Distributed Semi-Synchronous Channel Coordination for Multi-Channel Wireless Networks

Taewoon Kim, Jaeseon Hwang, and Hyuk Lim*

Abstract

In multi-channel wireless networks, multi-channel diversity can increase the number of concurrent transmissions and thus improve the throughput performance as data transmission on a wireless channel does not interfere with transmissions on the other non-overlapping channels. However, multi-channel coordination may cause severe performance degradation due to hidden terminals, missing receivers, or broadcast deafness problems if the channel usage information is not properly shared among the neighboring nodes. In this paper, we devise a semi-synchronous multi-channel coordination protocol that enables wireless nodes to: i) efficiently exchange channel and coordination information, and ii) reduce the overhead of channel switchings. In the proposed protocol, a rendezvous interval is set up in a distributed manner depending on the traffic rate and pattern, and each node independently switches its channel when it can complete its transmissions and then returns to the control channel within the rendezvous interval. This approach makes all nodes return to the control channel at almost the same time without incurring a severe synchronization overhead. Through subsequent analyses and simulation studies, we show that the proposed protocol effectively reduces the number of channel switchings, thereby achieving higher throughput in various multi-channel networking environments.

Index Terms

Multi-channel, channel coordination, medium access control, wireless networks.

* Corresponding author. Tel.: +82 62-715-2229. E-mail address: hlim@gist.ac.kr.

T. Kim is with Telecommunications Technology Association, Seohyeon-dong 267-2, Bundang-gu, Seongnam-city, Gyonggi-do, Republic of Korea.

J. Hwang and H. Lim are with the Department of Information and Communications, Gwangju Institute of Science and Technology, Gwangju 500-712, Republic of Korea.

I. Introduction

In the early stages of wireless networking, multiple non-overlapping channels were utilized with restricted functionalities. For example, in community networks, mobile clients are allowed to use the best channel among several available channels, and once a connection is established, the client should stay on that channel as long as the current connection lasts. In these networks, clients are not allowed (or able) to dynamically switch channels to obtain better conditions, such as to achieve a higher signal-to-noise ratio (SNR). Moreover, this limited operation is possible only when there is a central coordinator that arbitrates the channel usage.

Recently, wireless mesh networks have been drawing increased attention due to their ease of deployment and management, and so have been considered as an efficient solution for supporting diverse types of network-wide connectivity. As a result, this state-of-the-art technology has led to a rapid growth in device population. Unfortunately, this increased number of wireless devices in a limited area can potentially cause serious problems because of the sharing-based nature of wireless networks. Thus, the performance degradation problem due to the sharing of limited bandwidth resources has motivated a series of works that exploit channel diversity in a distributed networking environment. In this environment, transmissions on a channel do not interfere with other transmissions on non-interfering channels, thereby enabling an increase in the number of concurrent transmissions and improving network performance.

However, one problem in exploiting multiple non-overlapping channels is that existing single-channel protocols are found to be inappropriate, as they cannot effectively deal with a number of potential difficulties while operating in multi-channel networks. These difficulties specifically emerge when nodes are distributed over multiple non-interfering channels; for example, multi-channel hidden terminals [1], missing receivers [2], and broadcast deafness problems. Among existing multi-channel solutions, synchronous protocols such as [1], [3]–[10] show higher efficiency than asynchronous solutions [2], [11]–[16], even with no additional transceiver. When the network is synchronized, exchanging channel status and receiver information becomes much easier, which then contributes to the efficient coordination of multi-channel operation. Some synchronous protocols focus on reducing the amount of control message exchanging, resulting in higher channel utilization; others attempt to achieve tight synchronization in spite of the increase of complexity in both implementation and operation. Thus, in order to enhance the performance of multi-channel protocols, balancing of channel utilization against synchronization overhead is highly required.

In this paper, in order to efficiently coordinate multi-channel operation and thereby enhance channel utilization with minimal channel coordination overhead, we devise a *distributed and semi-synchronous multi-channel coordination* (DiSC) protocol using a single transceiver. Though synchronization has advantages in terms of the ability to coordinate multi-channel operations, it may also increase complexity in both implementation and operation. To balance the tradeoff between these two factors, *we reduce the level of synchronization while taking advantage of synchronous operation*. With DiSC, wireless nodes can rendezvous on a channel at *almost* the same time, but require no additional synchronization process. This rendezvous not only effectively handles multi-channel problems, but also increases channel utilization. In addition, DiSC reduces the channel switching overhead by performing multiple packet transmissions per channel switching. To the best of our knowledge, DiSC is the first semi-synchronous wireless multi-channel protocol, which is the main contribution of this paper.

The remainder of this paper is organized as follows. We provide a background for multiple channel coordination in Section II, and an overview of related work in Section III. In Section IV, we illustrate the design principles and aspects of the DiSC protocol in further detail, and then perform analytic and simulation studies in Section V and Section VI, respectively. Finally, we conclude this paper in Section VII by discussing future directions of this work.

II. Background

In this section, we briefly introduce the background of multi-channel coordination and summarize key issues that should be paid careful attention to when designing a multi-channel protocol.

A. Multi-channel Coordination

Multi-channel coordination for exploiting multiple channel diversity has received significant attention because it can increase the network throughput capacity by enabling multiple concurrent transmissions without interfering with each other. In order to exploit this potential of multi-channel diversity, competing nodes should use non-overlapping (so called *orthogonal*) channels to guarantee that communication on one channel does not interfere with any of the other channels. The number of non-overlapping and orthogonal channels is 3 and 12 in IEEE 802.11b and IEEE

802.11a [17], respectively. However, since the number of sender-receiver pairs is typically larger than the number of the orthogonal channels, a dedicated channel cannot be statically allocated to each pair. Therefore, it is important to coordinate the use of multiple channels among contending nodes in designing a multi-channel protocol.

B. Difficulties in Multi-channel Coordination

Unfortunately, without a central coordinator it is quite difficult to make a decision on how to allocate channels among contending nodes. Misguided multichannel allocation may lead to the following three problems: hidden terminals, missing receiver, and broadcast deafness.

1) *Multi-channel hidden terminal:* When sender-receiver pairs are exchanging control messages about their channel usage, some of neighboring nodes cannot overhear the control messages if they are on a different channel. Because of the incomplete channel usage information, a sender-receiver pair may attempt to use a data channel that is currently occupied by other nodes, thereby experiencing collisions with other nodes.
2) *Missing receiver:* When a sender fails to identify the channel where its intended receiver currently resides, the missing receiver problem occurs, resulting in a number of unsuccessful transmission attempts.
3) *Broadcast deafness:* When a node broadcasts a message, not all neighboring nodes within its transmission range reply to the broadcast message because some of them are on different channels, which can disturb routing or network management activities.

In a nutshell, most multi-channel coordination problems including above three problems arise because it is quite difficult that wireless nodes have full knowledge on channel usage of other nodes on multi-channel networks. It is mainly due to the limitation that wireless nodes are equipped with a half-duplex transceiver, and are restricted to listen or transmit only on a single channel at a time.

III. Related Work

In this section, we will summarize previous research efforts according to whether they rely on the synchronization among nodes for multichannel coordination or not. If the network is synchronized, exchanging channel status and receiver information becomes much easier, which then contributes to the efficient coordination of multi-channel operation. However, the synchronization among nodes may incur the increase of complexity in both implementation and operation, resulting in the degradation of channel utilization. Therefore, it is important how to balance channel utilization against synchronization overhead in multi-channel coordination protocols. In this section, we review synchronous and asynchronous multi-channel protocols.

A. Synchronous Protocols

So *et al.* proposed the multi-channel MAC (MMAC) [1] protocol, which adopts the IEEE 802.11 power saving mechanism (PSM) to synchronize clocks between neighboring nodes. MMAC separates time into two fixed sessions; one for negotiation and the other for data transmission. During the negotiation session, nodes exchange control packets so that sender-receiver pairs can switch to a data channel in the following session. Previously, Chen *et al.* [4] devised MAP, which allows the data transmission phase to have a variable length depending on the previous negotiation; in addition, MAP removes contentions during the data transmission phase using a scheduling algorithm. One advantage of these two protocols is that they only use one interface, so there is no additional hardware implementation cost compared to the commodity in 802.11 wireless devices. However, it should be noted that the negotiation phase of both protocols should be long enough to accommodate all requests, which limits the maximum achievable throughput performance.

Tzamaloukas *et al.* devised the channel hopping multiple access (CHMA) protocol [9], in which nodes continuously switch channels according to the common hopping sequence. In CHMA, if a sender succeeds in exchanging control messages with its intended receiver, both nodes stop hopping and start data transmission. When the pair completes its transmission, both nodes re-synchronize and follow the previous hopping sequence. Then, they further improved their work by guaranteeing collision-free transmissions of multicast and broadcast packets [9]. The main advantage of both protocols is that they require neither an additional interface nor a dedicated control channel. However, frequent channel hopping and the need of tight synchronization incur overhead in both implementation and operation.

