AMCM: Adaptive Multi-Channel MAC Protocol for IEEE 802.11 Wireless Networks

Paul Tan [†‡]
Institute for Infocomm Research[†]
Singapore
tanpaul@i2r.a-star.edu.sg

Mun Choon Chan
School of Computing[‡]
National University of Singapore
chanmc@comp.nus.edu.sg

Abstract

This paper presents AMCM, a traffic-adaptive multi-channel MAC protocol that increases the capacity of wireless network by enabling multiple concurrent transmissions on orthogonal frequency channels using a single half-duplex transceiver. AMCM is based on the IEEE 802.11 MAC but provides fine-grain, asynchronous coordination among locally interfering nodes for channel negotiation. By incorporating load-awareness, channel availability awareness and batch transmissions, our window-based approach achieves high channel utilization under varying load, while avoiding the control-window saturation problem as the number of channels increases. For single-hop scenarios, we show that, at low load, AMCM is comparable to IEEE 802.11 MAC, while under high load, AMCM delivers almost $N\times$ improvement gain over IEEE 802.11 MAC protocol, where N is the number of channels. AMCM also outperforms existing multi-channel MAC protocols [8, 11] by 100% and 150% respectively under high load at a lower hardware cost and complexity. In multi-hop scenarios, AMCM achieves performance improvement of 190% and 90% for both dense and sparse network over IEEE 802.11 MAC respectively. In both scenarios, AMCM achieves close to full utilization of all channels with good protocol efficiency.

1. Introduction

IEEE 802.11 [1] is the *de-facto* wireless networking standard for wireless local area network (WLAN). Currently, the standard has four specifications which includes IEEE 802.11, IEEE 802.11a, IEEE 802.11b and IEEE 802.11g. Each of these specifications differs in their operating frequency range, modulation scheme and transmission speed. The standard also supports the use of multiple channels. This enables multiple transmissions to take place simultaneously without causing interference to each other. Clearly, by exploiting multiple channels, the capacity of the wireless network can be increased. Unfortunately, the original MAC protocol is designed for single-channel wireless network and thus cannot capitalized on this multi-channel capability.

In this paper, we propose a new adaptive multi-channel CSMA/CA-based MAC protocol called AMCM. AMCM enables multiple concurrent transmissions over several frequency channels using a single half-duplex transceiver. AMCM is based on the IEEE 802.11 MAC. The main idea is to introduce a traffic-adaptive control window to provide fine-grain channel negotiations among locally interfering nodes. The protocol has several key features. Firstly, the protocol does not requires network-wide synchronization nor does it requires any dedicated control channel for channel negotiation purposes. Secondly, by dynamically adapting the size of the control window to varying traffic load, AMCM mimics single-channel IEEE 802.11 MAC during low load, while enabling multiple concurrent transmissions during high load with good protocol efficiency.

Extensive simulation results show that in single-hop (WLAN-like) scenarios, AMCM delivers almost $N\times$ improvement over the IEEE 802.11 MAC protocol, where N is the number of channels. In multi-hop scenarios, AMCM performs better in dense network as the demand for concurrent transmissions increases. The performance improvement achieved over IEEE 802.11 is 190% and 90% for dense and sparse network respectively. In high load and network density, AMCM achieves close to full utilization of all channels.

The paper is organized as follow. Related work is presented in Section 2, follow by the description of AMCM in Section 3. Simulation results are presented in Section 4. Finally, we conclude in Section 5.

2. Related Work

Several methods have been proposed to increase the capacity of wireless networks such as IEEE 802.11 DCF enhancements [2, 3, 4] , the use of directional antennas [5, 6, 7] and multi-channel MAC [8, 9, 10, 11, 12] protocols. In this paper, we will only focus on exploiting frequency diversity by leveraging on multiple orthogonal frequencies. The major differences among these multi-channel solutions are in their use of a single- or multiple transceivers and channel assignment or coordination mechanism.

So and Vaidya propose Multi-channel MAC (MMAC) [8], a single-transceiver solution which uses the *Ad Hoc Traffic Indication Messages (ATIM)* to perform channel reservation. MMAC requires nodes to be synchronized such that every node can start the beacon interval at about the same time. Unfortunately, this tight synchronization requirement can be a problem in multi-hop networks. Even

1-4244-0425-8/06/$20.00 ©2006 IEEE

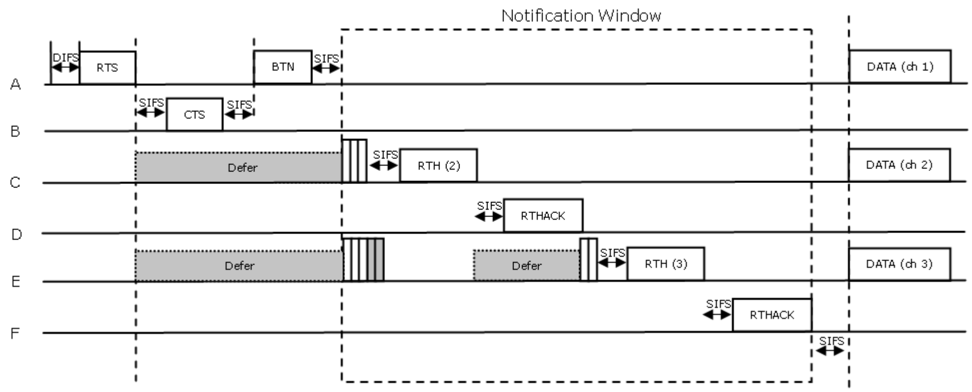


Figure 1. Operations of AMCM with 3 competing traffic flows (A→B, C→D, E→F)

though MMAC uses all available channels for data exchange, the overheads incurred by the periodic beacon transmissions and ATIM packets can result in lower performance gain over IEEE 802.11 MAC.

Another single-transceiver solution is SSCH [9]. It differs from ”rendezvous” channel coordination mechanism (such as [8]) whereby nodes periodically meet on the primary channel to perform channel negotiations. In contrast, SSCH adopts a pseudo-random sequence to allow nodes to decide which channel to switch for the next 10ms. This duration is chosen as a tradeoff between the channel switching overheads and forwarding delay in multi-hop wireless networks.

Nasipuri et al. [10] propose a *soft channel reservation*-based multi-channel CSMA protocol. It assumes that each node can listen to all N channels simultaneously. To transmit, the sender must first search for an idle channel. When more than one idle channel exists, the channel that was used during the last transmission is always preferred; thus *soft-reservation*. This protocol has low control overheads, but unfortunately increases the hardware cost and complexity of the node since N transceivers are required.

