Comparison of Multi-Channel MAC Protocols [*]

Jeonghoon Mo
School of Engineering
Information and
Communications University
Yusong, Taejon, 305-600,
Korea
jhmo@icu.ac.kr

Hoi-Sheung Wilson So
Department of Electrical
Engineering and Computer
Sciences
University of California
Berkeley, CA 94720, USA
so@eecs.berkeley.edu

Jean Walrand
Department of Electrical
Engineering and Computer
Sciences
University of California
Berkeley, CA 94720, USA
wlr@eecs.berkeley.edu

ABSTRACT

This paper compares, through analysis and simulation, a number of multichannel MAC protocols for wireless networks. We first classify these protocols into 4 categories based on their principles of operation. We then examine the effects of the number of channels and devices, channel switching times, and traffic patterns on throughput and delay. Our study focuses on a single collision domain.

Categories and Subject Descriptors: C.2.5 [Computer-Communication Networks]: Local and Wide-area Networks—*Access schemes*; C.4 [Performance of Systems]: Modeling techniques.

General Terms: Algorithms, Design, Performance.

Keywords: Multiple channels MAC, Multi-channel MAC, Wireless MAC protocols, Simulation, Analysis, Comparison.

1. INTRODUCTION

Researchers have proposed protocols that exploit the multiple channels available in 802.11 and other wireless networks to increase capacity. These protocols are for networks in which orthogonal channels, such as disjoint frequency bands, are available. Using a multichannel MAC protocol, different devices can transmit in parallel on distinct channels. The parallelism increases the throughput and can potentially reduce the delay, provided that the channel access time is not excessive. Protocols differ in how devices agree on the channel to be used for transmission and how they resolve potential contention for a channel. These choices affect the delay and throughput characteristics of the protocol.

There have been only limited studies compare these protocols under identical operating conditions. In this paper, we compare several existing and one new multi-channel MAC protocols. As may be expected, different protocols are preferable depending on the operating conditions. Our objective is to contribute to the understanding of the relative merits of different designs. We first use analytical models to gain insight into the strengths and weaknesses of the various protocols. We then use simulations to obtain more accurate comparisons assuming more realistic traffic patterns. We consider only the case of a single collision domain where all the devices can hear one another.

In Section 2, we describe the protocols that we compare in this paper. Section 3 presents simplified analytical models of the protocols. Section 4 discusses the numerical results that result from the analytical models and compares the performance of the protocols. Section 5 describes results of simulations of more realistic models of the protocols. Section 6 concludes the paper with some comments about the lessons of this study.

2. DESCRIPTION OF PROTOCOLS

There are many variations on multichannel protocols. Our first step is to classify them based on their general principles of operation. We then describe representative protocols of the different classes. We also comment on the many variations that have been proposed for such protocols.

2.1 Principles of Operation

Devices using a multichannel MAC protocol exchange control information in order to agree on the channel for transmitting data. In *single rendezvous* protocols, the exchange of control information occurs on only one channel at any time. That single control channel can become the bottleneck under some operating conditions. *Multiple rendezvous* protocols allow multiple devices to use several channels in parallel to exchange control information and make new agreements. This approach alleviates the rendezvous channel congestion problem but raises the challenge of ensuring the idle transmitter and receiver visit the same rendezvous channel.

In the subsequent sections, we describe the following protocols and their variations:

1. **Dedicated Control Channel** (single rendezvous using 2 radios): devices use one radio to constantly monitor the control channel.

2. **Common Hopping** (single rendezvous using 1 radio): devices hop together quickly and stop upon agreement for transmission.

---

[*] This work was supported in part by the National Science Foundation under Grant CNS-0435478 and in part by the BrOMA ITRC program of Korean MIC.

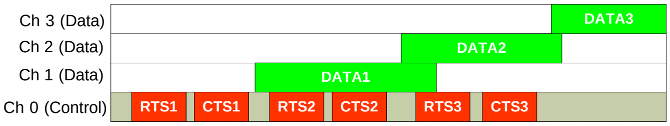
MSWiM'05, October 10–13, 2005, Montreal, Quebec, Canada.

3. **Split Phase** (single rendezvous using 1 radio): devices periodically tune to the control channel together.

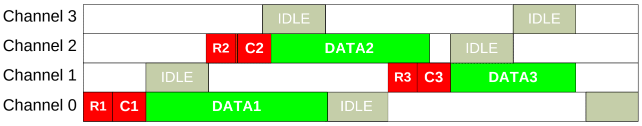
4. **McMAC** (multiple rendezvous using 1 radio): transmitter jumps to receiver's slow hopping sequence.