In contrast to the above protocols, McMAC [8] is quite different in that it can make parallel rendezvous on a different channel. In McMAC, a node performs periodic channel switching according to its pseudo-random hopping sequence. If there are pending messages in the queue, a sender temporary deviates from its default sequence and transmits to a receiver on another channel. In SSCH [3], nodes periodically tune to another channel according to their randomized hopping sequence. If a sender wants to transmit a packet to a receiver, it first tries to rendezvous with its corresponding recipient; the sender then changes its hopping schedule so that it can overlap with the receiver. Patel *et al.* [7] divided the network into several subnetworks, and then allocated different channel hopping sequences for each network. The transmission sequence is such that each subnetwork can rendezvous with other subnetworks during every channel hopping schedule.

Li *et al.* [5] proposed an on-demand multi-channel protocol on a clustered network. A different type of node, called aggregator, schedules the medium access, thereby making each sensor node operate in a contention-free manner. The scheduling mechanism is traffic adaptive and QoS-aware. Moreover, Automatic Repeat reQuest (ARQ), which is an error-control method for reliable data transmission that retransmits the lost data frames detected by acknowledgements and timeouts, is performed by making use of the unused spectrum to further improve throughput performance. Zhou *et al.* proposed MMSN [6], which assigns frequencies such that the nodes within 2-hop range use different channels. In addition to an efficient broadcasting support, they also introduced an optimal non-uniform backoff algorithm. TMCP is a tree-based approach introduced by Wu *et al.* in [10]. They first performed an empirical study on the number of available orthogonal channels under the consideration of the effects of interferences from IEEE 802.11 based networks. TMCP constructs disjoint trees and assigns different channels to each tree so as to exploit parallel transmissions among multiple trees.

B. Asynchronous Protocols

Shi *et al.* [2] proposed AMCP as a way of alleviating the starvation problem, which is the phenomenon that a few dominating flows take most of bandwidth; thereby the rest of flows get little or none. Each sender-receiver pair decides on a data channel according to its own internal channel table. When an agreement is reached, both nodes switch to the data channel and transfer one DATA/ACK; after that, they return to the control channel. Luo *et al.* devised CAM-MAC [13] which co-operatively exchanges channel and node information. In CAM-MAC, the basic operation is similar to AMCP, but when the requested channel or receiver is temporarily unavailable the neighbor nodes can notify the sender of this unavailability. Note that both protocols use only one transceiver. Conversely, DCA [14] and DPC [11] use two transceivers; one is fixed to the control channel, and the other dynamically tunes in to a data channel to exchange DATA/ACK. Both DCA and DPC follow the same basic operations as single-transceiver multi-channel protocols.

Nguyen *et al.* [16] extended the basic IEEE 802.11 RTS/CTS mechanism in order to avoid collisions in asynchronous ad hoc networks. After exchanging RTS/CTS, both sender and receiver update their channel usage information by sending an ATS (Announce To Send) packet. Once the sender and receiver return to the common channel, they only observe the common channel to avoid collision.

The major drawback of these asynchronous approaches is their low channel utilization. Nodes transfer only one data packet for two channel switchings. In addition, when the network is congested, the control channel may become bottlenecked. In this case, packet collisions can further reduce the number of successful negotiations, and thereby degrade the overall throughput.

DB-MCMAC [15] is somewhat different from other asynchronous protocols because it uses the same number of transceivers and channels. Therefore, it does not require a dedicated channel for exchanging control messages since every channel can be monitored. A node maintains per-neighbor queues, and any idle transceiver can dynamically detach a packet from its queue and transmit it. The advantage of this protocol is that the best channel for a receiver is used for transmission, with a high probability, because it tracks the state of each channel. However, the primary drawback of DB-MCMAC is the hardware constraint in which each transceiver has to simultaneously transmit and receive signals on the assigned channel.

Tanigawa *et al.* [12] introduced an additional transceiver to effectively utilize the multi-channel resources. While one transceiver is fixed to the control channel, the other is dynamically switched among data channels. A sender performs multiple packet transmission and dynamic receiver selection with a new buffer structure.

IV. Proposed Multi-Channel Coordination Protocol

We propose a *distributed semi-synchronous multi-channel coordination* (DiSC) protocol that efficiently exchanges channel information among competing nodes and thereby reduces the channel switching overhead.

The key feature of the DiSC protocol is its *semi-synchronization*, with which it achieves a certain level of synchronization without strict clock synchronization processes; thus, significantly reducing the degree of multi-channel coordination complexity. In the DiSC protocol, there is a dynamically established interval within which all sender-receiver pairs can independently perform negotiation and data transmission on the control and data channels, respectively, but must return to the control channel no later than the end of the interval, referred to as the *rendezvous interval*. Because all nodes are stationed on the control channel at the end of each rendezvous interval, the DiSC protocol achieves both efficient multi-channel information exchanges and fail-safe coordination of channel usage in a distributed and synchronous manner.

The DiSC protocol is designed under the following assumptions: (i) there are M non-overlapping channels, (ii) one of the channels is the control channel, which is known to all the nodes, and (iii) all other channels except the control channel are data channels that are to be used for payload delivery.

A. Semi-Synchronous Coordination

The DiSC protocol allows each node to independently perform negotiation, followed by data transmission; both constitute a transmission session. However, the sessions established by different nodes (on different channels) are roughly synchronized within a rendezvous interval. Here, the length of the rendezvous interval is determined by the first node that establishes a session among the contending nodes.

The DiSC protocol consists of two stages: *negotiation* and *data transmission*, and its overall procedure is as follows.

(p1) During the negotiation stage, a sender checks whether or not there exists an ongoing rendezvous interval.

(p2) If a session is not found, it sends a multichannel-request-to-send (mRTS) control packet to its intended receiver. Otherwise, it first makes a decision on whether or not it can complete data transmissions within the current rendezvous interval, and if possible, it transmits an mRTS.

(p3) If the receiver is available to receive as many packets as the sender notified, it sends a multichannel-clear-to-send (mCTS) control packet.

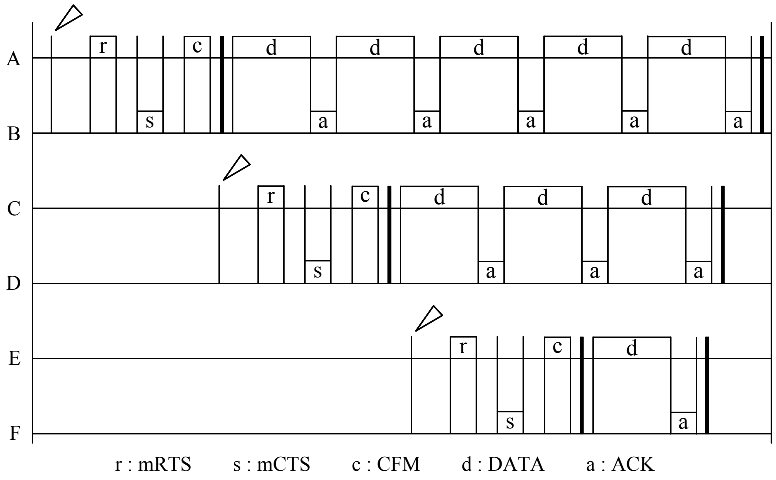


Fig. 1. Timing diagram of the DiSC protocol for three sender-receiver pairs. The bold lines indicate the channel switchings.

(p4) On receiving the mCTS packet, the sender broadcasts a confirmation (CFM) control packet, which includes the channel and duration information.

(p5) Both the sender and receiver switch their channel to the data channel and start their data transmissions.

(p6) After the data transmissions are completed, they return to the control channel. All the nodes will return almost at the same time even though they started at different instances within the rendezvous interval.

Fig. 1 presents the timing diagram of the negotiation and transmission operations of DiSC for three sender-receiver pairs. First, the node $\mathcal{A}$ starts to negotiate with the node $\mathcal{B}$ for data transmission. While these two nodes are negotiating on the control channel, the other nodes overhear the intended channel and time duration for the transmission session of $\mathcal{A} \rightarrow \mathcal{B}$. After $\mathcal{A}$ and $\mathcal{B}$ switch to a data channel, the nodes $\mathcal{C}$ and $\mathcal{D}$ start their negotiation. They decide to transfer only three packets so as to return to the control channel before the rendezvous interval ends. As illustrated in the figure, they immediately return to the control channel as soon as they complete their transmissions on the data channel. In this way, it is guaranteed that all three sender-receiver pairs will return to the control channel by the end of the rendezvous interval, in preparation for the negotiation of the next transmission session.

B. Procedure of the DiSC Protocol

In this section, we discuss further details of each procedure in the DiSC protocol.

Stage 1: Negotiation

We define two types of transmission sessions. The first session in the rendezvous interval is called the *prime session* S_{pr}; the sessions following S_{pr} in the same interval are then referred to as *inter-sessions* S_{in}.

1) *Determining the session type*

   The nodes should decide which type of session is to be established. To determine the session type, a sender node checks its channel table.[1] If there is no ongoing transmission session, it initiates a new rendezvous interval, and this transmission session between the sender and its receiver becomes S_{pr}. If a prime transmission session already exists, the node should establish S_{in}.