Wu et al. [11] propose an on-demand dynamic channel assignment protocol (DCA) which assigns a dedicated channel for control purposes, and other channels for data. As such, DCA requires each node to be equipped with *two* transceivers. The idea is to listen to both control and data channels at the same time. The channel assignment/negotiation is done during the RTS/CTS exchange. One of the advantages of DCA is the non-existence of *multi-channel hidden-node problem* since nodes always listen to the control channel. Apart from increased in per-node hardware cost, DCA requires the use of a dedicated control channel in IEEE 802.11b (3 channels) results in 33% of the total bandwidth as the control overhead and possible poor channel utilization. With higher number of channels available (e.g. IEEE 802.11a), *control channel saturation* [8] problem can arise since all data channel assignments/negotiations are performed over a single control channel.

Another multi-transceiver solution is MUP [12]. The authors proposed a new link layer protocol, called the Multi-radio Unification Protocol (MUP), that coordinates multiple IEEE 802.11 radios operating over multiple channels.

Our solution proposed in this paper is partly inspired by the work in [13]. The authors in [13] proposed the Medium Access via Collision Avoidance with Enhanced Parallelism (MACA-P). MACA-P extends the basic IEEE 802.11 MAC protocol to allow parallel transmissions using single channel. Unlike traditional IEEE 802.11, the actual data transmission is *delayed by a control phase interval*, which allows multiple sender-receiver pairs to synchronize/align their data transfers. Similarly, AMCM also leverages on the idea of exchanging control messages prior to the start of the multiple data transmissions over different channels.

In comparison, our multi-channel MAC protocol (AMCM) is simple, adaptive and cost-effective. It does not required periodic network-wide synchronization, but instead required only locally affected nodes to synchronized themselves to a common window. AMCM also scales with the number of channels and adapts well to varying traffic load with good protocol efficiency. As we will show later through extensive simulations, AMCM avoids control channel saturation problem by incorporating both *channel availability awareness* and *batch transmissions*.

3. Adaptive Multi-Channel MAC

In this section, we first give an overview of the design of AMCM follow by detail descriptions.

First, given a set of N (> 1) orthogonal channels, a *primary or default channel* is chosen and is assumed to be known to all nodes in the network. The rest of the channels (N - 1) are *secondary channels*.

Figure 1 shows an illustration of a typical operation cycle of the AMCM protocol. At the beginning of the cycle, node A (the sender) acquires the default channel through the use of RTS/CTS exchanges with node B. As a result, knowing that the default channel is not available, nodes C to F will attempt to communicate by acquiring one of the secondary channels. This acquisition is performed in a duration called *N*otification Window (NW) that begins after the reception of the *B*TN message sent by node A (sender of the winner of the primary channel). The duration of NW is announced in the RTS, CTS and BTN messages. Notification frames exchange during NW include information on the *s*econdary channel to be acquired and the *reservation duration* in which both nodes will spend on the new channel. Nodes that successfully acquired the secondary channels switch to their respective secondary channels only at the end of NW. Upon completion, these nodes switch back to the default channel. The cycle then repeats.

One key feature of AMCM is that nodes dynamically negotiate and switch channel in a *distributed and asynchronous* manner. There is no static negotiation period or pre-assigned dedicated channel for negotiation. Instead, the protocol let nodes dynamically synchronize/align themselves locally to a common notification window for secondary channel acquisition. In addition, the duration of NW and reservation duration per channel are adapted according to the traffic load and topology.

The three components of AMCM are *acquisition of secondary channel* (Section 3.1), *operating in secondary channel* (Section 3.2) and *return to primary channel* (Section 3.3). The details are presented in the following sections.

3.1. Acquisition of Secondary Channels

3.1.1. Behavior on Primary Channel

As shown in Figure 1, when a node has packets to transmit, it transmits RTS to its intended receiver, which then responds with CTS. In addition, upon receiving the CTS, the sender transmits a new frame called *Begin-To-Notify* (BTN).

This three-way handshake achieves the follow purposes: (1) reserves the primary channel for data transmission; (2) alleviates potential hidden-terminal problem, which is similar to IEEE 802.11 DCF and (3) announces to the neighboring nodes the upcoming NW and its duration.

Item (3) is key to the asynchronous operations of AMCM. Note that nodes overhearing RTS frame should not take it as a confirmation that the primary channel has been acquired by the RTS-sender since it is possible that the receiver might not respond with a CTS for some reasons. Instead, overhearing nodes should treat it as a *tentative confirmation* and only update their NAV accordingly. For this reason, only the CTS and BTN frames are taken as a confirmation (for both the neighbors of the CTS-sender and RTS/BTN-sender). Once a confirmation is overheard, nodes must now align themselves to the upcoming NW and also the duration advertised.

Nodes overhearing CTS frame will start NW after τ_{cts} seconds, while nodes which overhear BTN frame will start NW after τ_{btn} seconds, where

$$\tau_{cts} = 2 \times t_{SIFS} + t_{BTN} \tag{1}$$

$$\tau_{btn} = t_{SIFS} \tag{2}$$

where t_{BTN} is the time taken to transmit a BTN frame using base rate. These values are shown in Table 1.

Duration of NW is carried in the RTS, CTS and BTN messages to ensure that it is heard by all nodes in the RTS/CTS range. For convenient, we will call both the BTN-sender and CTS-sender *NW-Initiators*. Determining the duration of NW is discussed in more detail in Section 3.1.3.

Once neighboring nodes detect a busy primary channel during an attempt to transmit, it freezes its backoff timer and update the NAV accordingly. It also aligns itself to the upcoming NW. Cases whereby a CTS response frame is not received after sending RTS frame should not be perceived as a collision or exposed-terminal problem. In the latter case, it could be that the RTS-recipient might be inside NW and therefore is unable to respond. In this case, the RTS-recipient (inside NW) can inform the RTS-sender about the current NW, and also to notify it to switch channel. We termed this as receiver-initiated notification.

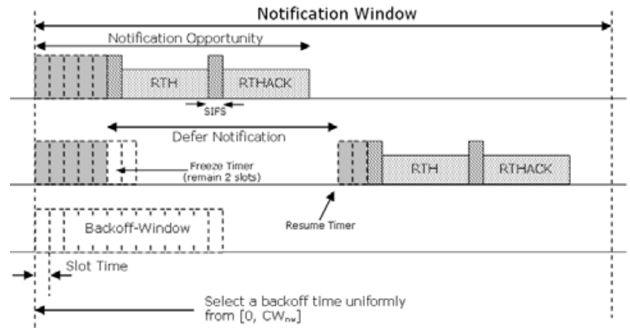


Figure 2. Contention-Window inside NW

In a multi-hop environment, it is also possible (though infrequent) that a node hears multiple NWs. In such cases, a node will accept only the first BTN/CTS message it receives and discard the rest.

