Dynamic Decentralized Multi-Channel MAC Protocols

Hulya Seferoglu
University of California Irvine
CA, USA
hseferog@uci.edu

Ashvin Lakshmikantha
UI Urbana-Champaign
IL, USA
lkshmknt@uiuc.edu

Ayalvadi Ganesh
University of Bristol
Bristol, UK
a.ganesh@bristol.ac.uk

Peter Key
Microsoft Research
Cambridge, UK
peterkey@microsoft.com

Abstract— **In this paper, we propose new dynamic decentralized multi-channel multiple access (MAC) protocols and study their performance. Our protocols build on slotted Aloha, but extend it in several ways to improve flow completion time and throughput, as follows: (i) channels are assigned to flows rather than packets to eliminate per packet collisions, thus the total number of collisions is reduced, and (ii) each flow owns or drops channels dynamically considering successful transmissions, thus the number of owned channels adapts to varying traffic. We present an analysis of the stability region and of flow completion times, for our algorithms, and show that one of them can achieve close to 100% throughput if flow sizes are large. We demonstrate by extensive simulations that, compared to current multi-channel MAC protocols, our algorithms improve flow completion time and throughput in wireless local area and mesh networks.**

I. Introduction

In traditional wireless local area networks (WLANs) and wireless mesh networks (WMNs), wireless nodes, access points, and mesh routers share a single channel. Due to growing channel demand, single channel wireless networks suffer from serious capacity degradation, [1]. Current wireless standards allow simultaneous operation over multiple non-overlapping channels; 802.11b permits three orthogonal channels in 2.4 GHz band, and 802.11a permits up-to twelve non-overlapping channels in the 5.0 GHz band. Simultaneous communication over multiple channels is achieved by equipping nodes with multiple interfaces operating on non-overlapping channels. However, how to assign channels to nodes is a problem not addressed by current MAC protocols, which are tailored for a single channel. In this paper, we propose dynamic decentralized multi-channel medium access (MAC) protocols as a solution to the channel assignment problem in WLANs and WMNs and compare their performance to some recently proposed multi-channel MAC protocols.

We consider the channel assignment problem for both WLANs and WMNs. In WLANs, an access point (AP) forwards incoming flows to their final destinations. In WMNs, a node communicates to others directly or over multiple hops. Although the communication mechanism is quite different in WLANs as compared to WMNs, the channel assignment problem is similar; each node should access a number of available channels to transmit its flow regardless of whether it is an AP, mesh router, or end point[1]. In view of this, we design multi-channel MAC algorithms independent of the underlying network and topology, which can be applied to both WLANs and WMNs.

We present two algorithms to improve flow completion times of short term flows and throughput. The first algorithm works as follows. When a flow is generated, its node attempts transmission over a randomly chosen channel. If the transmission is successful, the node owns the channel, and makes all other transmissions over this channel. In the second algorithm, a node can transmit over multiple channels simultaneously, and continuously attempts to acquire additional channels during its flow transmission. When a collision occurs, both algorithms drop the channel with some probability. The main points behind our algorithms are to (i) assign channels to flows rather than packets to eliminate per packet collisions, and (ii) acquire and drop channels dynamically to adapt to varying channel traffic.

In contrast to some other proposals, our algorithms don't use carrier sensing (CSMA). One motivation for this is that CSMA doesn't work well with hidden terminals, which are especially a problem in mesh networks, and we wanted a single solution that works in both the access point and mesh scenarios. Secondly, we wanted to determine the fundamental limitations of MAC protocols in a setting where carrier sensing isn't effective. There is nothing about our algorithms that precludes carrier sensing, incorporating which would only improve their performance.

Specifically, the contributions of this paper are as follows:

- Decentralized multi-channel MAC algorithms that improve flow completion time, throughput, and channel utilization.
- Theoretical analysis of the stability region and flow completion times of the proposed algorithms.
- A comprehensive performance study based on extensive simulations for different networks, number of available channels, and algorithm parameters.

[1]Channel assignment and routing are a joint problem in wireless mesh networks. However, we assume that these problems are independent by considering pre-defined routes. The effect of channel assignment on routing is out of the scope of this paper.

This paper is organized as follows. Section II discusses related work. Section III gives an overview of the system model. Section IV presents our multi-channel MAC algorithms and their theoretical analysis. Section V presents simulation results and demonstrates the benefits of the proposed algorithms over baseline schemes, in terms of flow completion time, throughput, and channel utilization. Section VI concludes the paper.

II. Related Work

There has been a significant amount of work on multi-channel MAC protocols in recent years [3]–[12]. Typically, these works assume that multiple orthogonal channels are provided by frequency division and develop a suite of protocols to exploit this channel diversity to achieve higher throughput. In [11], the authors make effective use of multiple channels by having a separate channel for reservation. This extends the idea of using short packets/slots in contention mode to reserve longer non-contending slots for the data, which goes back to early work on multiple access systems, summarized in [2]. As the channels are obtained by frequency division, their scheme requires two receivers. In [9], the authors describe CHMA, whereby all the channels are used efficiently using only a single transceiver. Extensions of these works can be found in [7], [10]. A Markov chain analysis of several existing multi-channel MAC protocols can be found in [13]. Channel diversity and receiver diversity are exploited in [14] to improve performance.

There has been a parallel body of work on developing protocols to allow users to transmit data simultaneously using code division [15]–[18]. These works consider adhoc networks and develop distributed algorithms to assign unique PN codes to all users. Note that this problem is quite different, akin to graph coloring. These papers focus on different technical challenges such as receiver design complexity, multiple access interference (MAI), transmitter design complexity, etc.

In most of the above work, the system is studied under the saturated assumption, namely, that the number of flows in the system is a constant and all the flows have infinite data to send. However, in reality, flows arrive and depart, making the network conditions more random.

Our work differs from that mentioned above in two respects. First, we do not use explicit reservations though our algorithms can be thought of as implementing implicit reservations. Consequently, we do not need to assume that all contending nodes can hear the reservation requests or replies. Second, we take a flow level view of the MAC protocol. Without requiring any specific higher layer information, a flow-aware MAC layer can perform significantly better in terms of flow level performance, as we demonstrate. The work that comes closest to ours in terms of approach is [6], which makes use of history to selectively transmit on specific channels on which the number of successful transmissions have been higher. The authors study a static network with a fixed number of long term flows via simulations. Thus, their work does not provide any theoretical guarantees on the performance of their algorithm. In our work, we present theoretical study of the performance of our algorithms which also takes into account flow arrivals and departures and evaluate effectiveness of out algorithms via extended simulations.

III. System Overview

We consider the multiple access problem when nodes are capable of transmitting over multiple channels simultaneously. Nodes may be access points or clients in a wireless LAN or nodes in wireless mesh networks.

We assume that there are N orthogonal channels available for simultaneous data transmission. Time is slotted and the length of each slot is sufficient for a node to transmit a single packet to another and receive an ACK from it. We consider a collision based model, i.e, if two or more packets are transmitted on the same channel in the same slot, then all these packets are lost. The nodes involved learn of this collision through not receiving an ACK.

We assume that nodes are capable of receiving and decoding data on all or a subset of the N channels simultaneously. While this assumption is made for analytical convenience, the cost of network interface cards (NICs) renders this impractical. We relax this assumption in our simulations and study the performance of our MAC protocols with a limited number of NICs.