2.2 Dedicated Control Channel

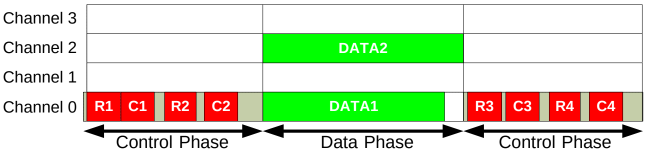
Every device has two radios. One radio is tuned to a channel dedicated to control messages; the other radio can tune to any other channel. In principle, all devices can overhear all the agreements made by other devices, even during data exchange. This system's efficiency is limited only by the contention for the control channel and the number of available data channels.



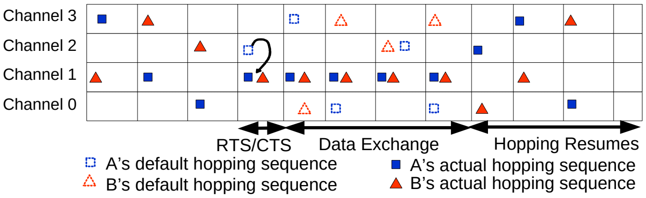
(a) Dedicated Control Channel Appraoch



(b) Common Hopping Approach



(c) Split Phase Approach



(d) McMAC

Figure 1: Basic operations of different MAC approaches.

Fig. 1(a) illustrates the operations of Dedicated Control Channel. In the figure, channel 0 is the control channel and channels 1, 2, and 3 are for data transmission. When device A wants to send to device B, it transmits an RTS (request-to-send) packet on the control channel. That RTS specifies the lowest-numbered free channel. Upon receiving the RTS, B responds with a CTS (clear-to-send) packet on the control channel, confirming the data channel suggested by A. The RTS and CTS packets also contain a Network Allocation Vector (NAV) field, as in 802.11, to inform other devices of the duration for which the sender, the receiver, and the chosen data channel are busy. Since all devices listen to the control channel at all times, they can keep track of the busy status of other devices and channels even during data exchange. Devices avoid busy channels when selecting a data channel.

Examples of this approach include DCA (Dynamic Channel Allocation) [10], DCA-PC (Dynamic Channel Allocation with Power Control) [11] and DPC (Dynamic Private Channel) [12].

The major advantage of Dedicated Control Channel is that it does not require time synchronization–rendezvous always happen on the same channel. The disadvantage of this protocol is that it requires a separate control radio and a dedicated channel, increasing cost and decreasing spectral efficiency when few channels are available.

2.3 Common Hopping

In this approach, devices have only one radio. Devices not exchanging data cycle through all channels synchronously. A pair of devices stop hopping as soon as they make an agreement for transmission and rejoin the common hopping pattern subsequently after transmission ends.

The Common Hopping protocol improves on Dedicated Control Channel in two respects: 1) it use all the channels for data exchange; 2) it requires only one transceiver per device. As shown in Fig. 1 (b), the hopping pattern cycles through channels 0, 1, 2 and 3. When device A wants to send to device B, it sends an RTS to B on the current common channel. If B receives the RTS properly, it returns a CTS on the same channel. Devices A and B then pause hopping and remain on the same channel during data transfer while the other idle devices continue hopping. When they are finished, devices A and B rejoin the common hopping sequence with all the other idle devices. It is possible that the common hopping sequence wraps around and visits the channel A and B are using before they finish data exchange. Idle devices sense the carrier and refrain from transmitting if it is busy.

While A and B are exchanging data, they are unaware of the busy status of the other devices. Hence, it is possible that a sender sends an RTS to a device that is currently busy on a different channel. Another issue with this approach is that devices hop more frequently. State-of-the-art integrated circuits implementations of tri-mode 802.11a/b/g radios require only about 30μsec for its voltage-controlled oscillator (VCO) to settle [13], but commercial off-the-shelf 802.11b transceivers require about 150 to 200μsec to switch channels [4]. Considering that an RTS in 802.11b takes only about $200 - 300\mu$sec, the hopping time penalty is not negligible. The approach also requires devices to have tight synchronization. Examples of this design approach include CHMA (channel hoping multiple access) [8] and CHAT (Channel Hopping multiple Access with packet Trains) [9].

2.4 Split Phase

In this approach, devices use a single radio. Time is divided into an alternating sequence of control and data exchange phases, as shown in Fig. 1 (c). During a control phase, all devices tune to the control channel and attempt to make agreements for channels to be used during the following data exchange phase.

If device A has some data to send to device B, it sends a packet to B on the control channel with the ID of the lowest numbered idle channel, say, i. Device B then returns a confirmation packet to A. At this point, A and B have agreed to use channel i in the upcoming data phase. Once committed, a device cannot accept other agreements that conflict with earlier agreements. (Note that when hidden nodes are prevalent, the sender and the receiver might have a very different view of which channels are free. A more sophisticated agreement protocol is then needed, as proposed in [6].)

