcal{C}_u(s_j-b)}]$ where u is on a data channel at s_j-b. If u switches to the control channel (i) before s_j, it will be suppressed by i's `McRTS` transmission until s_j, and thus the vulnerable period of i receiving `McCTS` is $[s_j, s_j+b]$, (ii) within $[s_j, s_j+b]$, this has been solved by Lemma 4.1, or (iii) after s_j+b, the probability to solve is obviously 1. Therefore,

$$\begin{aligned} \Pr[\mathcal{I}_u(s_j, s_j+b) | \overline{\mathcal{C}_u(s_j-b)}] =& \Pr[\Omega_u(s_j-b, s_j)]\; e^{-\lambda_c b} \\ &+ \Pr[\Omega_u(s_j, s_j+b)]\,(1 - \frac{b}{T_d} + \frac{1-e^{-\lambda_c b}}{\lambda_c T_d}) \\ &+ \{1 - \Pr[\Omega_u(s_j-b, s_j)] - \Pr[\Omega_u(s_j, s_j+b)]\} \times 1. \end{aligned}$$

According to (4.3), $\Pr[\Omega_u(s_j - b, s_j)] = \Pr[\Omega_u(s_j, s_j + b)] = b/T_d$. Then by substitution the proposition is proven. □

Lemma 4.4. *For a Poisson random variable K with mean value $\overline{K}$, and $0 < p < 1$,*

$$\mathbb{E}[p^K] = e^{-(1-p)\overline{K}}.$$

Proof.

$$\mathbb{E}[p^K] = \sum_0^\infty p^k \Pr(K = k) = e^{-\overline{K}} \sum_0^\infty \frac{(p\overline{K})^k}{k!} = e^{-(1-p)\overline{K}}.$$

□

Lemma 4.5. *For three random distributed nodes v, i and j,*

(a) $\mathbb{E}[K_{v\setminus i} | v \in \mathcal{N}_i] \approx 1.30n$.

(b) $\mathbb{E}[K_{v\setminus i} | v \in \mathcal{N}_{ij}] \approx 1.19n$.

(c) $\mathbb{E}[K_{ij}] \approx 1.84n$.

Proof. Let $A_s(\gamma)$ be the intersection area of two circles with a distance of γ between their centers, and $\gamma < R$, where R is the circles' radius. It can be derived from [72] that

$$A_s(\gamma) = 2R^2 \arccos\frac{\gamma}{2R} - \gamma\sqrt{R^2 - \frac{\gamma^2}{4}}.$$

Let $A_c(\gamma)$ be the complementary area of $A_s(\gamma)$, i.e., $A_c(\gamma) = \pi R^2 - A_s(\gamma)$, and let A_{ij} and $A_{v\setminus i}$ be the areas where $\mathcal{N}_{ij}$ and $\mathcal{N}_{v\setminus i}$ are located, respectively.

(a) See Figure 4.4a. Letting $\gamma = ||vi||$ where $v \in \mathcal{N}_i$, and $f(r)$ be its probability density function (pdf), we have $f(r)dr = 2\pi r dr/(\pi R^2)$, which gives $f(r) = 2r/R^2$. Thus

$$\mathbb{E}[A_{v\setminus i} | v \in \mathcal{N}_i] = \int_0^R A_c(r) f(r) dr \approx 1.30R^2,$$

and hence $\mathbb{E}[K_{v\setminus i} | v \in \mathcal{N}_i] \approx n \cdot 1.30R^2/R^2 = 1.30n$.

(b) Let $\gamma_1 = ||vi||$ where $v \in \mathcal{N}_{ij}$, and $f_1(r_1)$ be its pdf. To solve $f_1(r_1)$, we consider $v \in \mathcal{N}_{i\setminus j}$ instead (see Figure 4.4b):

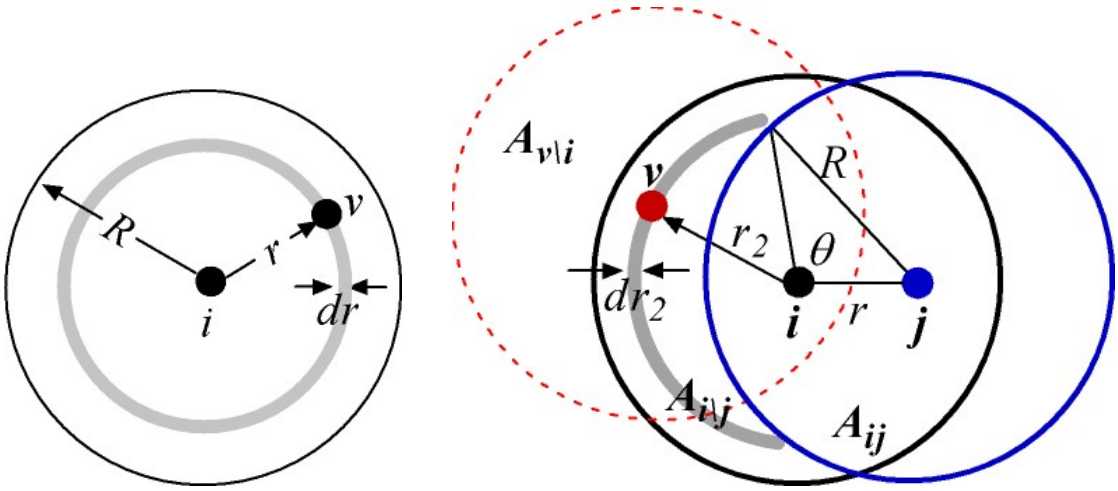
$$\begin{aligned}
\because\ \mathbb{E}[A_{v\setminus i} | v \in \mathcal{N}_i] &= p_1 \cdot \mathbb{E}[A_{v\setminus i} | v \in \mathcal{N}_{ij}] \\
&\quad + (1 - p_1) \cdot \mathbb{E}[A_{v\setminus i} | v \in \mathcal{N}_{i\setminus j}] \\
\text{where } p_1 &\triangleq \Pr[v \in \mathcal{N}_{ij} | v \in \mathcal{N}_i] = \frac{A_s(r)}{\pi R^2}, \\
\therefore\ \mathbb{E}[A_{v\setminus i} | v \in \mathcal{N}_{ij}] &= p_1^{-1} \mathbb{E}[A_{v\setminus i} | v \in \mathcal{N}_i] \\
&\quad - (p_1^{-1} - 1) \cdot \mathbb{E}[A_{v\setminus i} | v \in \mathcal{N}_{i\setminus j}]. \qquad (4.6)
\end{aligned}$$

To determine $\mathbb{E}[A_{v\setminus i} | v \in \mathcal{N}_{i\setminus j}]$, let $\gamma_2 = ||vi||$ where $v \in \mathcal{N}_{i\setminus j}$, and $f_2(r_2)$ be its pdf. It is determined by

$$f_2(r_2)dr_2 = \frac{2(\pi - \theta) r_2 dr_2}{A_{i\setminus j}} \text{ and } \cos\theta = \frac{r_2^2 + r^2 - R^2}{2r_2 r}.$$

Therefore

$$\begin{aligned}
\mathbb{E}[A_{v\setminus i} | v \in \mathcal{N}_{i\setminus j}] &= \int_{R-r}^R A_c(r_2) f_2(r_2) dr_2 \\
&= \int_{R-r}^R \frac{2r_2 A_c(r_2)}{A_c(r)} (\pi - \arccos\frac{r_2^2 + r^2 - R^2}{2r_2 r}) dr_2.
\end{aligned}$$



(a) $v \in \mathcal{N}_i$. (b) $v \in \mathcal{N}_{i\setminus j}$.

Figure 4.4: Deriving the pdf of distance $||vi||$.

Substituting this and $\mathbb{E}[A_{v\setminus i}|v \in \mathcal{N}_i] \approx 1.30R^2$ (by case (a)) into (4.6) solves $\mathbb{E}[A_{v\setminus i}|v \in \mathcal{N}_{ij}]$, which we denote by $M(r)$. Then we have

$$\mathbb{E}[K_{v\setminus i}|v \in \mathcal{N}_{ij}] = \frac{n}{R^2}\int_0^R M(r)f(r)dr \approx 1.19n.$$

(c) Proven by noticing that A_{ij} is complementary to the area corresponding to case (a). □

Based on Proposition 4.2 and 4.3, Lemma 4.4 and 4.5-(a), we can prove

$$\begin{aligned} p_{oh} &\approx p_{ctrl}\ \exp[-1.30n(1-p_{ni\text{-}oh})], \\ p_{succ} &\approx p_{oh}\exp[-1.30n(1-p_{ni\text{-}cts})]. \end{aligned} \tag{4.7}$$

Proof. Taking the expectation of $\Pr[\mathcal{O}(v \leftarrow i)]$ (given by (4.4)) over all neighboring (v, i) pairs using Lemma 4.4 and Lemma 4.5-(a):

$$p_{oh} \approx p_{ctrl}\ \mathbb{E}[p_{ni\text{-}oh}^{K_{v\setminus i}}] \approx p_{ctrl}\ \exp[-1.30n(1-p_{ni\text{-}oh})].$$

To solve for p_{succ}, notice that for a control channel handshake to be successful, (i) the `McRTS` must be successfully received by the receiver, with probability p_{oh}, and (ii) the `McCTS` must be successfully received by the transmitter, with probability $\mathbb{E}[p_{ni\text{-}cts}^{K_{i\setminus j}}]$ based on Proposition 4.3 (assuming that $p_{ni\text{-}cts}$ holds for nodes in $\mathcal{N}_{i\setminus j}$ independently, as an approximation). Therefore,

$$p_{succ} \approx p_{oh}\mathbb{E}[p_{ni\text{-}cts}^{K_{i\setminus j}}] \approx p_{oh}\exp[-1.30n(1-p_{ni\text{-}cts})].$$

□

Now we solve λ_c (together with λ_{cts}). From the perspective of a transmitter, the average number of successful control channel handshakes that it initiates per second is $p_{ctrl}\lambda_{rts}p_{succ}$. Since each successful control channel handshake leads to transmitting one data packet, we have

$$p_{ctrl}\lambda_{rts}p_{succ} = \lambda.$$

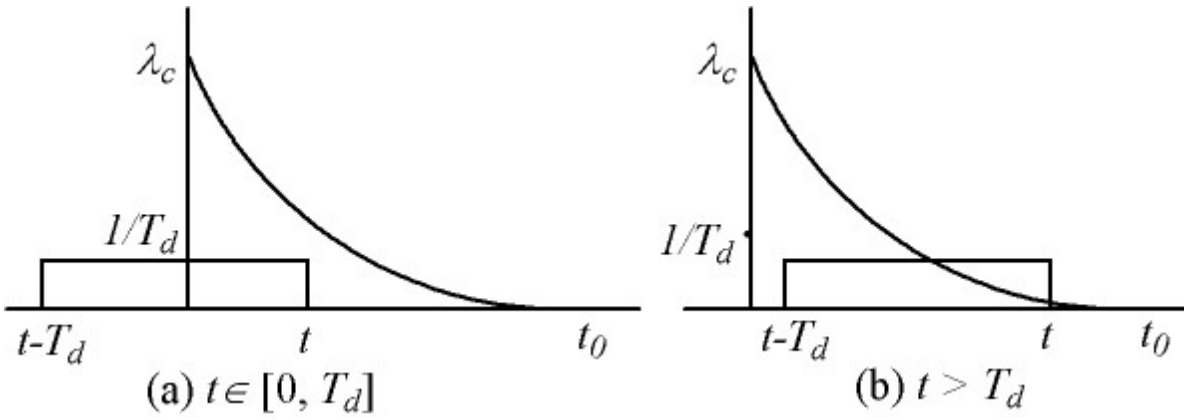


Figure 4.5: The convolution of $\frac{1}{T_d}$ ($t \in [0, T_d]$) and $\lambda_c e^{-\lambda_c t}$ ($t > 0$).

From the perspective of a receiver, it sends an `McCTS` when it successfully receives (overhears) an `McRTS` addressed to it, * and hence $\lambda_{cts} = \lambda_{rts}\, p_{oh}$. Then combining these with $\lambda_c = \lambda_{rts} + \lambda_{cts}$ yields

$$\lambda_c = \frac{\lambda(1 + p_{oh})}{p_{ctrl}\, p_{succ}} \text{ and } \lambda_{cts} = \frac{\lambda\ p_{oh}}{p_{ctrl}\, p_{succ}} \tag{4.8}$$

where p_{oh} and p_{succ} are given in (4.7).

4.3.3 Solving Equation 4.1 and Target Metric p_{co}

Based on the proof of (4.4), it can be derived that

$$\Pr[\mathcal{O}(v \leftarrow y) | \mathcal{O}(v \leftarrow x)] \approx p^{\star}_{ctrl}\ p^{K_{v \setminus y}}_{ni\text{-}oh}, \tag{4.9}$$

$$\text{where} \quad p^{\star}_{ctrl} \triangleq \Pr[\mathcal{C}_v(s_y) | \mathcal{O}(v \leftarrow x)].$$

Note that $p^{\star}_{ctrl} \neq p_{ctrl}$, because s_y is not an *arbitrary* time for v due to the effect form $\mathcal{O}(v \leftarrow x)$. The reason is that $\mathcal{O}(v \leftarrow x)$ implies $\mathcal{C}_v(s_x)$, and thus for $\mathcal{C}_v(s_y)$ to happen, v must stay *continuously* on the control channel during $[s_x, s_y]$ (otherwise, a switching will lead to v staying on the data channel for T_d, but $s_x + T_d > s_y$ since x's data communication is still ongoing at s_y, and hence $\mathcal{C}_v(s_y)$ can never happen).

It can be proven that

$$p^{\star}_{ctrl} = \frac{(w\lambda_c - \frac{1-w}{T_d})\ g(\lambda_c + \lambda_w) + \frac{1-w}{T_d}\ g(\lambda_w)}{1 - w + (w\lambda_c - \frac{1-w}{T_d})\ g(\lambda_c)} \tag{4.10}$$

where

$$g(x) = \frac{1 - e^{-xT_d}}{x}, \quad w = \frac{p_{ctrl} - p_{oh}}{1 - p_{oh}}, \text{ and}$$
$$\lambda_w = \lambda_{rts} p_{succ} + \lambda_{cts}.$$

Proof. Recall that node v must stay continuously on the control channel during $[s_x, s_y]$. Let $\tau_c = s_y - s_x$ and suppose v switches to a data channel at $s_x + \tau_w$, then we need $\tau_w > \tau_c$. Hence $p^{\star}_{ctrl} = \Pr(\tau_w > \tau_c)$, where $\tau_c \in [0, T_d]$.

Denote by $f_{\tau_c}(t)$ the pdf of an unbounded τ_c ($s_y \in (s_x, \infty)$). The fact that a MCC problem is created by x and y (at s_y) implies that y missed x's control message (at s_x). This is due to one of the following: (i) y is on the control channel at s_x but interfered, in which case $f_{\tau_c}(t)$ is $\lambda_c e^{-\lambda_c t}$ (ignoring the short interference period which is in the magnitude of b, while τ_c is in the magnitude of T_d), (ii) y is on a

*In a more sophisticated protocol model, a receiver may not respond even if it receives McRTS due to, e.g., disagreeing with the transmitter's channel selection. This behavior is modeled by ideal DISH and real DISH, which will be shown in Section 4.4 that it does not fundamentally change the results.

data channel at s_x, in which case y must switch to the control channel before s_y. Again see Figure 4.2, where t_1 and t_2 are now s_x and s_y, respectively. Let $\tau_1 = t_{sw} - s_x$ and $\tau_2 = s_y - t_{sw}$, then $\tau_c = \tau_1 + \tau_2$. Note that τ_1 is uniformly distributed in $[0, T_d]$, τ_2 is exponentially distributed with the mean of $1/\lambda_c$, and τ_1 and τ_2 can be regarded as independent. Therefore, $f_{\tau_c}(t)$ is the convolution of $\frac{1}{T_d}$ $(t \in [0, T_d])$ and $\lambda_c e^{-\lambda_c t}$ $(t > 0)$, which can be calculated by referring to Figure 4.5, to be $f^d_{\tau_c}(t) =$

$$\frac{1 - e^{-\lambda_c t}}{T_d}[u(t) - u(t - T_d)] + \frac{e^{-\lambda_c t}}{T_d}(e^{\lambda_c T_d} - 1)u(t - T_d).$$

where $u(\cdot)$ is the unit step function.

A weighted sum of the above cases (i) and (ii) gives

$$f_{\tau_c}(t) = w\ \lambda_c e^{-\lambda_c t} + (1 - w)\ f^d_{\tau_c}(t)$$

where w is the weight for case (i). To determine w, note that the probability of case (ii) is $1 - p_{ctrl}$, and the probability of case (i) is $p_{ctrl}(1 - p_{ni\text{-}oh}^{K_{y\setminus x}})$ (using (4.4)) whose mean is $p_{ctrl}(1 - \exp[-1.30n(1 - p_{ni\text{-}oh})])$. Therefore

$$w = \frac{p_{ctrl}[1 - e^{-1.30n(1-p_{ni\text{-}oh})}]}{p_{ctrl}[1 - e^{-1.30n(1-p_{ni\text{-}oh})}] + (1 - p_{ctrl})} = \frac{p_{ctrl} - p_{oh}}{1 - p_{oh}}.$$

Finally we compute $p^\star_{ctrl} = \Pr(\tau_w > \tau_c)$ using $f_{\tau_c}(t)$. Recall that $f_{\tau_c}(t)$ is the pdf of an unbounded τ_c but τ_c is in fact bounded within $[0, T_d]$, therefore its actual pdf is $f_{\tau_c}(t) / \int_0^{T_d} f_{\tau_c}(t)dt$. Assuming that τ_w is exponentially distributed with mean $1/\lambda_w$, we have

$$p^\star_{ctrl} = \mathbb{E}_{\tau_c \in [0,T_d]}\ \Pr(\tau_w > \tau_c) = \int_0^{T_d} e^{-\lambda_w t} \frac{f_{\tau_c}(t)}{\int_0^{T_d} f_{\tau_c}(t)dt} dt$$

which reduces to (4.10). For λ_w, noticing that it is the average rate of a node on the control channel switching to data channels, which happens when a node successfully initiates a control channel handshake via `McRTS` or sends a `McCTS`, we have $\lambda_w = \lambda_{rts} p_{succ} + \lambda_{cts}$. □

Combining (4.4) and (4.9) reduces (4.1) to

$$p^{xy}_{co}(v) \approx p_{ctrl}\ p^\star_{ctrl}\ p_{ni\text{-}oh}^{K_{v\setminus x} + K_{v\setminus y}},\ \forall v \in \mathcal{N}_{xy}. \tag{4.11}$$

Let $p^{xy}_{co}(\star)$ be the average of $p^{xy}_{co}(v)$ over all $v \in \mathcal{N}_{xy}$, i.e., $p^{xy}_{co}(\star)$ is the probability that an arbitrary node in $\mathcal{N}_{xy}$ is cooperative with respect to x and y, Using Lemma 4.5-(b),

$$p^{xy}_{co}(\star) \approx p_{ctrl}\ p^\star_{ctrl}\ \exp[-2.38n(1 - p_{ni\text{-}oh})]. \tag{4.12}$$

By the definition of p^{xy}_{co} in Table 4.1,

$$p^{xy}_{co} \approx 1 - \prod_{v \in \mathcal{N}_{xy}} [1 - p^{xy}_{co}(v)] \approx 1 - [1 - p^{xy}_{co}(\star)]^{K_{xy}}, \tag{4.13}$$

where the events corresponding to $1 - p^{xy}_{co}(v)$, i.e., nodes not being cooperative with respect to x and y, are regarded as independent of each other, as an approximation.

