

Starvation Mitigation Through Multi-Channel Coordination in CSMA Multi-hop Wireless Networks *

Jingpu Shi, Theodoros Salonidis, and Edward W. Knightly
Dept. of Electrical and Computer Engineering, Rice University
Houston, TX 77005

jingpu@ece.rice.edu, thsalon@ece.rice.edu, knightly@ece.rice.edu

ABSTRACT

Existing multi-channel protocols have been demonstrated to significantly increase aggregate throughput compared to single-channel protocols. However, we show that despite such improvements in aggregate throughput, existing protocols can lead to flow starvation in a multi-hop network, a phenomenon that also occurs with single-channel protocols. In this paper, we devise Asynchronous Multi-channel Coordination Protocol (AMCP), a distributed medium access protocol that not only increases aggregate throughput, but more importantly, addresses the fundamental coordination problems that lead to starvation. Based on AMCP's counter-starvation mechanisms, we analytically derive and experimentally validate an approximate lower bound on the throughput of any flow in an arbitrary topology. We also demonstrate that AMCP can deliver significantly higher per-flow throughput than both IEEE 802.11 and existing multi-channel solutions. In addition to its performance properties, AMCP is both simple in that it operates using the primitives of IEEE 802.11 DCF, and cost-effective in that it requires only a single half-duplex transceiver and no infrastructure support.

Categories and Subject Descriptors

C.2.1 [Computer-Communication Networks]: Network Architecture and Design—*Wireless communication*

General Terms

Design, Performance

Keywords

CSMA, CSMA/CA, Multiple Channels, Starvation

1. INTRODUCTION

Distributed CSMA-based random access protocols such as IEEE 802.11 DCF are well known to produce unfairness or even flow starvation when applied to multi-hop wireless networks. The main

*This research is supported by NSF ITR Grants ANI-0331620 and ANI-0325971, and by the Cisco ARTI program.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

MobiHoc'06, May 22–25, 2006, Florence, Italy.

Copyright 2006 ACM 1-59593-368-9/06/0005 ...\$5.00.

reason is that in multi-hop settings not all transmitters are within range of each other and hence may have a different view of the channel state.

Starvation can be addressed by appropriately scheduling interfering transmissions over multiple orthogonal channels. Indeed, if every node were equipped with a large number of channels and transceivers, the access problem would be eliminated and starvation would not occur. However, most commercially available wireless cards have a single transceiver and support a limited number of orthogonal channels.

Scheduled access methods [7, 16, 19, 25, 27, 29, 31] can operate under transceiver and channel constraints and can address starvation by coordinating transmissions across multiple channels (be it time slots, frequency bands, or spread spectrum codes) in an optimal manner. However, such techniques typically require global knowledge of topology and traffic requirements as well as some form of infrastructure support: either global time slot synchronization (TDMA) or pre-distribution of codes (CDMA). Such operational requirements cannot be easily supported in the distributed ad hoc network setting.

In this paper we utilize multiple channels to mitigate starvation under the constraint of a single half-duplex radio at each node and the absence of centralized knowledge or infrastructure support. Our solution, called Asynchronous Multi-channel Coordination Protocol (AMCP), uses the simple distributed access primitives of IEEE 802.11 DCF and provides analytical minimum rate guarantee for each flow in the network. Despite its simplicity, AMCP is designed to address the root cause of starvation of CSMA protocols in single-channel multi-hop wireless networks as well as the dual coordination problems that arise by the introduction of multiple channels. While previous multi-channel MAC protocols [1, 2, 18, 24, 30, 33] have been shown to increase aggregate network throughput, they do not provide mechanisms that prevent starvation in multi-hop wireless networks. We show that without proper coordination of transmissions, the aggregate throughput may increase with the number of channels, but certain flows may still receive zero throughput.

We first present that starvation in single-channel CSMA systems arises due to misaligned transmissions, which either cause a transmitter of a flow to defer through carrier sense for extended time periods, or cause collisions at the flow's receiver. Multi-channel wireless technologies have the potential to address starvation by moving the misaligned interfering transmissions to different channels. Still, achieving this goal is challenging, especially when each node can transmit or receive on only a single channel and link at a time. Although packets can be transmitted on different channels, transmissions in a multi-channel system are still not aligned. This results in the following generic multi-channel coordination problems: 1) Control packets sent on a certain channel fail to inform neighboring nodes currently communicating on a different channel and 2) Control packets intended for a certain receiver may fail be-

cause the receiver is currently on a different channel. These problems may also lead to starvation if not addressed appropriately.

In light of the above coordination problems we revisit the basic design principles of multi-channel MAC protocols. A fundamental design choice is whether to use a dedicated control channel or transmit both control and data information on all channels. AMCP utilizes a dedicated control channel to address both single-channel and multi-channel coordination problems and effectively alleviate starvation in a multi-hop wireless network. To combat the bottleneck caused by the control channel, we compute the maximum number of data channels that can be supported by the control channel as a function of the protocol parameters. This allows one to quantitatively perform appropriate sizing on the control channel capacity.

Next, we derive an approximate lower bound on the throughput of any AMCP flow in an arbitrary topology. The basic technique is to construct a hypothetical, low-throughput scenario on the control channel and to model the impact of the aggregate channel hopping pattern of the interfering flows. The lower bound depends on system parameters and the number of interfering nodes within the neighborhood of each flow. Therefore, it can be computed using only local information.

Through extensive simulations we demonstrate the properties of AMCP in both single-hop and multi-hop networks. We show that the throughput achieved by AMCP can approach the approximate lower bound in highly congested contention regions while being much higher in multi-hop scenarios. We design experiments to isolate and expose each fundamental single- and multi-channel coordination problem and show how AMCP addresses the issue and describe why existing multi-channel solutions do not. As AMCP switches channels at packet level, we evaluate via simulations the performance degradation of AMCP due to channel switching delay.

The remainder of the paper is organized as follows. In Section 2 we present the coordination problems that result in starvation in single channel systems, then point out the issues involved when multiple channels are used to address starvation. In Section 3 and 4 we present and analyze AMCP. In Section 5 we evaluate the performance of AMCP through simulations. Related work is discussed in Section 6 and Section 7 concludes.

2. MOTIVATION AND PROTOCOL DESIGN ISSUES

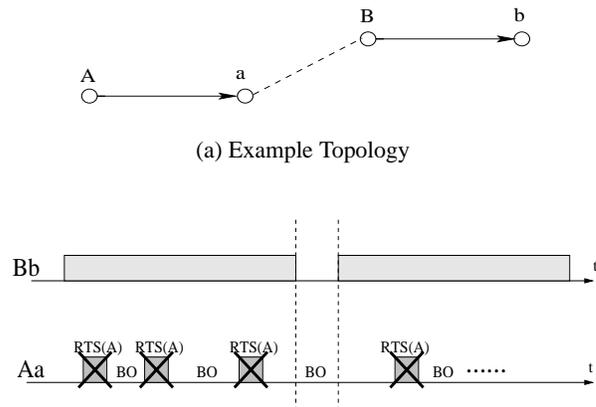
In this section, we first present Information Asymmetry (IA) and Flow-in-the-Middle (FIM), two coordination problems that have been shown to cause starvation in single-channel CSMA multi-hop wireless networks [12]. We then show that multiple channels can be used to address starvation and compare two broad classes of solutions. Then we study two generic coordination problems inherent in a multi-channel system, namely the Multi-Channel Hidden Terminal problem identified in [30] and the Missing Receiver problem which we identify in this paper. These multi-channel coordination problems manifest in both classes of solutions and may cause performance degradation if not addressed properly.

2.1 Starvation in CSMA single-channel multi-hop wireless networks

When all transmitters are within range of each other it can be shown that CSMA protocols provide fair access opportunities to all flows. Unfortunately, in a multi-hop topology where not all nodes are within range of each other, such protocols do not perform well, even if coordination enhancements such as RTS/CTS control packet exchanges [3] are used. More specifically, throughput distributions arise in which a few flows capture all bandwidth while many other flows get very low or even zero throughput. Such starvation phenomena are not merely due to having a different number of contenders for each flow, which is natural in a multi-hop topol-

ogy; rather, they are due to coordination problems when CSMA-based access is used in a multi-hop environment. Here we illustrate these coordination problems that cause starvation through two characteristic examples.