In the second phase, devices tune to the agreed channel and transfer data. The protocol allows multiple pairs to choose the same channel because each pair might not have enough data to use up the entire data phase. As a result, the different pairs must either schedule themselves or contend during the data phase. In the analysis, we assume that at most one device pair can be assigned to each channel, so there is no need for scheduling or contention. In the simulations, we assume random access as suggested in MMAC [6].

The advantage of this approach is that it requires only one radio per device. However, it requires time synchronization among all devices, though the synchronization can be looser than in Common Hopping because devices hop less frequently. Examples of this approach are MMAC [6] and MAP (Multichannel Access Protocol) [2]. Their main difference is that the duration of the data phase is fixed in MMAC whereas it is variable in MAP and depends on the agreements made during the control phase.

2.5 Multiple Rendezvous

Multiple rendezvous protocols differ from the previous three in that multiple device pairs can make agreements simultaneously on distinct channels. The main motivation is to overcome the single control channel bottleneck. However, since there are multiple rendezvous channels, special coordination is required so that two devices can rendezvous on the same channel. One solution is for each idle device to follow a “home” hopping sequence and for the sending device to transmit on that channel to find the intended receiver. Examples of this approach include SSCH (Slotted Seeded Channel Hopping) [1] and McMAC [5].

In SSCH there are as many hopping sequences that each device can follow as there are channels. Each sequence is uniquely determined by the seed of a pseudo-random generator. Each device picks multiple (e.g., 4) sequences and follows them in a time-multiplexed manner. When device A wants to talk to B, A waits until it is on the same channel as B. If A frequently wants to talk to B, A adopts one or more of B's sequences, thereby increasing the time they spend on the same channel. For this mechanism to work, the sender learns the receiver's current sequences via a seed broadcast mechanism.

In McMAC, each device picks a seed to generate an independent pseudo-random hopping sequence. When a device is idle, it follows its default hopping sequence as shown in Fig. 1 (d). Each device puts its seed in every packet it sends, so its neighbors eventually learn its hopping sequence. For simplicity, devices are assumed to hop synchronously. The hopping can be made less frequent than Common Hopping to reduce the channel switching and synchronization overhead. When device A has data to send to B, A flips a coin and transmits with some probability p during each time slot. If it decides to transmit, it tunes to the current channel of B (e.g., channel 1) and sends an RTS. If B does not reply with a CTS, either due to an error or because B is busy, then A tries again later by coin flips. If B replies with a CTS, both A and B stop hopping to exchange data. Data exchange normally takes several time slots. After the data exchange is over, A and B return to their original hopping sequence as if there was no pause in hopping.

SSCH and McMAC are similar in that they allow devices to rendezvous simultaneously on different channels. In the rest of the paper, we focus on McMAC as an example protocol in this category.

2.6 Protocol Variations

In this paper, we compare four generalized approaches to designing multi-channel MAC protocols. An actual protocol includes fine adjustments that deviate from the generalized scheme. Instead of incorporating all possible variations of the four schemes in our analysis and simulation, we briefly mention several proposed improvements and discuss their effects qualitatively.

For Dedicated Control Channel, it is possible to use the control channel for data transfer when all other channels are busy. For Split Phase, adaptation of the duration of data and control phases was proposed by [2]. [6] suggests advertising the number of packets for each destination in the rendezvous message to achieve better load balancing across channels. We have tried to optimize the values of the control and data phase durations in our simulations, but the results are omitted due to space constraints.

3. SYSTEM MODEL AND ANALYSIS

In this section, we describe and analyze simplified models of the protocols. The objective of the models is to enable numerical comparisons of the performance characteristics of the protocols under identical operating conditions. The following simplifications are made for all the protocols:

1. Time is divided into small slots, with perfect synchronization at slot boundaries;
2. Upon making an agreement, the devices can transmit only one packet (one may think of a ”packet” as the amount of data that can be transferred per channel agreement);
3. The packet lengths are independent and geometrically distributed with parameter q (i.e., with mean $1/q$);
4. Devices always have packets to send to all the other devices; in each time slot, an idle device attempts to transmit with probability p.

The first simplification enables us to use discrete time models. The second one reduces the efficiency of all the protocols since it requires a new agreement for every packet. The third simplification is needed to construct simple Markov models. We use the fourth simplification to identify the throughput of the protocols.