Thus p_{co} is determined by averaging p^{xy}_{co} over all (x, y) pairs that are possible to create MCC problems. It can be proved [73] that these pairs are neighboring pairs (x, y) satisfying (d_i denoting the degree of a node i)

(a) $d_x \geq 2$, $d_y \geq 2$, but not $d_x = d_y = 2$, or

(b) $d_x = d_y = 2$, but x and y are not on the same three-cycle (triangle).

This condition is satisfied by all neighboring pairs in a *connected* random network, because the connectivity requires a sufficiently high node degree ($5.18 \log N$ where N is the total number of nodes [74]) which is much larger than 2. Therefore, taking expectation of (4.13) over all neighboring pairs using Lemma 4.4 and Lemma 4.5-(c),

$$\begin{aligned} p_{co} &= 1 - \exp[-p_{co}^{xy}(\star)\overline{K_{xy}}] \\ &\approx 1 - \exp[-1.84n\ p_{co}^{xy}(\star)]. \end{aligned} \tag{4.14}$$

This completes the analysis.

4.3.4 Special Case: Single-Hop Networks

Now that all nodes are in the communication range of each other, we have $p_{ni\text{-}oh} = p_{ni\text{-}cts} = 1$ according to Proposition 4.2 and 4.3, which leads to $p_{succ} = p_{oh} = p_{ctrl}$ according to (4.7), and $p_{co}^{xy}(v) = p_{ctrl}\ p_{ctrl}^{\star}$ according to (4.11). Hence (4.13) reduces to

$$p_{co}^{xy} = 1 - (1 - p_{ctrl}\ p_{ctrl}^{\star})^{K_{xy}},$$

where K_{xy} is the number of all possible cooperative nodes with respect to x and y, leading to $K_{xy} = n - 4$. [†] So, as the average of p_{co}^{xy},

$$p_{co} = 1 - (1 - p_{ctrl}\ p_{ctrl}^{\star})^{n-4}, \tag{4.15}$$

where p_{ctrl} is given below, by solving equations (4.5), (4.7) and (4.8),

$$p_{ctrl} = \frac{1}{2}(1 - \lambda T_d + \sqrt{1 + \lambda T_d(\lambda T_d - 6)}\ ),$$

and $p_{ctrl}^{\star}$ is given below, by reducing (4.10) with $w = 0$,

$$p_{ctrl}^{\star} = \frac{g(\lambda_w) - g(\lambda_c + \lambda_w)}{T_d - g(\lambda_c)}$$

$$\text{where}\quad \lambda_c = \frac{1}{2}\left(\frac{1 - \sqrt{1 + \lambda T_d(\lambda T_d - 6)}}{\lambda T_d^2} - \frac{3}{T_d}\right),$$

$$\lambda_w = \frac{1 - \sqrt{1 + \lambda T_d(\lambda T_d - 6)}}{T_d} - \lambda.$$

4.4 Investigating p_{co} with DISH

We verify the analysis in both single-hop and multi-hop networks and identify key findings therein. We also investigate the correlation between p_{co} and network performance.

4.4.1 Protocol Design and Simulation Setup

Non-Cooperative Case

This is a multi-channel MAC protocol based on the protocol framework described in Section 4.2. Key part of its pseudo-code is listed below, where S_{ctrl} is the control channel status (FREE/BUSY) detected by the node running the protocol, S_{node} is the node's state (IDLE/TX/RX, etc.), L_{queue} is the node's current queue length, and they are initialized as FREE, IDLE and 0, respectively. The frame format of `McRTS` and `McCTS` is shown in Figure 4.6, where we can see that they carry channel usage information. A node that overhears `McRTS` or `McCTS` will cache the information in a *channel usage table* shown in Figure 3.7, where `Until` is converted from `Duration` by adding the node's own clock.

As is based on the system model, this protocol does not use a concrete DISH mechanism, i.e., cooperation is treated as a resource while not actually utilized.

[†] This means all nodes excluding x, x', y and y', where x' is the node that x is currently communicating with (on a data channel), and y' is the node that y *was* communicating with (on a data channel) when x was setting up communication with x' (hence y and y' missed x's control message). None of these four nodes can be a cooperative node.

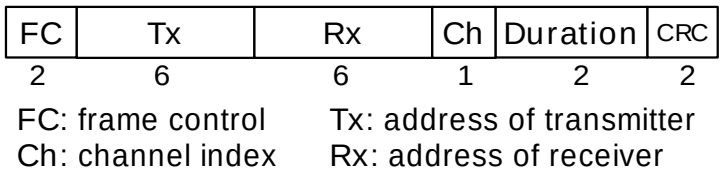


Figure 4.6: Frame format of `McRTS` and `McCTS`.

Tx	Rx	Ch	Until
T1	R1	1	00:15:36
T2	R2	3	00:16:01

Figure 4.7: Channel usage table.

Procedure 4.1 PKT-ARRIVAL

[Called when a data packet arrives]

1: enqueue the packet, L_{queue}++
2: **if** $S_{ctrl} = FREE \wedge S_{node} = IDLE \wedge L_{queue} = 1$ **then**
3: call ATTEMPT-RTS
4: **end if**

Ideal DISH

This protocol is by adding an ideal cooperating mechanism to the non-cooperative case. Each time when an MCC problem is created by nodes x and y and if at least one cooperative node is available, the node that is on the control channel, i.e., node y, will be informed without any message actually sent, and then back off to avoid the MCC problem.

Real DISH

In this protocol, cooperative nodes actually send cooperative messages to inform the transmitter or receiver of the MCC problem so that it will back off. We design a real DISH protocol by adapting CAM-MAC [71] . We change CAM-MAC such that $\|PRA\| + \|CFA\| = \|McRTS\| = \|PRB\| + \|CFB\| = \|McCTS\|$, where $\|\cdot\|$ denotes packet size.

Simulation Setup

There are six channels of data rate 1Mb/s each (the number of channels does not affect results as long as the network is kept stable). Data packets arrive at each node as a Poisson process. The uniform traffic pattern as in the model is used. Traffic load λ (pkt/s), node density n ($1/R^2$), and packet size L (byte) will vary in simulations. In multi-hop networks, the network area is 1500m×1500m and the transmission range is 250m. Each simulation is terminated when a total of 100,000 data packets are sent over the network, and each set of results is averaged over 15 randomly generated networks.

4.4.2 Investigation with Non-Cooperative Case

Verification of Analysis

The p_{co} obtained via analysis and simulations are compared in Figure 4.8. We see a close match between them, with a deviation of less than 5% in almost all single-hop scenarios, and less than 10% in almost all multi-hop scenarios. Particularly, the availability of cooperation is observed to be at a high level ($p_{co} > 0.7$ in most cases), which suggests that a large percentage of MCC problems would be avoided by exploiting DISH, and DISH is feasible to use in multi-channel MAC protocols. (**Finding 1**)

Specifically, Figure 4.8a and Figure 4.8c consistently show that, in both single-hop and multi-hop networks, p_{co} monotonically decreases as λ increases. The reasons are two folds. First, as traffic grows, each node spends more time on data channels for data transmission and reception, which reduces p_{ctrl} and hence the chance of overhearing control messages (p_{oh}), resulting in lower p_{co}. Second, as the control channel is the rendezvous to set up all communications, larger traffic intensifies the contention and introduces more interference to the control channel, which is hostile to messages overhearing and thus also reduces p_{co}.

Procedure 4.2 ATTEMPT-RTS

[Called by PKT-ARRIVAL or CHECK-QUEUE]

construct a set $\mathcal{F}$ of free channel indexes using channel usage table
if $\mathcal{F} \neq \phi$ **then**
  send `McRTS` with `CH`:=RANDOM($\mathcal{F}$)
else
  Timer $\leftarrow$ min(`until` $- now$)
  while $S_{ctrl} = FREE \wedge$ Timer not expired **do**
    wait {carrier sensing remains on}
  end while
  if Timer expired **then**
    call CHECK-QUEUE
  else
    call PASSIVE {receive a control message}
  end if
end if

Procedure 4.3 CHECK-QUEUE

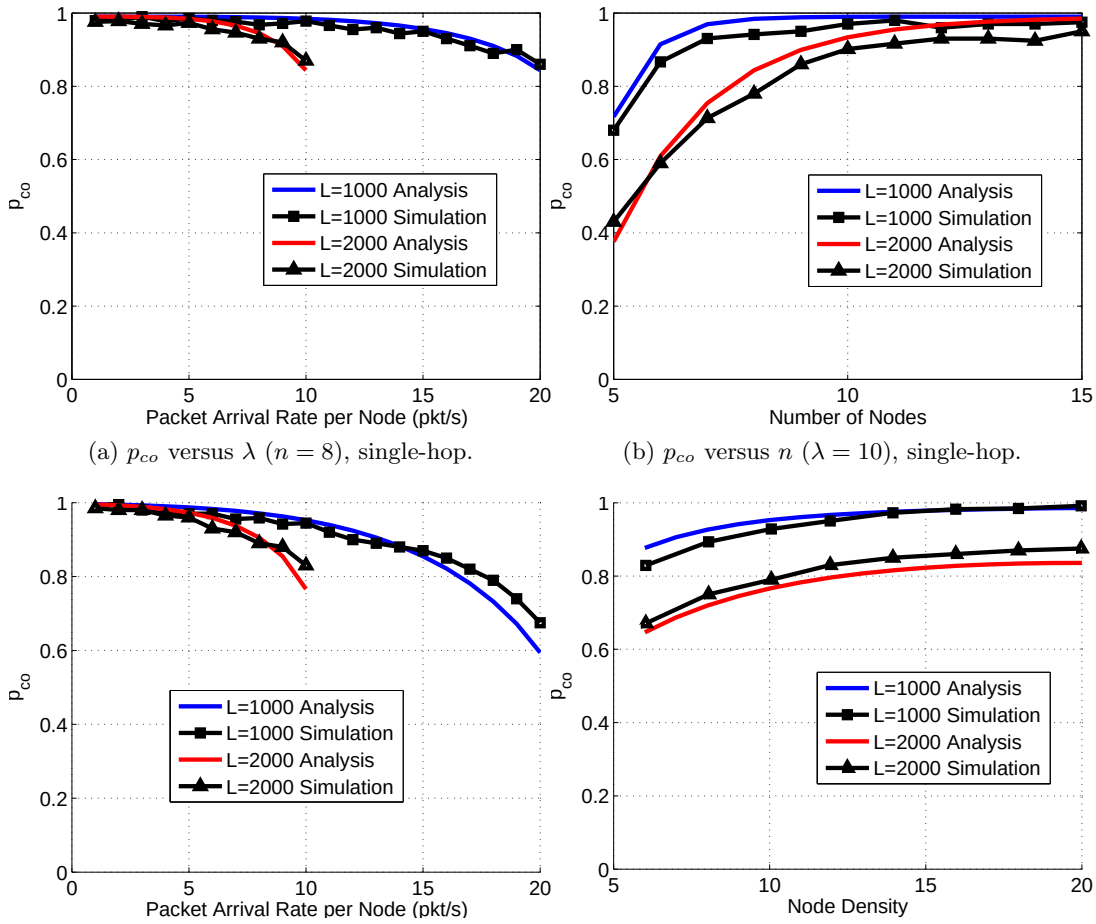
[Called when $S_{ctrl} = FREE \wedge S_{node} = IDLE$ changes from $FALSE$ to $TRUE$]

if $L_{queue} > 0$ **then**
  Timer $\leftarrow$ RANDOM$(0, 10b)$ {FAMA [75, 76]}
  while $S_{ctrl} = FREE \wedge$ Timer not expired **do**
    wait {carrier sensing remains on}
  end while
  if Timer expired **then**
    call ATTEMPT-RTS
  else
    call PASSIVE {receive a control message}
  end if
end if

Figure 4.8b and Figure 4.8d show that p_{co} monotonically increases as n increases, and is concave. The increase of p_{co} is because MCC problems are more likely to have cooperative nodes under a larger node population, while the deceleration of the increase is because more nodes also generate more interference to the control channel.

An important message conveyed by this observation is that, although a larger node density creates more MCC problems (e.g., more channel conflicts as data channels are more likely to be busy), it also boosts the availability of cooperation which avoids more MCC problems. This implies that the performance degradation can be mitigated. (**Finding 2**)

In both single-hop and multi-hop networks, a larger packet size L corresponds to a lower p_{co}. However, note that this is observed under the same *packet* arrival rate (pkt/s), which means actually a larger *bit* arrival rate for a larger L, and can be explained by the previous scenarios of p_{co} versus λ. Now if we consider the same *bit* arrival rate, by examining the two analysis curves in Figure 4.8c where we compare p_{co} with respect to the same $\lambda \cdot L$ product, e.g., $(\lambda = 5, L = 2000)$ versus $(\lambda = 10, L = 1000)$, and $(\lambda = 10, L = 2000)$ versus $(\lambda = 20, L = 1000)$, then we will see that a larger L corresponds to a *higher* p_{co}, which is contrary to the observation under the same *packet* arrival rate. The explanation is that, for a given bit arrival rate, increasing L reduces the number of packets and hence *fewer control channel handshakes* are required, thereby alleviating control channel interference. (**Finding 3**)



(a) p_{co} versus λ ($n = 8$), single-hop.

(b) p_{co} versus n ($\lambda = 10$), single-hop.

(c) p_{co} versus λ ($n = 10$), multi-hop.

(d) p_{co} versus n ($\lambda = 10$), multi-hop.

Figure 4.8: Impact of traffic load and node density, with different packet sizes. The value ranges of X axes are chosen such that the network is stable.

Dominating Impact Factor

The above results indicate that node density and traffic load affect the availability of cooperation in *opposite* ways. This section aims to find which one dominates over the other. In Figure 4.9, we plot the relationship of p_{co} versus λ and n, given $L = 1000$ and based on the *analytical* result for single-hop networks. We multiplicatively increase λ and n with the same factor (two), and find that, when increasing (λ, n) from (5,5) to (10,10), p_{co} keeps *increasing* from 0.865 to 0.999, and when increasing (λ, n) from (10,5) to (20,10), p_{co} keeps *increasing* from 0.724 to 0.943. Consistent results were also observed in other scenarios, i.e., $L = 2000$ in single-hop, and $L = 1000, 2000$ in multi-hop networks.

This investigation shows that n is the dominating factor over λ that determines the variation of p_{co}. This implies that DISH networks should have better scalability than non-DISH networks, since p_{co} increases when both traffic load and node density scale up. (**Finding 4**)

4.4.3 Investigation with Ideal DISH

For ideal DISH, we only present results in multi-hop networks, as the single-hop simulation results were similar and the discussion in this section applies to both sets of results.

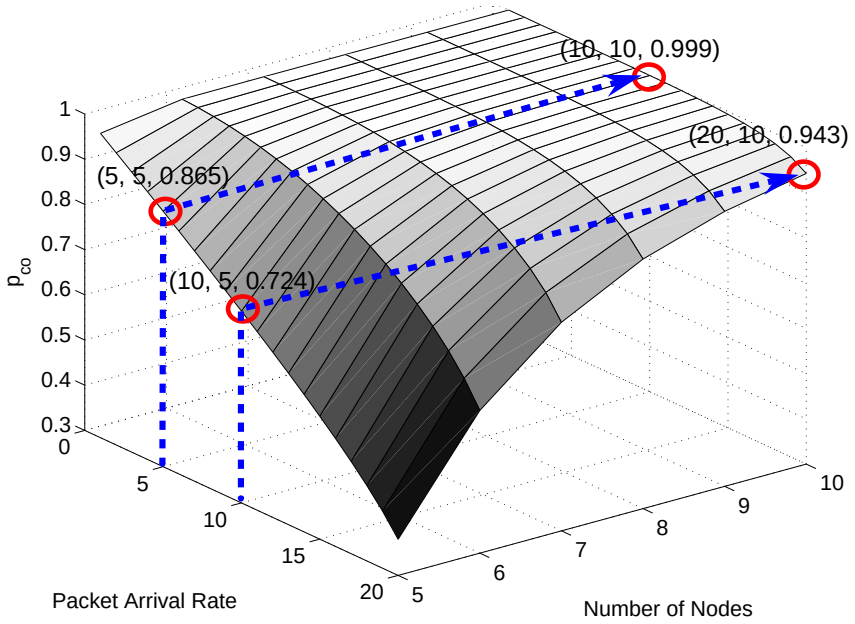


Figure 4.9: p_{co} versus λ and n. Each of the two arrows indicates a multiplicative increase of λ and n with the same factor (two).

Verification of Analysis

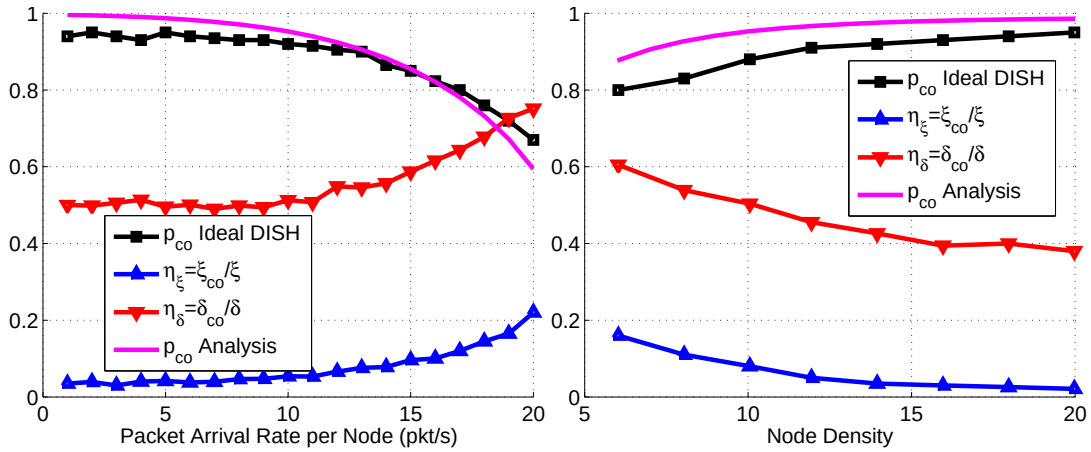
The results of comparison are shown in Figure 4.10, where p_{co} with ideal DISH well matches p_{co} of analysis. This confirms **Findings 1-3**, and we speculate the reasons to be as follows. With ideal DISH, a transmitter will be informed of a deaf terminal problem at times and thus back off for a fairly long time, which leads to *fewer* `McRTS` being sent. On the other hand, a node will also be informed of a channel conflict problem at times and thus re-select channel and retry shortly, which leads to *more* control messages being sent. Empirically, channel conflict problems occur more often, and hence there will be an overall *increase* of control messages being sent. This escalates interference and thus would lower down p_{co}. However, nodes will switch to data channels less frequently because cooperation suppresses a number of conflicting data channel usages. This makes nodes stay *longer* on the control channel and would elevate p_{co}. Consequently, p_{co} does not significantly change.