2) *Contention on the control channel*

   The length of S_{pr} should be large enough to support multiple data transmissions on the data channel because it limits the maximum duration of S_{in}; if the rendezvous interval is too short, it incurs frequent channel switchings. For example, in Fig. 1, the number of transmissions in S_{pr} (i.e., the transmissions of $\mathcal{A} \rightarrow \mathcal{B}$) is 5, and those for S_{in} are smaller. Because a sender cannot change its receiver on the data channel, the prime session should be established by a node that has a sufficient number of packets destined to a specific receiver. In order to give a higher priority to a node with a long queue, we exploit the contention mechanism in the IEEE 802.11 DCF, in which the contention window (CW) size of each node is inversely proportional to the number of packets destined to a specific receiver in its queue. We elaborate on this issue in Section IV-D1.

3) *Sending an mRTS packet*

   Prime session: The length of S_{pr} (T_{pr}) is the length of the rendezvous interval, which includes the backoff delay and negotiation time on the control channel, and the data transmission and channel switching time on a data channel. Let N_{pr} denote the number of transmissions that the sender performs during the data transmission stage. Then, the

[1] Each node updates its channel table by eavesdropping on the control packets on the control channel.

length of the transmission stage (τ_{tr}^{pr}) is given by

$$\tau_{tr}^{pr} = \min\left(N_{pr} \times T_{tr}, T_{max}\right), \tag{1}$$

where T_{max} is the maximum bound for S_{pr}, and T_{tr} is the duration for completing one transmission. Note that $T_{tr} = \sigma_{DATA} + \sigma_{SIFS} + \sigma_{ACK} + \sigma_{SIFS}$, where σ_{SIFS} is the SIFS slot length, and σ_{DATA} and σ_{ACK} are the times required for transferring DATA and ACK packets at a default transmission rate for each frame. Then, the sender sends its intended receiver an mRTS packet that includes the length of the transmission stage and a set of channels that are currently available to the sender.

Inter-session: S_{in} is allowed only when the transmissions can be completed before the rendezvous interval ends, so that all nodes can make a control channel rendezvous at almost the same time. Since the nodes learn the length of the rendezvous interval by overhearing control messages on the control channel, they can calculate how many packets can be transferred within the remaining duration. The number of packets that a sender can transfer (N_{in}) is derived as follows:

$$N_{in} = \left\lfloor \frac{T_{pr} - t_{passed} - \tau_{nego} - 2 \cdot d_{ch}}{T_{tr}} \right\rfloor, \tag{2}$$

where t_{passed} is the time elapsed since the ongoing S_{pr} starts, d_{ch} is the channel switching delay, and τ_{nego} is the required time for exchanging control messages with the intended receiver, and is given by

$$\tau_{nego} = \sigma_{mRTS} + \sigma_{SIFS} + \sigma_{mCTS} + \sigma_{SIFS} + \sigma_{CFM},$$

where σ_{mRTS}, σ_{mCTS}, and σ_{CFM} are the mRTS, mCTS, and CFM slot length, respectively. If N_{in} is non-zero, the sender sends an mRTS packet to its receiver. Otherwise, the sender stops its channel negotiation.

4) *Sending an mCTS packet*

When a receiver gets an mRTS packet, it first compares the received available channel list with its own channel table, and picks one of the channels that is commonly available to both the sender and the receiver in order to transmit the data. Because the channel information at both the sender and receiver may not be consistent, the sender sends a list of all available channels and the receiver selects one channel for data transmission.

The receiver also selects the highest transmission rate from the SNR of the mRTS packet sent by the sender, and re-computes the duration of the data transmission on the data channel. Finally, the receiver packages the channel, duration, and rate information in an mCTS packet, transmits it to the sender, and then switches to the data channel immediately.

5) *Send a CFM packet*

Upon receiving an mCTS packet, the sender includes both channel and duration information in a CFM (confirmation) packet, broadcasts it, and then switches to the data channel. The purpose of sending a CFM packet is to notify the channel and provide duration information to nodes in the vicinity of the sender.

Stage 2: Data Transmission

When the sender-receiver pair switches to a data channel, the sender performs carrier-sensing on the channel to check whether or not ongoing transmissions exist. Note that even though each node overhears the control packets on the control channel, the gathered channel usage information may be incomplete due to packet collisions and losses. To prevent potential collisions on the data channel from happening, the sender should sense the channel; if there is no ongoing transmission on the channel for DIFS slot time, it may then start to transmit data.

1) *Data transmission*

When the sender sends the first data packet on the data channel, it uses the rate instructed by the receiver during the negotiation stage. After that, the sender may adjust the transmit rate according to the SNR information from the receiver. This is because the channel status of the data channel may be different from that of the control channel. However, if the transmit rate is lower than that selected during the negotiation stage on the control channel due to lower SNR on the data channel, the sender will not be able to meet the rendezvous interval. In this case, it may give up some of the intended transmissions in order to return to the control channel in time.

2) *Returning to control channel*

After completing the data transmissions, which are scheduled during the negotiation session, all the nodes return to the control channel. Since each sender-receiver pair performs maximum number of transmissions before the current S_{pr} ends, neighboring nodes can meet

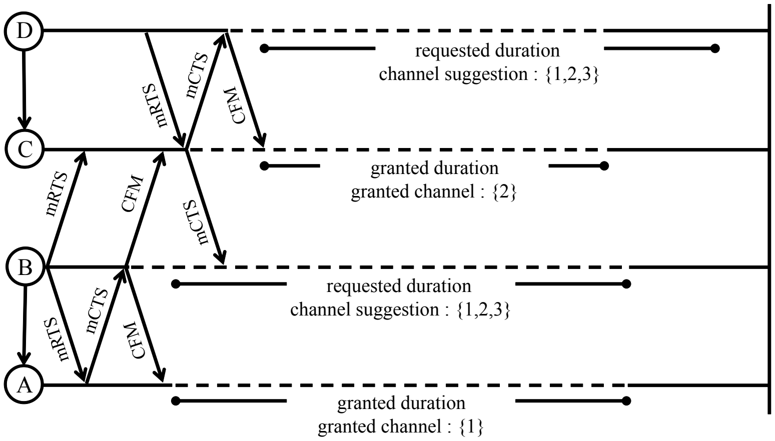


Fig. 2. An example of the semi-synchronous operation of DiSC. In each row of a node, a dotted or straight part stands for the presence on a control or a data channel, respectively.

each other on the control channel almost at the same time. When the current rendezvous interval ends, the nodes may initiate another rendezvous interval by repeating these steps again.

C. Example of the Semi-Synchronous Operation

To illustrate the overall procedure of DiSC, we provide an example with a simple topology consisting of four nodes as shown in Fig. 2. Here, it is assumed that the sender nodes $\mathcal{B}$ and $\mathcal{D}$ are backlogged, and the number of available data channels is three.

At the beginning, all nodes are on the control channel, and there is no ongoing negotiation on the control channel and no data transmission on the data channels. First, the node $\mathcal{B}$ triggers its negotiation stage by sending an mRTS packet to the node $\mathcal{A}$. The mRTS packet contains the channel list. Since there is no ongoing session, the node $\mathcal{B}$ requests the maximum duration for data transmission with a set of available channels, e.g., $\{1, 2, 3\}$, where each number indicates a channel index. Acquainted with the absence of a prime-session, the node $\mathcal{A}$ agrees to the duration by replying with an mCTS, including a randomly selected channel, e.g., $\{1\}$, from the received channel list, since all channels in the list are currently available. Upon receiving the mCTS, the node $\mathcal{B}$ broadcasts the agreements, i.e., which channel will be used and how long it will be, by transmitting a CFM, and then switches to the channel 1. The node $\mathcal{A}$ also switches to the channel 1 right after receiving the CFM. Then the nodes $\mathcal{A}$ and $\mathcal{B}$ perform the data transmissions on the data channel during the agreed duration.

The node $\mathcal{D}$ now performs the channel negotiation with the node $\mathcal{C}$. While the node $\mathcal{C}$ learns the channel usage information by overhearing the CFM on the control channel, the node $\mathcal{D}$ cannot. Because of the lack of channel information, the node $\mathcal{D}$ tries to trigger a prime-session and regards the channel 1 as an available candidate channel. However, the node $\mathcal{C}$ has the knowledge of the ongoing prime-session, so it reduces the transmission duration so that both nodes $\mathcal{C}$ and $\mathcal{D}$ can return to the control channel before the node $\mathcal{B}$ returns. In addition, the node $\mathcal{C}$ excludes the channel 1 being used by the nodes $\mathcal{A}$ and $\mathcal{B}$, and it selects another available channel. After the data transmissions, each sender-receiver pair on a different channel returns to the control channel at the end of its reserved duration. Here, the moments of each pair's return are close enough not to cause any multi-channel problems.

D. DiSC Features

DiSC has several distinguishing features for enhancing channel utilization as well as for preventing multi-channel problems, discussed in Section II-B.

1) Concurrent transmission sessions: Under the proposed protocol, once a prime transmission session is established, a number of concurrent inter-transmission sessions can be independently initiated on interference-free data channels. Note that if the length of the prime session is sufficiently long, the number of concurrent transmission sessions will be large, resulting in an increase of the aggregate throughput performance in the multi-channel network.

To increase the number of concurrent transmissions, a node having the largest number of backlogged packets destined to an intended receiver should trigger a prime session. The DiSC protocol gives a higher priority to such a node, allowing it to get the channel faster than neighboring nodes by adjusting the contention window (CW) as follows:

$$CW = \frac{CW_{min} \cdot q_{max}}{N_p} \approx \frac{CW_{min} \cdot q_{max}}{q/N_{neighbor}},$$

where N_p is the number of backlogged packets destined to a receiver, $N_{neighbor}$ is the number of neighboring nodes, and q_{max} and q are the maximum and the current queue sizes, respectively. Here, instead of counting the number of packets for each receiver, we obtain the approximate average number of packets per receiver.