Information Asymmetry (IA). The IA problem arises when the senders of two contending flows are not within radio range and have an asymmetric view of the channel state. Fig. 1(a) is an example topology of the IA problem ([3]), where the transmitter B of flow Bb is within radio range of the receiver a of flow Aa not in range of transmitter A . If both flows are backlogged, flow Bb will receive significantly higher throughput than flow Aa . This is because the transmitter B of flow Bb knows exactly when to contend for the channel (through the control packets sent by the receiver of flow Aa). On the other hand, sender A cannot sense the activity of flow Bb and has to discover an available time slot only through random back-off. Since for efficiency purposes the ratio of data transmission interval to the idle slot size is usually large, most of these random attempts occur during the transmission of flow Bb and result in collisions at receiver a . Repeated collisions trigger timeouts at sender A , which repeats doubling its contention window. As a result, the collision probability of flow Aa is close to 1, while the collision probability of flow Bb is close to 0. Figure 1(b) shows the channel state experienced by flow Aa .



(b) Flow activities: Flow Bb does not experience collisions. The random attempts of node A to find an idle interval within the transmissions of flow Bb result in RTS failures and exponential back-off.

Figure 1: Information Asymmetry (IA) example

Flow-in-the-Middle (FIM). The FIM problem arises when the sender of a flow senses the activity of neighboring nodes that are not within range with respect to each other. This behavior is illustrated in the three-link scenario of Fig. 2. If all flows are backlogged, the middle flow Bb will receive very low throughput, while the outer flows (Aa and Cc) will receive maximum throughput. This is not due to high loss probability, but rather to the lack of transmission opportunities for the middle flow. More specifically, when one of the outer flows (say flow Aa) captures the medium, the transmitter of the middle flow Bb will sense and defer but the transmitter of the other outer flow Cc will continue contending and initiate transmission. When flow Aa ends transmission, it will contend and initiate transmission, while flow Bb now defers due to flow Cc . Fig. 2(b) shows that the misaligned concurrent transmissions of the outer flows may be sensed by the transmitter B of the middle flow for extended periods of time. The middle flow has a chance to access the medium only when *both* outer flows are in the

back-off phase (the vertical lines interval in Fig. 2(b)). Unfortunately, such occurrences become increasingly rare especially as the ratio of data transmission interval to the back-off interval increases.

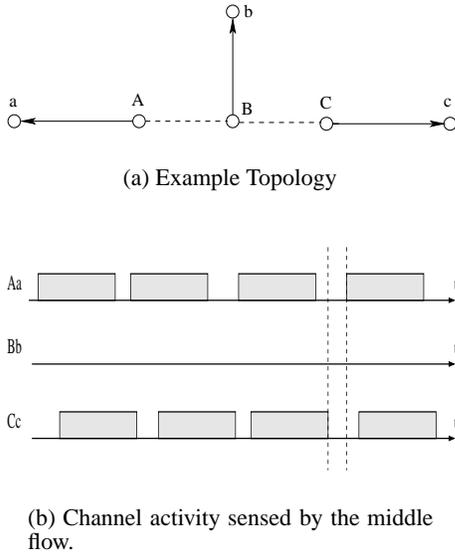


Figure 2: Flow In The Middle (FIM) example

Both IA and FIM problems are not specific to the 802.11 DCF access mechanism. They are generic coordination problems that arise due to the asymmetry of the multi-hop topology and due to the use of carrier sense. In a general topology the starving flows experience the combined effect of both IA and FIM problems and their throughput may even reach zero. For an analytical model of starvation phenomena in single-channel CSMA multi-hop networks, see [12].

For convenience, in the rest of the paper we use the term “advantaged flows” to refer to flows with geometry advantage (such as flow Bb in Fig. 1(a) and the outer flows Aa and Cc in Fig. 2(a)) and the term “disadvantaged flows” to refer to flows with geometry disadvantage (such as flow Aa in Fig. 1(a) and flow Bb in Fig. 2(a)). We also maintain the convention of using capital letter for the transmitter and lowercase letter for the receiver of each flow.

2.2 Starvation avoidance through multiple channels

In both the IA and FIM starvation scenarios, the disadvantaged flow is unable to identify an idle interval because transmissions are generally misaligned and their durations are much larger than the back-off interval.

Clearly, starvation would be eliminated if all transmissions occurred on orthogonal channels. Potential solutions can be classified into two approaches, as exemplified in Fig. 3 for the case of two flows. In the first approach, the entire transmission (including control and data transmissions) of each flow is scheduled on a different channel (Fig. 3(a)). The reason this approach can avoid starvation is straightforward: an advantaged flow will not starve a disadvantaged flow because they both transmit on different channels. In the second approach, control packets are transmitted on a separate control channel and data packets of different flows are distributed to different data channels (Fig. 3(b)). This approach also alleviates starvation: as the data packets have moved to different channels, contention occurs only on the control channel between

short control packets, whose length is comparable to the back-off interval.

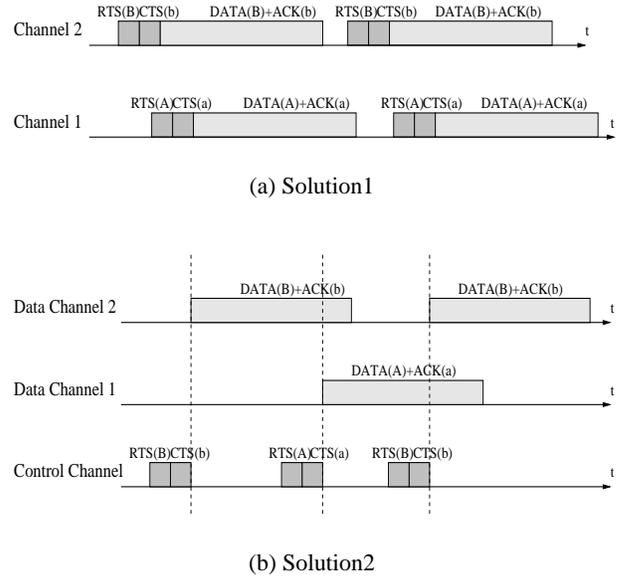


Figure 3: The use of multiple channels to address starvation.

Each approach has its advantages and disadvantages. The advantage of the first approach is that it does not require the overhead of a dedicated channel for control messages and can potentially reduce the contention between the advantaged flows and the disadvantaged flows to zero. However, it can lead to logical partition where two nodes within range are unable to communicate. This is a significant challenge, especially when a node has only one transceiver and can only transmit or listen to one channel at a time. In the second approach nodes immediately return to the control channel after finishing their data transmissions. The advantage of this approach is that nodes have a common channel (but not time) reference to coordinate their transmissions. The downside is that a dedicated control channel introduces overhead, which can be significant if its capacity is not appropriately designed.

2.3 Multi-channel coordination problems

Regardless of the solution approach, it is challenging to coordinate transmissions over different channels in an asynchronous setting where each node has a single radio transceiver. Transmissions occurring on different channels can still be misaligned. When a node communicates on a channel, it is not aware of the state on other channels. Hence, when it finishes communication it may attempt to exchange information with its neighbors while they are currently on other channels. To design an efficient protocol we must be able to accurately characterize this lack of coordination. We investigate two generic coordination problems, classified with respect to their effect on the intended actions of control packets.

In the **Multi-channel Hidden Terminal Problem**, control packets sent on a certain channel fail to inform neighboring nodes currently communicating on a different channel.

An instance of this generic problem was first identified in [30]. To illustrate this problem we use a “naive” protocol that is a straightforward extension to IEEE 802.11 DCF for a multi-channel setting: The RTS/CTS control packets are exchanged on a dedicated control channel and reserve data channels for data packets. Nodes return to control channel immediately after they finish their data transmissions.

Now we consider again the two-flow topology of the IA scenario in Fig. 1. In this example, we assume that the protocol operates with two data channels. As shown in Fig. 4, a control packet exchange of disadvantaged flow Aa may occur when the advantaged flow Bb transmits on data channel 2. Suppose Aa selects data channel 1 and initiates a transmission. When flow Aa transmits, flow Bb will return to the control channel. Since it has not heard the reservation of Flow Aa , it may select data channel 1. In this case, flow Aa will experience a collision, while the transmission of Bb succeeds. Flow Aa can be starved if there are many advantaged flows within its radio range.

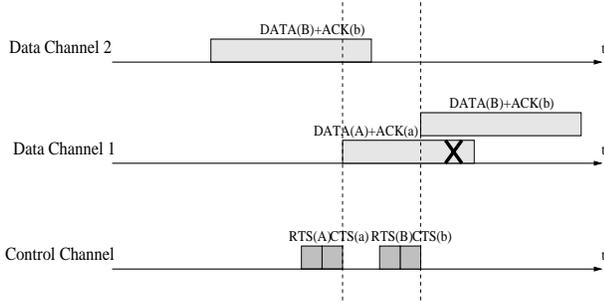


Figure 4: The Multi-Channel Hidden Terminal Problem.