As **Finding 4** is obtained via analysis which, as shown in Figure 4.10, matches the simulations with ideal DISH, it is automatically confirmed.

Correlation between p_{co} and Performance

We investigate how p_{co} correlates to network performance — specifically, data channel collision rate ξ, packet delay δ, and aggregate throughput S. We consider both stable networks and saturated networks under multi-hop scenarios.

In stable networks, we measure (ξ, δ) and (ξ_{co}, δ_{co}) when without and with cooperation (ideal DISH), respectively. Then we compute $\eta_\xi = \xi_{co}/\xi$ and $\eta_\delta = \delta_{co}/\delta$ to compare to p_{co} with ideal DISH. The first set of results, by varying traffic load λ, is shown in Figure 4.10a. We observe that the two *ascending* and *convex* curves of η_ξ and η_δ approximately *reflect* the *descending* and *concave* curve of p_{co}, which hints at a *linear* or *near-linear* relationship between p_{co} and these two performance ratios. That is, $\eta_\xi + p_{co} \approx c_1$, $\eta_\delta + p_{co} \approx c_2$, where c_1 and c_2 are two constants. The second set of results, by varying node density n, is shown in Figure 4.10b. On the one hand, η_ξ and η_δ decreases as n increases, which is contrary to Figure 4.10a. This confirms our earlier observations: n is amicable whereas λ is hostile to p_{co} (the smaller η_ξ and η_δ, the better performance cooperation offers). On the other hand, the correlation between p_{co} and the performance ratios is found again: as p_{co} increases on a concave curve, it is reflected



(a) Impact of λ ($n = 10$). (b) Impact of n ($\lambda = 10$).

Figure 4.10: Investigating p_{co} with ideal DISH in stable networks. This includes (i) verification of analysis, and (ii) correlation between p_{co} and (η_ξ, η_δ) (ratio of data collision, ratio of packet delay). $L = 1000$ bytes. Each Y axis represents multiple metrics.

by η_ξ and η_δ which decrease on two convex curves.

In saturated networks, we vary node density n and measure aggregate throughput without and with cooperation (ideal DISH), as S and S_{co}, respectively. Then we compute $\eta_S = S/S_{co}$ (note that this definition is inverse to η_ξ and η_δ, in order for $\eta_S \in [0, 1]$) to compare to p_{co} with ideal DISH. The results are summarized in Figure 4.11. We see that (i) p_{co} grows with n, which conforms to Finding 2, and particularly, (ii) the declining and convex curve of η_S reflects the rising and concave curve of p_{co}, which is consistent with the observation in stable networks. In addition, the p_{co} here is lower than the p_{co} in stable networks. This is explained by our earlier result that higher traffic load suppresses p_{co}.

In summary, the experiments in stable networks and saturated networks both demonstrate a strong correlation (*linear* or *near-linear* mapping) between p_{co} and network performance ratio in terms of typical performance metrics. This may significantly simplify performance analysis for cooperative networks via bridging the *nonlinear* gap between network parameters and p_{co}, and also suggests that p_{co} be used as an appropriate performance indicator itself. (**Finding 5**)

Note that this does not imply that the delay, the channel collision rate, and the throughput are linear with respect to each other, because the above stated relationship is for the performance *ratios* between DISH and non-DISH networks.

The explanation to this linear or near-linear relationship should involve intricate network dynamics. We speculate that the rationale might be that (i) MCC problems are an essential performance *bottleneck* to multi-channel MAC performance, and (ii) p_{co} is equivalent to the *ratio* of MCC problems that can be avoided by DISH. In any case, we reckon that this observation may spur further studies and lead to more thought-provoking results.

4.4.4 Investigation with Real DISH

For the same reason as mentioned for ideal DISH, we present the results for multi-hop networks only.

Verification of Analysis

As shown in Figure 4.12, the simulation and analytical results still match, with deviation less than 15%. This is explained by two underlying factors. On the one hand, since real DISH actually sends cooperative messages, the control channel will have more interference which tends to diminish p_{co}. On the other hand, these cooperative messages inform transmitters or receivers of conflicting channel selections, which leads to a reduced number of channel switchings. Hence nodes will stay longer on the control channel and

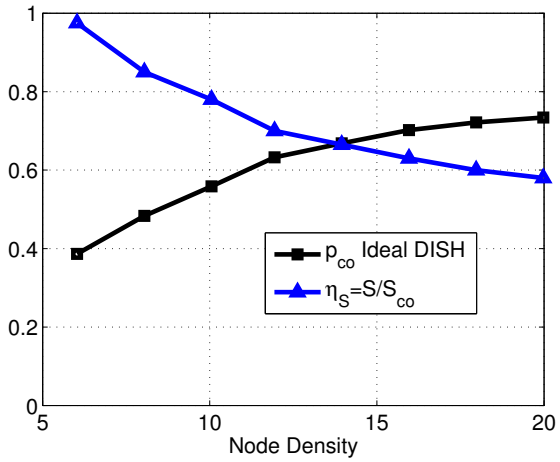
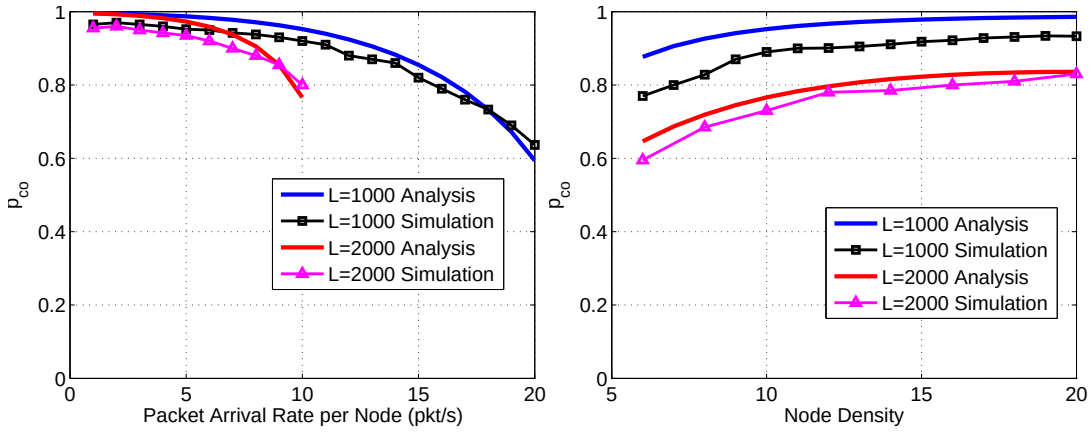


Figure 4.11: Investigating p_{co} with ideal DISH in saturated networks: correlation between p_{co} and η_S (throughput ratio). $L = 1000$ bytes. The Y axis represents multiple metrics.



(a) p_{co} versus λ ($n = 10$). (b) p_{co} versus n ($\lambda = 10$).

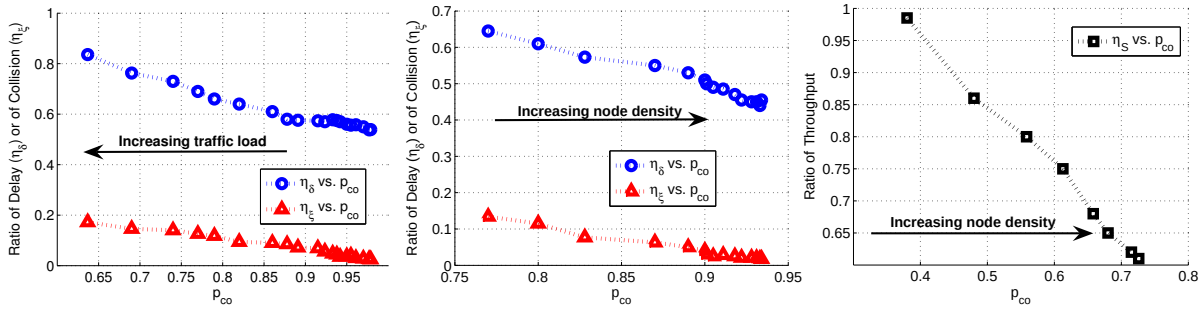
Figure 4.12: Verification of p_{co} with real DISH in multi-hop networks. $L = 1000$ bytes.

hence p_{co} tends to be elevated. Consequently, p_{co} is kept close to the analytical result. **Findings 1-4** are thus confirmed.

Correlation between p_{co} and Performance

We examine this issue using scatter plots which provide another point of view besides the direct representation in Section 4.4.3. These plots are given in Figure 4.13, where it is more apparent to see the near-linear relationship between p_{co} and $\eta_\xi, \eta_\delta, \eta_S$ (respectively). This confirms **Finding 5**.

An observation is that, while η_δ and η_S (the ratio of delay and throughput, respectively) in real DISH are slightly larger than those in ideal DISH, η_ξ (the ratio of data channel collision) is slightly smaller. Note that, by definition, the smaller these ratios are, the better the corresponding performance is (i.e., shorter delay, fewer collisions, and higher throughput, respectively). To explain this difference, first notice that the larger η_δ and η_S is simply because real DISH has to afford overhead for cooperation (physically send cooperative messages) while ideal DISH does not need to. Second, the smaller η_ξ, which counter-intuitively conveys fewer data channel collisions than ideal DISH, relates to two factors: (i) the transmissions of cooperative messages in real DISH lowers down the efficiency of cooperation than in ideal DISH, as explained before, and hence each use of a data channel in real DISH is slightly more likely



(a) η_δ and η_ξ versus p_{co}, varying λ ($n = 10$).

(b) η_δ and η_ξ versus p_{co}, varying n ($\lambda = 10$).

(c) η_S versus p_{co}.

Figure 4.13: Correlation between p_{co} and different performance metrics with real DISH in multi-hop networks.

to encounter collision, (ii) the cooperative messages suppress nearby nodes from *initiating* handshakes (via CSMA) and also interfere *ongoing* control channel handshakes, leading to fewer accomplished control channel handshakes per second, and hence fewer data channel usages than ideal DISH. The latter factor, according to the simulation results, outweigh the former factor, thereby explains the smaller η_ξ. In fact, these two factors also contribute to the longer delay and lower throughput in real DISH than in ideal DISH.

4.5 Channel Bandwidth Allocation

Our investigation shows that p_{co} is a meaningful performance indicator for DISH networks and captures several other critical performance metrics. In this section, we leverage p_{co} as an instrument and apply our analysis to solving an important issue in multi-channel operation, channel bandwidth allocation.

4.5.1 Problem Formulation

For a given amount of total channel bandwidth W, define a bandwidth allocation scheme $\mathcal{A}_m = (m, \sigma)$, where m is the number of data channels and $\sigma = w_c/w_d$, in which w_c is the bandwidth of the control channel and w_d is the bandwidth of a data channel. The objective is to obtain the optimal scheme $\mathcal{A}^*_m = (m, \sigma^*)$ which achieves the maximum p_{co} for a given m. We remark that:

- An implicit assumption of the problem formulation is that bandwidth is equally partitioned among all *data* channels.

- In the formulation, m is designated as an input parameter instead of a variable subject to optimization. This is because of the following: (i) in practice, the main consideration on choosing m is *system capacity* (the number of users a system can accommodate), (ii) if, otherwise, m is a variable subject to optimization, the formulation will be equivalent to $\mathcal{A} = (w_c, w_d)$ and its solution will be a *single* "universally optimal" m which generally does not fit into practical situations.

- There exists another bandwidth allocation problem, where data channel bandwidth w_d is fixed and only control channel bandwidth w_c can be adjusted (or vice versa). We do not investigate this problem because (i) from a practical perspective, radio frequency band is a regulated or highly limited resource and cannot be arbitrarily claimed, and (ii) even if the band can be arbitrarily claimed, then the solution becomes obvious — p_{co} will monotonically increase as w_c or w_d grows, as a consequence of using more resource.

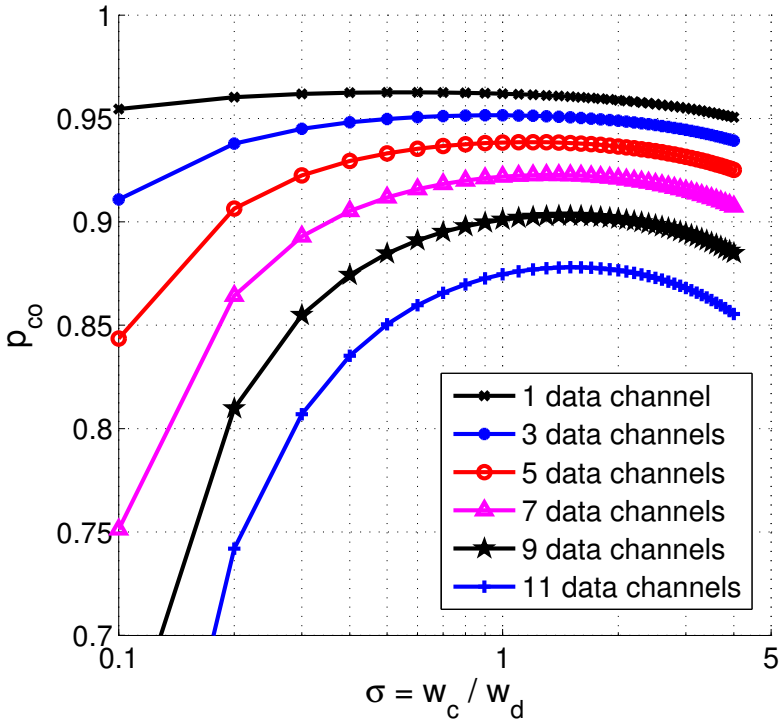


Figure 4.14: p_{co} versus σ under different m.

In the following, we solve the above optimization problem using the analytical results derived in Section 4.3. The feasibility of applying our analysis is based on the consistency between p_{co} of analysis and that of simulations across non-cooperative case, ideal DISH and real DISH, as shown in Section 4.4.

4.5.2 Solutions and Discussion

Denoting the size of a control packet and a data packet by l_c and L, respectively, we have[‡]

$$l_c = w_c b, \quad L = w_d T_d.$$

Combined with $W = w_c + m \cdot w_d$, we have

$$\frac{l_c}{b} + \frac{mL}{T_d} = W. \tag{4.16}$$

Then we rewrite σ as

$$\sigma = \frac{T_d l_c}{bL}. \tag{4.17}$$

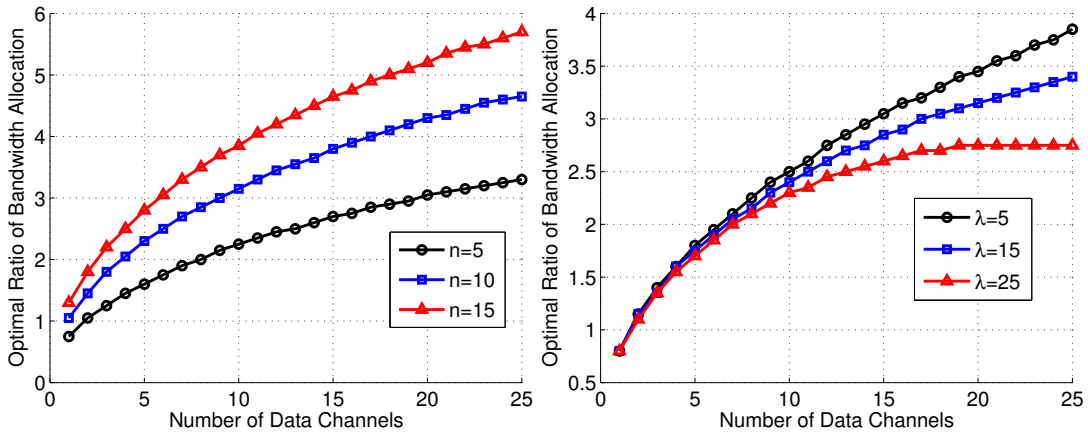
Next, for a given W and protocol-specified l_c and L, compute b and T_d using (4.16) and (4.17) for different combinations of m and σ. Then substitute each pair of b and T_d into the equations derived in Section 4.3 and calculate p_{co}. By this means, we will obtain a matrix of p_{co} which corresponds to different combinations of m and σ. Finally, comparing p_{co} for each m in order to find the maximum will obtain the optimal solution $\mathcal{A}^*_m = (m, \sigma^*)$ for a given m.

Using this method, we can demonstrate results given a set of parameters. Figure 4.14 is such a plot given the following parameters: $L = 2000$ bytes, $l_c = 34$ bytes (cf. Figure 4.6, plus PHY preamble and header), $W = 40$Mb/s, $n = 6$, $\lambda = 20$. We can see that p_{co} is concave and *not* monotonic with respect

[‡]We assume that ACK packet size is negligible compared to data packet size. Or alternatively, one can simply define L as the sum size of data and ACK packets.

to σ, and it reaches the maximum at a certain σ on each curve corresponding to a specific m. There are two counteractive factors attributing to this. First, as σ increases, the control channel is allocated more bandwidth, so the time needed for transmitting a control packet, and hence the vulnerable period of receiving a control packet, is being reduced. As such, the probability of successful overhearing will increase and hence p_{co} tends to be higher. Second, as W is fixed, increasing control channel bandwidth has to squeeze data channels simultaneously, which prolongs data packet transmission time and hence is to the effect of enlarging data packet size L. According to Section 4.4, a larger L leads to a lower p_{co}. (**Finding 6**)

In particular, we obtain $\mathcal{A}_1^* = (1, 0.55)$, $\mathcal{A}_3^* = (3, 0.95)$, $\mathcal{A}_5^* = (5, 1.15)$, $\mathcal{A}_7^* = (7, 1.35)$, $\mathcal{A}_9^* = (9, 1.45)$, $\mathcal{A}_{11}^* = (11, 1.5)$. This conveys the following message: when the number of channels is small, the control channel should be allocated less bandwidth than a data channel (i.e., $\sigma < 1$), whereas when there are more channels, the control channel should be allocated more bandwidth than a data channel (i.e., $\sigma > 1$). The rationale is that, as the number of channels increases, it becomes easier for a node to secure a data channel for data transmission, and thus fewer nodes will be waiting for free data channels on the control channel. This reduces the chances of having cooperative nodes. In order to counteract this effect, the control channel should be allocated more bandwidth to increase the probability of successful overhearing (by shortening the vulnerable periods of receiving control packets).



(a) Under different n ($\lambda = 10$). (b) Under different λ ($n = 6$).

Figure 4.15: σ^* versus m under different combinations of n and λ. $L = 1000$ bytes.