These simplifications allow us to form a Markov chain whose state X_t is the number of communicating pairs at time t. When $X_t = k$, $2k$ devices are involved in data transfer while the other $N - 2k$ devices are idle. The maximum number of pairs is bounded by the number of M_D of data channels and $\lfloor N/2 \rfloor$, half the number of devices. Accordingly, the state space of X_t is

$$\mathcal{S} := \{0, 1, \ldots, \min(\lfloor \frac{N}{2} \rfloor, M_D)\}.$$

The number M_D of data channels is equal to M for all approaches except Dedicated Control Channel for which $M_D = M - 1$ since a channel is reserved for control.

A state transition happens when new agreements are made or existing transfers end. Let $S_k^{(i)}$ and $T_k^{(j)}$ respectively denote the probability that i new agreements are made and the probability that j transfers terminate in the next slot when the state is k. The state transition probability p_{kl}

from state k at time t to l at time $t+1$ can be expressed as follows:

$$p_{kl} = \sum_{m=(k-l)^+}^{k} S_k^{(m+l-k)} T_k^{(m)}. \tag{1}$$

In this expression, m is the number of transfers that terminate and its value is between $(k-l)^+$ and k. At least $(k-l)^+$ transfers should terminate to have l pairs in the next slot and k is the maximum number of terminating transfers. Also, the probability $T_k^{(j)}$ that i transfers finish when the system is in state k is given by the following expression:

$$\begin{aligned} T_k^{(j)} &= Pr[j \text{ transfers terminate at time } t | X_{t-1} = k] \\ &= \binom{k}{j} q^j (1-q)^{k-j}. \end{aligned} \tag{2}$$

Equation (1) is further simplified in the single rendezvous protocols such as Dedicated Control Channel or Common Hopping because $S_k^{(i)} = 0$ for all $i > 1$. Indeed, at most one additional pair can meet in the next slot in a single rendezvous protocol. Accordingly, for such protocols, the equation becomes

$$p_{kl} = T_k^{(k-l)} S_k^{(0)} + T_k^{(k-l+1)} S_k^{(1)} \tag{3}$$

where $T_k^{(j)} = 0$ when $j < 0$.

The average utilization ρ per channel can be obtained as

$$\rho = \frac{\sum_{i \in \mathcal{S}} i \cdot \pi_i}{M} \tag{4}$$

where π_i is the limiting probability that the system is in state i and $\mathcal{S}$ is the state space of the Markov chain. One obtains π_i by solving the balance equations of the Markov chain. We then derive the total system throughput by multiplying ρ by the channel transmission rate and by M_D, the number of data channels.

3.1 Dedicated Control Channel

Devices constantly monitor the control channel and keep track of which devices and data channels are idle. When a device has packets to transmit to an idle receiver, it sends an RTS message for that idle receiver on the control channel. If it hears the RTS, the receiver replies to the sender with a CTS. Then both the sender and the receiver tune to the agreed channel to start transmission.

An agreement is made when exactly one idle device attempts to transmit an RTS on the control channel. Hence, the success probability $S_k^{(i)}$ in the next time slot, given that k pairs are communicating in the current slot, is:

$$S_k^{(i)} = \begin{cases} (N-2k)p(1-p)^{(N-2k-1)}, & \text{if } i = 1; \\ 1 - S_k^{(1)}, & \text{if } i = 0; \\ 0, & \text{otherwise.} \end{cases} \tag{5}$$

The transition probabilities (3) can be rewritten as follows:

$$p_{kl} = \begin{cases} 0 & \text{if } l > k+1 \\ T_k^{(0)} S_k^{(1)}, & \text{if } l = k+1; \\ T_k^{(k-l)} S_k^{(0)} + T_k^{(k-l+1)} S_k^{(1)}, & \text{if } 0 < l \le k; \\ T_k^{(k)} S_k^{(0)} & \text{if } l = 0. \end{cases} \tag{6}$$

We obtain the system throughput R_d as

$$R_d = (M-1) C \rho_d \tag{7}$$

where C is the channel capacity and ρ_d is the data channel utilization that we calculate using (4); the subscript d refers to the Dedicated Control Channel approach.

3.2 Common Hopping

The analysis of this protocol is very similar to that of the Dedicated Control Channel, but with three differences: 1) devices do not track the status of each other; 2) some slots are busy and unavailable for control messages; 3) the switching penalty is incurred whenever a device hops.

3.2.1 Information about Other Devices

Even if an RTS is sent without collision, the receiver may not respond because it might be busy on a different channel. When a device is sending or receiving, it cannot keep track of others. At best, idle devices can keep track of the agreements that others make. However, this information becomes stale once the devices become busy. We approximate this situation by assuming that the sender selects the receiver uniformly out of $N-1$ other devices and that the probability that the selected receiver is not busy is $\frac{N-2k-1}{N-1}$.

3.2.2 Busy Slots

The protocol can make a new agreement only when the current common hopping channel is idle. We model this effect by considering that the probability that idle devices can use a given slot to make an agreement is $(M-k)/M$ when k channels are busy.