Although in this example we used a naive protocol where all control messages are exchanged in a dedicated control channel, it is evident that the problem is also present when control messages are transmitted on different channels. The multi-channel hidden terminal problem limits the ability of control packets to block interfering flows. If no proper measures are taken it may result in very poor performance.

The **Missing Receiver Problem** arises when *control packets sent on a certain channel to access an intended receiver fail because this node is currently on a different channel (acting either as transmitter or receiver)*.

To illustrate the problem, we consider the simple three-node scenario of Fig. 5, where node A transmits to node B and node B transmits to node C . We first consider the naive protocol version where all control messages are transmitted on different channels. In Fig. 5, an access attempt of A for B on channel 1 will fail if B is on channel 2. Then node A will perform random back-off and retry on channel 1. Unless proper measures are taken, this problem will cause large packet delay for flow AB and decrease its throughput.

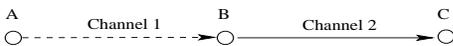


Figure 5: Missing Receiver Problem.

The problem also persists with a protocol that separates the control channel from data channels. Suppose A starts contending for B and B starts contending for C on the control channel. As long as one of them wins the contention, the other node will be able to synchronize and resume contention at the end of the data transmission. Unfortunately, synchronization is lost when the nodes count-down simultaneously. In this case, both nodes will not be able to hear each other's RTS while they transmit. Therefore the RTS from B to C succeeds, while the RTS from A to B fails. After this point, node A will try to discover node B using random back-off. This is difficult to occur since A will need to find a short interval where B returns for its own back-off on the control channel. It is more likely for B to contact A when it contends in the control channel for the next packet for C . In this case, A synchronizes with the end of transmission of B but it will already have a large back-off interval

and will not be able to compete fairly for B . Hence flow AB will starve if no proper measures are taken.

It is evident that similar inefficiencies arise in the other version of the Missing Receiver Problem, where node B acts as receiver on link BC .

Note that the Missing Receiver Problem does not exist in a single-channel system because A can carrier sense the data transmissions of B and immediately defer until the end of BC transmission.

3. ASYNCHRONOUS MULTI-CHANNEL COORDINATION PROTOCOL (AMCP)

We first illustrate the basic principles of AMCP and then present its implementation. Finally, we show how it addresses the multi-channel coordination problems.

3.1 Overview

Following the second approach of Section 2.2, AMCP uses a dedicated control channel on which nodes contend to reserve data channels by exchanging RTS/CTS packets according to 802.11 DCF. Upon successful control packet exchange, both the sender and the receiver switch to the reserved data channel, denoted by x , and transmit a data packet. After a data packet is successfully transmitted on channel x , the sender and receiver return to the control channel and set all channels as unavailable except x . They may contend for data channel x immediately or contend for other data channels after the timers of these channels expire.

The RTS/CTS control packets serve a dual purpose: first, they aid two link endpoints to negotiate on commonly available data channels; second, they inform neighboring nodes to set the overheard data channels piggy-backed in RTS/CTS as unavailable for an entire data transmission interval. However, a node overhearing an RTS/CTS will not always defer for the entire data transmission; under certain conditions, it may initiate contention after the overheard RTS/CTS.

The exact deferring rules (described in Section 3.2) implement an efficient coordination scheme where nodes stay on the control channel long enough to learn about which channels to compete, while at the same time not always waiting for the entire data packet transmission, thus increasing throughput. We proceed to describe the exact protocol operations.

3.2 Protocol Description

3.2.1 Structures and variables

We assume one control channel and N data channels, indexed from 1 to N . All channels are orthogonal with respect to each other. Each node has a single transceiver, hence it can either transmit or listen, but not both. Also it can listen to or transmit on one channel at a time. To execute AMCP, each node maintains the following structures and variables:

- A local N -entry *Channel Table*. Each table entry corresponds to a data channel and consists of a bit called *avail_bit* indicating channel availability, and a timer called *avail_timer* indicating the remaining time a channel is not available. Each time the channel becomes unavailable (*avail_bit* = 0), its timer is set to expire after a data transmission duration. When the timer expires, the corresponding channel becomes available (*avail_bit* = 1). By default, when a node joins the network all its *avail_bits* are set to zero.
- An integer *prefer* variable takes values from 0 to N . If non-zero, this variable indicates that a node prefers to compete for the data channel indexed by *prefer*. If zero it indicates no preference.

3.2.2 Reservation/transmission cycle

Initially all nodes reside on the control channel. We now describe the protocol actions that occur when node A has a packet intended to node a . We denote a neighboring node of A or a as node C .

Step 1: Channel selection. Node A selects a data channel by inspecting its channel table. Among the available data channels, the channel indexed by *prefer* is selected if *prefer* is non-zero and available. Otherwise one of the available data channels is randomly chosen. If no data channel is available, the node waits until any of the *avail_timers* expires.

Step2: Channel contention. Suppose that data channel x is selected. Node A inserts the index x to its RTS packet and contends on the control channel using the 802.11 DCF CSMA/CA mechanism. In AMCP, a control channel's NAV interval expires at the end of a RTS/CTS transmission, rather than the end of a DATA/ACK transmission as in IEEE 802.11.

Step 3: Channel negotiation. When node a receives the RTS packet, it inspects the status of channel x in its channel table. If x is available, node a replies to A with a *Confirming CTS* packet containing index x . Then, it switches to data channel x and waits for a DATA packet. If channel x is not available, node a replies to A with a *Rejecting CTS* packet containing index 0 and a list of its available data channels, and remains on the control channel.

If node A receives a *Confirming CTS*, it switches to channel x and transmits the DATA packet to a . If A receives a *Rejecting CTS*, it randomly selects a channel available in both its channel table and the channel list included in the CTS packet, then it inserts the index of this channel in a RTS packet and begins a new contention cycle on the control channel.

Step 4: Data transmission. Upon reception of the DATA packet, node a responds with an ACK on data channel x , then switches back to the control channel. Upon reception of the ACK packet, A also switches back to the control channel. The packet transmission has completed successfully.

Step 5: Setting channel availability. After A returns to the control channel it sets its *prefer* variable to x ; A also sets the *avail_bit* unavailable and starts *avail_timer* for all other data channels except x . Node a sets its *prefer* variable and *Channel Table* in the same way. Node A restarts step 1 if there is a packet in its transmission queue.

We note that errors in the transmitted control and data packets are handled with timeout mechanisms similar to 802.11. If a timeout occurs while a node resides on a data channel, the node returns to the control channel, sets its *prefer* variable to 0, sets the *avail_bit* unavailable and starts *avail_timer* for all data channels.

3.2.3 Overhearing nodes' deferral rules

Let C be a neighbor of either A or a . When C overhears an RTS packet, it first updates its channel table by setting its *avail_bit(x)* = 0 and sets *avail_timer(x)* to expire at the end of the full data packet transmission (for a duration equal to CTS + DATA + ACK). When node C hears a *Confirming CTS*, it sets its *avail_bit(x)* = 0 and starts *avail_timer(x)* in the same way. When it hears a *Rejecting CTS*, no action is needed. Note that this deferring rule is only with respect to channel x . Node C can compete for other available channels after deferring for the duration of an RTS/CTS exchange.

There is only one exception to the above deferring rules. When C wants to transmit to A and hears an RTS from A , intended to a , it will defer until the end of the entire transmission of flow Aa , and set its contention window size to the minimum value. Similarly, when C wants to transmit to a and hears a CTS from a intended to A , it will defer until the end of the entire transmission of flow Aa and set its contention window size to the minimum value. This scheme provides an opportunity for C to address the Missing Receiver Problem.

3.3 Addressing multi-channel coordination problems

To present how AMCP solves the coordination problems described in 2.3, we consider the topology in Figure 1(a) and suppose there are 2 data channels and 1 control channel.

Multi-channel Hidden Terminal Problem. Consider again the Multi-channel Hidden Terminal Problem example of Fig. 4. Recall that when flow Bb arrives on the control channel during Aa transmission on data channel 1, it does not have sufficient information about the state of channel 1 because it has not heard the RTS/CTS packet of flow Aa while transmitting its own data packet on data channel 2. If it selects channel 1 it will cause a collision to the disadvantaged flow Aa .