Then we investigate the relationship between σ^* (the optimal bandwidth allocation ratio) and m. In each set of computation, we use (4.16), (4.17) and the equation array derived in Section 4.3, compute the optimal ratio σ^* by search in each series of (σ, p_{co}) for each m. Also, a larger range of m (1..25) and a smaller step size (one) are used. We perform multiple sets of computation with different n and λ corresponding to different network scenarios.

Figure 4.15 presents these results. The first observation is that σ^* monotonically increases with m. This is consistent with the previous series of ($\mathcal{A}_1^*$, $\mathcal{A}_3^*$, ... $\mathcal{A}_{11}^*$). The second observation is that σ^* increases with n (Figure 4.15a) but decreases with λ (Figure 4.15b). This tells us that, to achieve a high availability of cooperation in a sparse network with heavy traffic, the control channel should be allocated much smaller bandwidth than in a dense network with light traffic. (**Finding 7**)

Figure 4.15 also shows that $\sigma^* > 1$ in most cases. This means that, for a DISH network to achieve larger p_{co}, it generally prefers larger bandwidth for the control channel than for *each* data channel. This is contrary to the prior approach of using a smaller frequency band for control ([77, 78]) or dividing total bandwidth equally among all channels (numerous studies) in non-cooperative multi-channel networks. (**Finding 8**)

4.6 Summary

DISH represents another dimension of exploiting cooperative diversity, in addition to data relaying, as a control-plane cooperative approach. This chapter gives the first theoretical treatment of this notion by addressing the availability of cooperation via a metric p_{co}. Instead of directly analyzing throughput which is an open problem in general and is much more complicated in a multi-channel DISH context, we first analyze p_{co} and then correlate it with other performance metrics including throughput, delay and collision rate. We conduct analysis in a multi-hop multi-channel wireless network, and investigate p_{co} with three different contexts of DISH: non-cooperative case, ideal DISH, and real DISH. Our analysis accurately captures the interaction between p_{co} and underlying parameters, and discloses important findings with respect to network dynamics. Our investigation reveals a near-linear relationship between p_{co} and typical performance metrics, which may greatly facilitate performance analysis for cooperative networks and also suggests p_{co} to be a useful performance indicator itself. Finally, our application of the analysis to solving a practical channel bandwidth allocation problem provides guidelines on bandwidth allocation for DISH networks.

Our study yields eight findings as listed in Section 4.1, which serve for different purposes. Findings 1 and 2 back up the feasibility and benefit of DISH. Findings 3 and 4 give hints on improving system performance by adjusting packet size, node density and traffic load. Finding 5 demonstrates the significance of the metric p_{co}. Findings 7 and 8 suggest ways of performance improvement from a system design perspective.

In the case of *mobile* ad hoc networks, a node v cannot become a cooperative node with respect to nodes x and y who create an MCC problem, if v fails to decode at least one control message from x and y due to its mobility . However, in most cases when the mobility level is not high, node v will still remain in the intersection region of x and y during the period x and y are sending their control messages, and thus is still able to cooperate. Also, another effect can compensate for the missing of cooperative nodes: a node, although not in the intersection range of x and y, first hears x's control message and then moves into y's range and hears y's control message. In this case, it can also identify the MCC problem and become a cooperative node. Our prior simulation on a real DISH protocol, CAM-MAC. with the random waypoint model showed that, when the moving speed is uniformly distributed in (0, 10] m/s (transmission range is 250m), the throughput decreases by only 3–8% in different scenarios.

This work attempts to encourage an insightful understanding of DISH, and based on our findings, it demonstrates that p_{co} is a useful metric capable of characterizing the performance of DISH networks, and also bears significant implications. We contend that DISH, as a new cooperative approach, is practical enough to be a part of future cooperative communication networks.

Chapter 5

Energy-Efficient DISH Strategies

5.1 Introduction

We have applied DISH to multi-channel MAC protocol design and proposed a protocol called CAM-MAC. It uses a single radio and is fully asynchronous, and significantly enhances system throughput. However, we suspect that the benefit from DISH comes at the cost of elevated energy consumption based on the following. First, nodes in such a cooperative protocol send (extra) cooperative messages for *information sharing*. Second, nodes have to stay awake during idle periods to receive control messages for *information gathering* (a prerequisite of information sharing).

For a quantitative understanding, we designed three protocols and evaluated them in terms of both throughput and energy consumption via simulations. The protocols are:

(a) *Cooperative*: a DISH protocol slightly improved from CAM-MAC [71] (details given in Section 5.2).

(b) *Autonomous*: the information sharing component is removed from *Cooperative*, i.e., neighbors do not send cooperative messages, while keeping the same handshake.

(c) *Autonomous-PSM*: the information gathering component is removed from *Autonomous*, and an ideal power saving mode (PSM) is used, where nodes sleep (switch off radio) whenever idle.

The single-hop simulation results show that *Cooperative* achieves 2.65 times throughput of *Autonomous* but consumes 2.94 times energy of *Autonomous-PSM*. This surged energy drainage presents a serious obstacle to putting DISH into practice.

To solve this problem, we realized that a major challenge is to cope with two contradicting factors: for energy conservation, nodes should be kept in sleep mode as much as possible during idle duty cycles, but for cooperation, nodes have to perform information gathering and sharing, which precisely happen during idle duty cycles.

In this study , we identify two candidate energy-efficient strategies, *in-situ energy conscious cooperation* and *altruistic cooperation*. The first strategy allows existing nodes to rotate the responsibility of cooperation such that nodes without the responsibility can sleep during idle periods. The second strategy deploys additional *solely cooperative nodes*, called altruists, to take over the responsibility of cooperation so that all existing nodes can sleep during idle periods. Applying these two strategies to *Cooperative* results in two new protocols:

(d) *Genie In-Situ*: rotating the responsibility of cooperation among nodes in an optimal manner (details given in Section 5.2); it gives a performance upper bound for in-situ energy conscious cooperation.

(e) *Altruistic*: introducing altruists whose only role is information gathering and sharing; existing nodes have the only role of carrying traffic. Altruists always stay awake.

We carry out a comparative study using the above five protocols. The highlights of our study are summarized below.

Table 5.1: Protocols and Role Assignments

Protocols	**Traffic**	**Gather**	**Share**	**PSM**
Autonomous	peer	peer	×	×
Autonomous-PSM	peer	×	×	✓
Cooperative	peer	peer	peer	×
Genie In-Situ	peer	peer	peer	✓
Altruistic	peer	altruist	altruist	peer

- *Optimal node deployment*: Altruist deployment is the first issue. We show that the optimal altruist density in random networks is near 1.31. (Section 5.3.2)
- *Cost efficiency*: This is a critical issue because altruistic cooperation uses additional nodes. However, we show that the increased cost substantially pays off — cost efficiency is more than doubled. (Section 5.4)
- *Throughput-energy trade-off*: We show that altruistic cooperation achieves the lowest energy consumption (20–60% of *Cooperative*) and comparable throughput to *Cooperative* simultaneously. (Section 5.5)
- *Bit-Meter-Price ratio*: We propose a metric called bit-meter-price ratio (BMP) which allows for a fair comparison of *cost efficiency* across different protocols and different networks. (Section 5.4.1)
- *Generality*: Instead of proposing a specific power-saving *protocol* like many other studies do, we propose *strategies* which, particularly altruistic cooperation, can be generally applied to perhaps all control-plane cooperative protocols.

This work is the first to address energy efficiency for control-plane cooperation. Our investigation supports that altruistic cooperation is the right strategy in exploring DISH. In-situ energy conscious cooperation, on the other hand, is an appropriate choice only (i) for applications with few nodes or light traffic, (ii) for applications that preclude using additional nodes, or (iii) if it can perform closely to the upper bound that we define. In addition, both strategies do not require multiple transceivers nor time synchronization.

5.2 System Model

5.2.1 Protocol Taxonomy and Design

In general, a MAC protocol has three roles that a node can take: (a) carry data traffic, (b) gather control information, and (c) share control information. Accordingly, we categorize five classes of MAC protocols based on feasible role combinations and the choices of using PSM, and design an example multi-channel MAC protocol for each category. They are listed in Table 5.1, where a *peer* refers to an existing node as opposed to an (additional) altruist.

Our assumptions are as follows. We consider a static ad hoc network in which each node is equipped with a single half-duplex transceiver that can dynamically switch among multiple orthogonal frequency channels but can only use one at a time. One channel is designated as a control channel and the others as data channels. For each data packet, a transmitter and a receiver perform a *control channel handshake* (like 802.11 RTS/CTS) to set up communication and, upon success, perform a *data channel handshake* (like DATA/ACK). The control channel handshake carries channel usage information (e.g., "who will use which channel for how long") in order to reserve a data channel, and hence a node may overhear this information and cache in its knowledge base. This knowledge base can be used for the node itself (selecting one from all free data channels randomly) and also for cooperation (identifying an

MCC problem which triggers cooperation). During idle periods, a node either senses the control channel or sleeps if allowed by PSM.

The five protocols are described below.

Cooperative

In this protocol, all nodes (peers) are responsible for cooperation. For the detailed protocol description, the reader is referred to Section 3.3.1.

Autonomous

Information sharing is removed from *Cooperative*, i.e., neighbors do not send cooperative messages, while keeping the same handshake.

Autonomous-PSM

Information gathering is removed from *Autonomous*, and an ideal power saving mode (PSM) is used, where nodes sleep whenever idle.*

Genie In-Situ

This protocol establishes an upper bound for in-situ energy conscious cooperation by rotating the responsibility of cooperation in *Cooperative* optimally. We do not use a real in-situ protocol because its complexity would negate any possible performance gain, as will be elaborated in Section 5.2.2. Rather, we use an upper bound as a benchmark for other protocols to compare against. In this genie protocol, nodes gather control information during idle periods *without consuming any energy*, and whenever an MCC problem is identified by multiple cooperative nodes, the node with the most helpful information† will send a cooperative message (energy consumed by this particular node will be counted) while the others will keep silent in sleep mode.

Altruistic

This protocol is the same as *Cooperative* but altruists are deployed, who exclusively perform information gathering and sharing and always stay awake. Existing nodes only carry traffic, behaving the same as in *Autonomous-PSM*.

Remark: *Autonomous* and *Autonomous-PSM* are formed by removing the cooperation components from *Cooperative*. This allows us, via a comparison against them, to see the impact of the cooperation components on energy consumption. In particular, *Autonomous-PSM* would consume very low energy and thus establishes itself as a near lower bound for the other protocols. However, they are not used as representatives of general non-cooperative protocols (such as MMAC, SSCH and AMCP) which represent a large variety of design paradigms. To be fair, each non-cooperative protocol embodying a specific paradigm should be allowed to devise its own strategy for energy efficiency. This paper focuses on DISH-based cooperative protocols only.

5.2.2 Qualitative Analysis

The qualitative properties given in this section will provide us with insights into the two energy-efficient strategies.

*A node is considered idle if it is not transmitting or receiving for its own sake, i.e., it is idly waiting, backing off, or overhearing messages not addressed to it.

†When a channel selected by, say, node u conflicts with multiple ongoing communications in the neighborhood, the communication that ends *last* carries the most helpful channel usage information because it tells the minimum necessary time for u to backoff.

In-Situ Energy Conscious Cooperation

Existing nodes rotate the responsibility of cooperation such that nodes without the responsibility can sleep during idle periods. There are two approaches to do this. In the *probabilistic approach*, each node decides to cooperate based on a fixed or varying probability, which is similar to probabilistic flooding [79–81] or probabilistic routing [82–84] in ad hoc networks, and some cluster-head rotating algorithms such as LEACH [85] and HEED [86] in WSNs. The other is the *voting approach*, where nodes periodically vote or elect some of them to cooperate, which is similar to prior work such as VCA [87], GAF [88], PANEL [89], and Span [90].

An obvious advantage of this in-situ strategy is no need of additional nodes. But on the other hand, it requires a runtime mechanism to determine the optimal (or near-optimal) cooperating probability or to vote the optimal (or near-optimal) cooperative nodes. This is difficult and will lead to high complexity explained as follows.

First, this runtime mechanism must be (i) distributed due to the lack of a central server, (ii) fair so as to balance energy consumption over all nodes, and (iii) adaptive to network dynamics (traffic, density, etc.).

Second, rotating the cooperative role would rely on message broadcast like in [79–81, 84, 87, 90], but broadcast is very *unreliable* in a multi-channel environment [32] because a broadcasting can only reach a *subset* of neighboring nodes. Solving this is shown by [32] to be non-trivial.

Third, it might be possible to avoid or reduce broadcast using geographic information, as in [82, 83, 88, 89]. However, this needs either the expensive GPS support or a distributed localization algorithm such as [91, 92] which adds *extra* overhead and complexity.

Fourth, the information acquisition process for rotating the responsibility of cooperation is more resource-consuming than usual neighbor discovery processes, because (i) cooperation needs both one-hop and two-hop neighbor information [71, 93], and (ii) the usual neighbor discovery only needs static information such as neighbor IDs or the number of neighbors, whereas this acquisition process needs both static and dynamic information (e.g., residual energy, traffic load, and link reliability, as in [84, 85, 90]). Consequently, there will be more frequent message exchanges and hence higher consumption of bandwidth and energy.

Finally, integrating this mechanism with an existing cooperative protocol involves significant modifications.

Therefore, the complexity, overhead, and unreliability of in-situ energy conscious cooperation would negate any possible performance gain it could achieve. Nonetheless, we establish a performance benchmark for this strategy using a *Genie In-Situ* protocol for other protocols to compare against.

Altruistic Cooperation

This strategy deploys altruists to take over the role of cooperation and always stay awake. Existing nodes (peers) only carry traffic and can sleep during idle periods. Apparently, the drawback is the need of additional nodes. However, it is rewarded by the following advantages.

First, it does not introduce any runtime mechanism and hence zero runtime overhead.

Second, its underlying cooperative protocol does not need redesign at all; implementing the strategy is very simple: one only needs to add a *constant boolean* flag in the protocol source code to differentiate the code for cooperation and the code for carrying traffic. This is our true experience in coding the simulation and the testbed.

Third, the multi-channel broadcast problem no longer exists because altruists always stay on the control channel and their cooperative messages only targets at those nodes on the control channel.

Fourth, this strategy is robust to network dynamics (node density, traffic load, residual energy, etc.); altruists do not need to adjust any parameter for circumstance changes. Even their deployment, as will be shown in Section 5.3, is also independent of the circumstances.

Fifth, altruists provide cooperation in a *guaranteed* rather than an *opportunistic* manner as of the in-situ strategy, as they are always available.

Finally, by this strategy, peers are nearly free to choose legacy power saving mechanisms (such as those reviewed in Section 2.5) because their only role is carrying traffic as in traditional networks.

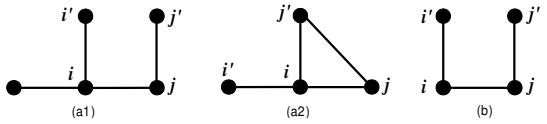


Figure 5.1: An illustration of Proposition 1. Subfigures (a1) and (a2) correspond to condition (a), subfigure (b) corresponds to condition (b), and the edges represent neighboring relationships.

5.2.3 Issues

We also carry out a quantitative study on the five protocols defined in Section 5.2.1, with respect to three critical issues:

(a) *Optimal node deployment* (foundation): finding the optimal altruist deployment scheme. (Section 5.3.2)

(b) *Cost efficiency* (key concern): examining if the increased cost due to additional nodes pays off. (Section 5.4)

(c) *Throughput-energy trade-off* (zooming-in): evaluating throughput and energy consumption specifically. (Section 5.5)

5.3 Optimal Node Deployment

5.3.1 Cooperation Coverage

This section establishes preliminaries for subsequent analysis and discussions.

Definition 5.1. An *unsafe pair* (UP) is a pair of peers that can create MCC problems between them. A *covered unsafe pair* (CUP) is an UP that both of the peers are covered by (i.e., within the radio range of) at least one common altruist.

The necessary and sufficient condition of forming an UP is given below.

Proposition 5.1. *In an undirected graph where vertices represent peers and edges represent peers' neighboring relationships, let d_i be the degree of an arbitrary peer i. If PSM is not used, two adjacent peers i and j form an UP if and only if:*

(a) $d_i \geq 2$, $d_j \geq 2$, and $d_i = d_j = 2$ does not hold, or

(b) $d_i = d_j = 2$, and i and j are not on the same three-cycle (i.e., triangle).

If PSM is used (peers sleep when idle), the above condition remains unchanged for the channel conflict problem, but changes to the following for the deaf terminal problem:
$d_i \geq 1$, $d_j \geq 1$, and $d_i = d_j = 1$ does not hold.

Proof. First consider the case without PSM.
Sufficiency: The condition is equivalent to that i and j can form two independent communication pairs, say p_i and p_j. For example, in Figure 5.1, p_i and p_j are (i, i') and (j, j'). Suppose p_i switches to a data channel ch_i when p_j is on a data channel ch_j, then the channel usage of ch_i is unknown to j. Therefore, after p_j switches back to the control channel and is setting up a new communication before p_i finishes communication on ch_i, (i) a channel conflict problem is caused if p_j selects channel ch_i to use, or (ii) a deaf terminal problem is caused if j initiates communication with i.

Necessity: Equivalently, we prove that i and j does not form an UP if d_i (or d_j) is 1 or i and j are on the same three-cycle. First, channel conflicts are not possible because only one communication pair can be formed. Second, deaf terminal problems are also not possible because, whenever the communication pair is performing a control channel handshake, the third node will be aware of it since, by our assumption, a node always listens to the control channel when idle (if without PSM).

The case with PSM is clear based on the above. Note that a receiver that is asleep (instead of communicating on a data channel) is not a deaf terminal by Definition 1. □

Definition 5.2. *Cooperation coverage* is the ratio between the number of CUPs and the number of UPs in a network. A network achieves *full cooperation coverage* if this ratio is 100%.

Proposition 5.2. *Consider networks using altruistic cooperation. Full cooperation coverage is*

(a) necessary for a multi-hop network, and

(b) necessary and sufficient for a single-hop network

to be free of MCC problems.

Proof. Necessity: By Definition 5.1, an UP cannot avoid MCC problems on its own, and hence it has to rely on external help, i.e., to become a CUP, to achieve this. As such, full cooperation coverage is necessary for the entire network to be free of MCC problems. This holds irrespective of single-hop or multi-hop networks.

Sufficiency: In a single-hop network, one altruist is enough to achieve full cooperation coverage. Since no more than one control channel handshake can be successful at any time, every MCC problem will be identified, and hence be prevented, by the altruist(s) via information sharing (collision may occur if there are more than one altruist, but the proposition does not change because the collision still indicates an MCC problem, as elaborated in the paper). □

5.3.2 Random Deployment Problem

Problem Statement: Consider an infinite random network with peers distributed according to a two-dimensional Poisson point process with density ρ_{peer}. The objective is to determine the density of altruists, ρ_{alt}, to guarantee a cooperation coverage of p_{cov} (say 90%).