2) Bulk transmission: Once a sender-receiver pair switches to a data channel, the sender may transfer multiple data packets to its receiver before the current S_{pr} ends. This operation contributes to throughput enhancement because it reduces the number of carrier sensings, channel switchings, and negotiations.

3) Transmission rate adaptation: The proposed protocol adopts a threshold based rate adaptation scheme. In other words, the transmission rate is increased/decreased if the measured SNR is above/below a predefined threshold. Initially, the rate for the first data transmission on the data channel is determined after the mRTS/mCTS packet exchange on the control channel, and it is adjusted after the first data transmission on a data channel. This adjusted rate is then used for the remaining data transmissions.

It should be noted here that the rate obtained during the negotiation stage may not be accurate because the SNR is not measured on the data channel. However, considering that the received signal strength is highly dependent on the distance between the sender and receiver, the first rate adaptation is expected to be a reasonable prediction.

4) Semi-synchronization: Gathering at the control channel at almost the same time is the key feature that decreases the complexity of multi-channel coordination. Once a prime session is initiated, nodes perform their independent *negotiate-and-switch* operations, and then return to the control channel by the end of the prime session. As a result, it is guaranteed that all the neighbor nodes are on the control channel at the end of each prime session, leaving all data channels vacant. Therefore, the nodes can easily exchange channel use information with each other. In addition, it can effectively prevent nodes from remaining on specific data channels and from starting to negotiate with a receiver that is already involved in another transmission. This significantly contributes to the coordination of multi-channel operation.

V. Performance Analysis

In this section, we investigate the achievable throughput of the proposed protocol, and then compare the analytically derived results with the simulation results. Our approach is comprised of three main tasks. First, we compute the expected backoff period, i.e., the amount of time wasted by the carrier sense multiple access (CSMA) channel contention mechanism among the contending nodes. Then, we aggregate the length of the transmission stages across all non-interfering data channels. Finally, we derive the achievable throughput. For simplicity of analysis, we assume a *single-cell* ad hoc network, in which every node is located within the transmission range of the other nodes.

A. Step 1: Backoff Counter Distribution

According to the IEEE 802.11 DCF channel acquisition mechanism, a node grasps the wireless channel when the randomly chosen backoff counter (or the contention window, cw) reaches zero. This backoff counter follows a uniform distribution, and collisions occur when the backoff counters of multiple nodes simultaneously expire. In this framework, a node that wants access to the channel sets a backoff counter only when the channel is idle for a DIFS interval (denoted by σ_{DIFS}), and sequentially decrements the backoff counter when the channel is idle for a slot time.

Suppose N senders try to gain access to the channel. Then, they independently set backoff counters within the given range $[0, CW_{max}]$. The individually chosen N backoff counters, which are independent and identically distributed (i.i.d.), are denoted by $x_1, x_2, \cdots, x_N$, where x_i is the backoff counter value of the ith sender. After sorting these values in ascending order, we get: $x_{(1)}, x_{(2)}, ..., x_{(N)}$, where $x_{(1)}$ is the smallest, and $x_{(N)}$ is the largest value among x_i. In this section, we refer to the ith smallest value $x_{(i)}$ as the ith order statistic.

Then, let $\mathcal{S} = (X_1, X_2, ..., X_N)$ be random variables [2] from the distribution function of $F(.)$, which is differentiable. Note that $F(.)$ follows a uniform distribution within the range $[0, CW_{max}]$. In this case, the ith order statistic of the sample space $\mathcal{S}$ is the ith smallest value, denoted as $X_{(i)}$. As such, the probability density function (pdf) of $X_{(i)}$ can be derived as

$$\begin{aligned} f_{X_{(i)}}(x) &= \frac{d}{dx} P\{\text{at least } i \text{ of the } X_{1..N} \text{ are less than or equal to } x\} \\ &= \frac{N!}{(i-1)!(N-i)!} F(x)^{i-1} f(x) (1-F(x))^{N-i}. \end{aligned}$$

From the density function of the backoff counter, we can compute the expected period that each sender should wait before accessing the channel when both N and CW_{max} are given.

B. Step 2: Aggregate Length of Data Transmission Stages

To compute the throughput for the proposed protocol, we first need to determine the number of bits transferred over the time elapsed in the context of the proposed algorithm. Under the

[2] Small x's indicate real values, and large X's indicate random variables.

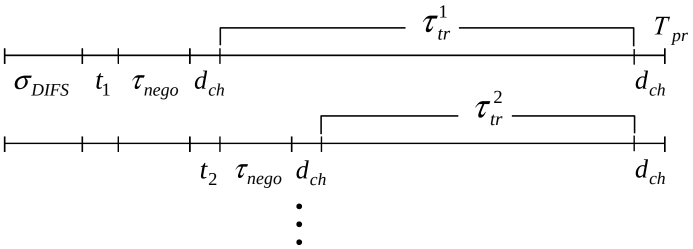


Fig. 3. Timeline of the DiSC operations for the first and second successfully negotiated sender-receiver pairs, which initiate the prime and first inter-sessions, respectively, within the same session.

assumption that the network is saturated, i.e., that every sender is backlogged, and that there is no idle period on the network except at the moment that the senders perform backoff procedures; Fig. 3 illustrates the timeline of both the prime and first inter-sessions. The first sender that grasps the channel establishes the prime session (the topmost timeline in Fig. 3), and the neighboring contending nodes then perform inter-transmissions within the ongoing prime session, of which the length T_{pr} is given by

$$T_{pr} = \sigma_{DIFS} + t_1 + \tau_{nego} + \tau_{tr}^1 + 2 \cdot d_{ch}, \tag{3}$$

where t_1 is the expected backoff period that expires first among the contending nodes (i.e., the 1st order statistic) and is equal to $E[f_{X_{(1)}}(x)]$, and τ_{nego} is the length of the negotiation stage. Note that both the prime and inter-sessions require the same amount of time slots for negotiation. In (3), τ_{tr}^1 is the length of the prime session's data transmission stage (i.e., τ_{tr}^{pr} in (1)), during which the sender-receiver pair does not occupy the control channel because this pair leaves the control channel upon completing the negotiation. Here, the τ_{tr}^1 for a backlogged sender is simply set at its maximum value (T_{max}).

Once the negotiation for the first pair ends, the other senders resume their backoff procedures. From the end of the first negotiation, t_2 is the time at which the second smallest backoff counter expires, where $t_2 = E[f_{X_{(2)}}(x)] - t_1$. As before, control messages are exchanged, and both the sender and receiver leave the control channel for at most τ_{tr}^2, as shown Fig. 3. The data transmission period of the 2nd sender-receiver pair is

$$\tau_{tr}^2 = T_{pr} - \sigma_{DIFS} - E[f_{X_{(2)}}(x)] - 2 \cdot \tau_{nego} - 2 \cdot d_{ch}.$$

In a similar way, the data transmission period of the ith sender-receiver pair can be obtained as

TABLE I

PARAMETERS FOR THE NUMERICAL VALIDATION OF IEEE 802.11B.

Parameter	Value
T_{max}	T_{tr}
d_{ch}	0
τ_{nego}	$\sigma_{mRTS} + \sigma_{SIFS} + \sigma_{mCTS} + \sigma_{SIFS}$

follows:

$$\tau_{tr}^{i} = T_{pr} - \sigma_{DIFS} - E[f_{X_{(i)}}(x)] - i \cdot \tau_{nego} - 2 \cdot d_{ch}.$$

C. Step 3: Achievable Throughput

We then compute the total bits transferred during a prime session. During a prime session, the number of concurrent sessions is given by

$$l = \min(m, M-1),$$

where M is the number of orthogonal channels, and m is the number of negotiations that are successfully made within the current session. Here, m is the smallest integer value such that $\tau_{tr}^{m+1} < T_{tr}$ for $m \leq N$. Note that if the residual time to the rendezvous interval is expected to be too short to perform one DATA/ACK exchange, no new negotiation is initiated on the control channel. As such, the total number of packets sent over the multiple orthogonal channels within a session is

$$\mathcal{K}_{total} = \sum_{i=1}^{l} \left\lfloor \frac{\tau_{tr}^{i}}{T_{tr}} \right\rfloor.$$

Finally, assuming that the average packet size is known, the achievable throughput $\mathcal{C}$ can be expressed as

$$\mathcal{C} = \frac{\mathcal{K}_{total} \cdot E[\psi]}{T_{pr}}, \tag{4}$$

where $E[\psi]$ is the average packet size.