We consider a dynamic connection-level model wherein each flow (or user) entering the system has a fixed amount of data that needs to be transmitted. For our purposes, this can be thought of as a single file. After successfully transmitting this file, the flow leaves the system. We assume that flows enter the system according to a Poisson process of rate $N\lambda$, i.e., the number of flows arriving per time slot is Poisson($N\lambda$). Recall that N is the number of channels, so λ can be thought of as the per-channel arrival rate. For purposes of mathematical analysis, we assume that the file sizes, measured in packets per file, are independent and identically distributed (iid) geometric random variables with mean $1/\mu$. We denote by $\rho = \lambda/\mu$ the normalized offered load; $\rho = 1$ is the maximum load that can be served with perfect scheduling.

IV. Multi-Channel MAC Protocols

In this section, we propose two multi-channel MAC protocols, Algorithm A and Algorithm B, and provide an analysis of their stability region and of flow completion times. We also present initial performance results based on Matlab simulations of an idealized model. The description and theoretical performance analysis of Algorithms A and B are in sections IV-A and IV-B respectively, while the initial performance evaluations are presented in section IV-C.

Since our protocols build on slotted Aloha, we briefly recall its natural extension to the multi-channel setting. In each time slot, each user picks a channel at random and attempts to transmit a packet on it with some fixed probability α, independent of the past. When all packets have been successfully transmitted, the user departs.

A. Algorithm A and Performance Analysis

When a new flow is generated, its node picks a channel at random and attempts to transmit over it with probability α, repeating the process in each time slot until a transmission is successful. The flow (equivalently, its node or user) is termed unsatisfied until it succeeds on some channel. At that point, it turns into a satisfied user and becomes the *owner* of that channel. A satisfied user continues to transmit over its owned channel and does not access other channels. If it suffers a collision in some time slot (which can only be caused by an unsatisfied user), it drops the channel with fixed probability p and becomes unsatisfied; with probability $1-p$, it retains the channel. If it doesn't suffer a collision, it completes the file transfer with probability μ in each slot (since file sizes are geometric with mean $1/\mu$) and leaves.

We now present a performance analysis when the channel drop probability p equals zero. Note that $p = 1$ corresponds to slotted Aloha. In practice, we would take the channel drop probability p to be small but non-zero in order to avoid deadlock due to channel errors, which can cause two users to think that they own the same channel.

Let $S(t)$ denote the number of satisfied users and $U(t)$ the number of unsatisfied users at the beginning of time slot t. We can easily compute the expected change in these quantities over time:

$$E[S(t+1)|S(t),U(t)] = S(t) - \mu\Big(1-\tfrac{\alpha}{N}\Big)^{U(t)}S(t) + \alpha(1-\mu)\Big(1-\tfrac{S(t)}{N}\Big)\Big(1-\tfrac{\alpha}{N}\Big)^{U(t)-1}U(t), \quad (1)$$

where the second term on the right refers to the number of satisfied users who leave the system as their file transmission is complete, and the third term to the number of unsatisfied users who successfully transmit, thereby changing status to satisfied. The pre-factor $1-\mu$ in the last term accounts for the fact that a fraction μ of the unsatisfied customers leave immediately upon transmitting a packet successfully. Likewise,

$$E[U(t+1)|S(t),U(t)] = U(t) + N\lambda - \alpha\Big(1-\tfrac{S(t)}{N}\Big)\Big(1-\tfrac{\alpha}{N}\Big)^{U(t)-1}U(t), \quad (2)$$

where the second term on the right denotes the number of new arrivals into the system, and the third term represents the decrease of the number of unsatisfied users who either become satisfied users or leave the system upon completing their file transmission.

Define $s(t) = S(t)/N$ and $u(t) = U(t)/N$. If N is large, the above equations heuristically lead us to consider the following difference equation model:

$$\begin{aligned}\Delta s(t) &= -\mu e^{-\alpha u(t)}s(t) \\ &\quad + (1-\mu)\alpha u(t)e^{-\alpha u(t)}(1-s(t)), \quad (3)\\ \Delta u(t) &= \lambda - \alpha u(t)e^{-\alpha u(t)}(1-s(t)). \quad (4)\end{aligned}$$

It can be shown using standard techniques that this model arises as the fluid limit when N tends to infinity, but we omit the details.

We now turn to identifying the equilibria of these dynamics. Setting $\Delta s(t)$ and $\Delta u(t)$ to zero, we get

$$s = \frac{(1-\mu)\alpha u}{\mu + (1-\mu)\alpha u}, \quad \alpha u e^{-\alpha u} = \frac{\lambda}{1-s}. \quad (5)$$

If these equations have a solution for given values of λ and μ, then the system can carry the corresponding offered traffic. If the system cannot carry the offered traffic, then the number of unsatisfied users will tend to infinity, reflected by equation (5) having no solution. We can thus identify the capacity region as a function of μ. We have:

Theorem 1: Suppose $p = 0$, i.e., users that have successfully acquired a channel never give it up until completing their file transmission. Then, for given μ, the largest λ for which equation (5) has a solution is given by $\lambda = z^2e^{-z}$, where z is the unique positive solution of the quadratic

$$(1-\mu)z^2 + \mu z - \mu = 0. \quad (6)$$

The corresponding value of the maximum achievable throughput, $\rho = \lambda/\mu$, is equal to $1/e$ at $\mu = 1$ and tends to 1 as the mean file size $1/\mu$ tends to infinity.

Remark. If $\mu = 1$, i.e., each file consists of a single packet, then Algorithm A reduces to slotted Aloha, so the maximum throughput of $1/e$ is as expected. The theorem says that Algorithm A achieves 100% throughput in the limit of infinitely large file sizes.

Proof. Letting $z = \alpha u$, we have from (5) that

$$\frac{1}{s} - 1 = \frac{\mu}{(1-\mu)z} \text{ and } 1-s = \lambda z^{-1}e^{z},$$

from which it follows that

$$s = \frac{\lambda(1-\mu)}{\mu}e^{z}, \quad ze^{-z} = \lambda + \frac{\lambda(1-\mu)}{\mu}z. \quad (7)$$

The latter equation may have no, one or two solutions for z depending on the values of λ and μ. For given $\mu \in (0,1]$, we want to find the largest λ for which it has a solution. For this λ, the line $x \mapsto \lambda + \frac{\lambda(1-\mu)}{\mu}x$ has to be tangent to the curve $x \mapsto xe^{-x}$ at some $x = z$. Therefore $(1-z)e^{-z} = \lambda + \frac{\lambda(1-\mu)}{\mu}$. Substituting this in (7), we get

$$ze^{-z} = \lambda + (1-z)ue^{-z}, \quad \text{i.e.,} \quad z^2e^{-z} = \lambda.$$

Substituting this in (7) again, $\lambda = \lambda z + \frac{\lambda(1-\mu)}{\mu}z^2$, which yields the quadratic

$$(1-\mu)z^2 + \mu z - \mu = 0. \quad (8)$$

This establishes the first claim of the theorem.

Substituting $\mu = 1$ in (8), we get $z = 1$ and hence $\lambda = z^2e^{-z} = 1/e$. At the other extreme, as μ tends to zero, the positive root of (8) satisfies $z = \sqrt{\mu} + O(\mu)$, so that $\lambda = z^2e^{-z} = \mu + O(\mu^{3/2})$. Hence, it follows that $\rho = \lambda/\mu$ tends to 1. □

The capacity region is a network-centric performance measure that describes the maximum flow that can be carried. We are also interested in the user-centric measure of how long it takes a single flow, with a specified file size f, to complete transmission. With perfect scheduling, it would require f time slots if a node can only transmit on one channel at a time.