Under AMCP, node B sets channel 1 as unavailable and sets a timer to expire after the duration of a RTS/CTS/DATA/ACK transmission (as specified in step 5 of protocol operations). Note that channel 1 may or may not actually be available, but node B sets it to unavailable, precisely because it does not have this information. Node B will compete for channel 1 only after the timer expires—by that time any transmission on channel 1 will have completed. If any RTS/CTS for channel 1 is heard during this period, node B will defer further but will have synchronized for contention on channel 1.

However, node B does not necessarily remain idle after the channel 1 timer is set. Since its transmission on channel 2 was successful, this channel is available and B will start contending immediately for this channel (through its *prefer* variable). Fig. 6(a) shows the scenario where B succeeds in capturing channel 2. In case B fails due to another flow Cc that contended for channel 2, it will also set a timer for channel 2 and defer contention until either of the two channel timers expires.

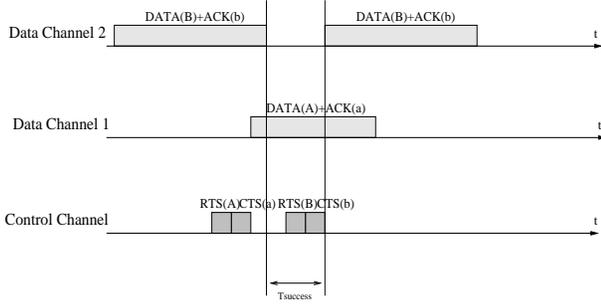
The worst-case scenario for flow Bb upon its arrival on the control channel is depicted in Fig. 6(b). Here flow Cc wins channel 2 and then flow Aa wins channel 1 before the channel 1 timer expiration. Although B has lost both contentions, it has synchronized on both data channels and will contend when either of these transmissions ends. Flow Bb has an advantage in capturing either channel in future access attempts: it can compete for both channels, counting down a single back-off counter. On the other hand, each of flows Aa and Cc will only compete for its preferred channel, according to a fresh back-off counter.

Summarizing, the simple waiting scheme of AMCP on the control channel effectively addresses the Multi-channel Hidden Terminal Problem by providing fair channel access opportunities to contending flows.

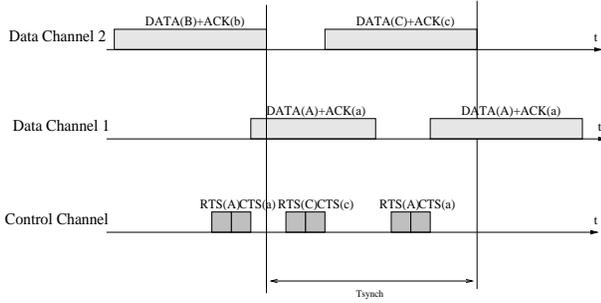
Missing Receiver Problem. Consider the scenario shown in Fig. 5, where A wants to transmit to B when B is transmitting to C on a different channel. AMCP handles the Missing Receiver Problem as follows. If A receives from B an RTS intended to C , A will defer until the end of the ongoing transmission of B and examine its back-off stage. If it is already in high back-off stage, A sets its contention window size to the minimum value. In this way, A will fairly contend for the attention of B when B is in idle state. In contrast, in the naive protocol, B will transmit many packets before A decrements its back-off counter to zero.

With AMCP, the key reason of A quickly synchronizing with B is that all control messages are transmitted on a dedicated control channel, where A can hear another RTS from B when A 's first RTS to B collides with the RTS of B to C .

In case node B is the receiver on link BC , node A performs the same actions as above when it hears the CTS of B to C . Therefore, AMCP effectively addresses both manifestations of the Missing Receiver Problem.



(a) Flow Bb contends and captures channel 2, upon return to the control channel.



(b) Flow Bb synchronizes to contend on both channel 1 and channel 2, upon return to the control channel.

Figure 6: AMCP addressing the Multi-channel Hidden Terminal problem.

4. PROTOCOL ANALYSIS

We now derive the analytical properties of AMCP. In Section 4.1 we derive the maximum number of data channels that can be supported by the control channel. In Section 4.2 we derive an approximate lower bound on the throughput achieved by any flow in an arbitrary topology.

4.1 Bottleneck Analysis

For any multi-channel protocol having a dedicated control channel, two potential bottlenecks exist: the number of data channels and the bandwidth of the control channel. Let M be the maximum number of data channels occupied by data transmissions when the control channel is saturated by control message exchanges. Let T_d be the duration of a successful data transmission (including DATA and ACK), T_r be the duration of an RTS packet, and T_c be the duration of an CTS packet, all expressed as time slots.

We observe that in steady state when the control channel is saturated, there are always M on-going transmissions on M data channels. Furthermore, within the time period of $T_d + T_r + T_c$, exactly M flows return to the control channel. Hence in steady state, M flows should successfully exchange control packets and switch to M respective data channels. Since the control channel is saturated, there is no idle interval between two successive RTS/CTS exchanges, if we neglect small overhead intervals, such as SIFS and DIFS. Thus M is given by,

$$M = \lfloor \frac{T_d + T_r + T_c}{T_r + T_c} \rfloor. \quad (1)$$

From Equation (1), we observe that M increases when the data transmission time T_d increases. For example, the control channel can drive more data channels if we reserve a data channel for multiple data packets. Also note that M derived above is for a single contention region. In a multi-hop network, M can be much larger because the control channel is spatially reused.

4.2 Lower Bound Analysis

In this section, we compute an approximate lower bound of per-flow throughput achieved by our protocol in an arbitrary multi-hop wireless network. We first construct a hypothetical, low-throughput scenario for a tagged flow, then compute its collision probability p by modeling the process by which control packets of other nodes arrive on the control channel as a Poisson process. We then apply the analytical model proposed in [13] to compute the throughput of the tagged flow, which serves as an approximate lower bound on the throughput achieved by any flows in an arbitrary topology.

Construction of the hypothetical scenario. We consider a tagged flow Aa that has N neighboring nodes in a network employing AMCP. We construct the scenario where flow Aa achieves very low throughput given N neighbors as the case that all of its N neighbors are backlogged and always transmit to receivers that are not in range of Aa using the minimum back-off window. We also assume that these N nodes are transmitting independently, in the sense that they do not sense and hence coordinate with each other's transmission. Furthermore, they are placed such that they are advantaged with respect to flow Aa . More specifically, we consider an IA scenario where these N nodes are within range of receiver a and out of range of transmitter A . In this scenario, most control packets of flow Aa will collide, thus forcing flow Aa to double its contention window. Notice that transmitter A is not able to sense the activity of the interfering flows. This scenario is hypothetical and only used to derive an approximate lower bound of the throughput of flow Aa given N neighboring nodes in its contention region.

Since in this scenario the interfering nodes transmit independently, their control packets arrive on the control channel independently. Consequently, we assume that the aggregate process formed by the control packet arrivals of the N interfering nodes is Poisson. While this process is not strictly Poisson, we validate the bound via simulations below.

Computation of the conditional packet loss probability. To compute the throughput of flow Aa in the hypothetical, low-throughput scenario, we first need to compute the collision probability p when node A attempts to transmit an RTS packet to a . Similar to [4], we refer to p as the conditional collision probability.

Let $X(t)$ be the Poisson process that represents the number of successful control packet arrivals of the N interfering nodes, given a starting point in time. Let α be the arrival rate of control packets and T be the arrival interval. Note that α is a deterministic value and T is a random variable.

We assume nodes can always find a data channel to transmit a data packet upon successful RTS/CTS exchange. The arrival rate α of $X(t)$ is given by:

$$\alpha = \frac{N}{T_d + T_r + T_c}. \quad (2)$$

Since $X(t)$ is a Poisson process, any interval T between two successive control packet exchanges of the interfering flows is exponentially distributed with the following CDF,

$$F_T(t) = P(T \leq t) = 1 - e^{-\alpha t}. \quad (3)$$

The RTS/CTS exchange between A and a will fail if it cannot fit within an idle gap $T - (T_r + T_c)$ between two successive control packet exchanges. This corresponds to the event $T - (T_r + T_c) < T_r$ (or $T < 2T_r + T_c$), which occurs with probability $p = F_T(2T_r + T_c)$. Combining with Equations (2) and (3), we de-

rive the final expression for the conditional packet loss probability p :

$$p = 1 - e^{-(2T_r+T_c)\frac{N}{T_d+T_r+T_c}}. \quad (4)$$

Throughput computation. We compute the throughput of the tagged flow Aa using a general model for backlogged flows sharing an 802.11 multi-hop network introduced in [13]. In that model, the channel view of each node comprises of a sequence of time intervals that correspond to 4 different states: (i) idle channel; (ii) channel occupied by successful transmission of the tagged station; (iii) channel occupied by a collision of the station; (iv) busy channel due to activity of other stations, detected by means of either physical or virtual carrier sensing (the NAV). The time intervals during which the station remains in each of the four states above are denoted by σ , T_s , T_c , and T_b , respectively.