3.1.2. Behavior Inside NW

Within a NW, all nodes competing for a secondary channel will now randomly select a backoff time in the interval [0,CW_{NW}]. The backoff time is decremented as long as the channel is sensed idle, stops when the channel is busy and resumes again when the channel is idle again. When the timer reaches zero, the node transmits RTH (*Request-to-Hop*) frame to its receiver, which then responds with RTHACK (*Request-to-Hop-Acknowledge*) frame. Note that each node maintains a separate backoff timer for this purpose. If nodes are unsuccessful in reserving a secondary channel during the current NW, they do not reset their timer, but instead resume it in the next NW. Nodes with no packet to transmit must update their channel usage list accordingly and remain idle throughout NW. Figure 2 shows an illustration of the contention and backoff inside a NW. Since all nodes are aware of the duration of NW, nodes will make sure that they do not transmit a RTH frame if the notification exchange (RTH/RTHACK) cannot be completed within the NW.

Both the RTH/RTHACK frames contain the information on the selected channel to switch to and the reservation duration on that new channel. Overhearing nodes will update their free-channel list accordingly to avoid selecting the same channel. These overhearing nodes also include those (new nodes) that are not aware of the current NW. Lastly, all nodes must stay on the primary channel throughout NW. This is to minimize any inconsistency in the channel usage list. The requirement for nodes to stay until the end of NW can pose a problem when the number of channels is large (such as IEEE 802.11a). Fortunately, this problem can be alleviated by adapting NW, which we will verify later through simulation in Section 8.

Again, in a multi-hop environment, it is possible (though unlikely) that a node receives RTH/RTHACK messages from different NWs. In such cases, nodes should still update their channel list accordingly. In addition, such conflict

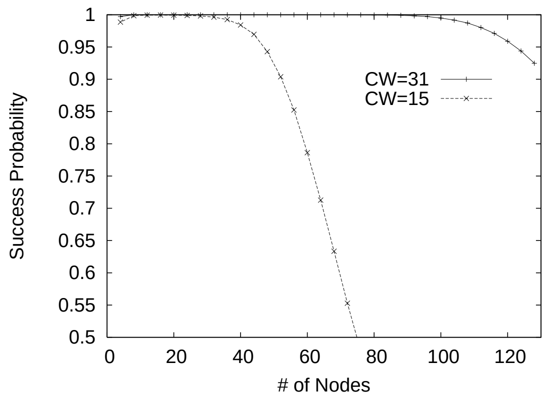


Figure 3. Probability of Acquiring Channel

can also lead to inconsistency in the status of free channel list among different nodes. However, since such collision are rare and collision on the secondary channel can be detected and resolved easily (see Section 3.2), the protocol is sufficiently robust and efficient.

One of the key parameters of AMCM is the value of CW_{NW}. CW_{NW} should be chosen such that it is large enough for a secondary channel to be acquired without collision in a single notification exchange and at the same time without incurring too much overhead. The collision probability depends on the number of flows contending for a secondary channel in a single collision domain. Through simulation and assuming a uniform random distribution, Figure 3 shows the probability that a secondary channel is successfully acquired when the number of nodes increases for CW_{NW} values of 15 and 31. While a value of 15 is sufficient for moderate number of nodes, a value of 31 ensures that even with more than 100 nodes contenting, the probability of success is still greater than 99%. Such flow density is sufficiently high for most network density settings and is used as the default value.

3.1.3. Adaptive Notification Window

The size of NW (γ) is decided by the RTS-sender and is broadcasted again by the CTS/BTN-sender. γ determines the number of RTH/RTHACK (notification) exchanges possible within one NW and should be a function of the number of free channels and traffic load.

At low load scenario whereby secondary channels are not required, γ should be small enough to reduce overhead on the default channel. During high load where multiple concurrent transmissions are desired, γ should be adjusted accordingly to utilize all available secondary channels. Ideally, it is desired that all available secondary channels be reserved during a single NW.

γ can be expressed in terms of number of *Notification Opportunity* (NOP). Ideally, each opportunity allows a secondary channel to be reserved/utilized. γ is calculated as follow:

$$\gamma = NOP \times t_{NOP} \tag{3}$$

$$t_{NOP} = CW_{NW} \times t_{slot} + t_{RTH} + t_{SIFS} + t_{RTHACK} \tag{4}$$

where t_{NOP} is the (maximum) size of each opportunity, t_{RTH} and t_{RTHACK} is the transmission time of RTH and RTHACK frame respectively. Assuming that one secondary channel is always acquired in each opportunity, then $NOP \leq N - 1$.

Algorithm 1 updateNOP - After NW expires

```
N_free ⇐ getNumFreeChannels()
if N_free > 0 then
    NOP ⇐ max(NOP − 1, NOP_min)
end if
unsuccessful_notify ⇐ isUnSuccessfulInNW()
```

Algorithm 2 computeNOP - Before transmitting RTS

```
NOP_old ⇐ NOP
if unsuccessful_notify then
    min(NOP + 1, NumChannels − 1)
    N_free ⇐ getNumFreeChannels()
    if N_free == 0 then
        NOP ⇐ NOP_old
    end if
end if
```

In this paper, we adopt a simple and effective algorithm to adapt NOP to both the traffic load and channel availability. The algorithm is described formally in Algorithm 1 and 2. NOP_{min} is 0 by default. The values of NOP are updated at the end of each NW and before the start of a NW. *NW-initiator* advertises the value of NOP using RTS, CTS and BTN frames on the primary channel and it is assumed that nodes have equal opportunity to acquire the primary channel.

The idea is to decrement NOP if there are any free secondary channels based on the current channel list information. Availability of free channel likely indicates that traffic load is low or no demand for secondary channels, thus γ or NOP can be reduced. In addition, nodes contending for a secondary channel will also check if they were successful in reserving a secondary channel during NW. Unsuccessful nodes will set their *unsuccessful_notify* flags to *true*.

On the other hand, NOP is incremented if the *NW-Initiator* was unsuccessful during the previous NW. However, in cases whereby there is no free channel available, the value of NOP should remain the same. As the channel reservation duration is included in the RTH/RTHACK messages, nodes can estimate the number of free channels (channel availability) during each NW, and can better determine the proper value of NOP.