For a new flow, the probability of acquiring a channel is $\alpha e^{-\alpha u}(1-s)$ in each time slot, independent of the past. Hence, the time to acquire a channel is geometric with mean $1/(\alpha e^{-\alpha u}(1-s))$. After getting a channel, the flow needs to transmit another $f-1$ packets. The time to successfully transmit each packet is also geometric, but with mean $e^{\alpha u}$. Hence, the mean flow completion time conditional on file size is given by

$$E[T|f] = \frac{e^{\alpha u}}{\alpha(1-s)} + (f-1)e^{\alpha u}, \tag{9}$$

which decomposes into a constant plus a multiple $e^{\alpha u}$ of the file size f. Here, (s, u) can be obtained by numerically solving (5).

As mentioned earlier, equation (5) has two solutions inside the capacity region, i.e., Algorithm A exhibits bistability, just like slotted Aloha. The smaller solution corresponds to the desirable equilibrium at which we want the system to operate. A large deviation can carry the system to the undesirable equilibrium, but this is rare. As with Aloha, the system may need to be flushed when this happens. There is good agreement between numerical results obtained using the smaller solution for (s, u) and simulations, suggesting that the system typically does stabilize in the neighborhood of the good equilibrium. See Section IV-C for details. We also find that, under light to moderate load, αu is small, $e^{\alpha u}$ is close to 1, and so the mean file transfer time is only a small multiple larger than the file size, which is the best achievable.

We can compute the unconditional mean flow completion time from (9). Since the mean file size is $1/\mu$, we have

$$E[T] = \frac{e^{\alpha u}}{\alpha(1-s)} + \frac{(1-\mu)e^{\alpha u}}{\mu}. \tag{10}$$

It can be readily verified using (5) that $\lambda E[T] = s + u$, as required by Little's law.

B. Algorithm B and Flow Completion Time Analysis

As in Algorithm A, each flow entering the system picks a channel randomly and attempts to transmit over it with probability α in each time slot. It gains ownership of a channel if it transmits successfully over it. However, unlike Algorithm A, users continue to behave as unsatisfied even if they own one or more channels. The algorithm works as follows. At the beginning of a slot, each user transmits over its owned channels with probability 1 and attempts one other randomly chosen channel with probability α, acquiring it if the attempt is successful. The user drops each of its own channels on which it suffered a collision with probability p, independent of its other decisions.

We now present a heuristic analysis of the flow completion time of Algorithm B when the channel drop probability p is zero. Let $A(t)$ denote the number of users in the system and $N(t)$ the number of owned channels at the beginning of time slot t. The expected change in $A(t)$ in one time slot is given by

$$\begin{aligned} &E[A(t+1)|A(t), N(t)] - A(t) \\ &\approx N\lambda - \mu\frac{N(t)}{A(t)}(1-\frac{\alpha}{N})^{A(t)-1}A(t), \end{aligned} \tag{11}$$

where the first term on the right is the number of newcomers to the system and the second term approximates (using the union bound) the number of users leaving the system after completing their file transmission. Similarly, the expected change of $N(t)$ over time is given by

$$\begin{aligned} &E[N(t+1)|A(t), N(t)] - N(t) \\ &\approx \alpha A(t)(1-\frac{\alpha}{N})^{A(t)-1}(\frac{N-N(t)}{N}) \\ &-\mu\frac{N(t)}{A(t)}(1-\frac{\alpha}{N})^{A(t)-1}\frac{N(t)}{A(t)}A(t), \end{aligned} \tag{12}$$

where the first term on the right represents new channels acquired by the $A(t)$ existing users and the second term represents the number of channels owned by users who departed during that slot.

Define $a(t) = A(t)/N$ and $n(t) = N(t)/N$. If N is large, then equations (11) and (12) suggest the following difference equation model:

$$\Delta a(t) = \lambda - \mu n(t)e^{-\alpha a(t)}, \tag{13}$$

$$\begin{aligned} \Delta n(t) &= \alpha a(t)e^{-\alpha a(t)}(1-n(t)) \\ &\quad -\mu\frac{n(t)}{a(t)}e^{-\alpha a(t)}n(t). \end{aligned} \tag{14}$$

We solve for the equilibrium values of a and n by setting $\Delta a(t)$ and $\Delta n(t)$ to zero. Thus, we get;

$$\rho = ne^{-\alpha a}, \quad a = \frac{\mu}{\alpha}\frac{n^2}{1-n}, \tag{15}$$

where $\rho = \lambda/\mu$. We can solve this numerically to find the channel occupancy and scaled number of users in the system in equilibrium.

In order to calculate flow completion time, we first calculate the expected time d_c to acquire a new channel and the expected time d_s until successful transmission on an owned channel. These times are geometric and their means are given by the reciprocal of the corresponding success probabilities:

$$d_c = \frac{1}{\alpha e^{-\alpha a}(1-n)}, \quad d_s = \frac{1}{e^{-\alpha a}}. \tag{16}$$

Suppose that a user owned K channels just prior to departure. The expected time to acquire them is Kd_c in expectation. In each time slot during which the user owned k channels, the expected number of successful transmissions is k/d_s. Thus, the number of successful transmissions prior to departure (which has to be equal to the file size) is no more than $K(K+1)/2$ times d_c/d_s in expectation. This argument suggests the approximation

$$\frac{d_c}{d_s}\frac{K(K+1)}{2} \approx \frac{1}{\mu} \tag{17}$$

Solving for K from the above, we estimate the expected flow completion time as $(K+1)d_c$.

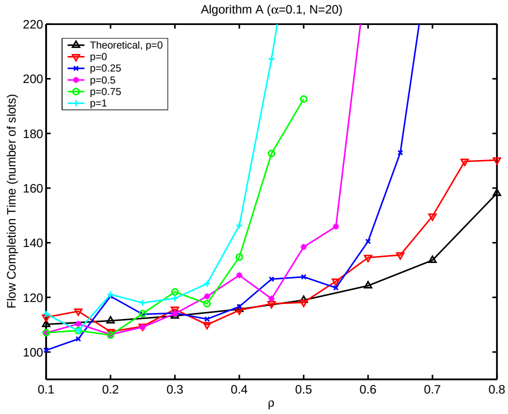


Fig. 1. Flow Completion Time of Algorithm A when $N = 20$, $\alpha = 0.1$

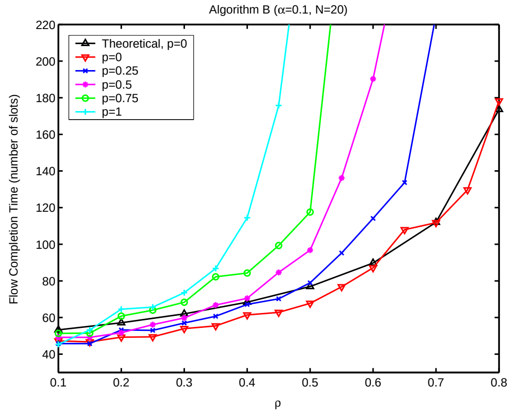


Fig. 3. Flow completion time of Algorithm B when $N = 20$, $\alpha = 0.1$

C. Initial Performance Evaluation

In this section we present initial simulations of Algorithms A and B in an ideal environment to show the consistency of the theory with simulations and to gain insight into their performance, measured by flow completion time.