Combining the effects described in the previous two points, the success probability $S_k^{(i)}$ now becomes:

$$S_k^{(i)} = \begin{cases} (N-2k)p(1-p)^{(N-2k-1)} \frac{N-2k-1}{N-1} \frac{M-k}{M}, & \text{if } i = 1; \\ 1 - S_k^{(1)}, & \text{if } i = 0; \\ 0, & \text{otherwise.} \end{cases} \tag{8}$$

The factor $\frac{N-2k-1}{N-1}$ in (8) is the probability that the receiver is one of the $N-2k-1$ idle devices other than the transmitter among the other $N-1$ devices. The last term $\frac{M-k}{M}$ is introduced to represent the chance that the common hopping channel is busy when k channels are busy.

3.2.3 Channel Switching Penalty

Let t_s and t_p denote the duration of one slot and the channel switching penalty respectively. One slot in the Common Hopping protocol has duration $t_s' = t_s + t_p$. Since the hopping penalty does not occur when devices transmit data in the other schemes, we can think of each slot as being slightly longer in Common Hopping, which results in a higher packet termination probability q. To make a fair comparison, one should choose $q' = q \cdot \frac{t_s + t_p}{t_s}$. The system throughput R_c is as follows:

$$R_c = MC\rho_c \tag{9}$$

where C is the channel capacity and ρ_c is the utilization of the system that can be calculated using (4). The subscript c denotes Common Hopping. Note that the multiplier is M instead of $M-1$ in Dedicated Channel.

3.3 Multiple Rendezvous

We analyze McMAC [5] in this section. In McMAC, each device has a home pseudo-random hopping sequence defined by a seed. Eventually, through broadcast, every device knows the seed of the other devices and can determine their home channel at any given time.

In McMAC, a device is unaware of which devices and channels are idle. We assume that any idle device is equally likely to have any channel as its current home channel and that these home channels are independent. When a device is idle, it attempts to transmit in the next time slot with probability p. In that case, the device chooses another device at random and goes to its channel.

To make j new agreements the following conditions must hold:

(1) The number A of devices that attempt to transmit should be at least i.

(2) The devices attempting to transmit should be on a channel without other attempts. Let O denote the number of channels where those isolated attempts take place; we call these channels "one-attempt channels."

(3) The channel where a device attempts to transmit should be idle. We designate by I the number of idle channels among the O one-attempt channels.

(4) Designate by J the number of transmitters out of I can find a receiver that is idle and does not attempts to transmit. Then $J = j$.

We compute the probability $S_k^{(i)}$ by conditioning on the values of O, I and A. We can write

$$S_k^{(j)} = \sum_{o,i,a} P[A=a]P[O=o|A=a]P[I=i|O=o,A=a] \\ \times P[J=j|I=i,A=a,O=o]$$

- $P[A=a] = \binom{N-2k}{a} p^a (1-p)^{(N-2k-a)}$ for $a \geq i$;
- $P[O=o|A=a]$ is the probability that o devices out of a are in a channel without other attempts given that a devices attempt. This probability is the same as the probability that o urns out of M contain exactly one ball after we throw a balls independently and uniformly in M urns. If we let Y_i^n be the number of urns with i balls at the n-th throw, then (Y_0^n, Y_1^n) is a Markov chain and the probability distribution of Y_1^a can be computed recursively over $n = 1, \ldots, a$ from the transition probabilities of that Markov chain.
- $P[I=i|O=o,A=a]$ is the probability that i channels out of o one-attempt channels are idle, which is equivalent to the probability that exactly $o-i$ out of o one-attempt channels are busy. Recall that k out of M channels are busy. We pick k busy channels out of M uniformly and also pick o one-attempt channels out of M uniformly, independently of each other. The intersection of the two sets corresponds to busy one-attempt channels. The conditional distribution of I is given by

  $$P[I=i|O=o,A=a] = P[I=i|O=o] = \frac{\binom{k}{o-i}\binom{M-k}{i}}{\binom{M}{k}}.$$

  The first equality comes from that I and A are conditionally independent given O. The denominator is the number of ways to select k busy channel out of M. The numerator is the number of ways to select $o-i$ one-attempt channels among k busy ones and to select i one-attempt channels among $M-k$ channels.
- We approximate the probability that a sender that attempts to transmit alone in an idle channel finds its receiver by $p_s = \frac{N-2k-a}{N-1}$. Moreover, we assume that the i transmitters are independently successful in finding their receivers. Let J be the number of successful senders who are alone in an idle channel and are able to find their receiver successfully. Then,

  $$P[J=j|I=i,O=o,A=a] = P[J=j|I=i,A=a] \\ = \binom{i}{j} p_s^j (1-p_s)^{(i-j)}.$$

3.4 Split-Phase Approach

In Split-Phase, time is divided into alternate control and data phases with durations c and d, respectively. Let $R(n)$ be the throughput of the n-th period. By ergodicity,

$$\lim_{n\to\infty} \frac{\sum_{i=1}^{n} R(n)}{n} \to R_s$$

where R_s is the expected throughput per period of the Split Phase approach.