According to the model in [13], the throughput of the tagged flow Aa is given by:

$$T_P = \frac{\tau(1-p)}{\tau(1-p)\bar{T}_s + \tau p\bar{T}_c + (1-\tau)(1-b)\sigma + (1-\tau)b\bar{T}_b}, \quad (5)$$

where τ is the probability that the node attempts to send a packet after an idle slot, b is the probability that the channel becomes busy after an idle slot due to activity of other nodes and p is the conditional packet loss probability.

The probability τ is a deterministic function of p and is given by [20]:

$$\tau = \frac{2q(1-p^{m+1})}{q(1-p^{m+1}) + W_0[1-p-p(2p)^{m'}(1+p^{m-m'}q)]}, \quad (6)$$

where $q = 1 - 2p$, W_0 is the minimum window size, m is the *maximum retry limit*, and m' is the backoff stage at which the window size reaches its maximum value. The average durations \bar{T}_s and \bar{T}_c are fixed and can be found in [4].

In this hypothetical scenario, the transmitter node A does not defer its transmission due to the activity of other nodes. Setting $b = 0$ in Equation (5) yields:

$$T_P = \frac{\tau(1-p)}{\tau(1-p)\bar{T}_s + \tau p\bar{T}_c + (1-\tau)\sigma + (1-\tau)}, \quad (7)$$

Using Equations (4) and (6), in Equation (7), we can now compute the throughput of the tagged flow Aa in the hypothetical scenario which serves as a lower bound approximation on the throughput achieved by any flow in an arbitrary topology as a function of number of interfering flows and system parameters.

Lower bound validation. We now validate the approximate lower bound with simulations obtained with *ns*. Both RTS/CTS packets and data packets are transmitted at 2 Mbps. We vary the number of flows N and place them in a $700m \times 700m$ area such that they belong to the same contention region. This means that only one flow can transmit successfully at a time, however it is not necessary that all transmitters or receivers are within range. For each N , we generate 10 data points each corresponding to the minimum rate achieved by a different contention region. Fig. 7 shows the minimum rates as data points and the lower bound as the analytical curve, as computed by our model.

We observe that in general the minimum rates are greater than the lower bound while in several cases the bound is tight.

5. PERFORMANCE EVALUATION

We evaluate AMCP in both single-hop and multi-hop topologies using the *ns-2* simulator with CMU wireless extensions. Unless otherwise specified we use the MAC parameters of Table 1. According to these parameters, the maximum rate achieved by a backlogged flow in isolation is 184 pkt/s. The simulator physical

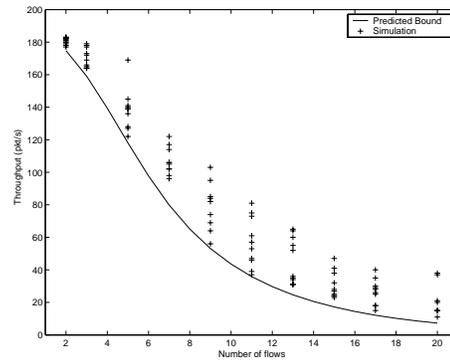


Figure 7: Comparison of lower bound to minimum throughput achieved in an arbitrary contention region as a function of the number of interfering flows.

layer parameters have been set so that the transmission range of each node is approximately 250m.

SIFS	10 μ s
DIFS	50 μ s
EIFS	364 μ s
σ	20 μ s
BasicRate	2 Mbps
DataRate	2 Mbps
PLCP length	192 bits @ 1 Mbps
MAC header (RTS,CTS,ACK,DATA)	(20,14,14,28) bytes @ BasicRate
Packet size	1000 bytes
(CW_{min} , CW_{max})	(31,1023)
Retry Limit (Short,Long)	(7,4)
Channel switching delay	224 μ s
MMAC ATIM window	20ms
MMAC Beacon interval	100ms

Table 1: MAC layer parameters

We begin with experiments on single-hop topologies to study the main protocol properties and illustrate the interplay between various parameters—number of channels, traffic load, control channel capacity, number of nodes, channel switching delay—that affect performance. Performance is measured in terms of aggregate throughput gain with respect to IEEE 802.11 DCF using a single channel.

We then move to multi-hop topologies, where we demonstrate the properties of AMCP: starvation mitigation, increase of aggregate utilization and addressing the fundamental coordination problems of both single channel and multi-channel systems, as elaborated in section 2. We also compare AMCP with MMAC, a single-radio, multi-channel protocol proposed in [30]. MMAC uses a globally synchronized control/data periodic frame (termed beacon interval). During the control subframe (termed ATIM window) flows contend on a default channel to reserve channels (including the default channel) for the data subframe. The flows that succeed in reserving a channel during the ATIM window contend during the data subframe using RTS/CTS 802.11 access mechanism. Our experiments use the same MMAC parameters as [30] (Table 1).

5.1 Single-hop topologies

In this series of experiments all nodes are within range of each other and are equally divided in a transmitter and receiver set. This yields a set of single-hop disjoint flows with distinct transmitter-receiver pairs. The case where a node is both sender and receiver is considered in the multi-hop experiments.

Effect of number of channels. Fig. 8 depicts the aggregate throughput achieved by AMCP as a function of the total number of channels for 15 backlogged flows (30 nodes). The capacity of

the control channel and each data channel is 2 Mbps. The case of AMCP with 2 channels is equivalent to single-channel 802.11, which provides the reference line in Fig. 8. The aggregate throughput increases linearly until 7 channels. After that point, it increases with a slower rate with additional channels; at 8 channels it reaches the limit of 1100 pkt/s where the control channel is saturated. This behavior agrees with our bottleneck analysis: for the parameters in this experiment, Equation (1) predicts that the control channel can drive up to 8 data channels.

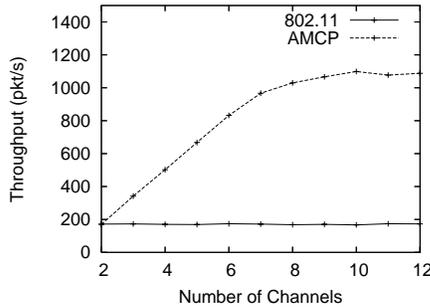


Figure 8: Throughput as a function of number of channels.

Effect of traffic load. We evaluate the performance of AMCP under non-backlogged conditions. Fig. 9 depicts the aggregate throughput of AMCP and IEEE 802.11 in a 15-flow topology as the input rate of each flow increases, when a total of 4 channels are used. Until 10 packets/s, the load is too low to exploit the additional data channels and AMCP yields similar performance to 802.11. After that point, channelization becomes effective and AMCP reaches an aggregate throughput gain equal to the number of data channels.

We note that existing multi-channel MAC protocols can achieve similar or slightly higher aggregate throughput than AMCP. For example, for 4 channels and under heavy load, DCA [33] also achieves three times the aggregate throughput of 802.11, similar to AMCP. This is because both AMCP and DCA dedicate a separate channel for control traffic. On the other hand, MMAC transmits control and data packets over 4 channels and achieves an additional gain of 20%-30%. However, DCA requires two radio transceivers per node and MMAC requires global synchronization. AMCP uses a single transceiver and no global synchronization.

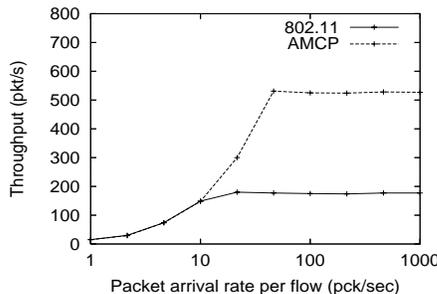


Figure 9: Aggregate throughput when arrival rate varies.

Effect of channel switching delay. Since AMCP switches channels at the packet level, channel switching delay due to hardware limitations can be a source of overhead. According to the IEEE 802.11 specification [14] this parameter can reach $224\mu s$. Fig. 10 shows a graceful decrease of aggregate throughput as channel switching delay increases from 0 to 5ms. At $224\mu s$, the throughput decrease is very small. This can be explained by the fact that $224\mu s$

is small compared to the duration of a data transmission. After 3ms, throughput goes below the single-channel maximum throughput of 184 pkt/s. For hardware with such high channel switching delays, the overhead can be addressed by reserving a channel for multiple data packets. Such functionality is easy to incorporate in the AMCP channel reservation mechanism.