Theorem 5.3. *The solution to Random Deployment Problem is given by*

$$\rho_{alt} > -\frac{\ln(1-p_{cov})}{(\frac{2\pi}{3}-\frac{\sqrt{3}}{2})r^2}. \tag{5.1}$$

Proof. Denote by p_{ij}^{cov} the probability that an arbitrary UP (i,j) is covered (i.e., is a CUP). By Definition 5.1, p_{ij}^{cov} is equivalent to the probability that there is at least one altruist in the common radio range of i and j, which is given by

$$p_{ij}^{cov} = 1 - e^{-\rho_{alt}A_{ij}}, \tag{5.2}$$

where A_{ij} is the intersected area of i and j's radio ranges, and can be proven to be

$$A_{ij} = 2r^2\theta - r^2\sin 2\theta, \tag{5.3}$$

where $\theta = \arccos\frac{d}{2r}$, d is the Euclidean distance between i and j, and r is the radio range.

The objective is equivalent to guaranteeing $p_{ij}^{cov} > p_{cov}$ for all UPs (i,j), which translates to

$$\min_{(i,j)} p_{ij}^{cov} > p_{cov}. \tag{5.4}$$

According to (5.2), p_{ij}^{cov} is a monotonically increasing function of A_{ij}, and hence is minimized by minimizing A_{ij}. To minimize A_{ij}, consider the minimization domain, namely all UPs. According to Theorem 5.1, forming an UP does not depend on the distance d between two neighbors, and hence $d \in [0, r]$. In addition, according to (5.3), A_{ij} is a monotonically decreasing function of d. Therefore, A_{ij} is minimized at $d = r$:

$$\min_{(i,j)} A_{ij} = A_{ij}|_{d=r} = (\frac{2\pi}{3}-\frac{\sqrt{3}}{2})r^2,$$

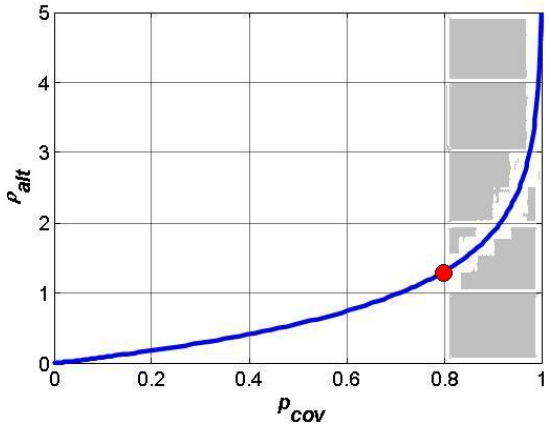


Figure 5.2: ρ_{alt} versus p_{cov} according to Theorem 5.3. Beyond the point (80%, 1.31), ρ_{alt} increases dramatically.

Table 5.2: Some Discrete Values of ρ_{alt} versus p_{cov}

p_{cov}	50%	60%	70%	80%	90%	95%	99%
$\rho_{alt} >$	0.56	0.75	0.98	1.31	1.87	2.44	3.75

and thus (5.4) resolves to

$$\begin{aligned}\min_{(i,j)} p_{ij}^{cov} &= 1 - \exp(-\rho_{alt} \cdot \min_{(i,j)} A_{ij}) \\ &= 1 - \exp[-\rho_{alt} \cdot (\frac{2\pi}{3} - \frac{\sqrt{3}}{2})r^2] \\ &> p_{cov},\end{aligned}$$

which can be reduced to Inequality (5.1). □

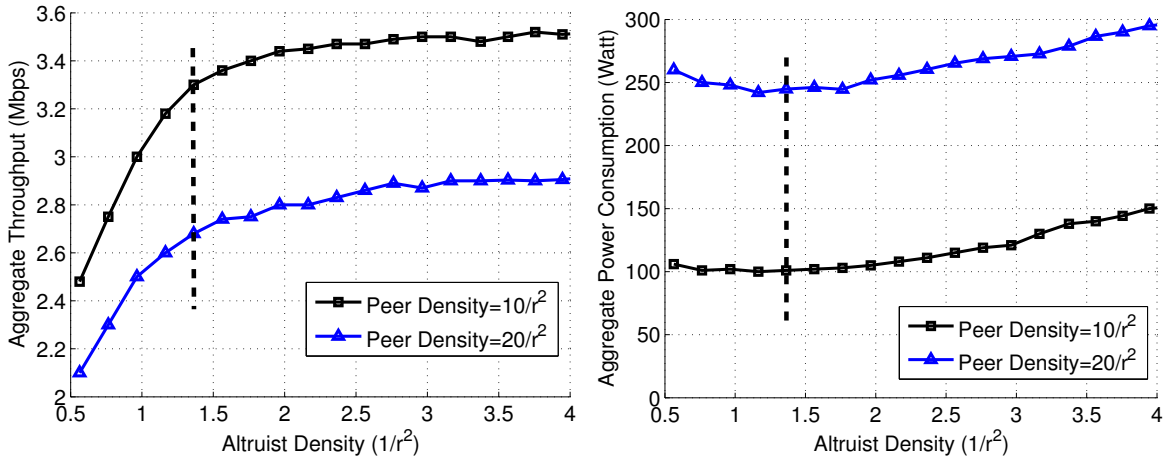
Remark: Theorem 5.3 gives the lower bound to the altruist density that guarantees a cooperation coverage of p_{cov}. Note that ρ_{peer} does not appear, implying that altruist deployment is independent of peer density. This significantly simplifies altruist deployment, since in many practical cases, the number of peers is unknown or varies. In addition, Theorem 5.3 tells that, in order to achieve full cooperation coverage, ρ_{alt} goes to infinity in multi-hop networks, while it is obvious to see that one altruist is sufficient for single-hop networks (will be shown in later simulations).

We plot the (ρ_{alt}, p_{cov}) relationship in Figure 5.2 and enumerate a series of specific values in Table 5.2. We see that ρ_{alt} dramatically increases beyond the point ($p_{cov} = 80\%$, $\rho_{alt} = 1.31$). This motivates us to conduct simulation to investigate the performance of altruistic cooperation particularly around this point.

Simulation Setup

To compute power consumption, we conducted a survey on commercial wireless cards. According to [94] (and with simple calculation), an IEEE 802.11 WaveLAN PC card consumes 1327/967/843/66 mW in TX/RX/IDLE/SLEEP state for the 2Mbps category, and 1346/901/741/48 mW for the 11Mbps category. A Cisco Aironet 350 series WiFi card [95] consumes 2250/1350/75 mW in TX/RX/SLEEP state. Other models including Intel Pro 2011, 3Com xJack, Compaq WL1000, and Siemens SS1021, all have the similar ratio. Therefore, we use a ratio close to the average of all the above, namely 25/18/15/1 ×50mW in TX/RX/IDLE/SLEEP state, to compute power consumption.

We randomly place nodes in a plane area. The radio transmission range is 250m and the interference range is 500m. The capture threshold is 6dB. In single-hop scenarios, the network area is 100m×100m and peers form disjoint source-destination pairs (i.e., flows). In multi-hop scenarios, the terrain is



(a) Aggregate throughput.

(b) Aggregate power consumption.

Figure 5.3: Finding the optimal altruist density in terms of both.

1500m×1500m and n peers form n non-disjoint flows randomly (each peer is the source of one flow and the destination of another flow). Shortest path routing is used.

There is one control channel and five data channels with bandwidth 1Mb/s each. Each source generates data packets according to a Poisson point process. Data payload size is 2KB. PLCP header is 6 bytes, PLCP preamble (short) is 9 bytes. The cooperative collision avoidance period is $35\mu s$. Channel switching delay is ignored because (i) it is common to all the protocols in this comparative study, (ii) it is about 80μs [29] and only amounts to the transmission time of 10 bytes on a 1Mb/s channel, and (iii) each data packet requires only two switchings (control$\xrightarrow{1}$data$\xrightarrow{2}$control), unlike channel hopping schemes [27–29] switching channels more frequently.

We terminate each simulation when a total of 100,000 data packets are sent over the network, and all results are averaged over 15 randomly generated networks.

Simulations

We conduct multi-hop network simulations for *Altruistic* by varying ρ_{alt} from $0.56/r^2$ to slightly more than $3.75/r^2$, which corresponds to varying p_{cov} from 50% to more than 99%. Peer densities ρ_{peer} of $10/r^2$ and $20/r^2$ are used,[‡] and each peer generates traffic at 25kb/s. The results are shown in Figure 5.3, where we see that (i) the throughput reaches a knee point at $\rho_{alt} = 1.3$–$2/r^2$ (Figure 5.3a) where the throughput levels off, (ii) the power consumption achieves the minimum also at $\rho_{alt} = 1.3$–$2/r^2$ (Figure 5.3b), and (iii) these two observations are irrespective of the value of ρ_{peer}.

These observations suggest that a judicious choice of ρ_{alt} is within the range of 1.3–$2/r^2$, and also confirm the independence between ρ_{alt} and ρ_{peer} shown by analysis. In this paper, we choose $\rho_{alt} = 1.31/r^2$ as a near-optimum solution, which corresponds to $p_{cov} = 80\%$.

In Figure 5.3a, the leveling off of throughput is because of the following. When ρ_{alt} is low, adding altruists convert many (uncovered) UPs into CUPs, thereby resolving many MCC problems and leading to a sharp increase of throughput. As ρ_{alt} continues to increase, more and more UPs are *redundantly* covered, and thus adding altruists becomes less efficient in converting UPs into CUPs, which slows down the throughput growth. Note that this also conforms to our analysis in (5.1), where ρ_{alt} increases exponentially as $p_{cov} \in (0, 1)$ increases.

In Figure 5.3b, the convexity of power consumption is due to two factors. On the one hand, adding altruists contributes a near-linear increase of aggregate power consumption to the network. On the other hand, more altruists increase cooperation coverage and prevent more MCC problems, which cuts down packet retransmissions and thus saves energy.

[‡]In addition to $10/r^2$ and $20/r^2$, the peer densities $5/r^2$ and $30/r^2$ were also simulated and yielded consistent results.

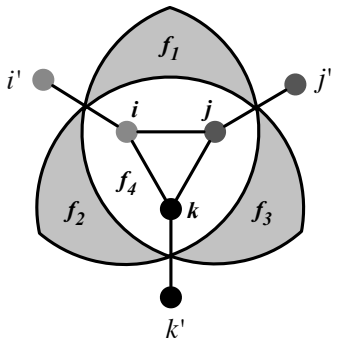


Figure 5.4: An illustration of Theorem 5.4. The edges represent neighboring relationships and the arcs represent radio ranges.

5.3.3 Arbitrary Deployment Problem

Problem Statement: Consider a network with peers forming a given topology on a plane. The objective is to determine the minimum number and the locations of altruists such that full cooperation coverage is achieved.

Theorem 5.4. *Arbitrary Deployment Problem is NP-hard.*

Proof. **Step 1: Identify UPs**

This step is to obtain a set U of all the UPs in the network by identifying UPs according to Proposition 5.1. As an example, see a six-node network shown in Figure 5.4. There are three UPs, that is, $U = \{(i,j),(j,k),(i,k)\}$.[§]

Step 2: Construct Orphanage Set

This step is to construct a set of all orphanages $\mathcal{H} = \{H_i | i = 1, 2, ...p\}$ in the network. The definition of orphanage depends on a notion called face.

Definition 5.3. A *face* is a region bounded by the (circular) radio boundaries of the peers that form UPs (there is no boundary inside a face). We say that *a face covers an UP*, if an altruist on any point of this face covers this UP.

For example, in Figure 5.4, i, j and k are all the peers that form UPs, f_1, f_2, f_3 and f_4 are all the faces, where, e.g., f_1 covers UP (i,j). Note that $f_1 \cup f_4$ is not a face.

Definition 5.4. An *orphanage* is the maximum set of UPs covered by a face. Rigorously, an orphanage H is a set of UPs ($H \subseteq U$) covered by a face f_H, and $\forall u \in U \setminus H$, u is not covered by f_H.

For example, in Figure 5.4, $H_1 = \{(i,j)\}$ and $H_4 = \{(i,j),(j,k),(i,k)\}$ are two orphanages covered by faces f_1 and f_4, respectively. But $H_4' = \{(i,j),(i,k)\}$ is not an orphanage. There are totally four orphanages.

By definition, there is a one-to-one mapping between each orphanage and its covering face. Thus, finding all the orphanages in a network can be done by finding all the faces that covers at least one UP. This problem is equivalent to the target coverage problem [96] in sensor networks, and is shown by [97] that the number of such faces is bounded by $|U|(|U|-1)+2$ and these faces can be found in time $O(|U|^3)$ by simply finding all the intersecting points of the circles (e.g., there are six such points in Figure 5.4).

Step 3: Formulate Problem

With U and $\mathcal{H}$, two problems can be posed:

(a) Decision problem: given U, $\mathcal{H}$ and an integer k, determine whether a subset $\mathcal{C} = \{H_i | i = 1, 2, ...q\} \subseteq \mathcal{H}$ exists, such that $\bigcup_{i=1}^{q} H_i = U$ and $q \leq k$.

[§]When using PSM *and* considering deaf terminal problems, $(i',i),(j',j),(k',k)$ are also UPs. We leave out this special case for a concise and more illustrative example.

(b) Optimization problem: given U and $\mathcal{H}$, minimize $k = |\mathcal{C}|$ over all possible $\mathcal{C} = \{H_i | i = 1, 2, ...q\} \subseteq \mathcal{H}$, subject to $\bigcup_{i=1}^{q} H_i = U$.

Since each orphanage $H_i \in \mathcal{H}$ corresponds to a unique face containing an altruist, q ($q \leq p$) is the minimum number of altruists that achieve full cooperation coverage.

The above two problems are the variants of the set cover problem defined by Karp [98]; the decision problem is NP-complete and the optimization problem is NP-hard.

□

Remark: In the proof, we have converted this problem into the classic *set cover problem* (Karp [98]) well studied in the literature, and there exist a number of greedy algorithms for finding the approximate solutions to the problem [99]. Particularly in our deployment case, these algorithms can be executed offline and hence do not introduce any runtime overhead. Also, a lower bound to the approximation ratio that such greedy algorithms can achieve in polynomial time was established recently by Alon et al. [100] to be $c \cdot \ln n$, where c is a constant and n is the number of elements to cover, i.e., UPs in our case.

A plausible thought is that, if the area is human-accessible, we can carefully deploy altruists such that the *entire region* is covered, thereby achieving full cooperation coverage irrespective of the peer topology. If this is true, we will be able to show that the minimum number of altruists needed to cover a $w \times h$ rectangular area is $\lceil w/\sqrt{2}r \rceil \cdot \lceil h/\sqrt{2}r \rceil$. However, this argument does not hold because covering an entire region does not guarantee covering every (unsafe) *pair* of neighbors by a *common* altruist.

5.4 Cost Efficiency

5.4.1 Bit-Meter-Price Ratio

We define a metric called BMP to evaluate cost efficiency:

$$BMP \triangleq \frac{\overrightarrow{F} \cdot \overrightarrow{D} \cdot b_0}{(N_p + N_a) \cdot \max(P_p^{max}, P_a^{max})}, \tag{5.5}$$

where $\overrightarrow{F}$ is a vector of end-to-end throughput for all flows (source-destination pairs), $\overrightarrow{D}$ is a vector of Euclidean distances for all source-destination pairs, N_p and N_a are the total number of peers and altruists, respectively, P_p^{max} and P_a^{max} are the maximum power consumption of all peers and all altruists, respectively, $b_0 = e_0/c_0$, and e_0 and c_0 are the initial energy and the unit cost of a node, respectively.

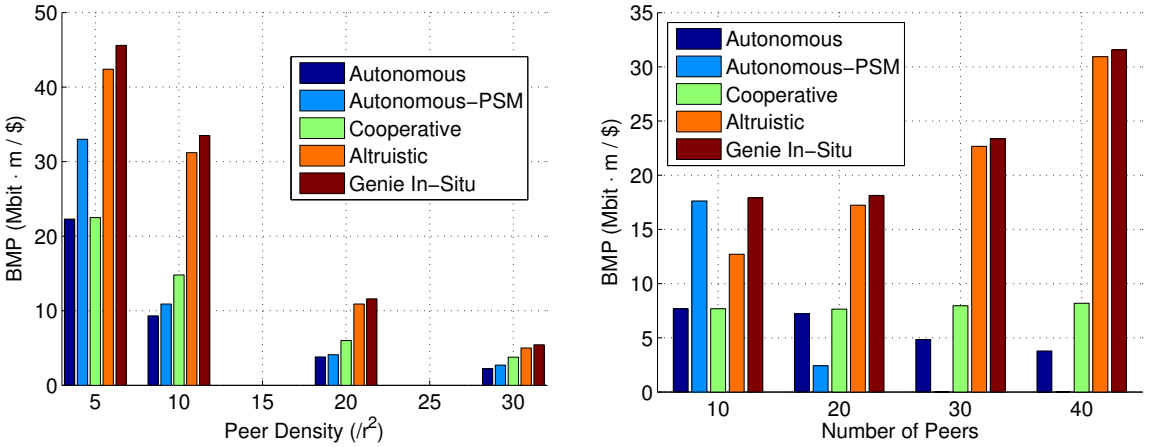
BMP can be understood as Throughput × Distance × Lifetime / Price, where

$$\text{Lifetime} : L \triangleq \frac{e_0}{\max(P_p^{max}, P_a^{max})}, \tag{5.6}$$

$$\text{Price} : C \triangleq c_0 \cdot (N_p + N_a). \tag{5.7}$$

Therefore, BMP gives the total amount of successfully delivered data scaled by distance, during the network's operational time, normalized by system resources. The unit is bit·m/$. In the above, lifetime L is defined as the time until any node (peer or altruist) runs out of energy. It can also be defined in terms of peers only, i.e., e_0/P_p^{max}, as peers are essential to operating a network. In fact, this latter definition leads to a *higher* BMP for altruistic cooperation, but we do not adopt it. In addition, e_0 and c_0 are the same for altruists and peers as they are basically the same devices.

BMP is applicable to networks with different geographic areas (taken into account by $\overrightarrow{D}$), sizes of population (by $N_p + N_a$), models of devices (by b_0),[¶] topologies (random or arbitrary), and networks without altruists (simply set $P_a^{max} = 0$ and $N_a = 0$). It is also independent of whether a protocol is cooperative or autonomous, single- or multi-channel. Plus, from a system design perspective, it captures the trade-off among three key factors: sustainable data rate, operational time, and economic cost. Ultimately, BMP allows for a fair comparison of cost efficiency across different protocols and different networks.



(a) Multi-hop networks.

(b) Single-hop networks.

Figure 5.5: Evaluating cost efficiency. The higher BMP, the more cost-efficient.

In this study, since all protocols use the same devices, b_0 does not affect comparison and we set $b_0 = 1$.