We designate by K_n the random number of agreements made during the n-th control phase and we define

$$\phi_i^c := P(K_n = i), \text{ for } i = 0,1,2,\ldots,\gamma := \min\left(c, \lfloor \frac{N}{2} \rfloor\right). \tag{10}$$

The above probability can be computed using the following recursive relationship on the duration c:

$$\phi_i^c = p_{suc}\phi_{i-1}^{c-1} + (1-p_{suc})\phi_i^{c-1}, \tag{11}$$

where p_{suc} is the probability that an agreement is made in a slot. Note that $\phi_0^c = (1-p_{suc})^c$.

Let $R(i,d)$ denote the average throughput in the duration of d when i agreements are made in the control phase. Then the channel utilization ρ_s can be computed using the following equation:

$$\rho_s = \frac{1}{(c+d)M} \sum_{i=1}^{\gamma} \phi_i^c R(i,d). \tag{12}$$

The system throughput R_s is

$$R_s = MC\rho_s \tag{13}$$

where C is the channel capacity.

The throughput $R(i,d)$ when i new agreements are made in the control channel cannot be larger than Md. If we assume perfect distribution of i agreements over M channels, each channel has either l or $l+1$ agreements where $l := \lfloor \frac{i}{M} \rfloor$. Since the packet size is geometrically distributed, the channel that has l agreements can be utilized up to $\min\left(\sum_{j=1}^{l} Y_j, d\right)$ slots where the $Y_j's$ are geometrically distributed random variables. Therefore, the averaged throughput can be upper-bounded as follows:

$$R(i,d) \leq (M-r)E\left[\min\left(\sum_{j=1}^{l} Y_j, d\right)\right] + rE\left[\min\left(\sum_{j=1}^{l+1} Y_j, d\right)\right] \tag{14}$$

where r is the number of channels with $l+1$ agreements. Thus $(M-r)$ is the number of channels with l agreements.

The right side of (14) is an upper bound since we assume perfect alignments among l pairs or $l+1$ pairs in each channel. The real throughput can be smaller due to collision among l pairs. However, we can use the right side as an approximation when the value l is small.

If c is too short, the multiple channels may not be utilized; On the other hand, if c is too long, the packet may suffer the long delay and under utilization. Therefore, choosing appropriate values for c and d is crucial to the performance of this scheme.

4. NUMERICAL RESULTS

In this section, we compare the different protocols using the analytical models that we described so far. Several approximations were made in the analysis presented in the previous section. To verify that such approximations are reasonable, we compared the analytical results to Monte Carlo simulations of the protocols. We call this set of simulations SimLite to distinguish them from more sophisticated simulation scenarios to be introduced later. We also vary the parameters such as the number of channels, channel switching time, number of devices and so on to observe their effects on the throughput.

We created two simulation scenarios: 802.11b and 802.11a. In the 802.11b scenario, there are 20 devices, 3 channels, and each channel has a data rate of 2 Mbps. In the other, there are 40 devices, 12 channels with a rate of 6 Mbps each. The packets generated have a random length with a geometric distribution with average length of 1 KB or 10 KB. One should think of the packet length as the amount of data a device can transfer after each channel agreement since there is no queueing. The bar charts of Fig. 2 shows that the analytical models are in close agreement with SimLite for different packet lengths in both scenarios.

4.1 802.11b Scenario

Fig.2 (top) shows the throughput under the 802.11b scenario. The x-axis is the transmission probability p of each device and y-axis is throughput.

Dedicated Channel Protocol. The protocol can use only two channels out of three for data which limits the maximum throughput to 4 Mbps which is much less than Common Hopping or McMAC. Given that constraint, however, Dedicated Control Channel achieves 3.8Mbps.

Parallel Rendezvous Although the control channel is not saturated, McMAC still performs the best for both short and long packets because (i) it uses all 3 channels for data transfer, and (ii) it does not have any quantization overhead of Common Hopping since it can reuse a channel as soon as the previous transfer has finished. Common Hopping cannot reuse a channel until the common hopping sequence has wrapped around to it. For long packets, McMAC achieves a remarkable 5.5 Mbps out of 6 Mbps.

Split-phase Approach. The throughput of Split Phase is the lowest for 1 KB packets and close to the lowest for 10 KB packets, even though we optimized the control and data durations c and d to be 20 msec and 40 msec. This is because only three agreements can be made for the three channels during the control phase.