D. Validation

In this section, we compute the throughput performance of the proposed protocol numerically and compare it with the simulation results to validate our throughput analysis. Here, we assume a *single cell* network where wireless nodes are located close to each other so that the transmission

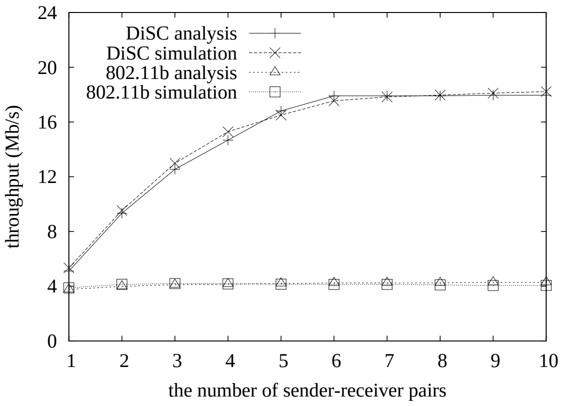


Fig. 4. Throughput results of both analysis and simulation for the proposed protocol (DiSC) and IEEE 802.11b with respect to the number of sender-receiver pairs on a single-cell network with $T_{max} = 5 \cdot T_{tr}$ and 10 orthogonal channels.

of each node can be heard by the other nodes. Here, the control and data packets are transmitted at 2 Mb/s and 11 Mb/s, respectively, the average packet size is 1000 bytes, and all the contending nodes are backlogged. Note that the MAC/PHY parameters are configured to the default values of IEEE 802.11b [17], except for the number of orthogonal channels. For comparison, we take the conventional (single-channel) IEEE 802.11b, with the analytic throughput for IEEE 802.11b being approximately obtained from (4), based on the parameters described in Table I.

1) Achievable throughput with regards to the number of contending nodes: First, we investigate the achievable throughput with respect to the number of contending nodes. Fig. 4 shows the analytical and simulation throughput results for both the proposed algorithm and IEEE 802.11b with respect to the number of sender-receiver pairs. The number of orthogonal channels is set at 10, and the number of sender-receiver pairs varies from 1 to 10. Note that the number of channels is sufficiently large and does not limit the throughput performance.

In Fig. 4, it is seen that the simulation and analysis results of DiSC are almost the same; i.e., the numerical model of DiSC is very accurate. In addition, the analytical results of IEEE 802.11b is in agreement with the simulation results. In case of IEEE 802.11b, the throughput performance remains constant in all the cases, since it does not exploit multiple channel diversity. However, the proposed protocol can effectively utilize multiple channels, thereby achieving significant throughput improvement.

Here, one might wonder why the aggregated throughput of the proposed protocol increases

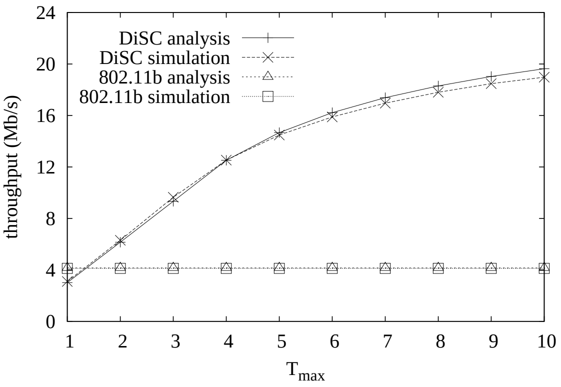


Fig. 5. Throughput results of both analysis and simulation for the proposed protocol (DiSC) and IEEE 802.11b with respect to T_{max} on a single-cell network with 4 sender-receiver pairs and 5 orthogonal channels.

and levels off from the six sender-receiver pairs case. This is due to the parameter of T_{max}, which limits the maximum length of a prime-session. The effect of T_{max} on the behavior of the proposed protocol will be evaluated in Section VI in detail.

2) Achievable throughput with respect to T_{max}: Finally, we study the effects of T_{max} on the achievable throughput. Here, T_{max} is the maximum length of a prime session's data transmission stage, and limits both the number and the length of inter-sessions. The throughput result with respect to T_{max} for the four sender-receiver pairs is shown in Fig. 5. Again, the throughput performance of the numerical models of both the proposed protocol and IEEE 802.11b precisely match their corresponding simulation results.

VI. Performance Evaluation

To evaluate the performance of DiSC, and then to compare it with MMAC, AMCP, and the legacy IEEE 802.11b DCF (with single-channel), we performed simulation studies using the network simulator (NS2) [18] under various network topologies and conditions.

- MMAC allows synchronized nodes to perform channel negotiations and multiple data transmissions during fixed intervals of two sessions under the similar operation of IEEE 802.11 PSM.
- AMCP allows each sender-receiver pair to independently (i.e., asynchronously) perform the channel negotiation immediately followed by data transmissions.

Since MMAC and AMCP are the most representative synchronous and asynchronous protocol, respectively, they can be selected for performance comparison with DiSC, which is a semi-synchronous protocol that has a certain level of synchronization based on asynchronous operation. In addition, none of the protocols requires any central coordinator, and allows each node to use a single transceiver in multi-channel wireless networks.

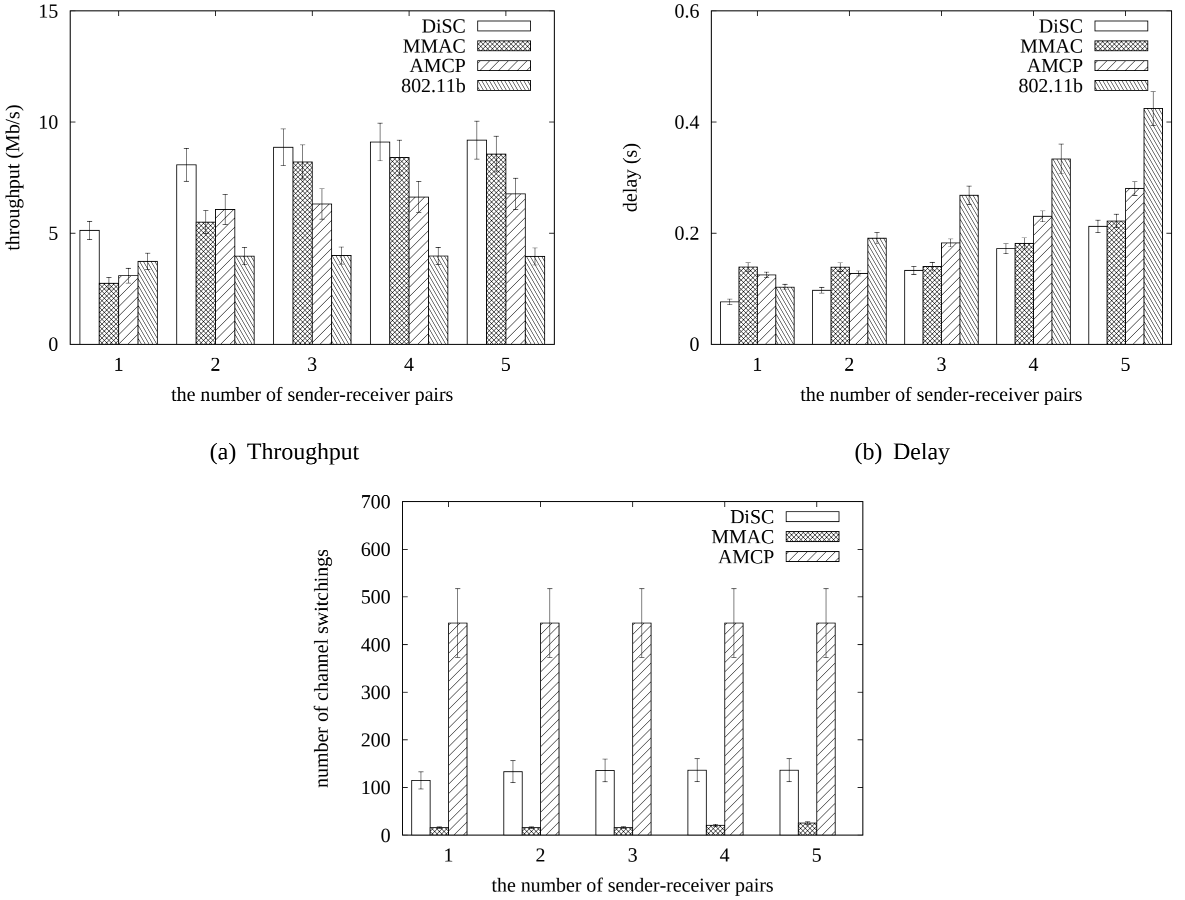
We set the transmission and interference ranges to 100 m and 220 m, respectively. The MAC/PHY parameters are configured to the default IEEE 802.11b parameters with 3 orthogonal channels (if not mentioned differently). The other protocol-specific parameters of MMAC and AMCP are set as described in their original studies: [1] and [2], respectively. For DiSC, T_{max} is set to $5 \cdot T_{tr}$. The data/basic rate is set at 11/2 Mb/s, and each node is configured to generate 11 Mb/s of constant bit rate (CBR) traffic with 1000-byte-long packets. These packets are to be transferred over the user diagram protocol (UDP) protocol. Note that each simulation runs for more than 30 seconds, and the reported values in each figure represent the average of 30 simulation runs.

A. Network topologies

1) Single-cell network: In this scenario, each sender-receiver pair is one hop away, and every node is within the transmission ranges of the other nodes. Therefore, all nodes can sense the transmissions from the other nodes on the network. Here, the number of sender-receiver pairs is varied from 1 to 5, and each pair serves a single UDP flow. The packet sizes are randomly chosen from 500 to 1500 bytes.