We assume that there are N channels in the system and that all flows can access all N channels. Users communicate to an access point over uplink channels in a WLAN setup, and depart after completing their file transmissions. We simulated Algorithms A and B in Matlab, without considering the physical channel or any layer interactions. We also neglected any overhead due to packet headers, propagation, and ACKs. In the following, we present the simulations results and compare with the theoretical analysis.

Fig. 1 shows the flow completion time of Algorithm A in number of slots with respect to the load ρ when the number of channels is $N = 20$, the channel attempt probability[2] is $\alpha = 0.1$, and the flow size is geometrically distributed with mean $1/\mu = 100$. From the figure, we can see that the mean flow completion time is around 100 to 120 for low loads. Since Algorithm A owns at most one channel at a time, the flow completion time has to be bigger than the flow size. The figure shows that it is not much bigger; the overheads added by the time to acquire a channel and by collisions are quite small at low loads. The figure also includes the theoretical flow completion time for $p = 0$ calculated in section IV-A. It matches fairly well with the corresponding simulation.

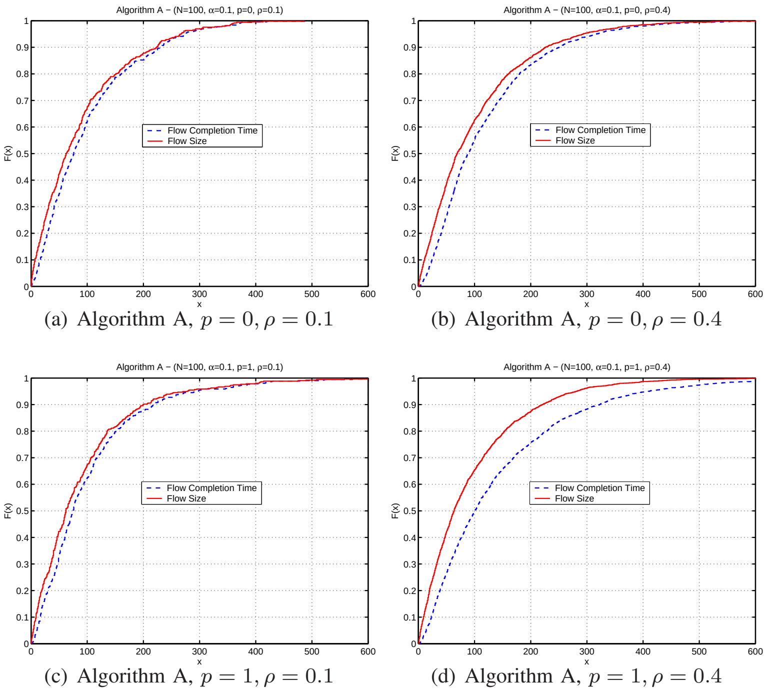
The figure shows that the stability region shrinks as the channel drop probability p increases. This is because increasing p increases the number of unsatisfied users and hence the frequency of collisions. As a result, the probability of owning a channel decreases, the flow completion time increases for a given load and the system becomes unstable at lower loads.

Fig. 2 shows the cumulative distribution functions (CDF) of flow completion times under Algorithm A, for different values of p and ρ. The number of channels is $N = 100$, the channel attempt probability is $\alpha = 0.1$, and the mean flow size is $1/\mu = 100$. The CDF of flow size is also plotted in the same figures for comparison, as it is the flow completion time under perfect scheduling. The gap between the two curves indicates the overhead suffered by Algorithm A. The plots show that this overhead is typically small, but increases with p and ρ, as expected. This is because it takes longer to find a free channel and there are more collisions at higher loads. Increasing p also increases the collision probability. Thus, in Fig. 2-(d) when both ρ and p are high, the flow completion time is around 1.5 times the flow size, whereas the overhead is only about 10% in the other figures.

Fig. 3 shows the flow completion time of Algorithm B with respect to ρ when the number of channels is $N = 20$, the channel attempt probability is $\alpha = 0.1$, and the mean flow size is $1/\mu = 100$. The figure shows that the flow completion time is much smaller in its stable region as compared to the flow completion times of Algorithm A shown in Fig. 1. The reason is that Algorithm B allows users to own and transmit over multiple channels at the same time and complete their flows in time less than its flow size. The penalty is that the flow completion time increases more steeply with load because the algorithm, being more aggressive, causes more collisions. The figure also includes the analytical calculation of flow completion time for $p = 0$ presented in section IV-B. This is seen to match the simulation quite well. The small gap could be due to the simplifying assumptions in the analysis.

Fig. 4 shows the cumulative distribution function (CDF) of flow completion times in Algorithm B when $N = 100$, $\alpha = 0.1$, and $1/\mu = 100$. Fig. 4-(a) shows that flow completion time is smaller than the flow size (except for very small flows, which don't gain enough benefit from acquiring multiple channels to offset the cost incurred in finding them). This shows the effectiveness of Algorithm B over Algorithm A, especially for large flows. In Fig. 4-(b) and Fig. 4-(c), the characteristic of the flow completion time curves does not change much as compared to Fig. 4-(a). However, when both p and ρ increase as in Fig. 4-(d), the flow completion time performance gets worse as compared to the other three figures. However, even in

[2]We have simulated our algorithms for different values of α and observed that the stable region of both Algorithms A and B is larger for small α values, so we set $\alpha = 0.1$ for all the simulations in the remainder of the paper.



(a) Algorithm A, $p = 0, \rho = 0.1$

(b) Algorithm A, $p = 0, \rho = 0.4$

(c) Algorithm A, $p = 1, \rho = 0.1$

(d) Algorithm A, $p = 1, \rho = 0.4$

Fig. 2. CDF of flow completion times of Algorithm A when $N = 100$, $\alpha = 0.1$, and (a) $p = 0, \rho = 0.1$, (b) $p = 0, \rho = 0.4$, (c) $p = 1, \rho = 0.1$, and (d) $p = 1, \rho = 0.4$

Fig. 4-(d) the flow completion time is comparable with the flow size, which is significantly better than that achieved by Algorithm A. This result supports the effectiveness of Algorithm B even at higher loads and higher channel drop probabilities.

V. Performance Evaluation

In this section, we evaluate flow completion time performance of aforementioned multi-channel MAC algorithms; Algorithm A and Algorithm B, via extensive simulations. We compare them with a baseline algorithm; *Dedicated Control Channel* (DCC).

A. Simulation Setup

The simulations were carried out in Glomosim simulation environment by modeling its MAC and physical layers for WLAN and WMN simulations.

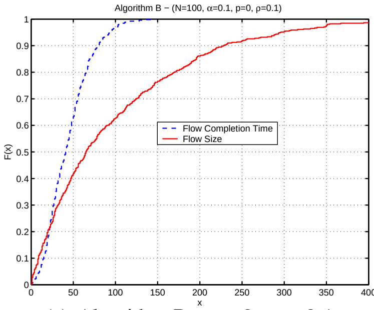
The MAC layer is the extended version of slotted Aloha according to Algorithm A and Algorithm B as explained in IV. Basically, when a new flow is generated, one of the nodes becomes active and generates a flow. Each active user sends a packet in a fixed time slot over its channel(s), or attempts to get a channel with probability $\alpha = 0.1$. Slots are set large enough to transmit a packet and receive its acknowledgement (ACK). If an ACK is received over an attempted channel, the channel is owned. On the other hand, if an ACK is not received for an owned channel, the channel is dropped with probability p.