4.2 802.11a Scenario

The bottom bar chart of Fig. 2 shows the throughput of the four different protocols in the 802.11a scenario.

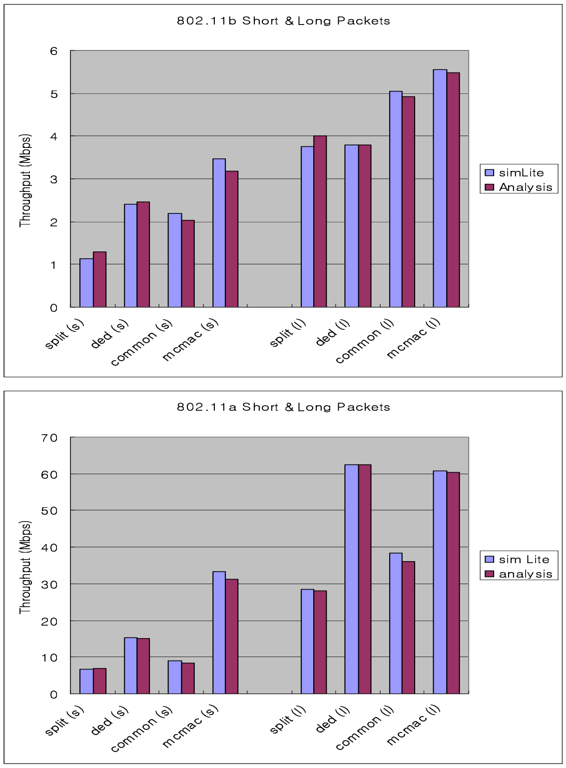


Figure 2: Throughput predicated by the analysis vs. simulation. Top: 802.11b setting Bottom: 802.11a setting.

Parallel Rendezvous. McMAC achieves about 31Mbps in the small packet size case (left bars) and 60Mbps for the large packet size case (right bars). Note that Dedicated Control Channel performs slightly better than McMAC when packets are large, but recall that it uses two radios.

Control Channel Congestion. In single rendezvous protocols, since there is only one control channel, a combination of short packets and a large number of channels can cause control channel congestion as shown on the left half of the plot. The best possible throughput for single rendezvous channel protocols can be estimated as follows. On average, one agreement is made every $1/P_{succ} \approx e = 2.718$ slots. After each agreement, a 1 KB packet is transferred on average. Since it takes 6.8 slots to transfer 1 KB packet (200μ sec per slot), the maximum throughput is $(6.8/e) \times 6$Mbps $\approx$ 15Mbps. Dedicated Control Channel, Common Hopping and Split Phase achieve 15Mbps, 9Mbps, and 8Mbps respectively.

Dedicated Channel with Long Packets. Observe that Dedicated Control Channel achieves more than 63Mbps when the average packet size is 10Kbyte. With large average packet size, the control channel no longer is a bottleneck. Moreover, 1 out of 12 channel is a small overhead.

4.3 Number of Channels

Fig. 3 shows the throughput results with different numbers of channels (x-axis) and for two average packet lengths. The number of devices in each simulation is 6 times the number of channels. We assume that the bandwidth per channel is 2Mbps. The slot time is 812μs; the channel switching time is 100μs. For Split Phase, we assume the control and data durations to be 20 msec and 40 msec respectively.

McMAC scales well with an increasing number of channels while the single rendezvous protocols do not. The top plot of Fig. 3 shows that when packets are short, the throughput of all single channel protocols (Split Phase, Dedicated Control Channel, Common Hopping) flatten out when the number of channels is more than six due to control channel congestion. With longer packets, the flattening of the curves occurs later with more available channels. However, the bottleneck of the control channel is only delayed, but *not* avoided.

The simulation results of [6] (Fig. 11) show that the performance of Split Phase is better than Dedicated Control Channel as the number of channel increases. The authors compare MMAC, a Split Phase algorithm with a Dedicated Control Channel approach, DCA, with 3 to 6 channels. The aggregate throughput of MMAC with 6 channels was about 3.7Mbps while that of the DCA was 2.3Mbps. Their results are opposite to our findings shown in Fig.3. The discrepancy stems mainly from two differences. First, in the channel scaling study of [6] (Fig. 11), the authors assume a sender can send multiple packets to the same receiver within each data phase due to queueing. Second, they assume that the traffic among different devices are disjoint, such that each receiver will only communicate with one sender. When the traffic is non-disjoint but queueing is still present, [6] also shows the performance gap between the two approaches narrows.