Fig. 6 shows the throughput, delay, and channel switching rate of DiSC, MMAC, AMCP, and IEEE 802.11b with respect to the number of the sender-receiver pairs. Note that 95% of confidence intervals (CI) are marked. The channel switching rate is measured as the number of channel switchings per 1 Mb data transfer. As shown in Fig. 6(a), in the case of the legacy IEEE 802.11b, since the traffic rate at which each sender is generating is high enough to fill the bandwidth, the wireless channel is already saturated with only one sender-receiver pair, and thus there is no further throughput improvement even though the number of sender-receiver pairs increases. This shows the inherent inability of multi-channel usage of conventional single channel wireless networks.

On the other hand, multi-channel protocols such as DiSC, MMAC, and AMCP can effectively



(a) Throughput

(b) Delay

(c) Channel switching rate

Fig. 6. Simulation results of DiSC, MMAC, AMCP, and IEEE 802.11b on a single-cell network with 95% of confidence intervals.

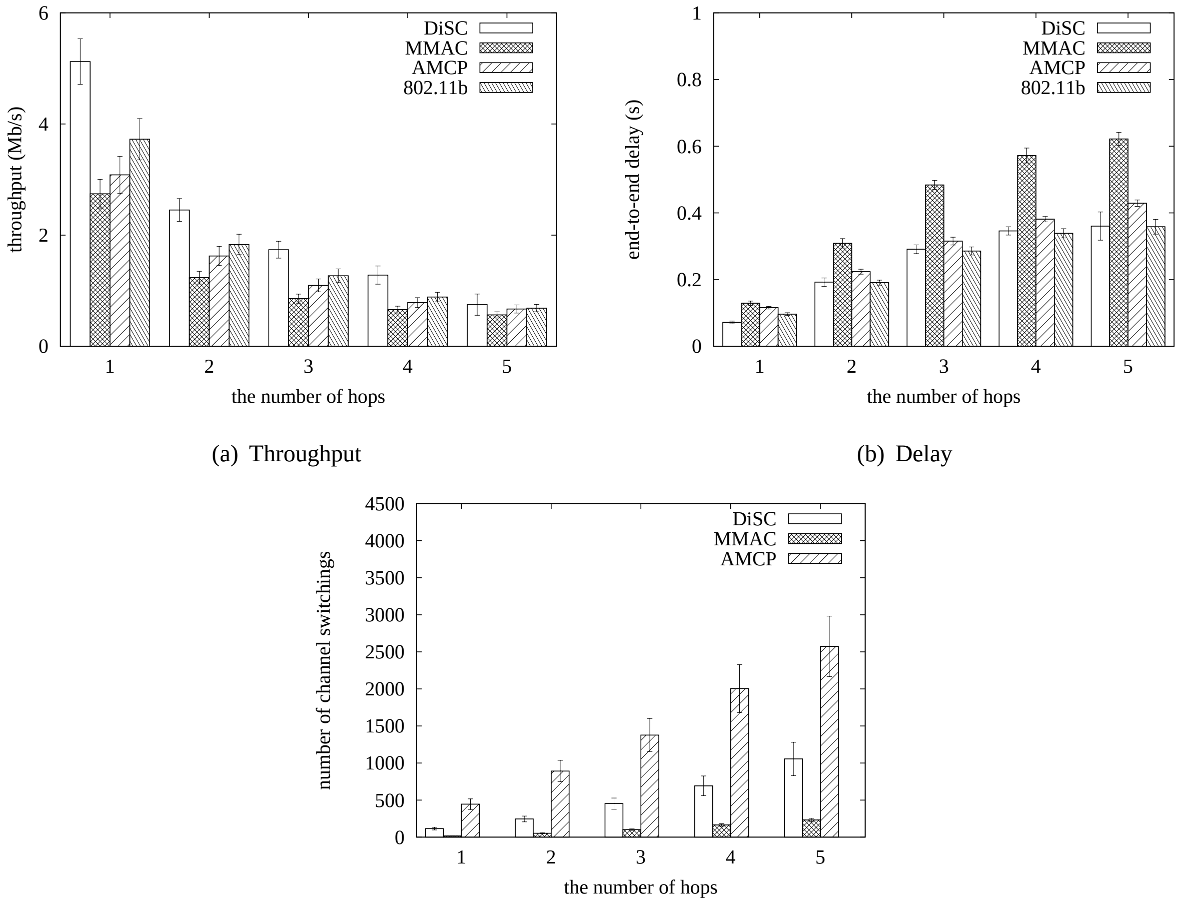
exploit multiple non-overlapping channels, and therefore achieve throughput enhancements. In particular, DiSC shows the highest throughput performance—even in the single-pair case in which both MMAC and AMCP turned out to be less efficient than IEEE 802.11b. These results imply that there is a performance penalty incurred by multi-channel operations of MMAC and AMCP, and that the overhead becomes larger when the number of acting sender-receiver pairs gets smaller. This inefficiency for both protocols is due in part to the following protocol-specific policies: i) AMCP incurs a delay overhead for each data transmission to avoid the multi-channel hidden terminal problem in an asynchronous manner; and ii) MMAC uses fixed lengths for the negotiation and data exchange intervals, and the default values of the intervals are too large for a small number of sender-receiver pairs.

In Fig. 6(b), DiSC achieves the lowest delay for all the cases of this scenario. We observe that the delay performance of IEEE 802.11b is better than that of MMAC and AMCP when the number of sender-receiver pairs is just one, but gets worse as the number of sender-receiver pairs increases. This result shows that the multi-channel protocols have the capability of distributing the traffic load over the multiple channels. On the contrary, under IEEE 802.11b, the wireless channel easily becomes congested as the number of sender-receiver pairs increases, resulting in longer delay of data delivery.

Fig. 6(c) shows the channel switching rates of DiSC, MMAC, and AMCP. Because MMAC periodically switches channels with a fixed interval, it gives the smallest channel switching rate. In comparison with AMCP, DiSC achieves the smaller channel switching rate because the bulk transmission in DiSC effectively reduces the unnecessary channel switchings by enabling multiple data transmissions on a data channel.

2) Multi-hop network: Next, we evaluate the performances of the four protocols on a chain network in which nodes are deployed in a row, and the distance between adjacent nodes is set to 90 m (i.e., one-hop away). In this chain scenario, there is only one flow—which travels n-hops—and all other configurations are exactly the same as in the previous single-cell scenario. The number of hops (n) varies from 1 to 5.

Fig. 7(a) shows the end-to-end throughput results for the chain topology with respect to the number of hops. Note that 95% of CI's are marked. Here, each case in this scenario has $(n + 1)$ nodes. As observed in the single-cell simulations, DiSC generally outperforms the other protocols, with IEEE 802.11b being the second-ranked scheme. In the single-hop network, the performance was seen to vary as the number of sender-receiver pairs increased, with multi-channel protocols tending to show better performance than IEEE 802.11b. However, the throughputs of MMAC and AMCP are slightly lower than that of IEEE 802.11 DCF even though they are able to exploit multiple channels for data transmissions. One reason for these lower throughputs is that because MMAC is a synchronous protocol and uses a fixed value for the channel reservation time, channel utilization is highly affected by the traffic load of each node, which gradually decreases along the path towards the destination. This uneven traffic load on the path causes MMAC to operate inefficiently on multi-hop networks. In AMCP, each node defers its channel negotiation for a certain interval to prevent the multi-channel hidden terminal problem after switching to the control channel. Unfortunately, the interval is too short to fully