To summarize, for a *NW-Initiator*, the value of NOP: (1) is decrement if there is free channel at the end of the previous NW and the *unsuccessful_notify* flag is not set. (Lower bound is NOP_{min}); (2) is increment if the *unsuccessful_notify* flag is set and there are free channels available. (Upper bound is $N - 1$); (3) otherwise, it remains the same.

3.1.4. Channel Switching Threshold (CST)

Within a single notification opportunity, nodes indicate the reservation duration, $T_{Reserve}(i)$, they will stay on the new channel i after NW elapses. Clearly, a longer duration on the new channel increases the total throughput by allowing more DATA packets to be transmitted. On the other hand, a shorter duration allows other nodes to contend for the secondary channel sooner and also to communicate with them. However, the overhead increases with smaller threshold, resulting in lower channel utilization.

CST prevents nodes with high sending rate (source-traffic) from occupying he new channel for too long, avoid unreachability[1] problem and also to minimize any unfairness problem.

Specifically, RTH-sender inspects its packet queue for the first few packets designated for the same destination node (next hop) and sets the duration ($T_{Reserve}(i)$) as the time it takes to transmit these packets. CST specifies the maximum number of packets in which a node is allowed to transmit on the negotiated channel. Assuming packets are of the same size,

$$T_{Reserve}(i) = t_{data/ack} \times min\{N_{queue}, \mathit{CST}\} + t_{rts/cts} \tag{5}$$

where N_{queue} is the number of first few packets in the queue designated for the destination node and $t_{data/ack}$ is the duration for a single DATA/ACK transmission. $t_{rts/cts}$ is the time taken for a single RTS/CTS exchange on the secondary channel.

When a node is transmitting to many destinations (such as relays in wireless ad-hoc networks), N_{queue} may be small. One possible approach to increase N_{queue} is to employ per-destination queuing such that even packets that are queued behind other packets can be transmitted. In general, a small CST ensures node reachability and is more fair, whereas a larger CST allows more packets to be transmitted per channel switch and also reduces the switching frequency. Unfortunately, if the threshold is too small, nodes can spend most of their time switching across channels, incurring more overheads and thus not utilizing the channels efficiently. The impact of CST settings will be investigated using simulation in Section 4.2.4.

3.2. Operating in Secondary Channel

Once switching over to the secondary channel, nodes perform an initial channel sensing for activity and then a single RTS/CTS exchange to avoid potential hidden-terminal problem. Once the exchange is completed, nodes initiate their data transmission with a series of DATA-ACK exchanges. We term this batch transmission. NAV information is also included in the RTS/CTS/DATA frame indicating the remaining channel reservation duration.

By default, nodes on the secondary channel return back to the default (primary) channel after the reservation duration which they advertised during NW. However, nodes can still return prematurely when a collision is detected. In cases where nodes detect busy channel during the initial channel sensing, both nodes must return. However, since this is probably due to incorrect channel usage list, therefore both nodes can defer their return until they overhear the NAV information contains in subsequent DATA transmission. This information can then be used to update the nodes' current channel list.

In AMCM, each node maintains one data structure, *Neighbor Channel List* (NCL) to update the status of all channels. For each channel *i*, NCL[i] contains address of the sender, address of the receiver and channel reservation duration.

Every node in the network maintain their own *NCL*. Upon overhearing control frames, nodes refresh their list accordingly to obtain an up-to-date information of all channels. In fact, this list is equivalent to the *NAV* in IEEE 802.11 MAC but is meant for multiple channels. Nodes also use this list to avoid transmission/notification to any node specified in the list.

We adopt a simple channel selection based on the NCL. In AMCM, RTH-senders choose the next available channel based on NCL. When a receiver receives a RTH frame inside NW, it consults the NCL and checks if the proposed channel is available. If the proposed channel has been reserved, the receiver responds with a *negative* RTHACK (channel ID is 0).

3.3. Return to Primary Channel

Since returning nodes (and in fact new nodes) may not be aware of the state of the primary channel and secondary channels, they must remain silence (to avoid collision) until they either overhear an upcoming NW initiation (i.e. RTS/CTS/BTN), current NW (RTH/RTHACK) or at least the duration of a maximum transfer unit (MTU).

When nodes return prior to an upcoming NW, nodes can overhear and decode the NAV information inside RTS/CTS/BTN frames and therefore defer themselves and align to the upcoming NW. Otherwise, if they return in the midst of NW, nodes remain silence until they can correctly deduce the state of all channels. When nodes overhear either RTH/RTHACK frames, they should align to the current NW. Node can then contend inside NW to reserve any free secondary channel.

Note that it is possible that returning nodes miss an earlier reservation and therefore try to reserve the same channel due to inconsistent NCL. However, when these nodes switch over to the new channel, they will eventually sense a busy channel due to ongoing data transmissions. In this case, nodes will update their channel list accordingly before switching back to the default channel.

4. Performance Evaluation

In this section, we evaluate AMCM through simulation against both single-channel (original IEEE 802.11 MAC) and multi-channel (MMAC[8], DCA[11]) MAC protocols.

For evaluation purpose, we use aggregate throughput, average packet delay, control overhead and fairness index as our performance metrics. For ease of comparison, we defined throughput ratio as the ratio of the throughput achieved by AMCM over the throughput achieved by IEEE 802.11MAC. Similarly, delay ratio is defined as the ratio of the average packet delay achieved by IEEE 802.11 MAC over AMCM. Control overhead is defined as the total number of control packets per DATA packet delivered. This metric measures the protocol efficiency of AMCM (under varying load) since it relies on additional messages exchange for both channel negotiation and collision avoidance.

Ideally, given N channels, the optimum aggregate throughput of a multi-channel MAC protocol should be

[1]nodes operating on the secondary channels will still be reachable within a short time frame

Transmission rate	2 Mbps
Transmission range	250m
Slot time	20μsec
SIFS	10μsec
Synchronization time	192μsec
RTS/CTS/BTN/ACK	20 bytes (272μsec)
RTH/RTHACK	32 bytes (320μsec)
Size of one NOP	960μsec

Table 1. Simulation Parameters

$N \times \alpha$, where α is the per-channel saturated throughput. Unfortunately, this idealistic improvement cannot be easily realized due to the control overheads incurred in some multi-channel MAC protocol ([8, 11, 9]) and the control-channel saturation problem [11]. As we will demonstrate later, besides achieving significant performance improvement in high load, AMCM achieves almost ideal linear performance scale up with the number of channels and works well in low load by reducing the control overhead adaptively.