In wireless channel, two-ray path loss model is used as a propagation path loss model. The model uses free space path loss for near sight and plane earth path loss for far sight. Free space path loss is proportional to the distance between transmitters and receivers, which depends on simulated topology, and inversely proportional to wavelength which is determined by operating frequency of selected channel. Earth path loss is proportional to square of distance between transmitters and receivers, and inversely proportional to receiver and transmitter antenna heights which are set to $1.5m$ for both transmitters and receivers. For the fading model, Rayleigh fading channel with Rayleigh variant 0.63 is employed. Multiple orthogonal channels are generated around 2.4 GHz band by separating each channel by 5 MHz. The bandwidth of each channel is set to 2 Mbps, and the number of channels is $N = 20$.

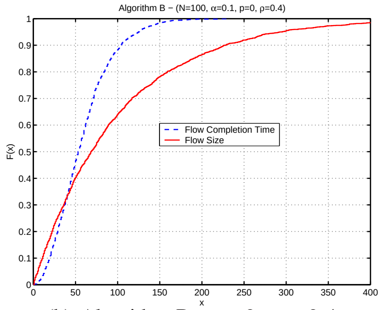
Downlink and uplink traffic is modeled by short term flows with fixed size (1500 bytes) packets. Flow size is geometrically distributed and flows arrive according to a Poisson distribution characterized by parameters; number of channels (N) and load (ρ). The mean flow size is set to $1/\mu = 100$. We assumed similar load levels for all users, and the buffer sizes are assumed to be large enough to avoid overflows.

We consider WLAN and WMN simulations in different topologies shown in Fig. 5 and Fig. 6, respectively. In Fig. 5: (a) every client placed around a wheel sends its flow to an AP in the center, (b) all clients are randomly placed in a terrain and they try to transmit to an AP, (c) all clients placed around a wheel try to communicate to each other via an AP. In Fig. 6: (a) all clients placed randomly in a terrain communicate to each other directly (over one hop), and (b) all clients placed in a grid topology communicate to each other over one or two hops.

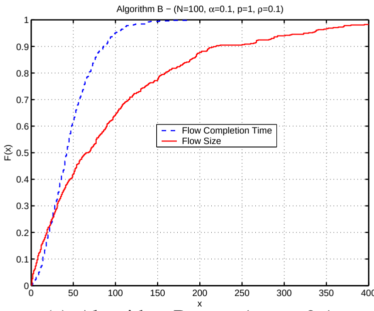
We compare Algorithm A and Algorithm B with a baseline multi-channel MAC protocol; Dedicated Control



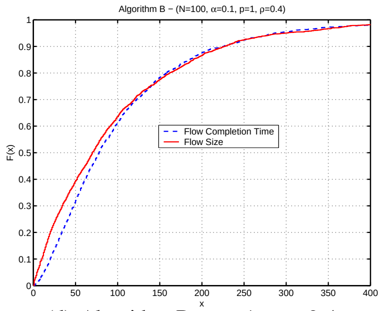
(a) Algorithm B, $p = 0, \rho = 0.1$



(b) Algorithm B, $p = 0, \rho = 0.4$

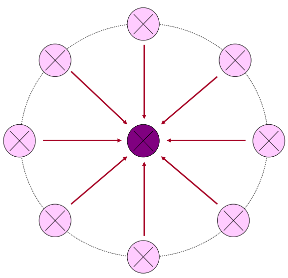


(c) Algorithm B, $p = 1, \rho = 0.1$

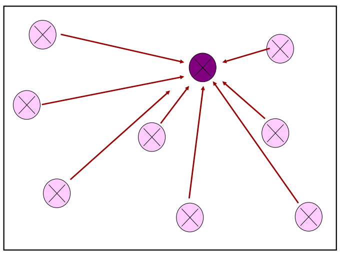


(d) Algorithm B, $p = 1, \rho = 0.4$

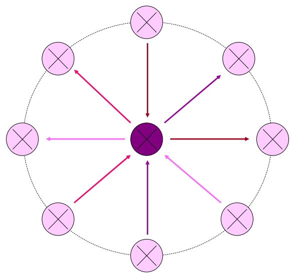
Fig. 4. CDF of flow completion times of Algorithm B when $N = 100$, $\alpha = 0.1$, and (a) $p = 0, \rho = 0.1$, (b) $p = 0, \rho = 0.4$, (c) $p = 1, \rho = 0.1$, and (d) $p = 1, \rho = 0.4$



(a) One Hop Wheel Topology

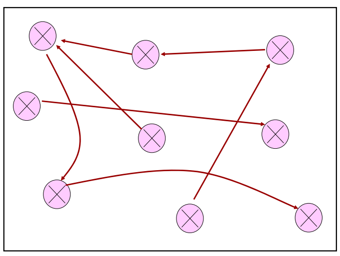


(b) One Hop Random Topology

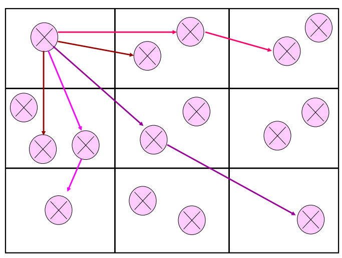


(c) Two Hop Wheel Topology

Fig. 5. WLAN topologies.



(a) One Hop Random Topology



(b) One/Two Hop Grid Topology

Fig. 6. WMN topologies.

Channel (DCC). DCC reserves one channel (called as control channel) for channel reservation messages. When a flow is generated, its user sends an RTS message including requested channel number and network allocation vector (NAV) over the control channel to its destination. All users in the system hear the requested channel number, since they continuously monitor the control channel. When a corresponding CTS message is sent back, the user that has sent the RTS message owns the channel, and all other users confirm that the requested channel is reserved, and they do not attempt to own this channel for NAV duration. DCC exhibits good flow completion time and throughput improvement compared to some other efficient multi-channel MAC protocols, [19].

B. Simulation Results

1) WLAN - One Hop Wheel Topology: In this part, $M = 40$ nodes including AP and clients are placed in a $100m \times 100m$ terrain according to wheel topology where circle radius is $25m$, Fig. 5-(a).

Fig. 7 shows flow completion time performance of Algorithm A and Algorithm B for one-hop transmissions from clients to an AP. Flow completion time results are given for various channel drop probabilities, $p = 0.01$, $p = 0.5$, and $p = 1$. Fig. 7-(a) shows that flow completion time of Algorithm A is $7.5ms$ in the stable region. A slot size is set to $6.5ms$ considering average round trip time (RTT). Thus, we can calculate the flow completion time as 115 slots. It is observed that the flow completion time is comparable with both analysis results and the basic simulation results given in Fig. 1. Fig. 7 (b) shows flow completion time of Algorithm B as $3.5ms$ corresponding to 55 slots. This value is consistent with both analysis results and basic simulation results presented in Fig. 3. Fig. 7 also shows that when the channel drop probability p increases, stable region of the system decreases which was also noted in Fig. 1 and Fig. 3. Thus, we can conclude that lower channel drop probabilities should be selected to enjoy large stable regions, for both Algorithm A and Algorithm B.