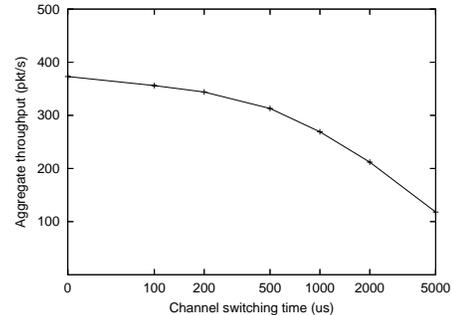


Figure 10: Effect of channel switching delay.

5.2 Multi-hop topologies

In this series of experiments we compare the performance of AMCP, MMAC, and single-channel 802.11 in static and mobile multi-hop topologies using both single-hop and multi-hop flows. We also consider specific scenarios that isolate inefficiencies that arise in the design of multi-channel protocols, namely the random channel selection problem due to collisions of control packets and the head-of-line (HOL) problem due to lack of packets to fill a channel reservation window.

5.2.1 Single-hop flows

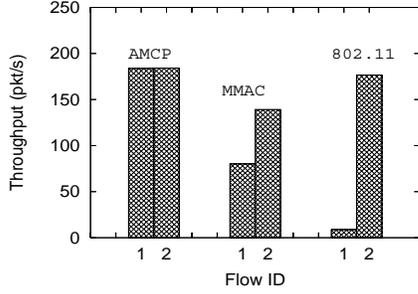
Single-channel starvation scenarios. We first investigate the ability of AMCP and MMAC to address the IA and FIM coordination problems (Fig. 1 and Fig. 2, respectively). These scenarios can easily be addressed by random channel selection if a large number of channels are available. Here we consider the case when a total of three channels are available.

We observe from Fig. 11(a) that AMCP provides equal and maximum throughput to each flow, despite that, topologically, flow *Bb* has more information about the channel. Furthermore, the simulation shows that the two flows persist transmitting on different channels. This is a desired property and shows that AMCP successfully separates the two flows and reduces their interaction.

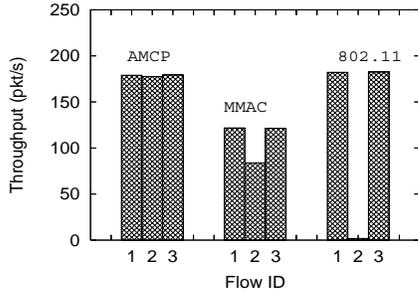
Under MMAC, flow *Bb* achieves 80% the maximum throughput of 802.11 and AMCP. This is the maximum throughput allowed by MMAC since the ATIM window is 20% of the beacon period. However, the key observation is that the disadvantaged flow *Aa* receives only $2/3$ of the maximum MMAC throughput. This is because the IA problem still exists in both the control subframe and the data subframe: the ATIM packet size is comparable to the back-off window size;¹ since its control packets collide, the transmitter of flow *Aa* is not informed about channel reservations in its neighborhood and is forced to perform random channel selection. *Aa* may choose the same data channel as *Bb* and, consequently, its data packets may be destroyed due to the IA problem.

Similarly, in Fig. 11(b) AMCP provides equal and maximum throughput to all flows. As in the IA scenario, the flows quickly coordinate and keep transmitting on the right channels: flow *Aa* and flow *Cc* on one data channel and the middle flow *Bb* on the other. In contrast, MMAC does not equalize the throughputs but is again

¹To allow more ATIM control packet exchanges in the 20 ms ATIM window, the back-off window size can not be set too small.



(a) IA scenario (Fig. 1(a)). 1 and 2 denote flow Aa and Bb , respectively.



(b) FIM scenario (Fig. 2(a)). 1, 2 and 3 denote flow Aa , Bb and Cc , respectively.

Figure 11: AMCP performance in the basic single-channel starvation scenarios.

subject to random channel selection: in this case, the transmitter of the middle flow is not able to decode the colliding ATIM control packets of the outer flows.

We now explain how AMCP addresses the IA and FIM problems. In the IA scenario, flow Bb does not experience collisions and will persist transmitting on one of the two data channels (e.g. channel 1). The receiver of flow Aa will be informed about this decision through the control packets of flow Bb . The transmitter A of flow Aa starts without any knowledge of which channel to use. Since contention in the control channel has been reduced by the removal of data packets, it is easier for A to access the receiver. In case it picked channel 1, A will be informed by the receiver and will compete and acquire channel 2 in its next access attempt. After that point both flows will continue transmitting on different channels.

In the FIM scenario flows do not experience collisions and therefore prefer to transmit on the same channel. Since the outer flows are not within range there may be an undesirable situation where they have preference for channels 1 and 2, respectively. In this case the middle flow Bb is blocked but only temporarily, until its channel timers expire. It will then contend on the control channel for any of the two data channels. When it acquires any of the two channels (e.g. channel 1) the outer flows are informed and will compete for channel 2. From this point on, since the flows do not experience collisions they will continue transmitting on orthogonal channels.

Arbitrary topology / single-hop flows. We now consider an arbitrary topology of 100 nodes placed in a $2000m \times 2000m$ area. The nodes are arbitrarily divided in 50 disjoint single-hop flows. Fig. 12 depicts the per-flow throughput under AMCP, MMAC,

802.11 as well as the AMCP lower bound, all with respect to the AMCP throughput sorted in decreasing order.

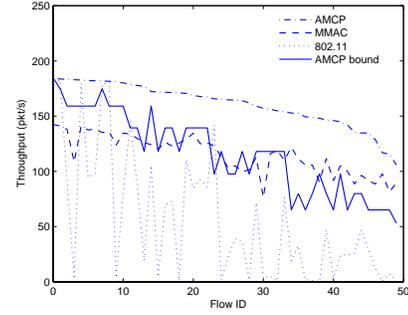


Figure 12: Per-flow throughput in an arbitrary topology of single-hop backlogged flows using 12 channels.

As expected, both AMCP and MMAC achieve higher aggregate throughput than 802.11. Furthermore, 802.11 clearly starves since 16 out of 50 flows receive close to zero throughput.

AMCP achieves higher per-flow throughput than MMAC and 802.11. Under AMCP, all flows receive above 105 pkt/s and each flow receives higher throughput than its predicted lower bound. Under MMAC all flows receive throughput above 75 pkt/s, yet always lower than AMCP; furthermore, 27 out of 50 flows receive throughput below the corresponding AMCP lower bound. Part of this inefficiency is due to the 20% ATIM window overhead; however, the flows with much lower throughput indicate that the random channel selection problem can be a source of inefficiency even if several channels (12 in this case) are available in the system.

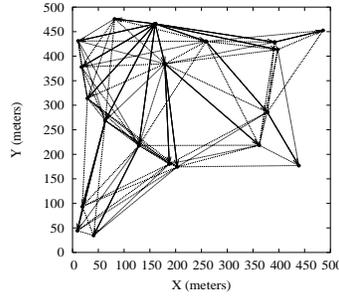
5.2.2 Multi-hop flows

Next, we move to more sophisticated scenarios involving multi-hop flows. Multi-hop flows induce non-disjoint single-hop flows which include the missing receiver problem and the head-of-line (HOL) problem in addition to the problems we have experimented so far. We first consider a scenario that isolates and illustrates these two additional problems. Finally, we consider an arbitrary scenario where all the problems are present and also evaluate the effect of mobility.

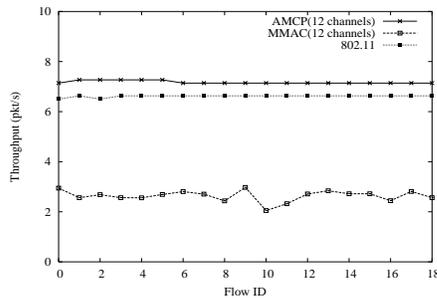
Download scenario. In the static 20-node topology of Fig. 13(a), a designated gateway node sends traffic to all other nodes through a tree structure. In this download scenario multiple channels are of little help because the bottleneck is the radio constraint at the root node. The maximum per-flow fair rate is $184 / 19 = 9.68$ pkt/s.

The per-flow throughputs under backlogged conditions are shown in Fig. 13(b). Two key observations are in place. First, AMCP delivers close to maximum per-flow throughput in a scenario where the missing receiver problem is strongly present.