4.1. Simulation Model

We simulate AMCM using Glomosim in both single-hop (Section 4.2) and multi-hop (Section 4.3) wireless networks. In each simulation, all nodes are configured with the same MAC protocol, operating at a raw data rate of 2 Mbps and a transmission range of 250m. For multi-channel MAC protocols, we assumed 3 orthogonal channels. The channel switching overhead is negligible and is ignored in this study. All nodes running AMCM are equipped with a single half-duplex transceiver. For traffic type, each source node generates and transmits constant-bit rate (CBR) traffic sending 1000 packets per second (pps). By default, γ, CW_{NW} and CST are configured to be 5 notification opportunities, 31 time-slots and 100 packets respectively. Each simulation run lasts for a duration of 300 seconds, and all results are averaged over 10 independent runs. The parameters used in this paper are summarized in Table 1.

4.2. Single-Hop

For single hop scenario, nodes are randomly placed within a square area and are within wireless coverage of each other. Therefore, every source node can reach its destination in a single-hop. We randomly select half of the nodes to be sources, while the other half to be destinations. We do not consider cases where a node sends to multiple destinations. The impact of the following parameters are studied: (i) traffic load, (ii) number of channels and (iii) CST.

4.2.1. Impact of Number of flows

In this section, we keep the traffic rate per flow constant (at 1000pps) and study the capacity of the wireless network as the number of flows increases. Since nodes are within each other's transmission and interference range, therefore only one communication flow can exist at any given time for the single-channel (IEEE 802.11) case.

From Figure 4, we observed that with only a single flow and packet size of 64 bytes, static AMCM (without NW adaptation) performs much worst (throughput ratio of 0.75) compared to IEEE 802.11 MAC due to the overheads of the BTN frames and large (static) NW. The throughput obtained in this case were 120kbps, 85kbps and 110kbps for IEEE 802.11, static AMCM and AMCM[2] respectively. Hence, the use of adaptive NW mechanism is crucial in reducing the performance gap at low load as the overhead is reduced to only BTN frames exchange. The throughput ratios of AMCM (with NW adaptation) over IEEE 802.11 are 1.0 for both 64 and 1500 byte packets.

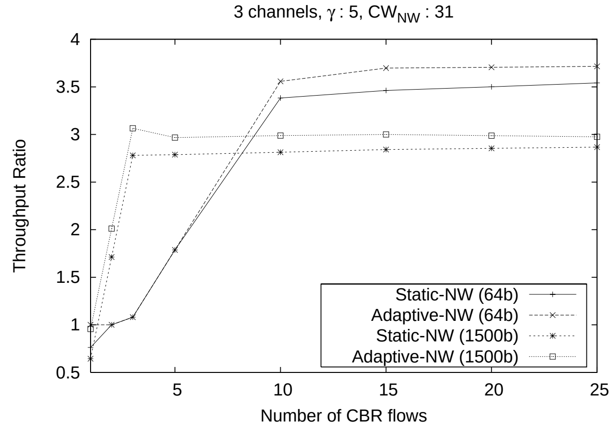


Figure 4. WLAN: Impact of number of traffic flows

As the number of traffic flows increases, the performance gain achieved by AMCM becomes more significant. For IEEE 802.11 MAC, the saturated throughput (one-hop capacity) of the network is only 320kbps (1.6Mbps) for 64 (1500) bytes packet, while AMCM delivers aggregate throughput of up to 1.1Mbps (4.8Mbps) for packet size of 64 (1500) bytes. Compared to IEEE 802.11 MAC, AMCM achieved significant throughput gain of 3.5 and 3 for 64 and 1500 bytes packet respectively.

Interestingly, we observed that the achievable throughput gain with smaller packet is higher than larger packet and *the improvement factor can exceed the number of channels available*. This is because in the secondary channel, only a single RTS/CTS is performed before a batch transmission of only DATA-ACK packets. The relative benefit of such batch transmission is larger for small packet, therefore giving rise to an improvement factor that can be larger than the number of channels.

The result shows that for a single flow, AMCM with the use of NW adaptation can be almost as efficient as IEEE 802.11 MAC. In addition, as the number of flows increases, the network operates in the saturated region and the aggregate throughput approaches $N \times \alpha$, where N and α are the number of orthogonal channels and per-channel saturated throughput respectively.

4.2.2. Impact of Traffic Load

In this section, we vary the network load with different packet sending rate while keeping the number of flows

[2] AMCM always means with NW adaptation

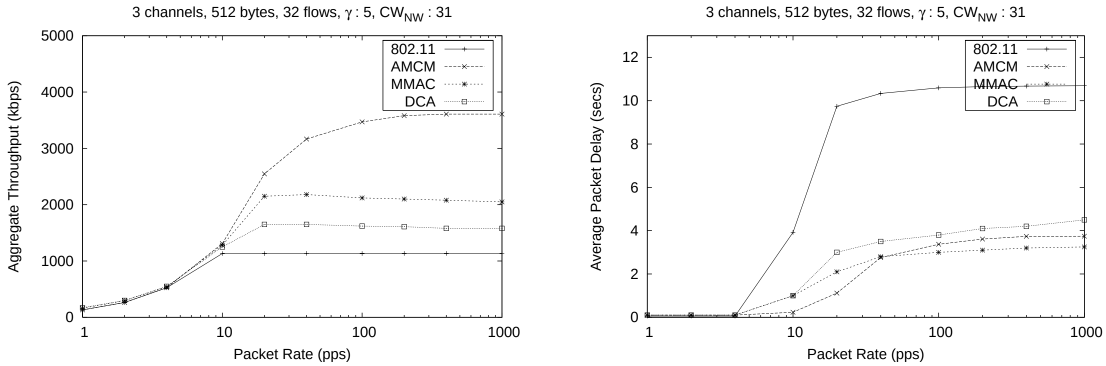
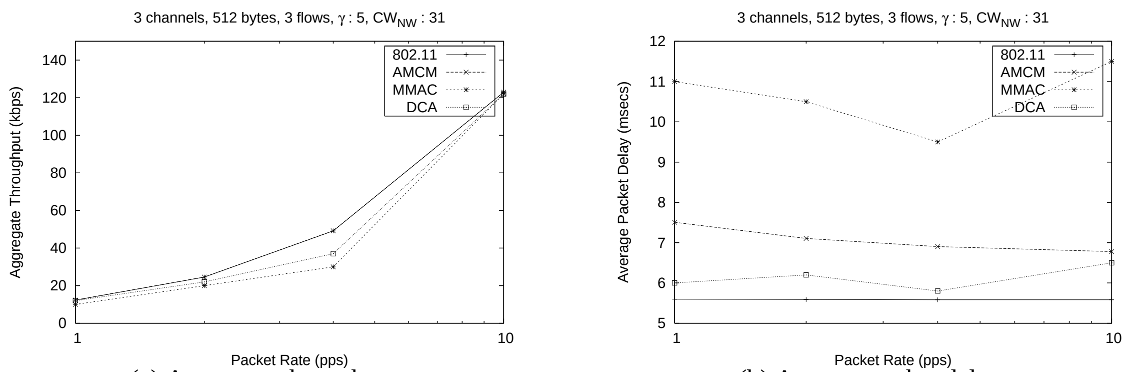


Figure 5. WLAN: Impact of traffic load on aggregate throughput and delay



(a) Aggregate throughput (b) Average packet delay

Figure 6. WLAN: Performance impact under low load

constant (32) and compare AMCM with single- and multi-channel (DCA [11] and MMAC [8]) MAC protocols.