Fig. 8 shows flow completion times for Algorithm A, Algorithm B, and DCC when the channel drop probability is set to $p = 0.1$. The figure shows that completion time improvement of Algorithm A and DCC in the stable region is similar, and Algorithm B performs better than both. The flow completion time of Algorithm A and DCC are almost the same, because the goal of both Algorithm A and DCC is to own a channel and transmit over it. DCC achieves this goal by monitoring the control channel and keeping track of the owned channels while Algorithm A achieves this goal in an ad-hoc manner. Although Algorithm A and DCC exhibits similar flow completion time characteristics, Algorithm A is more practical than DCC, because DCC has some problems in practice such as hidden terminal problem and its need to monitor the control channel continuously while Algorithm A is totally ad-hoc. On the other hand, flow completion time for Algorithm B is lower than both Algorithm A and DCC

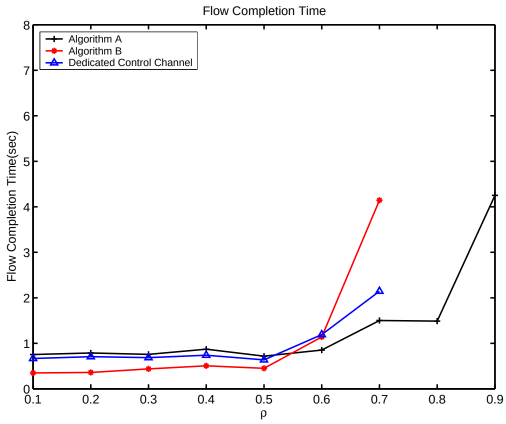
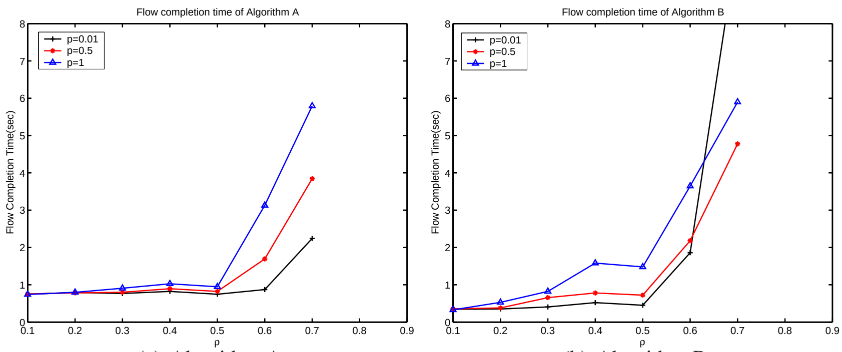


Fig. 8. WLAN - One Hop Wheel Topology. Flow completion time of Algorithm A, Algorithm B, and DCC when $N = 20$, $\alpha = 0.1$, $p = 0.1$, $1/\mu = 100$, and $M = 40$.

in the stable region, because Algorithm B can own more than one channel while Algorithm A and DCC can get at most one. Similar to Algorithm A, Algorithm B is also an ad-hoc algorithm and does not have the practical limitations of DCC. We can conclude that Algorithm A and Algorithm B have flow completion times lower than or equal to DCC, and our algorithms are more suitable for practical applications as compared to DCC.

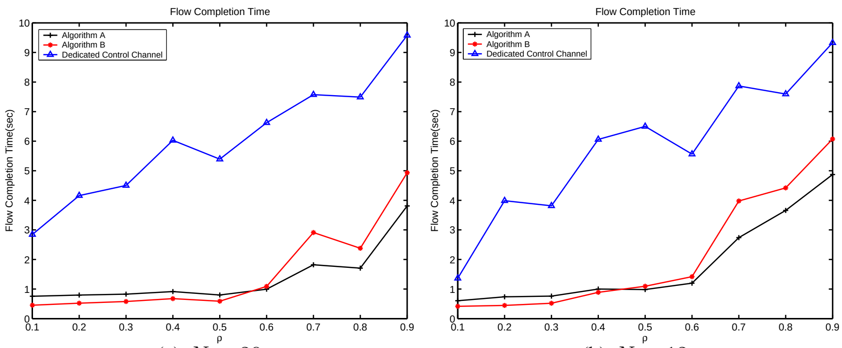
So far we have assumed that the number of interfaces at each node is equal to the number of channels in the system. However, current technology can only allow a limited number of interfaces per node. Fig. 9 shows flow completion time of Algorithm A, Algorithm B, and DCC for (a) $N = 20$ and (b) $N = 12$ when each node is equipped with either two or three interfaces, chosen randomly. Fig. 9-(a) shows that Algorithm A and Algorithm B exhibits similar flow completion time performance for limited number of interfaces as compared to the scenario where the number of interfaces is equal to the number of channels (as shown in Fig. 8). However, DCC exhibits poor flow completion time performance when the number of interfaces is limited. The reason is that DCC uses one channel, i.e. one interface, as a control channel, and uses either one or two channels as data channels. This reduces its chance to find an idle channel in the system. Thus, some users should wait sometime until their channels become idle. This increases the flow completion time of DCC while Algorithm A and Algorithm B still have enough interfaces for transmission. This result proves that DCC exhibits good flow completion time performance only when the number of interfaces is large. We know that 802.11a permits up to $N = 12$ non-overlapping channels with a few interfaces per node. In order to show that our algorithms improve flow completion time even with current technology limitations, we have simulated Algorithm A, Algorithm B, and DCC when the number of channels is $N = 12$ and the number of interfaces are either two or three per node. Fig. 9-(b) shows that flow completion times of both Algorithm A and Algorithm B are better as compared to DCC.

2) WLAN - One Hop Random Topology: In this topology, $M = 40$ nodes are randomly distributed in a



(a) Algorithm A (b) Algorithm B

Fig. 7. WLAN - One Hop Wheel Topology. Flow Completion time of (a) Algorithm A and (b) Algorithm B when $N = 20$, $\alpha = 0.1$, $1/\mu = 100$, and $M = 40$.



(a) $N = 20$ (b) $N = 12$

Fig. 9. WLAN - One Hop Wheel Topology. Flow Completion Time of Algorithm A, Algorithm B, and DCC when the number of interfaces is either 2 or 3, $\alpha = 0.1$, $p = 0.1$, $1/\mu = 100$, $M = 40$, and (a) $N = 20$ (b) $N = 12$.

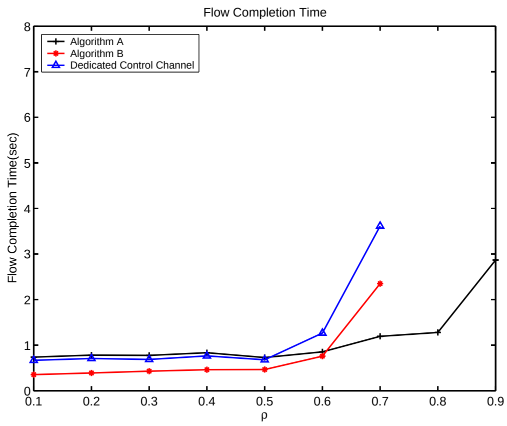


Fig. 11. WLAN - One Hop Random Topology. Flow completion time of Algorithm A, Algorithm B, and DCC when $N = 20$, $\alpha = 0.1$, $p = 0.1$, $1/\mu = 100$, and $M = 40$.

$100m \times 100m$ terrain and one of the nodes is selected as an AP, Fig. 5-(b).