Second, MMAC delivers substantially lower throughput than both AMCP and 802.11. This is not due to the missing receiver problem because MMAC uses synchronized contention. It is also not due to the random channel selection problem because the number of channels is not the bottleneck in this scenario. The problem arises because each node intends packets to multiple outgoing neighbors. During the 20ms control subframe, each node contends for the link corresponding to the HOL packet in its queue. Upon success, for the next 80ms-data subframe it will contend and transmit in the reserved channel only for this link. Hence, the data subframe can be fully utilized only if a sufficiently high number of packets of this link immediately follow the HOL packet. Unfortunately this is not likely to happen if this node intends packets to multiple neighbors and is the source of inefficiency in this scenario.



(a) Topology: Gateway downloads to all other nodes through a tree structure. Arrows denote direction of transmissions and dotted edges denote interference.



(b) Per-flow throughputs. AMCP and MMAC use 12 channels.

Figure 13: Download scenario.

There appears to be no easy solution to the HOL problem. On one hand a node could be allowed to reserve a channel for multiple links during the data subframe. This would require both significant changes to the MAC protocol as well as sophisticated queue management that would increase protocol complexity. On the other hand, the data subframe can be reduced to fit packet transmissions of a single link. However this increases the overhead due to the control subframe. Optimal sizing of the global control/data subframe is hard to perform without a-priori knowledge of traffic requirements. In addition, no sizing would suit all nodes in the network.

The HOL problem is not specific to MMAC. It exists in any multi-channel protocol that attempts to reserve a channel for several packet transmissions (e.g. SSCH [2]). If not addressed properly, it can produce substantial overhead that counter-balances the gain due to multiple channels. On the other hand, the HOL problem is not present in AMCP because contention occurs on a per-packet basis.

Multi-hop flows and mobility. To study mobility and the joint effects of the above factors, we consider a mobile scenario of 50 nodes in a $1000m \times 1000m$ area and form 10 multi-hop flows with arbitrary source-destination pairs. We use the random waypoint mobility model where nodes move at 1 m/s.

To test MAC protocol performance we need to operate at relatively high loads. Under such conditions, a dynamic MANET routing protocol can cause frequent route changes due to lost routing packets, which in turn can have a dominating degrading effect in overall performance. To decouple the effect of routing, we pre-compute shortest path routes based on the initial topology and keep

the routes fixed during each run. We then consider only the experiments where no route breakages occurred. In this way, we can test how the MAC protocols react to mobility viewed as changes of the network contention regions.

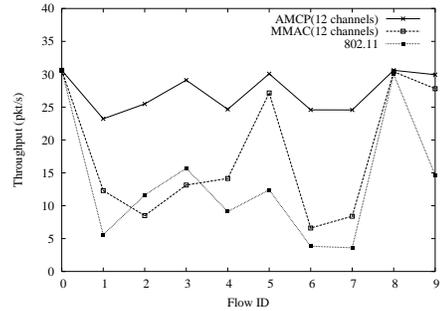


Figure 14: Per-flow throughput in an arbitrary topology of multi-hop flows using 12 channels.

Fig. 14 shows the achieved throughputs under a per-flow UDP load of 30 pkt/s. Each data point is the average of 10 mobility scenarios. AMCP appears robust in terms of delivery ratio since each flow achieves throughput close to 30 pkt/s. In contrast, several flows receive much lower throughput under MMAC and 802.11. Overall, MMAC outperforms 802.11. However, in 7 out of the 10 flows it receives substantially lower throughput than AMCP and flows 6 and 7 receive very low throughput similar to 802.11. This inefficiency is due to the superposition of ATIM window overhead, the random channel selection problem and the HOL problem.

6. RELATED WORK

There is an extensive body of work on distributed MAC protocols for multi-hop wireless networks each operating under different environments and assumptions. In this section we review existing MAC protocols in view of the starvation problem by broadly classifying them based on 1) number of channels used (single-channel/multi-channel) and 2) the access method they use (contention-based access or scheduled access).

6.1 Single channel protocols

Distributed CSMA-based MAC protocols that provide fair access by enhancing coordination of transmissions over a single channel have been proposed in [10, 11, 15, 17, 22, 23, 26]. Although a fair access protocol may address starvation it may not always be able to meet a desired minimum rate posed by the system. Topology-transparent protocols are an alternative to CSMA-based access for provision of minimum rate guarantees [5]. Each node uses a different transmit-listen periodic schedule that has been computed based on projected number of neighbors (a system parameter). By construction, the schedule of each node possesses an interesting mathematical property: at any time, at least one transmit slot will be received conflict-free by all neighbors of this node. An operational problem is that a mechanism is needed to distribute these schedules to the nodes. Also the schedules are static in the sense that nodes cannot modify their transmission decisions without compromising the minimum rate guarantee.

The above protocols operate over a single channel. When multiple channels are supported by the system and need to be exploited for both fairness and aggregate throughput increase, the problem of efficiently coordinating transmissions, especially under transceiver limitations, becomes much harder.

6.2 Multi-channel protocols

Scheduled access. Distributed scheduled access (TDMA) protocols [6, 8, 9, 21, 34, 35]. can operate with a single transceiver, support both single or multiple channels and provide fair or minimum rate guarantees by establishing and maintaining conflict-free transmission schedules.

Each node typically uses a slotted periodic frame synchronized to a global clock and split in a control and data subframe. Control messages are exchanged during the control subframe and reserve conflict-free slots in the data subframe for data packets. There are several concerns with the control/data frame approach. First, global slot synchronization is not easy to support in a distributed ad hoc network setting. Second, due to network logical partitions in a mobile setting a mechanism is needed to help nodes maintain consistent views of the position of the control portion [32]. These problems can be addressed by asynchronous scheduled access techniques that use local slot synchronization on each link and a coordination mechanism that allows transmitting control or data packets at any slot of the TDMA schedule [28]. However, to maintain rate guarantees additional support (such as directional antennas or a channel distribution mechanism) is required.

Compared to distributed scheduled access protocols, AMCP is a lightweight solution with the more modest goal of maintaining a certain minimum level of performance.

Contention-based access. Distributed contention-based CSMA multi-channel protocols have been proposed in [1, 18, 24, 30, 33]. The works in [24] and [18] assume nodes can receive packets on all channels simultaneously. Current hardware does not support listening on an arbitrary number of channels. The DCA protocol [33] uses a separate transceiver for the control channel. This is an expensive solution considering that control traffic is much lower than data traffic. The additional transceiver could instead be used to double network data throughput. Indeed, MUP [1] uses two transceivers each assigned to a different channel and executing a separate instance of IEEE 802.11 DCF; Packets are scheduled on the transceiver (and channel) that experience the least contention. It is evident that MUP would experience starvation within each channel much like a single-radio / single-channel IEEE 802.11 DCF system.

MMAC [30] uses a single transceiver and resolves the access problem using a synchronized control/data periodic frame. Since it uses globally synchronized contention, MMAC avoids the Multi-channel Hidden Terminal and Missing Receiver problems. It is also able to use more channels than solutions that use a dedicated control channel like AMCP. However, it inherits all the problems of scheduled access with a synchronized control/data frame structure. Furthermore, as demonstrated in Section 5, its performance can be severely degraded due to the random channel selection and HOL problems.

SSCH [2] is a single-transceiver, multi-channel protocol. Each node hops between channels using a 13-hop pseudo-random sequence. Within a channel hop duration a node uses IEEE 802.11 DCF to transmit data or control packets (which advertise its channel hopping schedule) to its neighbors. The channel hopping sequences have been designed such that any two nodes will overlap in at least one of the 13 hops. Each node uses the channel hopping schedules of its neighbors to transmit to them, by tuning the corresponding hops in its own hopping schedule. SSCH follows the first general solution approach we presented in Section 2. It avoids the control channel bottleneck by distributing both control and data packets to different channels. However, since it reserves a channel for multiple data transmissions on the same link it is subject to the HOL problem. Furthermore, since every node may decide to change its hopping schedule to transmit to others, the Missing Receiver problem can be extremely severe. In the three-node example of Fig. 5, even if node A knows the hopping schedule of B , its access attempt

on a certain hop may fail when B has tuned to the hopping schedule of C . This will result in several RTS failures of A during the 10ms duration of this hop. This problem becomes more severe when B has several outgoing links. In AMCP, node B will coordinate with node A through the control channel, irrespective of the number of its outgoing links.

Regardless of assumptions on hardware or infrastructure support, all protocols in [1, 2, 18, 24, 30, 33] have focused on increasing aggregate network throughput and do not provide any form of analytical per-flow throughput guarantees.