From Figure 5, AMCM achieved similar performance as the rest when the network load is low. As the packet rate increases, AMCM outperforms both DCA and MMAC, which saturates at around 1.5Mbps and 2Mbps respectively using packet size of 512 bytes. Saturation throughput for IEEE 802.11 is only 1.2Mbps. As reported in [8], DCA begins to suffer from control-channel saturation problem with high traffic load. In contrast, AMCM does not rely on a dedicated control channel, but instead adapt NW to increase the channel utilization. At 1000pps, AMCM achieves an aggregate throughput of 3.8Mbps. Unlike DCA and MMAC, AMCM achieves better utilization of all the three channels, and also achieved almost $3\times$ throughput improvement over its single-channel counterpart.

Figure 5 also illustrates the effectiveness of AMCM in terms of packet delay. AMCM incurs much smaller end-to-end packet delay than IEEE 802.11 MAC, and is also lower than both DCA and MMAC protocols below 40pps where it is already supporting a much higher throughput.

Figure 6 illustrates the performance under low load using only 3 CBR traffic flows. We observed that at very low load, AMCM delivers identical throughput performance as IEEE 802.11 MAC but slightly higher average packet delay due to BTN message overhead (γ = 0). Nevertheless, the average delay is much lower than MMAC and slightly larger than DCA. Note that DCA requires two transceivers.

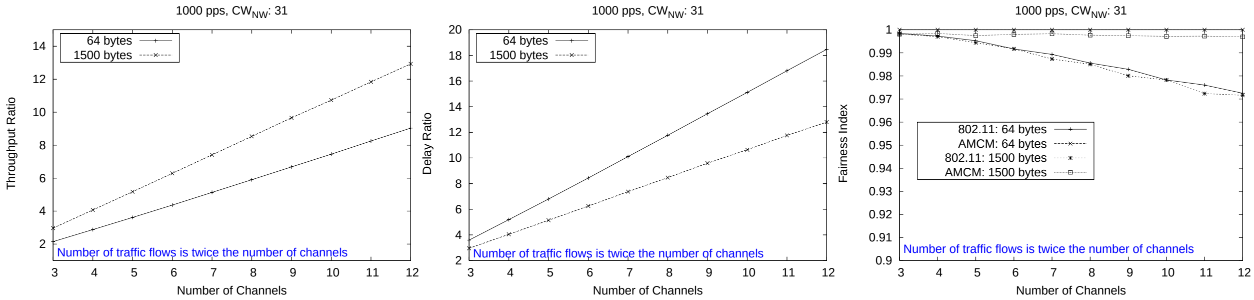
Figure 7 shows the control overhead ratio for varying traffic load. AMCM on average requires 4 control packets for a DATA packet delivery during low load. At higher load, AMCM incurs only 2 control packets for each DATA packet delivered. This is due to lower collisions rate and more useful DATA packets delivered while on the secondary channel (i.e. batch transmission). For single-channel IEEE 802.11 MAC, 3 control packets are required and it increases slightly with more flows due to collisions. Interestingly, AMCM incur slightly lower overheads with increasing flows. This is because nodes are aware of their neighboring channel availability information and therefore will not compete/negotiate for channel during NW.

The results demonstrate the ability of AMCM to mimic single-channel IEEE 802.11 at low load, while able to almost fully utilize all available channels during high load with good protocol efficiency.

4.2.3. Impact of number of channels

In this experiment, we study the performance impact with different number of channels (N). γ is initialized to N notification opportunities (NOP). In order to utilize all channels, *the number of traffic flows is set to twice the number of channels simulated*.

Our simulation result from Figure 8a shows that as the number of channels increases, AMCM is able to adapt γ dynamically to both the traffic load and the availability of secondary channels. The throughput gain over IEEE 802.11 MAC increases with the number of channels (and flows) for both small and large packet size. With 12 channels (24 flows), AMCM achieves almost a throughput ratio of 13 and 9 for small and large packet size respectively. More im-



(a) Throughput (b) Delay (c) Fairness

Figure 8. WLAN: Impact of number of channels

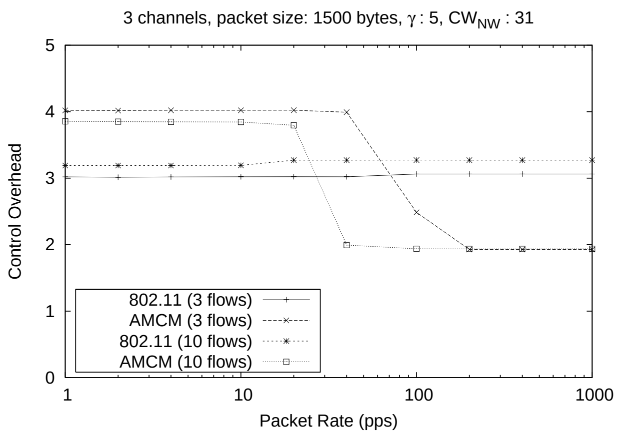


Figure 7. WLAN: Comparsion of control overhead against IEEE802.11

portantly, this observation also demonstrates that AMCM does not suffer from the control-window saturation problem as the number of channels increased. For up to 12 channels, AMCM delivers a performance gain that scales linearly with the number of channels.

In terms of packet delay, Figure 8b shows that AMCM also achieves significant reduction when compared to IEEE 802.11 MAC. With 3 channels, the reduction is 3.8 and 3 for 64 bytes and 1500 byte packets respectively. This reduction also increases with the number of channels. With 12 channels, the reduction is 18 and 13 for 1500 byte and 64 byte packets respectively.