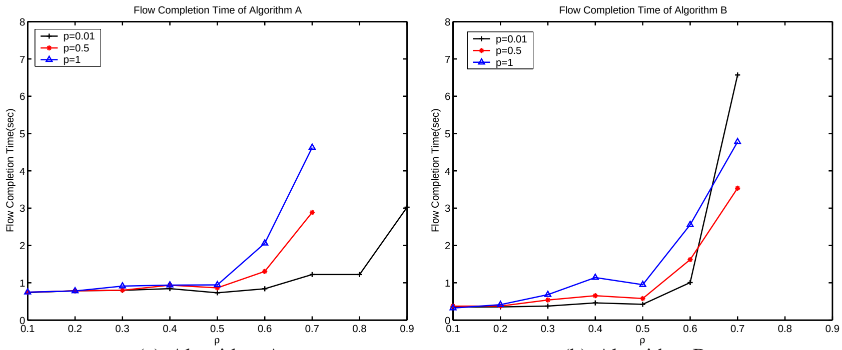
Fig. 10 shows that flow completion time performance of Algorithm A and Algorithm B is consistent both in one hop wheel and one hop random topologies for different channel drop probabilities, p. Furthermore, Fig. 11 shows that flow completion time characteristics of Algorithm A, Algorithm B, and DCC in one hop random topology follow a similar trend as in one hop wheel topology for $p = 0.1$. This result indicates the effectiveness of our algorithms in a different topology.

3) WLAN - Two Hop Wheel Topology: In this part, $M = 39$ nodes including AP and clients are placed in a $300m \times 300m$ terrain according to wheel topology where circle radius is $75m$, Fig. 5-(c).

Fig. 12 presents flow completion times of Algorithm A, Algorithm B, and DCC for channel drop probability, $p = 0.1$. As it is seen flow completion times of Algorithm A and DCC diverge quickly at lower loads. The reason is that AP is the bottleneck of the system in this topology, because although all clients try to access to AP to communicate to their destinations, AP can use at most one channel to handle total traffic. Thus, packets wait a lot in AP's buffer, and flow completion time increases boundless. On the other hand, this problem does not exist in Algorithm B as shown in Fig. 12, because Algorithm B allows nodes to own more than one channel. It is observed from simulations that AP can own up-to 5-10 channels when Algorithm B is used and handles traffic successfully. It is important that Algorithm B solves bottleneck-node problem, because it is a common problem in both WLANs (APs may be bottleneck) and WMNs (intermediate nodes may be bottleneck).

4) WMN - One Hop Random Topology: In this topology, $M = 40$ nodes are distributed randomly in a $100m \times 100m$ terrain. In this WMN setting there is no access point; nodes select their destinations randomly and transmit to them directly (over one hop), Fig. 6-(a).

Fig. 13 shows flow completion times of Algorithm A and Algorithm B for different channel drop probabilities, p. The figure shows that the performance curves of Algorithm A and Algorithm B follow a similar trend as in WLAN topologies, i.e., our algorithms exhibit similar



(a) Algorithm A (b) Algorithm B

Fig. 10. WLAN - One Hop Random Topology. Flow completion time of (a) Algorithm A and (b) Algorithm B when $N = 20$, $\alpha = 0.1$, $1/\mu = 100$, and $M = 40$.

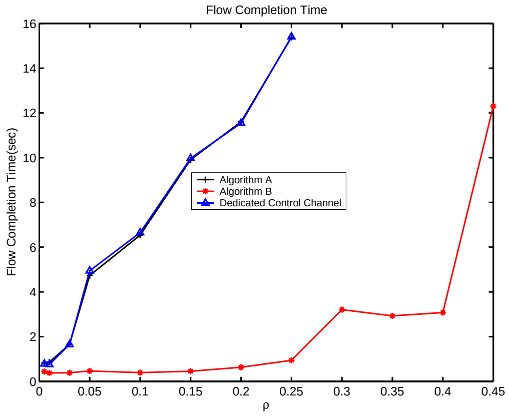


Fig. 12. WLAN - Two Hop Wheel Topology. Flow completion time of Algorithm A, Algorithm B, and DCC when $N = 20$, $\alpha = 0.1$, $p = 0.1$, $1/\mu = 100$, and $M = 39$.

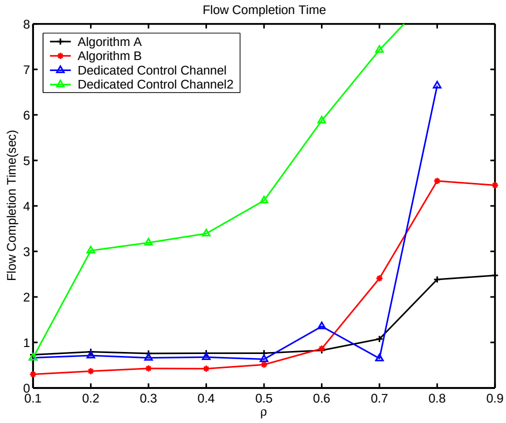


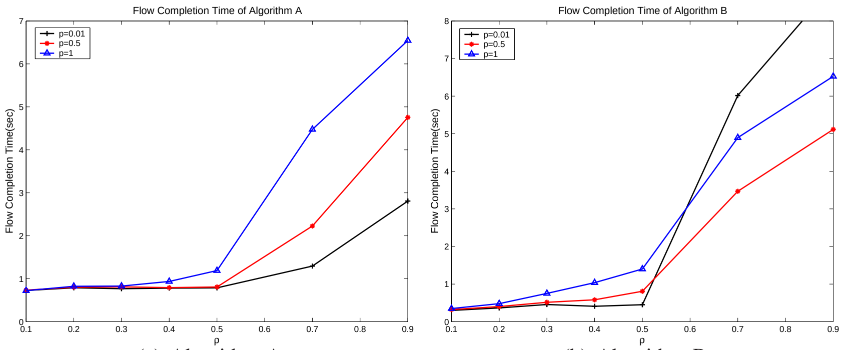
Fig. 14. WMN - One Hop Random Topology. Flow completion time of Algorithm A, Algorithm B, DCC, and DCC2 when $N = 20$, $\alpha = 0.1$, $p = 0.1$, $1/\mu = 100$, and $M = 40$.

flow completion time performance in both WLAN and WMN topologies. This observation is intuitive because we have designed our algorithms as independent from underlying network and topology.

Fig. 14 shows flow completion time of Algorithm A, Algorithm B, DCC, and DCC2 for one hop random topology in WMN. DCC2 is Dedicated Control Channel (DCC) in which nodes begin to monitor the control channel when they become active instead of continuously monitoring the control channel. As in the previous simulation results, Algorithm A and Algorithm B exhibit similar flow completion time improvements while Algorithm B is better. Flow completion time improvement of DCC2 is worse than Algorithm A, Algorithm B, and DCC, because active nodes in DCC2 do not have enough information on previously reserved channels, so they can attempt to already owned channels, i.e., a channel may be owned by more than one user. This increases collisions and flow completion time. We have presented DCC2 results in here in addition to DCC results to demonstrate the effectiveness of our algorithms as well as Dedicated Control Channel when there is limited information about channel occupancies. Since our algorithms are ad-hoc, channel occupancy knowledge does not have any effect on flow completion times. However, it is crucial for Dedicated Control Channel. This results prove that our algorithms are more suitable for practical applications.