5.4.2 BMP evaluation

We conduct simulations on all the five protocols, measure the parameters in (5.5), and then compute BMP for them. In *Altruistic*, according to Section 5.3.2, we set altruist density to be $1.31/r^2$ in multi-hop networks, and deploy one altruist in single-hop networks. Every source generates traffic at 25kb/s and 160kb/s in multi-hop and single-hop networks, respectively.

The results are shown in Figure 5.5. The key observation is that, apart from *Genie In-Situ*, *Altruistic* is the clear winner among all the protocols; its BMP is more than twice that of the other protocols in most cases. Compared to *Genie In-Situ*, the BMP of *Altruistic* is only marginally lower. In fact, for a real in-situ energy conscious cooperation (without genie), the complexity and overhead for voting cooperative nodes or optimizing the probability of cooperation will most likely negate this marginal advantage of *Genie In-Situ* over *Altruistic*.

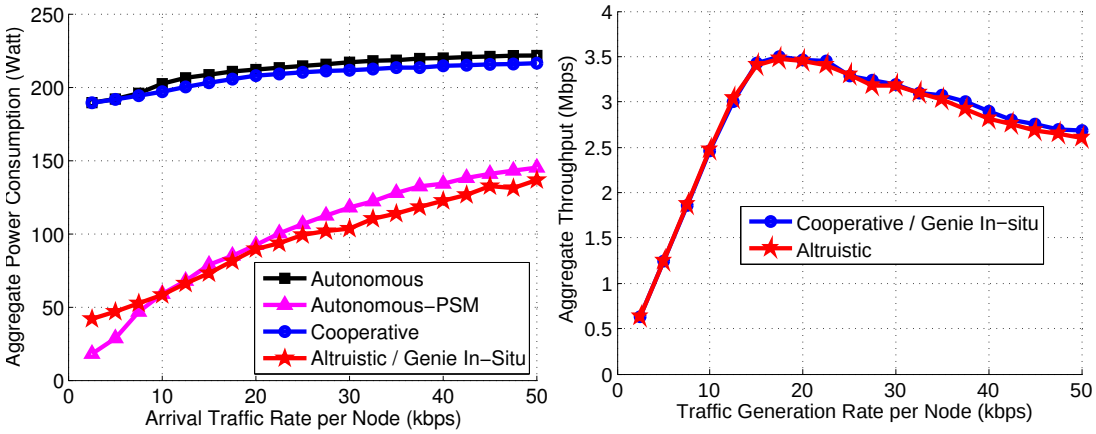
For a more precise understanding, we examine BMP component by component. Taking a multi-hop network with peer density $10/r^2$ (in Figure 5.5a) as an example, we compare *Altruistic* and *Cooperative* in terms of four components of BMP:

- Throughput and Distance, $\overrightarrow{F} \cdot \overrightarrow{D}$: 3822 Mbit·m/s for *Altruistic* and 3826 Mbit·m/s for *Cooperative*, which are almost equal. The reasons are that (i) $\overrightarrow{F}$ is almost the same for the two protocols because the cooperation coverage in *Altruistic* (80%) is speculated to be close to the probability that an MCC problem can obtain cooperation in *Cooperative* (a theoretical analysis of the probability of obtaining cooperation can be found in [101] which uses a simplified system model), and (ii) $\overrightarrow{D}$ is statistically the same for the two protocols since the same network and the same routing algorithm are used.

- Lifetime L: $\frac{e_0}{0.301}$ sec for *Altruistic* and $\frac{e_0}{0.718}$ sec for *Cooperative*. *Altruistic* has a longer lifetime (2.385 times that of *Cooperative*) because (i) most nodes are peers which can employ sleep-wake duty cycling to save power, and (ii) the few altruists only send small control packets (INV) when necessary and thus do not considerably contribute to power consumption.

- Price C: $407c_0$ for *Altruistic*, 13% higher than $360c_0$ for *Cooperative*.

¶Each individual network is homogeneous, i.e. with the same e_0 and c_0. However, it is simple to extend (5.5) to accommodate heterogeneous networks.



(a) Power consumption.

(b) Throughput.

Figure 5.6: Evaluating throughput-energy trade-off in multi-hop networks. Since *Genie In-Situ* was observed to perform very closely to *Cooperative* in terms of throughput and *Altruistic* in terms of power consumption, we combine the corresponding curves for a clear visualization.

Therefore, according to

$$BMP = \frac{\overrightarrow{F} \cdot \overrightarrow{D} \cdot L}{C},$$

and since $e_0/c_0 = b_0 = 1$, we have

$$BMP_{altruistic} = 31.2Mbit \cdot m/\$,$$

$$BMP_{cooperative} = 14.8Mbit \cdot m/\$,$$

which translates to a significant ratio of 2.11 ($BMP_{altruistic}$ / $BMP_{cooperative}$).

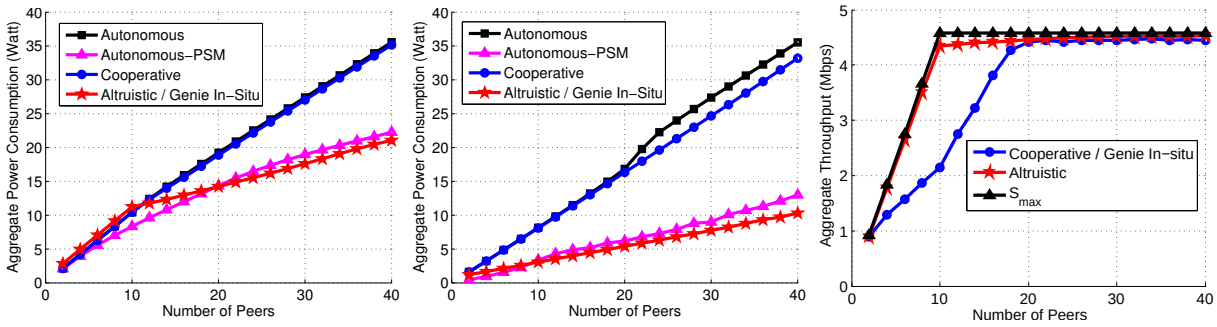
In Figure 5.5b, the single-hop results give a clearer demonstration of the effect from cooperation. As the number of peers increases, the BMP of *Autonomous* slightly decreases due to the lack of information sharing, and the BMP of *Autonomous-PSM* dramatically drops (and even reaches near zero) due to the lack of both information sharing and gathering, while the BMP of *Cooperative* nearly maintains. On the other hand, the BMP of *Altruistic* and *Genie In-Situ* both increase, thanks to energy conservation.

Our BMP evaluation demonstrates that the performance gain of altruistic cooperation well offsets its increased cost due to additional nodes; its cost efficiency more than doubles that of other protocols. In-situ energy conscious cooperation, as another possible choice, is only justifiable if it can closely approach the upper bound via prudent design.

5.5 Throughput-Energy Trade-off

This section addresses the third issue by zooming into the performance of throughput and aggregate energy consumption (including altruists). The same as in Section 5.4.2, the altruist density is $1.31/r^2$ in multi-hop networks and only one in single-hop networks.

Figure 5.6 summarizes the results for multi-hop networks with peer density $10/r^2$ (the results for $20/r^2$ are similar). For power consumption as shown in Figure 5.6a, a remarkable 40–80% energy saving is seen by comparing *Altruistic* against *Cooperative* or *Autonomous*. At higher traffic load, *Altruistic* even slightly outperforms *Autonomous-PSM* which intuitively should consume the least energy. The reasons are twofold. First, altruists incur insignificant power usage because they are sparse and the small cooperative messages are only received by a few peers (other peers are asleep or on data channels). Second, altruists significantly reduce retransmissions caused by MCC problems which happen more frequently under higher traffic. The second factor also explains why *Cooperative* consumes slightly



(a) Power consumption (saturated traffic). (b) Power consumption (light traffic). (c) Throughput (saturated traffic).

Figure 5.7: Evaluating throughput-energy trade-off in single-hop networks.

less power than *Autonomous*. The throughput shown in Figure 5.6b clearly show that the altruistic cooperation preserves the throughput benefit of the original cooperative protocol.

Figure 5.7 summarizes the results for single-hop networks. Figure 5.7a and Figure 5.7b show energy consumption under saturated and light traffic (160kb/s per source), respectively.[‖] *Altruistic* is observed to conserve energy substantially. For example, it consumes only 30% power of *Cooperative* when the number of nodes is 40 (Figure 5.7b). In addition, we see that *Altruistic* even slightly outperforms *Autonomous-PSM* again, which is explained in the multi-hop results.

In Figure 5.7c, it is noteworthy that *Altruistic* even outperforms *Cooperative* and *Genie In-Situ* when the number of peers is less than 20. The reason is that (i) at high traffic load, the small number of peers will stay on data channels most of the time, making *Cooperative* and *Genie In-Situ* lack of cooperative nodes, but (ii) on the contrary, *Altruistic* has a dedicated cooperative node guaranteeing full cooperation coverage, which also enables *Altruistic* to approach S_{max} very closely. S_{max} is the maximum throughput derived in [93] based on the common handshake used by all the five protocols in this study:

$$S_{max} = \frac{\min(m, n_f) \cdot T_{payload} \cdot W}{T_{cca}^{min} + T_{ctrl} + T_{data} + T_{sw}}, \tag{5.8}$$

where m is the number of data channels, n_f is the number of flows, W is the data channel bandwidth, $T_{payload}$ is the transmission time of data payload, T_{cca}^{min} is the minimum CCA duration, T_{ctrl} and T_{data} are the duration of a successful control/data channel handshake, and T_{sw} is channel switching delay.

In summary, our simulations demonstrate that *Altruistic* is able to achieve the lowest energy consumption while preserving the throughput benefit of the cooperative protocol.

5.6 Reflections

5.6.1 Limitation

Altruistic cooperation becomes less effective when there are only a few peers (relative to the number of channels) or traffic is light. For example, the BMP of *Altruistic* is lower than *Autonomous-PSM* in Figure 5.5b at 10 nodes (five data channels, low traffic) and in Figure 6.5 at 2 nodes (two data channels, high traffic). This is due to the extra energy consumption and cost incurred by altruists while channel contention is very mild. In such scenarios, in-situ energy conscious cooperation may be a better choice. Other appropriate scenarios for the in-situ strategy are those precluding additional nodes and when prudent design can achieve the genie upper bound, as mentioned in Section 5.4.2.

We do not include the manpower cost in node deployment because it is hard to characterize in reality. However, via appropriate simplification, one can still incorporate it into the price model of BMP

[‖] *Autonomous-PSM* and *Autonomous* perform differently even under high traffic load. This is because PSM allows a node to switch off radio even if it is always backlogged, e.g., when it finds that all data channels are busy.

as follows: (1) Assuming that manpower cost is additive, i.e., it adds a constant cost of c_{man} to the total price, simply redefine the price (equation (5.7)) as $C \triangleq c_0 \cdot (N_p + N_a) + c_{man}$. (2) Assuming that manpower cost is multiplicative, i.e., it is proportional to the number of nodes, then redefine the price as $C \triangleq (c_0 + c_{man})(N_p + N_a)$, where c_{man} stands for *unit* manpower cost.

5.6.2 Protocol Overhead

The cooperative protocol used in this paper employs a four-way handshake which appears to have very high overhead. This overhead, however, pays off in most cases, via the cooperation gain, as shown in our simulations and experiments. Nevertheless, reducing the overhead is still desired especially when data packets are too small. There are several ways to do this. For example, (1) IEEE 802.11 allows to use a 9-byte rather than an 18-byte PLCP preamble for each packet, (2) each packet can use 1–2 byte node IDs instead of 6-byte universal MAC addresses, because a MAC protocol only needs neighborhood-wide instead of network-wide unique identifiers, and (3) packet train is a very effective way to amortize overhead, as used by MMAC, SSCH, and WiFlex. We have adopted (1) and would like to incorporate (2) and (3) in future implementations.

5.6.3 Fairness

A possible concern is that altruists may be over-burdened since they are always awake, and hence drain energy must faster than peers. To address this unfairness, it is possible to combine the altruist strategy and the in-situ strategy such that altruists also rotate the role of cooperation. However, this *hybrid* strategy will sacrifice *simplicity* as the primary feature of the altruist strategy. In fact, having altruists always awake is not necessarily energy unfair because the evaluation (via both simulation and testbed) of BMP, whose definition (5.5) takes energy fairness into account (by P_p^{max} and P_a^{max}), shows that altruistic cooperation performs well in most cases. Nevertheless, the fairness could become a pronounced problem under some (e.g., non-uniform) traffic patterns and due to non-linear protocol operations, which merits future study.

5.6.4 Using Multiple Radios

It is possible to use multiple radios to exploit multi-channel diversity, such as in [18–23], and it is also possible to build multi-radio MAC protocols using DISH. Hence, a parallel research of this study could be designing energy-efficient strategies for these multi-radio protocols or, leaving out generality, directly designing energy-efficient multi-radio cooperative MAC protocols. In that context, some concepts in this study, e.g., cooperation coverage and BMP, can still apply.

5.6.5 Summary

DISH enhances the system throughput significantly when applied to multi-channel MAC protocols. However, energy consumption can be tripled due to its two inherent components, information gathering and sharing. In this paper, we propose two energy-efficient strategies to solve this problem. Our comparative study shows that altruistic cooperation, although extremely simple (zero runtime overhead and no protocol re-design), achieves low energy consumption and preserves the throughput benefit. It is also cost efficient by more than offsetting the additional cost by its substantial performance gain. The other strategy, in-situ energy conscious cooperation, is suitable for applications with few nodes or light traffic, and those that preclude using additional nodes.

The key to the success of altruistic cooperation is twofold. First, the use of dedicated cooperative nodes provides cooperation in a *guaranteed* rather than an *opportunistic* manner. Second, the use of altruists exempts the resource-consuming tasks, i.e., information gathering and sharing, from *all* nodes to only *a few*.

This study gives the first treatment on energy efficiency for DISH-based cooperative multi-channel MAC protocols. We contend that DISH is worth pursuing and altruistic cooperation is a simple and effective way to explore it.

Chapter 6

Hardware Implementation

For a further and more realistic validation, we implemented the protocols used in our simulations on COTS hardware and conducted experiments. These protocols include CAMMAC-RAND, CAMMAC-MRU, UNCOOP-RAND, and UNCOOP-MRU, which are used in Chapter 3, and *Cooperative*, *Autonomous*, *Autonomous-PSM*, and *Altruistic*, which are used in Chapter 5. To be best of our knowledge, these prototypes are the first full implementation of asynchronous multi-channel MAC protocols for ad hoc networks (the related work is reviewed in detail in Section 2.6).

6.1 Implementation

6.1.1 Platform Selection

We chose a micro-controller (MCU) based platform with an ASIC radio, instead of (i) an FPGA-based platform, which is more expensive and requires hardware description language (HDL) in programming, or (ii) a software radio, whose MAC source node is not fully available. Among the ASIC radios, we chose 802.15.4 radios instead of 802.11 radios because 802.11-radio based devices (such as laptops and PDAs) have higher cost and bigger size than 802.15.4 devices, and 802.11-based development software (such as HostAP [102] and MadWifi [103]) has more limited MAC layer control than the software available to 802.15.4, such as TinyOS [104]. Eventually, we chose TelosB Mote [61], which is a MCU platform with an ASIC radio (CC2420 [105]) as our hardware platform and TinyOS 2.0 as our software platform. TinyOS has almost full control over the MAC layer, and its component-based architecture and C-like programming enables rapid development.

This platform choice suffices for a comparative study but, as a caveat, is not suitable for establishing benchmarks for multi-channel WiFi cards.

6.1.2 Overcoming Limitations

There are two limitations of our chosen hardware. First, the maximum packet size that CC2420 supports is only 127 bytes. Therefore, in substitution for each data packet, we transmit a *sequence of fragments*, with inter-fragment intervals τ counted as actual payload: each interval τ corresponds to a payload of τW bits, where W=250kb/s is channel rate. Also, intermediate fragments are treated as pure payload without frame headers and footers. The second limitation is that the accuracy of timing on TelosB motes is not reliable at the microsecond level while reliable at the millisecond level. We circumvent this by *proportionally* prolonging all protocol intervals such as inter-frame spacings up to milliseconds. For example, to transmit a 2-Kbyte data packet, a node transmits a sequence of 20 fragments with the length of 30 bytes each (including preamble) and the 19 intervals of 8 ms each. This results in a total of $\sim$175ms to transmit a data packet (each fragment needs 100–200 μs to be sent in the air after assembled in memory). Under the same setting, a control channel handshake lasts for $\sim$9 ms. This ratio (175:9) is close to that in our simulations.

Figure 6.1: Detecting packet collision via an interleaved fragment sequence, where `TX/RX` IDs are *alternate* and `seq`'s are *inconsecutive*.

6.1.3 Collision Detection

Interestingly, the above methods used to overcome the limitations enable us to devise a very simple yet accurate technique for packet *collision detection*, which can also be used in many other scenarios. Collision detection is useful in that it benefits collision avoidance, flooding, channel selection, and data aggregation algorithms [106] in differentiating between the two causes of *packet corruption*: packet collision and channel imperfection (such as noise and fading). A typical prior technique is CRC checking, which unfortunately does not solve the detection problem because it only indicates packet *corruption*. Other solutions such as using link quality indicator (LQI) and/or received signal strength indicator (RSSI) are empirical in essence and lack in accuracy. According to [107–109], it is still controversial whether RSSI or LQI is a better indicator for link quality.

Our technique is by detecting an *interleaved fragment sequence*. The key idea is based on (i) the fragmented data transmission and (ii) the fact that the fragment interval (8 ms) is much larger than the fragment transmission time (<1 ms). Therefore, an interleaved fragment sequence in which the fragments are sent by more than one nodes, as illustrated by Figure 6.1, indicates data packet collision (recall that intervals are actually payload in our case). Hence to detect packet collision, a node simply needs to check fragment headers.

6.2 Experiments and Results

For visualization purposes, we use the three LEDs on each TelosB mote to indicate specific events of interest (a maximum of $2^3 = 8$ events). For example, a blue LED indicates an ongoing control channel handshake, a green LED indicates an ongoing data channel handshake, and a red LED indicates transmitting a cooperative message. Other events are indicated by LED combinations.

6.2.1 Experiments on CAM-MAC

Recall from Chapter 3 that CAM-MAC allows neighboring nodes to send cooperative messages (INV) to help transmitter-receiver pairs avoid MCC problems. A snapshot of our experiments on CAM-MAC is shown in Figure 6.2.

In the experiments, nodes are configured as disjoint flows in an indoor area, and source nodes are always backlogged. Three channels are used as one control channel and two data channels, each with bandwidth 250kbit/s.* The transmission power is 0dBm which is the maximum on CC2420. All nodes are within the radio range of each other, which is the same setting as used by [32, 59, 60]; we leave multi-hop experiments as future work due to practical constraints such as cost and complexity. In collecting statistics for each of the four protocols, each single data point is by averaging over 6 experiments and each experiment runs for 360 actual seconds.