In order to explore how well the channel capacity is shared among all the flows, the Jain's Fairness Index [14] is used. This fairness index is defined as:

$$Fairness\ Index, f(x) = \frac{(\sum_i^m x_i)^2}{m \sum_{i=1}^m x_i^2} \qquad (6)$$

where *m* is the number of flows, and x_i is the throughput for flow *i*. Ideally, perfect fairness of *f(x) = 1* is desired. In the worst case, *f(x)* = $\frac{1}{M}$, where *M* is the number of contending nodes.

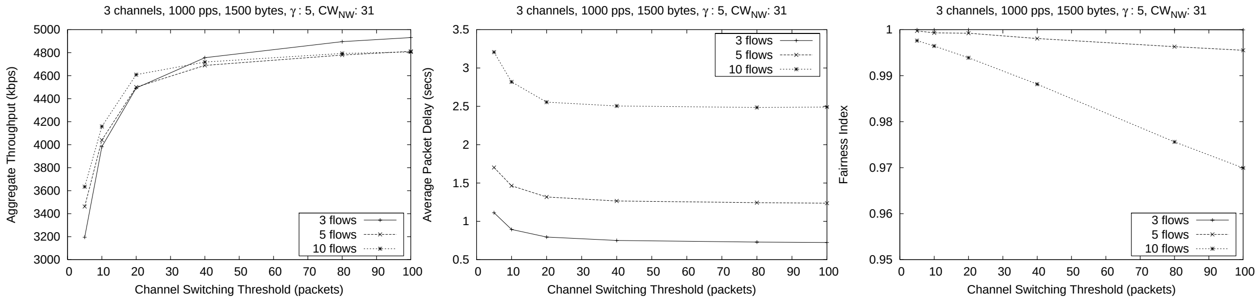
Figure 8c shows the fairness index of AMCM and IEEE 802.11 MAC. The results show that as the number of channels increase, AMCM distributes the capacity evenly among all traffic flows. For large packet size, the fairness index is lower compared to small packet size. This is due to the increase in reservation duration on the secondary channel. As the duration increases, nodes on the primary channel must wait longer in order to acquire the channel once it is free. Overall, AMCM still outperforms IEEE 802.11 MAC as the number of traffic flows increases for the two packet sizes shown.

From these observations, we conclude that AMCM can adapt very well to different number of channels and provide high spatial reuse factor with an increase of number of channels. We also noted that the packet size determines the degree of tradeoff between high throughput, lower delay and fairness.

4.2.4. Impact of Channel Switching Threshold

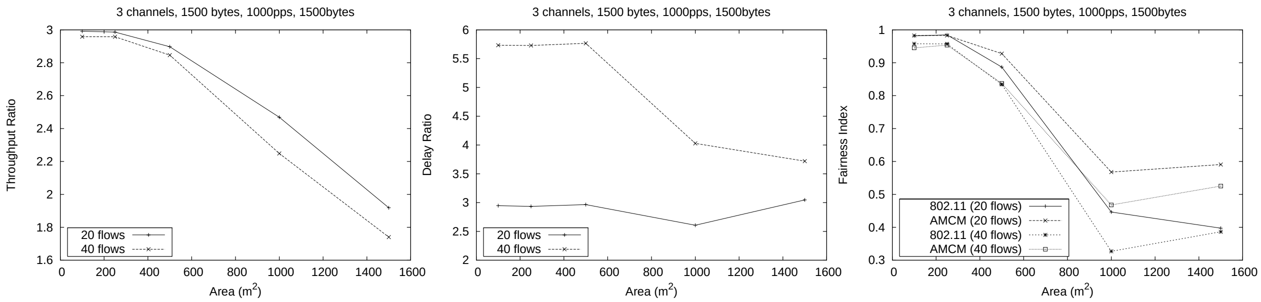
Recall that switching nodes calculate and advertise their reservation duration on the selected secondary channel during NW. This duration is calculated as described in Section 3.1.4. All simulations described above consists of traffic patterns which are disjoint (i.e unique source/destination pairs). Ideally, once a sender succeeds in capturing a secondary channel during NW, it would preferably want to take advantage of this opportunity to *flush* its packet queue for the intended receiver so as to increase the overall throughput and channel utilization. The value of CST determines the maximum period two nodes can reserve on the secondary channel. In this experiment, we study the effect of CST on both the throughput and delay for disjoint traffic pattern.

From Figure 9a, as expected, the aggregate throughput increases with the threshold value since more DATA packets can be transmitted in a single switch to the secondary channel. We observed that with smaller threshold, the aggregate throughput is higher with more flows since the overhead of NW is amortized by having more transmissions on the secondary channels. In general, the throughput performance increases rapidly with smaller threshold (≤ 20). As the threshold increases beyond 40 packets, the aggregate throughput for both 5 & 10 flows decreases since 3 channels are used for this simulation. For packet delay, as the threshold increases, average packet delay also decreases since the aggregate throughput has increased by a sufficiently large amount. An increase of CST beyond certain threshold does not increase the performance gain further and in fact can increase the average delay slightly if the load is increased further.



(a) Aggregate throughput (b) Delay (c) Fairness

Figure 9. WLAN: Impact of CST



(a) Throughput (b) Delay (c) Fairness

Figure 10. Multi-hop: Effects of Network Density

Figure 9c shows the fairness property of AMCM. As the threshold increases, the fairness index decreases. In addition, the degree of unfairness becomes more severe as the network load increases.

The results in this section shows that there is a minimum threshold in which significant performance is achieved. Operating below this threshold results in reduced channel utilization due to increased NW-related overheads. Once CST increases beyond 40-50 packets, the rate of improvement decreases and is bounded by the number of channels (3 in this case). However, larger CST can also result in more unfairness.

4.3. Multi-hop

For simulations in the multi-hop network, nodes are randomly placed in an area. In order to eliminate the effect of routing, the source and destination nodes are within communication range of each other. In addition, the source-destination pairs are unique. Therefore, a node can only be part of one flow. Even though communicating nodes are within communication range, this scenario is unlike the previous single-hop scenario since the level of interference and traffic within a local interference region varies significantly throughout the wireless network.

First, we study the performance of AMCM under different network densities. In this experiment, we randomly placed 100 nodes in a square area. We simulated using 20/40 CBR traffic flows generating 1000 packets (1500 bytes) per second. Throughout the simulation (300 secs), all nodes are configured with three channels.

Figure 10 shows the performance of AMCM and IEEE 802.11 MAC. Figure 10a shows that AMCM achieved higher spatial reuse in dense network. In fact, scenarios with areas smaller than 200m by 200m are simply the single-hop scenarios. We achieved almost 3$\times$ throughput gain over single-channel IEEE 802.11 MAC in such cases. However, as the network becomes more sparse, the gain decreases to $\approx$1.8 for an area of 1500m^2. For such sparse networks, the number of contending flows in some locations may be insufficient to exploit all three channels and hence cannot achieve sufficient improvement compared to single-channel.