5) WMN - One/Two Hop Grid Topology: In this topology, $M = 39$ nodes are distributed to $300m \times 300m$ terrain which is divided into 9 equal sized grids, Fig. 6-(b). $M = 39$ nodes are divided into six $4-node$ and three $5-node$ sets. Each set is assigned to a grid randomly and the nodes in a set are randomly placed in each grid. When a flow is generated (i.e. when a node becomes active for transmission), the node selects its destination randomly. If the source and destination nodes are in the same grid or in the contiguous grids, the transmission is achieved directly (over one hop). Otherwise, a node is randomly selected from one of common neighborhood grid(s) of the source and destination nodes to route packets, i.e., the selected node becomes a relay for two-hop transmission.

Fig. 15 shows the flow completion time of Algorithm A, Algorithm B, and DCC for one/two hop grid topology in WMN. It is seen from the figure that the flow completion time performance of Algorithm A and Algorithm B are comparable with their performances in different WLAN and WMN topologies. However, DCC's flow completion time is worse than both Algorithm A and Algorithm B in this topology due to hidden terminal problem. In DCC, hidden terminals are unable to listen all channel reservations over the control channel, so it is possible that some channels are owned by multiple



(a) Algorithm A

(b) Algorithm B

Fig. 13. WMN - One Hop Random Topology. Flow completion time of (a) Algorithm A and (b) Algorithm B when $N = 20$, $\alpha = 0.1$, $1/\mu = 100$, and $M = 40$.

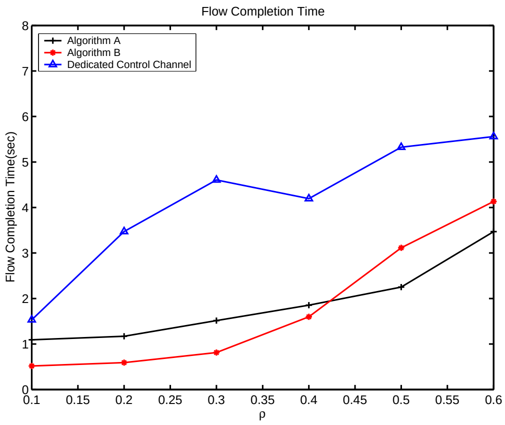


Fig. 15. WMN - One/Two Hop Grid Topology. Flow completion time of Algorithm A, Algorithm B, and DCC when $N = 20$, $\alpha = 0.1$, $p = 0.1$, $1/\mu = 100$, and $M = 39$.

users. This increase the number of collisions and the flow completion time. This result shows that our algorithms are not affected by hidden terminal problem, hence they are more practical.

VI. Conclusion

In this paper, we considered multi-channel medium access (MAC) protocols. We proposed two new multi-channel MAC algorithms, called Algorithm A and Algorithm B, and presented a theoretical analysis of their performance. In particular, we showed that Algorithm A can achieve 100% throughput in the limit of large flow sizes, in a single access point setting. In this setting, we observed a good match between the theoretical flow completion time analysis and basic simulation results. We then presented extended simulation results of Algorithm A and Algorithm B for different topologies of WLAN and WMN using the Glomosim simulation environment. We compared flow completion times of Algorithm A and Algorithm B with the Dedicated Control Channel (DCC) algorithm. The results show that our algorithms show comparable or better performance as compared to DCC without having some of the practical limitations of DCC.

References

[1] P. Gupta and P. Kumar, "Capacity of Wireless Networks," *in IEEE Trans. on Information Theory*, vol. 46, pp. 388-404, March 2000.

[2] D. Bertsekas and R. Gallagher, "Data Networks," Prentice Hall, 2nd edition, 1992.

[3] N. Jain, S. Das, and A. Nasipuri, "A multi-channel CSMA MAC Protocol with Multiple Receiver-Based Channel Selection for Multi-Hop Wireless Networks," *in Proc. of IEEE ICCCN*, 2001.

[4] Z. Ji, Y. Yang, J. Zhou, M. Takai, and R. Bagrodia, "Exploiting Medium Access Diversity in Rate-Adaptive Wireless LANs," *in Proc. of ACM MOBICOM*, 2004.

[5] V. Kanodia, A. Sabharwal, and E. Knightly, "A Multi-Channel Opportunistic Auto-Rate Media Access Protocol for Ad-Hoc Networks," *in Proc. of IEEE BROADNETS*, 2004.

[6] J. So and N. Vaidya, "Multi-Channel MAC for Ad-Hoc Networks: Handling Multi-Channel Hidden Terminals Using a Single Transceiver," *in Proc. of ACM MOBIHOC*, 2004.

[7] Z. Tang and J. Garcia-Luna-Aceves, "Hop Reservation Multiple Access (HRMA) for Multi-Channel Packet Radio Networks," *in Proc. of IEEE IC3N98, Seventh International Conference on Computer Communications and Networks*, 1998.

[8] M. K. Tsatsanis, R. Zhang, and S. Banerjee, "Network-Assisted Diversity for Random Access Wireless Networks," *in IEEE Trans. on Signal Processing*, 2000.

[9] A. Tzamaloukas and J. Garcia-Luna-Aceves, "Channel-Hopping Multiple Access," *in Proc. of IEEE ICC*, pp. 415-419, 2000.

[10] A. Tzamaloukas and J. Garcia-Luna-Aceves, "Channel-Hopping Multiple Access with Packet Trains for Ad-Hoc Networks," *in Proc. of IEEE Mobile Multimedia Communications*, Tokyo, 2000.

[11] S. Wu, C. Lin, Y. Tseng, and J. Sheu, "A Dynamic Multi-Channel MAC for Ad-Hoc LAN," *in Proc. of International Symposium on Parallel Architectures, Algorithms and Networks (ISPAN 00)*, pp. 232-233, Dec. 2000.

[12] S. Wu, Y. Tseng, C. Lin, and J. Sheu, "A Multi-Channel MAC Protocol with Power Control for Multi-Hop Mobile Ad-Hoc Networks," *The Computer Journal*, 2002.

[13] H. W. So, J. Walrand, "Design of a Multi-Channel MAC Protocol for Ad-Hoc Wireless Networks," Technical Report No. UCB/EECS-2006-17, Feb. 2006.

[14] M. Cao, V. Raghunathan, and P. R. Kumar, "Cross Layer Exploitation of MAC Layer Diversity in Wireless Networks," *in Proc. of IEEE ICNP*, 2006.

[15] A. Bertossi and M. A. Bonuccelli, "Code Assignment for Hidden Terminal Interference Avoidance in Multi-Hop Packet Radio Networks," *in IEEE Trans. on Communications*, vol. 3(4), pp. 441-449, Aug. 1995.

[16] J. Garcia-Luna-Aceves and J. Raju, "Distributed Assignment of Codes for Multi-Hop Packet Radio Networks," *in Proc. of IEEE MILCOM*, vol. 1, pp. 450-454, 1997.

[17] L. Hu, "Distributed Code Assignment for CDMA Packet Radio Networks," *in IEEE Transactions onf Networking*, vol. 1, pp. 668-667, 1993.

[18] A. Muqattash and K. Marwan, "CDMA Based MAC Protocol for Wireless Ad-Hoc Networks," *in Proc. of IEEE MOBIHOC*, pp. 153-164, June 2003.

[19] J. Mo, H. S. W. So, J. Walrand, "Comparison of Multi-Channel MAC Protocols," *in IEEE MSWiM*, Oct. 2005.