(a) Throughput

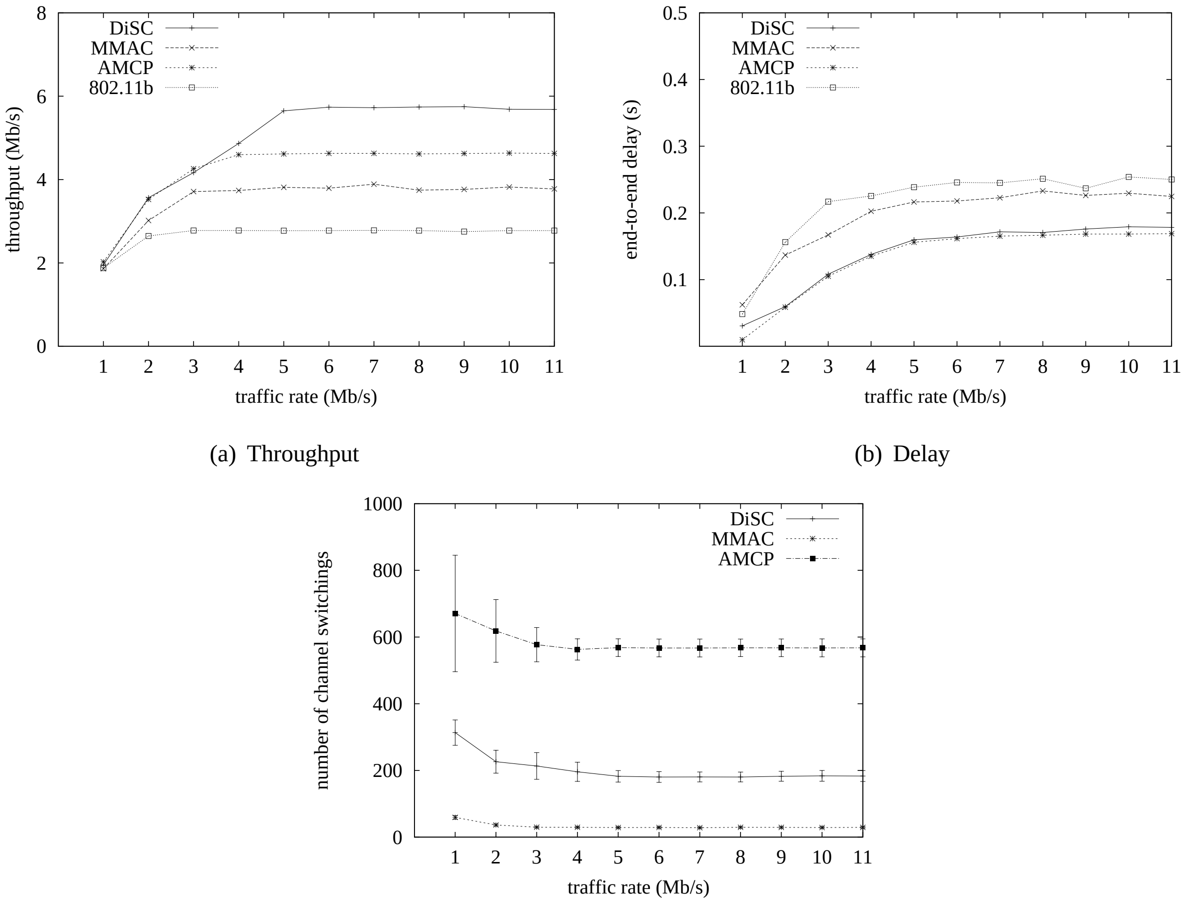
(b) Delay

(c) Channel switching rate

Fig. 7. Simulation results of DiSC, MMAC, AMCP, and IEEE 802.11b on a multi-hop network with 95% of confidence intervals.

prevent the hidden terminal problem in multi-hop networks where the intra-flow interference is severe.

Fig. 7(b) shows the delay performances. The throughput performance on multi-hop network rapidly decreases with respect to the number of hops. Here, DiSC and IEEE 802.11b have the smallest delays, and MMAC has the largest delay. Because the traffic rate near the destination node on the multi-hop path becomes quite low, the delay performances on multi-hop topology show a trend similar to that of single-cell network with one flow in Fig. 6(b). While IEEE 802.11b and AMCP transmit only one packet once they grasp a channel for data delivery MMAC reserves a data channel for a fixed period and transmits more than one packet; until the period is over, awaiting packets cannot be forwarded to the next-hop destination. Therefore, the delay



(a) Throughput

(b) Delay

(c) Channel switching rate

Fig. 8. Simulation results of DiSC, MMAC, AMCP, and IEEE 802.11b in a random network with respect to the offered traffic load.

performance of MMAC can be improved if the period is appropriately adjusted at each hop.

Fig. 7(c) shows the channel switching rates. Overall channel switching rates are almost linearly proportional to the number of hops. As like in Fig. 6(c), MMAC achieves the smallest channel switching rate, and AMCP gives the largest channel switching rate.

3) Random network: To evaluate the throughput performance on more general topologies, we construct a random network where both single-hop and multi-hop flows coexist. On the random network, five sender-receiver pairs are randomly selected, and they establish UDP flows with variable-sized packets.

Fig. 8(a) shows the aggregate throughput performances of DiSC, MMAC, AMCP, and IEEE 802.11b with respect to the offered traffic load. The aggregate throughput represents how many bits are delivered from source nodes to the destination node during unit time. For light traffic load, it can be seen that all protocols show almost equal throughput performance; however, as the loads are increased, the throughput of the protocols become saturated. We observed that the throughput for DiSC becomes saturated at a much higher traffic load than in the other protocols. Fig. 8(b) shows the average delay performances. The average packet delay is the average duration for successfully delivered packets between the source and the destination node. DiSC and AMCP achieve almost the same delay performance while MMAC and IEEE 802.11b have comparatively larger delays. Fig. 8(c) shows the average channel switching rates. As the offered traffic load increases, the average channel switching rates decrease and eventually level off, and MMAC and AMCP have the largest and smallest switching rates, respectively.

B. Impact of the simulation parameters

1) Effect of T_{max} : In the proposed protocol, T_{max} is the upper limit of the prime-session's length. When a sender which triggers a prime session has sufficient packets to send, the prime session allots the time duration of T_{max} to transmit packets. As T_{max} becomes longer, the achievable throughput will increase because a longer prime session allows more inter-sessions and also reduces the backoff and negotiation overhead by performing bulk transmissions.

However, in multi-hop networks a long transmission session may incur a long delay; as a result, it may work poorly with transmission control protocol (TCP) flows because the increased round-trip-time (RTT) will reduce the available bandwidth due to its congestion prevention mechanism [20]. Moreover, in sparse networks where the node density is very low, a long prime session does not provide much benefit because there will be few or no inter-sessions. Therefore, it is important to select an appropriate value for T_{max} that reflects the actual network conditions. Currently, dynamically adjusting T_{max} is left as our future work, which can contribute to increasing throughput and decreasing delay.

Fig. 9 illustrates the aggregate throughput of the DiSC, compared to MMAC, AMCP, and IEEE 802.11b with respect to the length of T_{max} and the number of data channels. The simulation was performed on a single-cell network with four sender-receiver pairs. Since an additional channel for exchanging control messages is required for both DiSC and AMCP, the actual number of orthogonal channels is larger than the number of data channels by 1. Here, the x-axis is the length of T_{max} in a T_{tr} unit, where T_{tr} is equal to the duration of *completing* one UDP packet

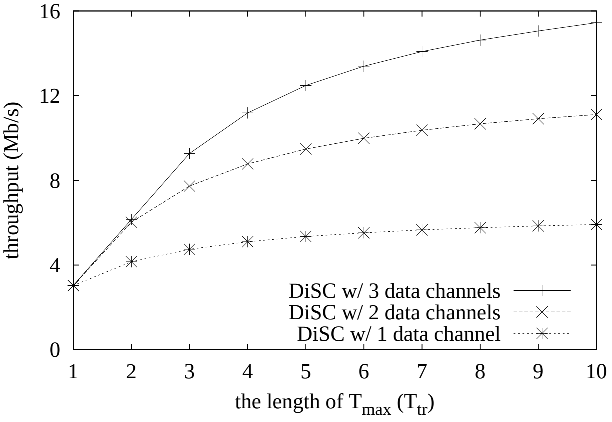


Fig. 9. The aggregate throughput with respect to both the length of T_{max} and the number of available data channels on a single-cell network with 4 sender-receiver pairs.

transmission (i.e., $\sigma_{DATA} + \sigma_{SIFS} + \sigma_{ACK} + \sigma_{SIFS}$, where DATA and ACK are transferred at data and basic rates, respectively).

When there is only one data channel, only a prime session can be established; yet the proposed algorithm still achieves throughput enhancement. This throughput gain comes from the reduced backoff and negotiation overhead provided by performing bulk transmissions. With two data channels, DiSC also shows considerable throughput enhancement; as T_{max} becomes larger, it achieves almost double the throughput compared to that of the single data channel case. When $T_{max} = T_{tr}$, the throughput is same as in the one data channel network, because the prime session (S_{pr}) is too short to permit any inter-session (S_{in}). However, when $T_{max} = 2 \cdot T_{tr}$, S_{pr} can allow one S_{in}, which is sufficient time to complete one UDP packet transmission. Here, the ratio of the transferred bytes for S_{in} to those for S_{pr} is $\frac{1}{2}$. Assuming that all senders are backlogged and that any node can trigger either S_{pr} or S_{in}, S_{in} can hold $(n-1)$ packet transmissions when S_{pr} holds n packets. Therefore, as T_{max} becomes larger, the ratio ($\frac{n-1}{n}$) approaches 1; i.e., with two data channels and a sufficiently long T_{max}, DiSC can achieve almost two times the throughput compared to a single data channel network.

Similarly, for three data channels, the number of packets transferred over the prime-session S_{pr}, the first inter-session $S_{in}^{(1)}$, and the second inter-session $S_{in}^{(2)}$ are $n, (n-1)$, and $(n-1-\alpha)$,

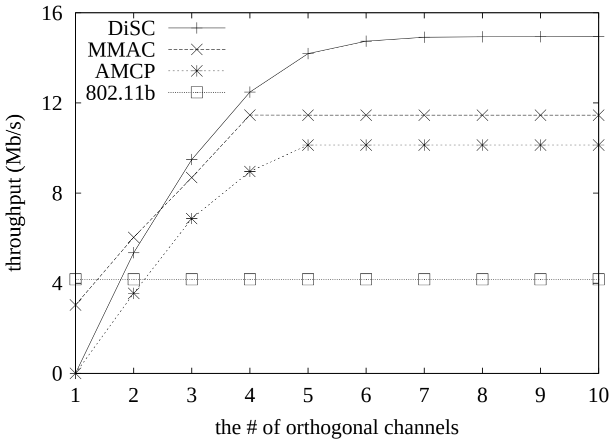


Fig. 10. The aggregate throughput for a single-cell network with respect to the number of data channels with $T_{max} = 5 \cdot T_{tr}$ and 4 sender-receiver pairs.

respectively.[3] If T_{max} is sufficiently large, we can expect three times the throughput compared to the single data channel case. Since this network is configured to hold three concurrent transmissions, the throughput increase is significant up to $T_{max} = 3 \cdot T_{tr}$. Also, it should be noted that with subsequent increases in the number of data channels, DiSC still achieves throughput enhancements; though the amount of increment gradually decreases. Again, this improvement originates from the ability to perform bulk transmissions.