7. CONCLUSION

We have presented AMCP, a distributed medium access protocol that utilizes multiple channels to address starvation in a multi-hop wireless network. AMCP is simple, yet effectively addresses the root cause of starvation and prevents further coordination problems inherent in a multi-channel system. AMCP resolves the above problems by asynchronously coordinating transmissions over a dedicated control channel.

Using a simple analytical model we have shown that AMCP provides an approximate lower throughput bound for a flow within any contention region of an arbitrary topology. Our experiments for arbitrary topologies have shown that AMCP utilizes multiple channels to yield much higher per-flow throughput than the lower bound and a significant aggregate throughput gain with respect to a single-channel system. Its benefits are achieved using minimal amount of resources: only a single half-duplex radio transceiver and no infrastructure support such as global time synchronization or channel pre-distribution mechanisms.

ACKNOWLEDGMENTS

We thank our shepherd, Prasant Mohapatra, for his guidance and helpful feedback. We also thank Omer Gurewitz for his helpful discussions.

8. REFERENCES

- [1] A. Adya, P. Bahl, J. Padhye, A. Wolman, and L. Zhou. A Multi-Radio Unification Protocol for IEEE 802.11 Wireless Networks. In *Proc. IEEE International Conference on Broadband Networks (Broadnets) 2004*, San Jose, CA, USA, October 2004.
- [2] P. Bahl, R. Chandra, and J. Dunagan. SSCH: slotted seeded channel hopping for capacity improvement in IEEE 802.11 ad-hoc wireless networks. In *Proc. ACM MobiCom*, New York, NY, USA, September 2004.
- [3] V. Bharghavan, A. Demers, S. Shenker, and L. Zhang. MACAW: A Media Access Protocol for Wireless LANs. In *Proc. ACM SIGCOMM*, London, UK, 1994.
- [4] G. Bianchi. Performance analysis of the IEEE 802.11 distributed coordination function. *IEEE Journal on Selected Areas in Communications*, 18(3):535–547, March 2000.
- [5] I. Chlamtac, A. Farago, and H. Zhang. Time-spread multiple-access (TSMA) protocols for multihop mobile radio networks. *IEEE Transactions on Networking*, 6(5):804–812, December 1997.
- [6] I. Chlamtac and A. Lerner. Fair Algorithms for Maximal Link Activation in Multihop Radio Networks. *IEEE Transactions on Communications*, 35, July 1987.
- [7] T.A. ElBatt and A. Ephremides. Joint Scheduling and Power Control for Wireless Ad-hoc Networks. In *Proc. IEEE INFOCOM*, New York, NY, USA, June 2002.
- [8] A. Ephremides and T.V. Truong. Scheduling Broadcasts in Multihop Radio Networks. *IEEE Transactions on Communications*, 38(4), April 1990.

- [9] A. Ephremides, J.E. Wieselthier, and D.J. Baker. A design concept for reliable mobile radio networks with frequency hopping signaling. *Proc. of IEEE*, 75(1):56–73, 1987.
- [10] J.J. Garcia-Luna-Aceves and A. Tzamaloukas. Floor Acquisition Multiple Access (FAMA) in Single-Channel Wireless Networks. *ACM Mobile Networks and Applications Journal, Special Issue on Ad-Hoc Networks*, 4:157–174, 1999.
- [11] J.J. Garcia-Luna-Aceves and Y. Wang. Throughput and Fairness of Collision Avoidance Protocols in Ad Hoc Networks. *Ad Hoc Networks: Technologies and Protocols*, Kluwer Academic Publishers, 2004.
- [12] M. Garetto, T. Salonidis, and E. Knightly. Modeling per-flow throughput and capturing starvation in CSMA multi-hop wireless networks. In *Proc. IEEE INFOCOM*, Barcelona, Spain, 2006.
- [13] M. Garetto, J. Shi, and E. Knightly. Modeling Media Access in Embedded Two-Flow Topologies of Multi-hop Wireless Networks. In *Proc. ACM MobiCom*, Cologne, Germany, August 2005.
- [14] IEEE 802.11 Working Group. Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications. 1997.
- [15] S. Lu H. Luo and V. Bharghavan. A new model for packet scheduling in multihop wireless networks. In *Proc. ACM MobiCom*, Boston, MA, USA, August 2000.
- [16] B. Hajek and G. Sasaki. Link Scheduling in Polynomial Time. *IEEE Transactions on Information Theory*, 34(5):910–917, September 1988.
- [17] X. Huang and B. Bensaou. On max-min fairness and scheduling in wireless ad-hoc networks: Analytical framework and implementation. In *Proceedings of ACM MobiHoc*, Long Beach, CA, October 2001.
- [18] N. Jain and S.R. Das. A Multichannel CSMA MAC Protocol with Receiver-Based Channel Selection for Multihop Wireless Networks. In *Proc. IC3N*, Scottsdale, AZ, USA, October 2001.
- [19] M. Kodialam and T. Nandagopal. Characterizing the Achievable Rates in Multihop Wireless Networks. In *Proc. ACM MobiCom*, San Diego, CA, USA, September 2003.
- [20] A. Kumar, E. Altman, D. Miorandi, and M. Goyal. New insights from a fixed point analysis of single cell IEEE 802.11 WLANs. In *Proc. IEEE INFOCOM*, Miami, FL, March 2005.
- [21] C.R. Lin. On-demand QoS routing in Multihop mobile networks. In *Proc. IEEE INFOCOM*, Anchorage, AK, April 2001.
- [22] H. Luo, J. Cheng, and S. Lu. Self-coordinating localized fair queueing in wireless ad hoc networks. *IEEE Transaction on Mobile Computing*, 3(1), January-March 2004.
- [23] T. Nandagopal, T. Kim, X. Gao, and V. Bharghavan. Achieving MAC layer fairness in wireless packet networks. In *Proc. ACM MobiCom*, Boston, MA, August 2000.
- [24] A. Nasipuri and S. Das. A Multichannel CSMA MAC Protocol for Multihop Wireless Networks. In *Proc. IEEE Wireless Communications and Networking Conference (WCNC)*, New Orleans, September 1999.
- [25] M.J. Neely, E. Modiano, and C.E. Rohrs. Dynamic Power Allocation and Routing for Time Varying Wireless Networks. *IEEE Journal on Selected Areas in Communications*, 23(1):89–103, Jan 2005.
- [26] T. Ozugur, M. Naghshineh, P. Kermani, C.M. Olsen, B. Rezvani, and J.A. Copeland. Balanced media access methods for wireless networks. In *Proc. ACM MobiCom*, Dallas, TX, USA, October 1998.
- [27] S. Ramanathan. A Unified Framework and Algorithm for Channel Assignment in Wireless Networks. In *Proc. IEEE INFOCOM*, Kobe, Japan, September 1997.
- [28] T. Salonidis and L. Tassiulas. Distributed Dynamic Scheduling For End-to-end Rate Guarantees In Wireless Ad Hoc Networks. In *Proc. ACM MobiHoc*, Urbana-Champaign, IL, USA, May 2005.
- [29] S. Sarkar and L. Tassiulas. End-to-end bandwidth guarantees through fair local spectrum share in wireless ad-hoc networks. In *Proc. Control and Decision Conference (CDC)*, Maui, HI, USA, December 2003.
- [30] J. So and N. Vaidya. Multi-Channel MAC for Ad Hoc Networks: Handling Multi-Channel Hidden Terminals Using A Single Transceiver. In *Proc. ACM MobiHoc*, Tokyo, Japan, May 2004.
- [31] L. Tassiulas and S. Sarkar. Maxmin fair scheduling in wireless networks. In *Proceedings of IEEE INFOCOM '02*, New York, NY, June 2002.
- [32] Y.C. Tseng, C.S. Hsu, and T.-Y. Hsieh. Power-saving protocols for IEEE 802.11-based Multi-hop Wireless Networks. In *Proc. IEEE INFOCOM*, New York, NY, USA, June 2002.
- [33] S.-L. Wu, C.-Y. Lin, Y.-C. Tseng, and J.-P. Sheu. New Multi-Channel MAC Protocol with On-Demand Channel Assignment for Mobile Ad Hoc Networks. In *Proc. International Symposium on Parallel Architectures, Algorithms and Networks (I-SPAN)*, Dallas / Richardson, TX, USA, December 2000.
- [34] C. Zhu and M.S. Corson. A Five-Phase Reservation Protocol (FPRP) for Mobile Ad Hoc Networks. *Wireless Networks*, 7(4):371–384, Oct 1998.
- [35] C. Zhu and M.S. Corson. QoS routing for mobile ad hoc networks. In *Proc. IEEE INFOCOM*, New York, NY, June 2002.