The experimental results are presented in Figure 6.3. When the number of nodes is four, the two MRU protocols have about twice throughput of the two RAND protocols. This is because MRU strategy in effect assigns each pair a *dedicated* data channel, while RAND strategy encounters channel conflicts with probability 0.5 at each selection (there are two data channels). The reason why CAMMAC-RAND and UNCOOP-RAND perform the same is that, any time when a transmitter-receiver pair selects a channel conflicting with the other pair, there is no additional node on the control channel to cooperate.

When the number of nodes is six, throughput of all protocols sharply declines except for CAMMAC-RAND. This is similar to the simulation results in Figure 3.10c at 12 nodes, and the explanation is that the deprivation of MRU channels, due to one more node pair, degrades MRU strategy to RAND strategy.

*Using three channels can show different trends from simulation results (especially for UNCOOP) rather than simply produce a scaled version. This is more clearly demonstrated in the experiments with *Altruistic* and will be explained in detail in Section 6.2.2.

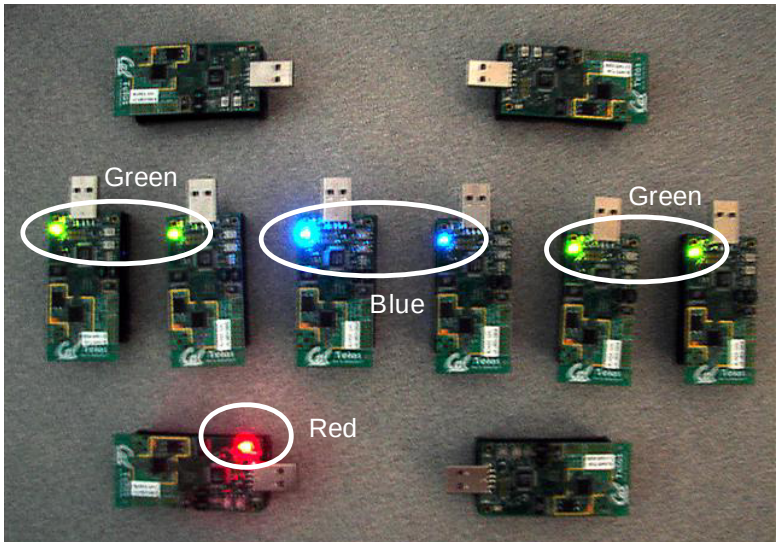


Figure 6.2: A snapshot of an experiment on CAM-MAC with 10 nodes. The four "green nodes" are two transmitter-receiver pairs communicating on two different data channels. The two "blue nodes" are performing a control channel handshake (specifically, a PRA has just been sent from one to the other). This creates a channel conflict problem because the only two data channels are already being in use. At this moment, a neighboring node, indicated by the red LED, identifies this (via the PRA) and sends a cooperative message (INV). After this, the two blue nodes will backoff to discontinue the control channel handshake, thereby preventing the data collision. Note that other three idle nodes may also identify the MCC problem, but the cooperation collision avoidance period takes effect and only one node will send INV in this case.

CAM-MAC achieves a moderate throughput of ∼260kbit/s and is more than UNCOOP because the two more nodes can occasionally provide cooperation.

Note that this performance degradation at six nodes is consistent with the comparison with MMAC shown in Figure 3.14a, where CAM-MAC has only marginal improvement over MMAC in the case of six nodes and two data channels.

Beyond six nodes, the throughput of CAM-MAC recovers since cooperative nodes become often available. RAND and MRU strategies do not make much difference due to the reason described in the simulation results. Finally at saturation, CAM-MAC gains more than *triple* the throughput of UNCOOP.

In summary, our experimental results further justify the value of DISH.

6.2.2 Experiments on Energy Efficiency

Recall from Chapter 5 that altruistic cooperation uses altruists as solely cooperative nodes to take over the responsibility of cooperation from normal nodes. Such a snapshot is shown in Figure 6.4.

In our experiments, nodes are randomly placed in a 10m×10m roof area, and the transmission power is set at 0dBm. By this setting, all nodes are within the radio range of each other, which is also used by [32, 59, 60]. To do multi-hop experiments, a large number of nodes are required to demonstrate the impact of a small ρ_{alt} on a large ρ_{peer} as shown in Section 5.3.2 and Section 5.5. Nodes are configured as disjoint flows, and source nodes are always backlogged. Three channels with data rate 250kb/s each are used, i.e., one control channel and two data channels. To compute power consumption, we trace the TX/RX/IDLE state of each node to accumulate the sojourn time in each state, and at the end of each experiment, do a weighted sum using the same power rates as in the simulation setup. For protocols using power saving, IDLE is treated as SLEEP and peers do not cache overheard information. Alternatively, it may be possible to actually put motes into sleep, e.g., by developing B-MAC [57] or X-MAC [58] into a multi-channel version, and then measure the actual battery drainage. However, accurately measuring

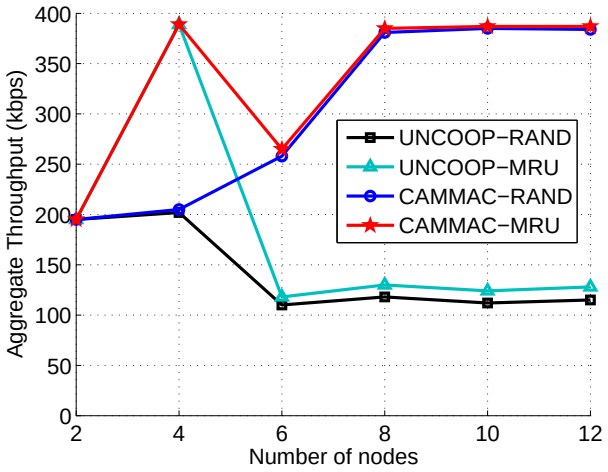


Figure 6.3: Experimental results. The maximum utilizable bandwidth is 500kbit/s.

the energy consumption of sensor nodes is difficult [62], and is also not necessary, because (i) our energy model is WiFi-based and not for sensor devices, (ii) otherwise a real sleep-wake schedule has to be used and loses generality as discussed in Section 2.5, and (iii) this is a comparative study and the goal is not to establish benchmarks for TelosB. This same approach was also used by [62].

In collecting statistics for the four protocols, every data point is by averaging over 8 experiments and each experiment runs for 600 actual seconds.

Figure 6.5 summarizes the experimental results of cost efficiency. *Altruistic* is again the clear winner among the protocols. The only exception is in the case of two peers, where *Autonomous-PSM* performs the best. The reason is simple: the only one transmitter-receiver pair does not have any channel contenders, and hence cooperation is not needed and adding an altruist only increases cost and energy consumption.

Figure 6.6 gives the results for throughput-energy tradeoff. The power consumption is shown in Figure 6.6a, where we can see that *Altruistic* consumes the lowest power among all the protocols when there are a sufficient number of nodes. It is notable that, although both experiments and simulations use the same power consumption rates, the experimental statistics are consistently lower than the simulation results (Figure 5.7a). This is because, in our hardware implementation, the protocol intervals are prolonged to overcome the inaccurate timing of hardware, as described in Section 6.1. Hence, the IDLE state appears more often whereas the TX/RX state appears less than they do in simulations. For throughput shown in Figure 6.6b, *Altruistic* performs better than *Cooperative* when the number of nodes is small, thanks to the always available cooperation provided by the altruist.

In summary, the experimental results confirm that *Altruistic* achieves high throughput and low energy consumption simultaneously, and is the most cost-efficient among all the protocols under comparison. The work also shows that multi-channel MAC protocols can be indeed implemented and really work on COTS hardware, with a single radio and asynchronously.

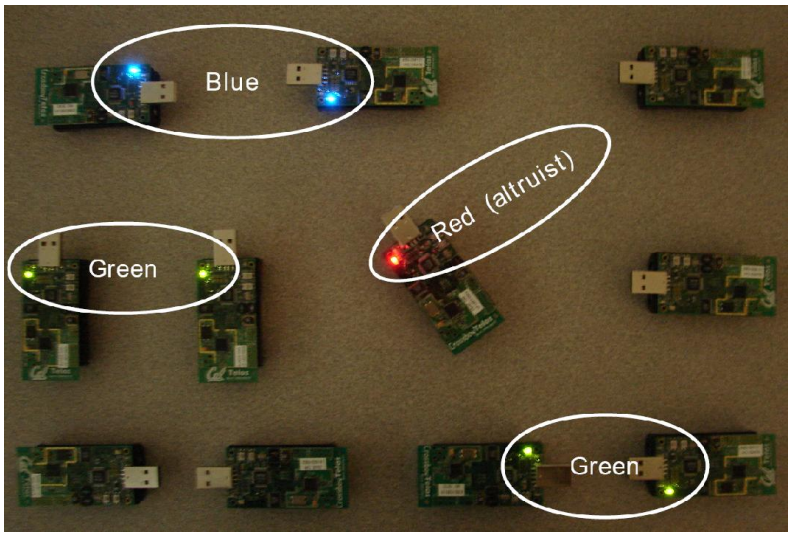


Figure 6.4: A snapshot of a trial indoor experiment on *Altruistic* with 11 nodes. The four "green nodes" and the two "blue nodes" are performing data and control channel handshakes, respectively, the same as in Figure 6.2. The blue nodes are going to cause a channel conflict to one pair of the green nodes. At this moment, the altruist, indicated by the red LED, identifies this and sends a cooperative message. The two blue nodes will then back off to terminate the control channel handshake and thereby avoid data collision.

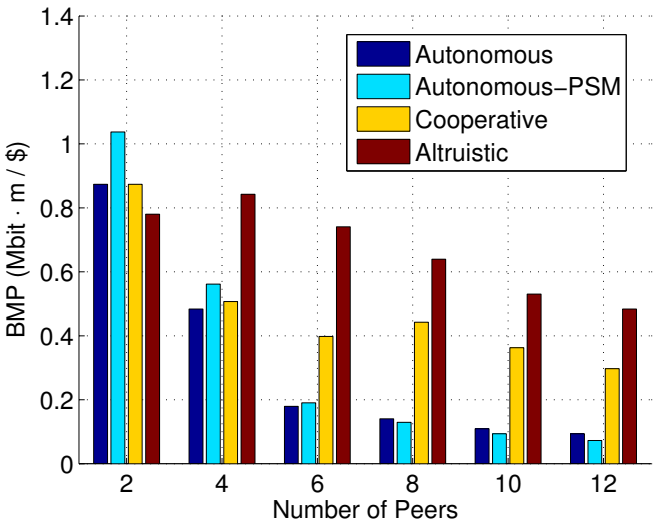
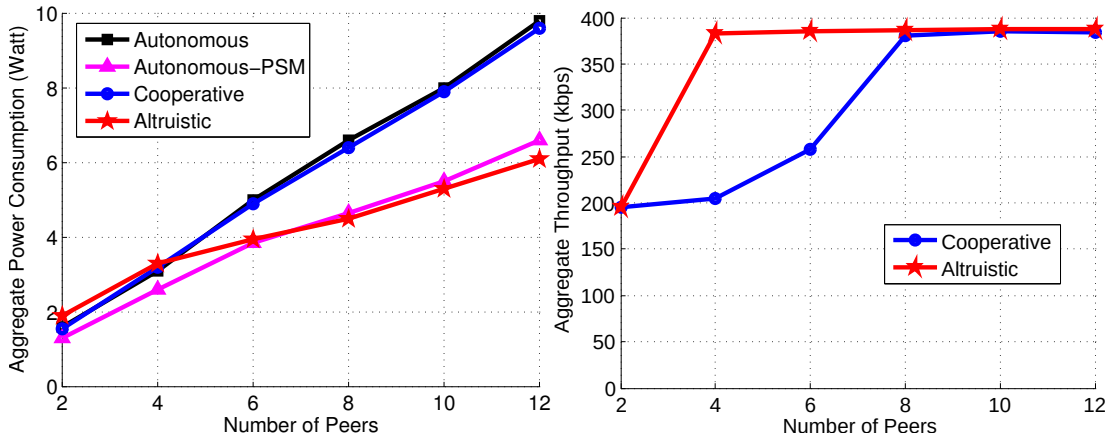


Figure 6.5: Experimental results of cost efficiency.



(a) Power consumption.

(b) Throughput.

Figure 6.6: Experimental results of throughput-energy tradeoff.

Chapter 7

Conclusion

In this thesis, we introduce distributed information sharing (DISH), a distributed flavor of control-plane cooperation, as a new and general approach to wireless network protocol design. The notion of control-plane cooperation augments the conventional understanding of cooperation which sits at the data plane as a relaying mechanism. For a proof of concept, we apply DISH to multi-channel MAC protocol design, taking approaches of analysis, simulation, and implementation. Our study demonstrates that (i) CAM-MAC is a cheap (single radio and asynchronous) yet very productive DISH-based protocol to support multi-channel operation, (ii) the metric p_{co}, which captures the availability of cooperation, is capable of characterizing the performance of DISH networks and bears significant implications, and (iii) altruistic cooperation is a very simple yet effective strategy that dramatically reduces energy consumption for DISH protocols while preserving the throughput benefit, and is cost efficient as well. Ultimately, DISH is shown to be a *viable* approach to cooperative wireless networking.

Recalling the twofold purpose of our study as stated in Section 1.2, we have proved that DISH is indeed a feasible and viable idea, and have proposed a multi-channel MAC protocol (CAM-MAC armed with altruistic cooperation) which is indeed practical.

7.1 DISH Applications

DISH can apply to a wide range of scenarios. The below lists a few possible applications in addition to multi-channel MAC protocol design.

- *Cognitive DISH*: Cognitive radio is a paradigm for wireless communication in which either a network or a wireless node changes its transmission or reception parameters to communicate efficiently avoiding interference with licensed or unlicensed users [110]. Its purpose is for the efficient use of frequency spectrum. A cognitive radio network consists of licensed (or primary) users and unlicensed (or secondary) users, and embodies two principles: (1) primary users need not care about secondary users when using the spectrum, (2) secondary users access the spectrum opportunistically with the constraint that they should not cause harmful interference to primary users. These increase the need of information about user status, and hint that DISH could be applied to facilitate the user detection and resource allocation. This motivated our recent work [111].

- *Cooperative directional antenna (CDA)*: Directional antennae are brought forth to reduce the interference region and prolong the transmission distance. By somehow imitating wired communication, they are expected to achieve better performance than omni-directional antennae. However, this technique is accompanied with two new problems: (i) due to the directionality, a transmission can only silence a small area and hence would have more hidden nodes, and (ii) due to the uncertainty of which direction an antenna is facing toward, a receiver may miss packets addressed at it from a transmitter. In both cases, DISH could be used to request neighboring nodes to take care of the hidden nodes or the deaf receiver.

- *DI-meSH*: A wireless mesh network consists of mesh routers and mesh clients, where mesh routers form the infrastructure to provide high-bandwidth access to the Internet, and mesh clients connect to the Internet via mesh routers. On the other hand, unlike a centralized network, (i) mesh clients can communicate to each other directly or via multi-hop, and (ii) mesh routers are also connected wirelessly via multi-hop. Therefore, a mesh network is a hybrid environment with both centralized and distributed control, where DISH could be incorporated to reinforce and improve the distributed communications.
- *Solving the traditional hidden terminal problem*: The hidden terminal problem [112] is caused by the ignorance of a receiver in the interference range of a (hidden) transmitting node. In this scenario, DISH can be introduced to enable idle neighbors to inform the hidden node to postpone its transmission.

7.2 Impact of DISH

We believe that DISH has significant implications to cooperative wireless networking. First, DISH represents a new dimension of cooperative diversity and would provoke new ways of devising cooperative mechanisms on the control plane. Second, DISH essentially builds a *distributed knowledge base* where individual nodes can store information into and retrieve information from. There is no special hardware requirement for this, making it easy to deploy such systems.

7.3 Challenges

There are also challenges to the notion of DISH. First, we have assumed a cooperative environment where each node is willing to help. However, in some realistic scenarios, misbehaving nodes exist and may cause adverse consequence without an appropriate solution. Second, applying DISH to wireless sensor networks is confronted with a problem that sensor devices only support very small data packets (e.g., less than 127 bytes on Mica family and Telos platforms). Hence the overhead of sending cooperative messages becomes prominent, and a solution is yet to be found. Third, throughput or delay analysis for DISH networks would be hard as such analysis in a non-DISH network is still open problems in general, but it is useful and worth pursuing. Our analysis in terms of p_{co} should provide a useful reference to studies toward this end.

Finally, the research community and especially the industry may be keen to see how DISH works on real WiFi devices. Currently we are carrying out a Proof-Of-Concept (POC) project [113] funded by the Nation Research Foundation (NRF). This project is to implement CAM-MAC on laptops with wireless multi-channel cards and to investigate relevant issues in realistic environments.

Bibliography

[1] S. Mirsky, "Einstein's hot time," *Scientific American Magazine*, September 2002.

[2] A. Host-Madsen, "Capacity bounds for cooperative diversity," *IEEE Trans. on Information Theory*, vol. 52, no. 4, pp. 1522–1544, Apr. 2006.

[3] J. N. Laneman and G. W. Wornell, "Distributed space-time-coded protocols for exploiting cooperative diversity in wireless networks," *IEEE Transactions on Information Theory*, vol. 49, no. 10, October 2003.

[4] A. Sendonaris, E. Erkip, and B. Aazhang, "User cooperation diversity–part I: System description," *IEEE Trans. Commun.*, vol. 51, no. 11, December 2003.

[5] ——, "User cooperation diversity–part II: Implementation aspects and performance analysis," *IEEE Trans. Commun.*, vol. 51, no. 11, Dec. 2003.

[6] H. Zhu and G. Cao, "rDCF: A relay-enabled medium access control protocol for wireless ad hoc networks," in *IEEE Infocom*, 2005.

[7] A. Azgin, Y. Altunbasak, and G. AlRegib, "Cooperative MAC and routing protocols for wireless ad hoc networks," in *IEEE GLOBECOM*, 2005.

[8] S. Moh, C. Yu, S.-M. Park, and H.-N. Kim, "CD-MAC: Cooperative diversity MAC for robust communication in wireless ad hoc networks," in *IEEE ICC*, June 2007, pp. 3636–3641.

[9] P. Liu, Z. Tao, S. Narayanan, T. Korakis, and S. S. Panwar, "CoopMAC: A cooperative MAC for wireless LANs," *IEEE Journal On Selected Areas In Communications*, vol. 25, no. 2, pp. 340–354, 2007.

[10] A. Scaglione, D.Goeckel, and J. N. Laneman, "Cooperative communications in mobile ad-hoc networks: Rethinking the link abstraction," *IEEE Signal Processing Magazine, Special Issue on Signal Processing for Wireless Ad hoc Communication Networks*, June 2006.

[11] Y.-W. Hong, W.-J. Huang, F.-H. Chiu, and C.-C. Kuo, "Cooperative communications in resource-constrained wireless networks," *IEEE Signal Processing Magazine*, vol. 24, no. 3, pp. 47–57, May 2007.

[12] "Bluetooth," http://en.wikipedia.org/wiki/Bluetooth.

[13] "Carrier sense multiple access with collision avoidance," http://en.wikipedia.org/wiki/CSMA/CA.