2) Effect of number of orthogonal channels (M): The number of orthogonal channels (M) may not be an adjustable parameter in practice. For example, IEEE 802.11b supports three orthogonal channels, IEEE 802.11a supports four times as many channels. Therefore, we can have a higher degree of freedom in utilizing multiple orthogonal channels with 802.11a than with 802.11b, when necessary.

Fig. 10 presents the aggregate throughput of the four protocols with respect to the number of orthogonal channels. A single-cell network is constructed with four sender-receiver pairs. The figure shows that in the case of IEEE 802.11b, which does not utilize multiple orthogonal channels, M does not affect the throughput performance. On the other hand, MMAC shows a linear increase of the throughput up to $M = 4$. When $M = 1$, four sender-receiver pairs of

[3] A non-negative integer α is determined according to the specific network specifications; e.g., transmission rate, packet size, and so on.

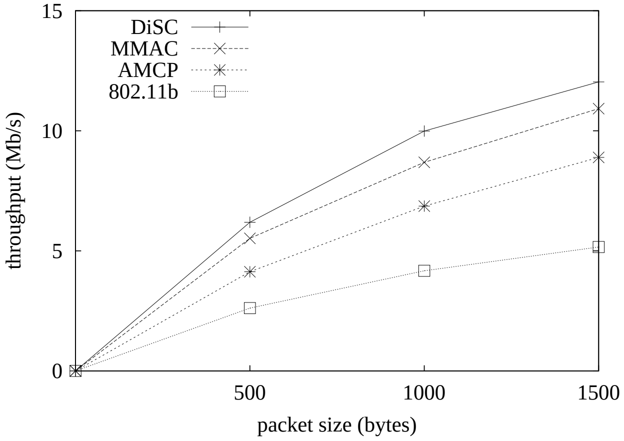


Fig. 11. Aggregate throughput with respect to different packet sizes on a single-cell network with 4 sender-receiver pairs.

MMAC operate on the same channel. However, when M is increased to 4, each pair then uses a different channel, and MMAC achieves four times the aggregate throughput compared to that when $M = 1$. However, the figure also shows that the subsequent addition of channels fails to further improve the throughput, and thus are deemed as being unnecessary. In the case of AMCP and DiSC, the throughput increases up to $M = 5$, and DiSC outperforms AMCP in all cases. The reason why the throughput of both protocols becomes saturated in the five orthogonal channel network is that the control channel is not used for the data transmissions.

3) Effect of packet size: In real networks, packets are typically of different sizes [21] according applications, encryption schemes, and so on. As such, it is important for a network to be able to handle packets of different sizes well. In this regard, to investigate the effect of packet size on the aggregate throughput, we consider three cases: packet sizes of 500 bytes, 1000 bytes, and 1500 bytes.

Fig. 11 presents the throughput results of the four protocols with respect to the packet size on the single-cell network. Clearly, DiSC achieves the highest throughput regardless of the packet size, and that the throughput increases as the packet size gets larger.

VII. Conclusion and Future Work

In this paper, we devised a distributed and semi-synchronous multi-channel coordination protocol (DiSC) for distributed CSMA/CA networks that can effectively manage multi-channel operations. In short, DiSC not only achieves performance enhancement compared to existing multi-channel protocols, but also mitigates multi-channel coordination problems such as multi-channel hidden terminals, missing receivers, and broadcast deafness. These advantages are derived from its semi-synchronous operation, performed with no additional synchronization process. In this protocol, neighboring nodes that have independently switched to different channels can rendezvous on the control channel when their reserved time expires; the nodes thus rendezvous at almost the same time. As such, implementation and operation overhead due to synchronization are eliminated. In addition, DiSC requires neither a central coordination device/scheme nor any additional transceiver, so the hardware implementation complexity remains the same as other off-the-shelf 802.11-compatible devices.

Note that there are a number of issues that remain as topics of future research. Notably, when arranging a set of candidate channels on the negotiation stage, the proposed protocol uses a random mechanism, i.e., a node randomly selects channels among available ones. However, if there is certain information that could improve the possibility of choosing a better channel, a larger degree of multi-channel diversity could be exploited; it is thought that accumulating channel usage information or measuring the channel status in advance could assist this procedure. In addition, we are planning to perform an empirical verification of the proposed protocol. Since DiSC does not need additional transceivers nor synchronization, setting up and configuring a test-bed for DiSC should not increase either the complexity of the hardware or software implementation.

REFERENCES

[1] J. So and N. Vaidya, "Multi-channel mac for ad hoc networks: handling multi-channel hidden terminals using a single transceiver," in *Proceedings of ACM MobiHoc*, 2004, pp. 222–233.

[2] J. Shi, T. Salonidis, and E. Knightly, "Starvation mitigation through multi-channel coordination in CSMA multi-hop wireless networks," in *Proceedings of ACM MobiHoc*, 2006, pp. 214–225.

[3] P. Bahl, R. Chandra, and J. Dunagan, "SSCH: Slotted seeded channel hopping for capacity improvement in IEEE 802.11 ad-hoc wireless networks," in *Proceedings of ACM MobiCom*, 2004, pp. 216–230.

[4] J. Chen, S. Sheu, and C. Yang, "A new multichannel access protocol for IEEE 802.11 ad hoc wireless LANs," *Proceedings of IEEE PIMRC*, vol. 3, 2003.

[5] H.-H. C. Cheng Li, Pu Wang and M. Guizani, "A Cluster Based On-demand Multi-Channel MAC Protocol for Wireless Multimedia Sensor Networks," in *Proceedings of IEEE ICC*, 2008, pp. 2371–2376.

[6] T. Y. T. H. J. A. S. Gang Zhou, Chengdu Huang and T. F. Abdelzaher, "MMSN: Multi-Frequency Media Access Control for Wireless Sensor Networks," in *Proceeedings of IEEE INFOCOM*, 2006.

[7] J. Patel, H. Luo, and I. Gupta, "A cross-layer architecture to exploit multi-channel diversity with a single transceiver," *Proceedings of IEEE INFOCOM Minisymp.*, 2007.

[8] H. So, W. Walrand, and J. Mo, "McMAC: A Parallel Rendezvous Multi-Channel MAC Protocol," in *Proceedings of IEEE WCNC 2007*, 2007, pp. 334–339.

[9] A. Tzamaloukas and J. J. G. luna aceves, "Channel-hopping multiple access with packet trains for ad hoc networks," in *Proceedings of IEEE MoMuC*, 2000.

[10] T. H. J. L. Yafeng Wu, John A. Stankovic and S. Lin, "Realistic and Efficient Multi-Channel Communications in Wireless Sensor Networks," in *Proceedings of IEEE INFOCOM*, 2008.

[11] W. Hung, K. Law, and A. Leon-Garcia, "A dynamic multi-channel MAC for ad hoc LAN," in *Proceedings of 21st Biennial Symposium on Communications*, 2002, pp. 31–35.

[12] Y. Tanigawa, H. Tode, and K. Murakami, "Multi-channel MAC protocols with two transceivers pursuing effective use of vacant resources," in *Proc. of IEEE CCNC*, 2010.

[13] T. Luo, M. Motani, and V. Srinivasan, "CAM-MAC: A cooperative asynchronous multi-channel MAC protocol for ad hoc networks," in *Proceedings of BROADNETS*, 2006, pp. 1–10.

[14] S. Wu, C. Lin, Y. Tseng, and J. Sheu, "A new multi-channel MAC protocol with on-demand channel assignmentfor multi-hop mobile ad hoc networks," in *Proceedings of I-SPAN*, 2000, pp. 232–237.

[15] M. Cao, V. Raghunathan, and P. Kumar, "Cross-layer exploitation of MAC layer diversity in wireless networks," in *Proceedings of IEEE ICNP*, 2006, pp. 332–341.

[16] D. Nguyen, J. Garcia-Luna-Aceves, and K. Obraczka, "Collision-free asynchronous multi-channel access in ad hoc networks," in *Proceedings of the 28th IEEE conference on Global telecommunications*. IEEE Press, 2009, pp. 5453–5458.

[17] "Medium access control (MAC) and physical layer (PHY) specifications."

[18] "The network simulator ns-2," http://www.isi.edu/nsnam/ns/.

[19] J. Li, C. Blake, D. De Couto, H. Lee, and R. Morris, "Capacity of ad hoc wireless networks," in *Proceedings of ACM MobiCom*, 2001, pp. 61–69.

[20] M. Mathis, J. Semke, J. Mahdavi, and T. Ott, "The macroscopic behavior of the TCP congestion avoidance algorithm," *ACM SIGCOMM Computer Communication Review*, vol. 27, no. 3, pp. 67–82, 1997.

[21] R. Sinha, C. Papadopoulos, and J. Heidemann, "Internet packet size distributions: Some observations," *Online, Oct*, 2005.