Figure 10b shows the delay performance against IEEE 802.11 MAC. As the network becomes more sparse, the delay ratio decreases. In a 100m^2 network, AMCM achieves almost a reduction of 3 and 5.5 for 20 and 40 flows respectively. In general, AMCM achieved lower packet delay in both network configurations compared to IEEE 802.11 MAC. This is true since the use of multi-channel MAC also helps to alleviate both contentions and collisions, which can significantly reduce the time spent in performing backoff and retransmissions.

Figure 10c shows the Jain's fairness index of AMCM and IEEE 802.11 MAC protocol. In dense network whereby hidden terminal problem is not dominant, AMCM achieves good fairness index similarly to IEEE 802.11 MAC. As the network becomes sparse, AMCM suffers similarly to IEEE 802.11 since we use similar RTS/CTS/BTN mechanism to alleviate hidden-terminal problem. In addition, for sparse network whereby both nodes' location and traffic might not be uniformly distributed, unfairness can occurred in certain locations (*hotspots*) where channel access is higher, as compared to other less-contention locations.

Finally, as an indication of how well AMCM utilizes all available channel resources, Figure 11 shows the chan-

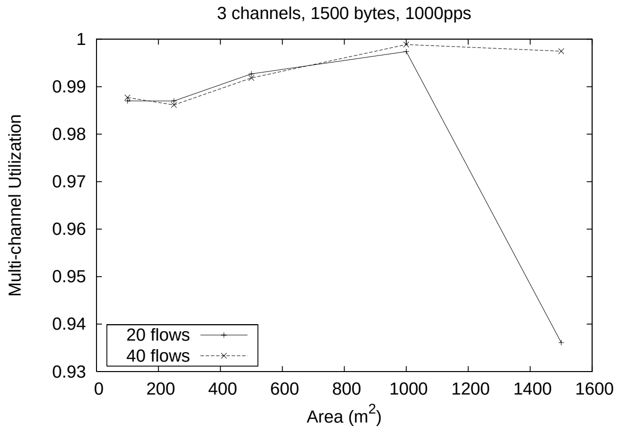


Figure 11. Multi-hop: Multi-Channel Utilization

nels utilization. When the demands for multiple channel is higher especially in dense network, we observe that AMCM utilizes all available channels efficiently. Load is distributed evenly over all channels as indicated by a fairness index close to 1. Even as the network becomes sparse and there are smaller number of flows, AMCM can still achieve high channel utilizations and spreads the load evenly among the channels. .

The results in this section demonstrate that AMCM is able to perform well in a multi-hop network scenario and delivers close to the maximum throughput possible with the utilizations for all channels are close to 1 in most cases. However, the amount of improvement over IEEE 802.11 MAC depends on the amount of spatial reuse possible. The main factor affecting the number of channels that can be utilized is the density of active flows. Such density depends on the network density and the number of active flows in the network.

5. Conclusions

In this paper, we proposed a new multi-channel CSMA/CA-based MAC protocol (AMCM) to enable multiple concurrent transmissions over several orthogonal channels. We performed extensive simulations to study the performance under both infrastructure WLAN (single-hop) and multi-hop wireless networks. We have several observations. Firstly, our traffic-adaptive window-based scheme adapts well to varying traffic load, and thus achieved high channel utilization. Secondly, given a *N-channel* wireless networks, we showed that our single transceiver solution achieved nearly $N\times$ performance gain over single-channel network. Lastly, AMCM has the ability to mimic single-channel IEEE 802.11 at low load, while utilizing all available channels during high load with good protocol efficiency.

References

[1] ANSI/IEEE. 802.11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, 2003.

[2] F. Cal, M. Conti, and E. Gregori, Dynamic Tuning of the IEEE 802.11 Protocol. ACM/IEEE Transactions on Networking, 2000.

[3] Y. Kwon,Y . Fang,an d H. Latchman, A Novel MAC Protocol with Fast Collision Resolution for Wireless LANs. In Proc. of INFOCOM, March-April 2003.

[4] M Heusse, F Rousseau, R Guillier, A Duda, Idle Sense: An Optimal Access Method for High Throughput and Fairness in Rate Diverse Wireless LANs. In Proc. of the ACM SIGCOMM Computer Communication Review, October 2004.

[5] R.R. Choudhury, X. Yang, R. Ramanathan, N.H. Vaidya, Using directional antennas for medium access control in ad hoc networks. In Proc. of the ACM MOBICOM, September 2002.

[6] Y.B. Ko, V. Shankarkumar, N.H. Vaidya, Medium Access Protocols using directional antennas in ad hoc networks. In Proc. of the IEEE INFOCOM, March 2000.

[7] A. Nasipuri, S. Ye, J. You, R. Hiromoto, A MAC protocol for mobile ad hoc networks using directional antennas. In Proc. of the IEEE WCNC, Chicago, IL, September 2000.

[8] J. So and N. H. Vaidya, A Multi-Channel MAC Protocol for Ad Hoc Wireless Networks. In Proc. of ACM Mobihoc, May 2004.

[9] Paramvir Bahl, Ranveer Chandra, and John Dunagan, SSCH: Slotted Seeded Channel Hopping for Capacity Improvement in IEEE 802.11 Ad-Hoc Wireless Networks. In Proc. of ACM Mobicom, 2004

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

[11] S.-L. Wu, C.-Y. Lin, Y.-C. Tseng and J.-P. Sheu, A New Multi-Channel MAC Protocol with On-Demand Channel Assignment for Multi-Hop Mobile Ad Hoc Networks. In Proc. of Int'l Symposium on Parallel Architectures, Algorithms and Networks (I-SPAN), 2000.

[12] A. Adya, P. Bahl, J. Padhye, A. Wolman, and L. Zhou, A multi-radio unification protocol for IEEE 802.11 wireless networks. In Proc. of First BroadNets 2004.

[13] A. Acharya, A. Misra, and S. Bansal, MACA-P : A MAC for concurrent transmissions in multi-hop wireless networks. In Proc. of the First IEEE PerCom 2003 Conference, pages 505508, Mar. 2003.

[14] R. Jain, D.M. Chiu, W.R. Hawe, A Quantitative Measure of Fairness and Discrimination for Resource Allocation Shared Computer Systems, Digital Equipment Corporation technical report TR-301, 1984.