[14] C.-C. Chen, E. Seo, H. Kim, and H. Luo, "Select: Self-learning collision avoidance for wireless networks," *IEEE Transactions on Mobile Computing*, vol. 7, no. 3, pp. 305–321, March 2008.

[15] F. Cal, M. Conti, and E. Gregori, "Tuning of the IEEE 802.11 protocol to achieve a theoretical throughput limit," *IEEE/ACM Transactions on Networking*, vol. 8, no. 6, Dec. 2000.

[16] M. Anand, E. B. Nightingale, and J. Flinn, "Self-tuning wireless network power management," in *ACM MobiCom*, 2003, pp. 176–189.

[17] D. Liu, Y. Zhang, and H. Zhang, "A self-learning call admission control scheme for CDMA cellular networks," *IEEE transactions on neural networks*, vol. 16, no. 5, pp. 1219–1228, 2005.

[18] 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 *I-SPAN*, 2000.

[19] A. Nasipuri, J. Zhuang, and S. R. Das, "A multichannel CSMA MAC protocol for multihop wireless networks," in *WCNC*, 1999.

[20] A. Nasipuri and J. Mondhe, "Multichannel CSMA with signal power-based channel selection for multihop wireless networks," in *IEEE VTC*, 2000.

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

[22] A. Adya, P. Bahl, J. Padhye, and A. Wolman, "A multi-radio unification protocol for IEEE 802.11 wireless networks," in *IEEE Broadnets*, 2004.

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

[24] J. Chen, S. Sheu, and C. Yang, "A new multichannel access protocol for IEEE 802.11 ad hoc wireless LANs," in *PIMRC*, 2003.

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

[26] J. Zhang, G. Zhou, C. Huang, S. H. Son, and J. A. Stankovic, "TMMAC: an energy efficient multi-channel MAC protocol for ad hoc networks," in *IEEE ICC*, 2007.

[27] A. Tzamaloukas and J. Garcia-Luna-Aceves, "Channel-hopping multiple access," in *IEEE ICC*, 2000.

[28] ——, "Channel-hopping multiple access with packet trains for ad hoc networks," in *IEEE Device Multimedia Communications*, 2000.

[29] P. Bahl, R. Chandra, and J. Dunagan, "SSCH: Slotted seeded channel hopping for capacity improvement in IEEE 802.11 ad-hoc wireless networks," in *ACM MobiCom*, 2004.

[30] R. Vedantham, S. Kakumanu, S. Lakshmanan, and R. Sivakumar, "Component based channel assignment in single radio, multi-channel ad hoc networks," in *ACM MobiCom*, 2006.

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

[32] H.-S. W. So, G. Nguyen, and J. Walrand, "Practical synchronization techniques for multi-channel MAC," in *ACM MobiCom*, 2006.

[33] L. Huang and T.-H. Lai, "On the scalability of IEEE 802.11 ad hoc networks," in *ACM MobiHoc*, 2002, pp. 173–182.

[34] Y.-C. Tseng, C.-S. Hsu, and T.-Y. Hsieh, "Power-saving protocols for IEEE 802.11-based multi-hop ad hoc networks," in *IEEE Infocom*, 2002.

[35] R. Maheshwari, H. Gupta, and S. R. Das, "Multichannel mac protocols for wireless networks," in *IEEE SECON*, 2006.

[36] Maxim Integrated Products Inc., *MAX2820, MAX2820A, MAX2821, MAX2821A 2.4GHz 802.11b Zero-IF Transceivers Data Sheet rev. 04/2004*, Sunnyvale, California, USA, 2004.

[37] J. W. Lee, J. Mo, T. M. Trung, J. Walrand, and H.-S. W. So, "Wiflex: Multi-channel cooperative protocols for heterogeneous wireless devices," in *IEEE WCNC*, 2008.

[38] S.-L. Wu, C.-Y. Lin, Y.-C. Tseng, and J.-P. Sheu, "A multi-channel MAC protocol with power control for multi-hop mobile ad hoc networks," *The Computer*, vol. 45, no. 1, pp. 101–110, 2002.

[39] N. Shacham and P. King, "Architectures and performance of multichannel multihop packet radio networks," *IEEE JSAC*, vol. 5, no. 6, p. 1013C1025, 1987.

[40] P. Kyasanur and N. H. Vaidya, "Routing and link-layer protocols for multi-channel multi-interface ad hoc wireless networks," *Mobile Computing and Communications Review*, vol. 10, no. 1, pp. 31–43, Jan. 2006.

[41] N. Abramson, "The Aloha system—another alternative for computer communications," in *AFIPS*. IEEE, 1970, pp. 281–285.

[42] L. Kleinrock and F. A. Tobagi, "Packet switching in radio channels: Part I–carrier sense multiple-access modes and their throughput-delay characteristics," *IEEE Trans. Commun.*, vol. COM-23, no. 12, pp. 1400–16, Dec. 1975.

[43] IEEE 802.11 Working Group, *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications*, 1999.

[44] P. Karn, "MACA - a new channel access method for packet radio," in *9th ARRL Computer Networking Conference*, London, Ontario, Canada, 1990, pp. 134–140.

[45] V. Bharghavan, A. J. Demers, S. Shenker, and L. Zhang, "MACAW: A media access protocol for wireless LAN's," in *ACM SigComm*, 1994, pp. 212–225.

[46] G. Bianchi, "Performance analysis of the IEEE 802.11 distributed coordination function," *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 3, pp. 535–547, March 2000.

[47] Y. S. Han, J. Deng, and Z. J. Haas, "Analyzing multi-channel medium access control schemes with ALOHA reservation," *IEEE Trans. on Wireless Comm.*, 2005.

[48] P. Kyasanur and N. H. Vaidya, "Capacity of multi-channel wireless networks: Impact of number of channels and interfaces," in *ACM MobiCom*, August 2005.

[49] M. Kodialam and T. Nandagopal, "Characterizing the capacity region in multi-radio multi-channel wireless mesh networks," in *ACM MobiCom*. New York, NY, USA: ACM Press, 2005, pp. 73–87.

[50] F. A. Tobagi and L. Kleinrock, "Packet switching in radio channels: Part III–polling and (dynamic) split-channel reservation multiple access," *IEEE Trans. Commun.*, vol. COM-24, no. 8, pp. 832–845, August 1976.

[51] J. Wang, Y. Fang, and D. Wu, "A power-saving multi-radio multi-channel MAC protocol for wireless local area networks," in *IEEE Infocom*, 2006.

[52] G. Zhou, C. Huang, T. Yan, T. He, J. Stankovic, and T. Abdelzaher, "MMSN: Multi-frequency media access control for wireless sensor networks," in *IEEE Infocom*, 2006.

[53] X. Chen, P. Han, Q. He, S. Tu, and Z. Chen, "A multi-channel MAC protocol for wireless sensor networks," in *IEEE Intl. Conf. on Comp. Inf. Tech. (CIT)*, 2006.

[54] K. R. Chowdhury, N. Nandiraju, D. Cavalcanti, and D. P. Agrawal, "CMAC – a multi-channel energy efficient MAC for wireless sensor networks," in *IEEE WCNC*, 2006, pp. 1172–1177.

[55] W. Ye, J. Heidemann, and D. Estrin, "Medium access control with coordinated adaptive sleeping for wireless sensor networks," *IEEE/ACM Transactions on Networking*, vol. 12, no. 3, pp. 493–506, 2004.

[56] T. van Dam and K. Langendoen, "An adaptive energy-efficient MAC protocol for wireless sensor networks," in *ACM SenSys*, 2003, pp. 171–180.

[57] J. Polastre, J. Hill, and D. Culler, "Versatile low power media access for wireless sensor networks," in *ACM SenSys*, NY, USA, 2004, pp. 95–107.

[58] M. Buettner, G. V. Yee, E. Anderson, and R. Han, "X-MAC: A short preamble MAC protocol for duty-cycled wireless sensor networks," in *ACM SenSys*, 2006, pp. 307–320.

[59] C. Chereddi, P. Kyasanur, and N. H. Vaidya, "Design and implementation of a multi-channel multi-interface network," in *ACM REALMAN*, 2006.

[60] H.-S. W. So, J. Walrand, and J. Mo, "McMAC: A parallel rendezvous multi-channel MAC protocol," in *IEEE WCNC*, 2007.

[61] J. Polastre, R. Szewczyk, and D. Culler, "Telos: Enabling ultra-low power wireless research," in *ACM/IEEE IPSN/SPOTS*, April 2005.

[62] Y. Kim, H. Shin, and H. Cha, "Y-MAC: An energy-efficient multi-channel MAC protocol for dense wireless sensor networks," in *ACM/IEEE IPSN*, 2008, pp. 53–63.

[63] H. Cha, S. Choi, I. Jung, H. Kim, H. Shin, J. Yoo, and C. Yoon, "RETOS: Resilient, expandable, and threaded operating system for wireless sensor networks," in *ACM/IEEE IPSN*, 2007.

[64] Tmote Sky, "http://www.sentilla.com/pdf/eol/tmote-sky-datasheet.pdf."

[65] Y. Wu, J. A. Stankovic, T. He, and S. Lin, "Realistic and efficient multi-channel communications in wireless sensor networks," in *IEEE Infocom*, 2008, pp. 1867–1875.

[66] H. K. Le, D. Henriksson, and T. Abdelzaher, "A practical multi-channel media access control protocol for wireless sensor networks," in *ACM/IEEE IPSN*, 2008, pp. 70–81.

[67] J. Mo, H. W. So, and J. Walrand, "Comparison of multichannel mac protocols," in *ACM MSWiM*, 2005.

[68] J. Broch, D. A. Maltz, D. B. Johnson, Y.-C. Hu, and J. Jetcheva, "A performance comparison of multi-hop wireless ad hoc network routing protocols," in *ACM MobiCom*, New York, NY, USA, 1998, pp. 85–97.

[69] I. F. Akyildiz, T. Melodia, and K. R. Chowdhury, "A survey on wireless multimedia sensor networks," *Computer Networks*, no. 51, pp. 921–960, 2007.

[70] Crossbow Technology Inc., http://www.xbow.com.

[71] T. Luo, M. Motani, and V. Srinivasan, "CAM-MAC: A cooperative asynchronous multi-channel MAC protocol for ad hoc networks," in *IEEE Broadnets*, San Jose, CA, USA, October 2006.

[72] E. W. Weisstein, "Circular segment—from mathworld," http://mathworld.wolfram.com/CircularSegment.html.

[73] T. Luo, M. Motani, and V. Srinivasan, "Altruistic cooperation for energy-efficient multi-channel MAC protocols," in *ACM MobiCom*, Montreal, QC, Canada, 2007.

[74] F. Xue and P. R. Kumar, "The number of neighbors needed for connectivity of wireless networks," *Wireless Networks*, vol. 10, no. 2, pp. 169–181, 2004.

[75] C. L. Fullmer and J. J. Garcia-Luna-Aceves, "Floor acquisition multiple access (FAMA) for packet-radio networks," in *SIGCOMM*, New York, NY, USA, 1995.

[76] J. J. Garcia-Luna-Aceves and C. L. Fullmer, "Floor acquisition multiple access (FAMA) in single-channel wireless networks," *Mobile Networks and Applications*, vol. 4, no. 3, pp. 157–174, 1999.

[77] P. Kyasanur, J. Padhye, and P. Bahl, "On the efficacy of separating control and data into different frequency bands," in *IEEE Broadnets*, October 2005.

[78] X. Yang, N. H. Vaidya, and P. Ravichandran, "Split-channel pipelined packet scheduling for wireless networks," *IEEE Transactions on Mobile Computing*, vol. 5, no. 3, pp. 240–257, 2006.

[79] S.-Y. Ni, Y.-C. Tseng, Y.-S. Chen, and J.-P. Sheu, "The broadcast storm problem in a mobile ad hoc network," in *ACM MobiCom*, 1999.

[80] M. B. Yassein, M. O. Khaoua, L. M. Mackenzie, and S. Papanastasiou, "Improving the performance of probabilistic flooding in manets," in *International Workshop on Wireless Ad-hoc Networks (IWWAN)*, May 2005.

[81] Q. Zhang and D. P. Agrawal, "Dynamic probabilistic broadcasting in manets," *J. Parallel Distrib. Comput.*, vol. 65, pp. 220–233, 2005.

[82] Y. Yu, R. Govindan, and D. Estrin, "Geographical and energy aware routing: A recursive data dissemination protocol for wireless sensor networks," Dept. Computer Science, UCLA, Tech. Rep. UCLA/CSD-TR-01-0023, May 2001.

[83] C. L. Barrett, S. J. Eidenbenz, L. Kroc, M. Marathe, and J. P. Smith, "Parametric probabilistic sensor network routing," in *ACM WSNA*, 2003, pp. 122–131.

[84] T. Roosta, M. Menzo, and S. Sastry, "Probabilistic geographic routing in ad hoc and sensor networks," in *International Workshop on Wireless Ad-hoc Networks (IWWAN)*, May 2005, pp. 122–131.

[85] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," in *IEEE HICSS*, 2000.

[86] O. Younis and S. Fahmy, "Distributed clustering in ad-hoc sensor networks: A hybrid, energy-efficient approach," in *IEEE Infocom*, Hong Kong, China, March 2004.

[87] M. Qin and R. Zimmermann, "VCA: An energy-efficient voting-based clustering algorithm for sensor networks," *Journal of Universal Computer Science*, vol. 12, no. 1, pp. 87–109, 2007.

[88] Y. Xu, J. Heidemann, and D. Estrin, "Geography-informed energy conservation for ad hoc routing," in *ACM MobiCom*, Sept. 2001.

[89] L. Buttyn and P. Schaffer, "PANEL: Position-based aggregator node election in wireless sensor networks," in *IEEE MASS*, Pisa, Italy, Oct. 2007.

[90] B. Chen, K. Jamieson, H. Balakrishnan, and R. Morris, "Span: An energy-ecient coordination algorithm for topology maintenance in ad hoc wireless networks," in *ACM MobiCom*, Sept. 2001.

[91] J. Bruck, J. Gao, and A. Jiang, "Localization and routing in sensor networks by local angle information," in *ACM MobiHoc*, May 2005.

[92] A. Caruso, S. Chessa, S. De, and A. Urpi, "GPS free coordinate assignment and routing in wireless sensor networks," in *IEEE Infocom*, March 2005.

[93] T. Luo, M. Motani, and V. Srinivasan, "Cooperative asynchronous multichannel MAC: Design, analysis, and implementation," *IEEE Transactions on Mobile Computing*, vol. 8, no. 3, pp. 338–52, March 2009.

[94] L. M. Feeney and M. Nilsson, "Investigating the energy consumption of a wireless network interface in an ad hoc networking environment," in *IEEE Infocom*, 2001.

[95] Cisco Systems Inc., "Cisco aironet 350 series client adapters," http://www.cisco.com/en/US/products/hw/wireless/ps4555.

[96] C.-F. Huang and Y.-C. Tseng, "The coverage problem in a wireless sensor network," *Mobile Networks and Applications*, vol. 10, no. 4, pp. 519–528, August 2005.

[97] P. Berman, G. Calinescu, C. Shah, and A. Zelikovsky, "Power efficient monitoring management in sensor networks," in *IEEE WCNC*, 2004.

[98] R. M. Karp, "Reducibility among combinatorial problems," in *Complexity of Computer Computations*. New York, USA: Plenum Press, 1972.

[99] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*, 2nd ed. New York: MIT Press and McGraw-Hill, 2001.

[100] N. Alon, D. Moshkovitz, , and M. Safra, "Algorithmic construction of sets for k-restrictions," *ACM Transactions on Algorithms (TALG)*, vol. 2, no. 2, pp. 153–177, April 2006.

[101] T. Luo, M. Motani, and V. Srinivasan, "Analyzing DISH for multi-channel MAC protocols in wireless networks," in *ACM MobiHoc*, Hong Kong, China, 2008.

[102] HostAP, http://hostap.epitest.fi/.

[103] MadWifi, http://madwifi.org/.

[104] TinyOS Community Forum, http://www.tinyos.net.

[105] Chipcon Corporation, "CC2420 2.4 GHz Zigbee/802.15.4 RF Transceiver," http://www.chipcon.com.

[106] K. Whitehouse, A. Woo, F. Jiang, J. Polastre, and D. Culler, "Exploiting the capture effect for collision detection and recovery," in *EmNets*, 2005.

[107] J. Zhao and R. Govindan, "Understanding packet delivery performance in dense wireless sensor networks," in *ACM SenSys*, 2003.

[108] N. Reijers, G. Halkes, and K. Langendoen, "Link layer measurements in sensor networks," in *IEEE MASS*, 2004.

[109] D. Lal, A. Manjeshwar, F. Herrmann, E. Uysal-Biyikoglu, and A. Keshavarzian, "Measurement and characterization of link quality metrics in energy constrained wireless sensor networks," in *IEEE GlobeCom*, 2003.

[110] "Cognitive radio," http://en.wikipedia.org/wiki/Cognitive_radio.

[111] T. Luo and M. Motani, "Cognitive DISH: Virtual spectrum sensing meets cooperation," in *IEEE SECON*, Rome, Italy, June 2009.

[112] F. A. Tobagi and L. Kleinrock, "Packet switching in radio channels: Part II–the hidden terminal problem in carrier sense multiple-access and the busy-tone solution," *IEEE Trans. Commun.*, vol. COM-23, no. 12, pp. 1417–33, December 1975.

[113] NRF POC Awardees - 1st Grant Call (Sep 2008), "DISH: Enabling cooperative multi-channel communication for wireless ad hoc networks," http://www.nus.sg/ilo/faculty/NRF_POC_Awardees_POC1.html.

Publications

Journal Papers

- T. Luo, M. Motani, and V. Srinivasan, "Cooperative Asynchronous Multi-channel MAC: Design, Analysis, and Implementation," ***IEEE Transactions on Mobile Computing***, Vol. 8, No. 3, pp. 338-52, March 2009.
- T. Luo, V. Srinivasan, and M. Motani, "Energy-Efficient Strategies for Multi-channel MAC protocols in Ad Hoc Networks," ***IEEE/ACM Transactions on Networking***, under the third round of review.
- T. Luo, V. Srinivasan, and M. Motani, "A Metric for DISH Networks: Analysis, Implications, and Application," ***IEEE Transactions on Mobile Computing***, under the second round of review.

Conference Papers

- T. Luo, M. Motani, and V. Srinivasan, "Analyzing DISH for multi-channel MAC protocols in wireless networks," in ***ACM MobiHoc***, Hong Kong, China, 2008.
- T. Luo, M. Motani, and V. Srinivasan, "Altruistic cooperation for energy-efficient multi-channel MAC protocols," in ***ACM MobiCom***, Montreal, QC, Canada, 2007.
- T. Luo, M. Motani, and V. Srinivasan, "CAM-MAC: A cooperative asynchronous multi-channel MAC protocol for ad hoc networks," in ***IEEE Broadnets***, San Jose, CA, USA, October 2006.