4.4 Average Packet Size

Fig. 4 shows the performance with different average packet sizes (x-axis) when the number of channels is 3 (top) and 12 (bottom). The throughput of every schemes increases with the size of packets, which can be explained from the increase in utilization per agreement. In the 802.11b case, Dedicated Control Channel becomes worse than Split Phase with packet size larger than 5Kbytes. Dedicated Control channel is bounded by 4Mbps since one channel is used for control. The performance of Split Phase depends heavily on the packet sizes. Its throughput increases to 3.5Mbps with 5Kbyte packet from 1.2Mbps with 1Kbyte packet. It is recommended that the Split Phase transmit as much as possible when a mobile gains access to a channel to maintain reasonable channel utilization. However, even when the packet size is large, it is worse than the others in the 12-channels case.

4.5 Channel Switching Penalty

Fig. 5 shows the performance with increasing switching penalty from 100μs to 2000μs when the number of channels is 3 (top) and 12 (bottom) respectively. Commercial off-the-shelf 802.11b transceivers require about 100 to 200μsec to switch channels. The throughput of Common Hopping and McMAC decreases faster whereas that of Split Phase and Dedicated Control Channel are almost insensitive to the switching time. This is because the first two approaches are based on hopping and incur a penalty every time they hop. When the switching penalty is high, Dedicated Control Channel and Split Phase are more efficient.

5. SIMULATION RESULTS

5.1 Simulation Model

We developed a slotted-time simulator to compare the different protocols in more realistic settings. There are three major differences between the simulations and the analytical model. The first is the existence, in the simulations, of per-destination packet queues in each device to study the effects of queueing and delay. Second, the traffic models in the simulations are more realistic. Third, several protocol improvements are added in the simulations. The assumptions and key features of the simulator are as follows:

- *Coarse-Grained Discrete Time*: The simulator works in discrete time. Each slot corresponds to the time required for making an agreement (e.g., RTS and CTS).

- *Modeling of CSMA/CA Medium Contention*: In the analytical model, we approximated the transmission success probability using the slotted ALOHA model. The carrier-sense feature in today's radios in effect reduces the penalty of a collision from hundreds of microseconds (packet time scales) to carrier-sense slots (microseconds). We introduce a parameter, the *contention success probability* p_{succ}, to model the medium contention mechanism. During each time slot, with probability p_{succ}, a winner is chosen randomly among all the devices contending for a channel. By picking appropriate values of p_{succ} we believe that this model can approximate the behavior and performance of CSMA/CA. One may argue that this model is an oversimplification of the situation. However, since this approximation is used for all models, it should result in a fair comparison.

- *Queueing of Packets*: Each device stores packets in per-destination queues. To improve performance, once a device starts sending data, it continues until either its queue for this destination becomes empty or a preset *medium occupancy limit* is reached, or, in the case of Split Phase only, the data phase has ended.

- *Traffic Models*: We use two classes of traffic: constant bit rate (CBR) and file transfers. Fixed-size Packets arrive periodically in both directions of a CBR connection. Independently, random file transfer requests with a geometrically distributed size are generated between all pairs of devices.

- *Split-Phase:* For the Split-Phase protocol, to increase performance, we allow multiple pairs to share a channel in the same data phase if no more free channels are available, as in MMAC [6]. Since multiple pairs may share a channel, contention resolution is required in the data phase. Additionally, devices that did not make any channel agreement can go to the last (and usually the least utilized) channel to send data.

5.2 Impacts of Receiver Contention

For multi-channel MAC, the ideal traffic pattern is a disjoint one in which each sender only communicates with one receiver, and vice versa. When the traffic is non-disjoint, it is possible for a sender to pick a busy receiver, thereby wasting throughput. To investigate the impact of receiver contention, in this section, we vary the number of simultaneous CBR connections per device. Each device sets up CBR connections to δ distinct destinations. The first device chooses δ different destinations. The rest of the nodes choose up to δ destinations if they are involved in less than δ

connections. Only destinations who have less than δ connections are eligible. A larger value of δ corresponds to more contention for receiver devices, and likely lower efficiency and longer delays.

Fig. 6 shows the delay vs. throughput of various schemes as δ increases under the 802.11b settings. Fig. 7 shows the delay vs. throughput under the 802.11a settings. The top two graphs correspond to $\delta = 1$ and the bottom ones to $\delta = 5$. We summarize our observations as follows.

5.2.1 *Throughput Degradation*

Given a fixed load, as δ increases, the average amount of data sent over each pair of devices decreases. Therefore, one expects a shorter communication to take place after each channel agreement. As a result, the frequency for making channel agreements increases, thereby decreasing the throughput for all schemes.

Moreover, for Split Phase, a further decrease in the throughput results as δ increases because each sender has to visit more receivers to deliver its packets. When these receivers have chosen different channels during a data phase, a sender can only deliver a fraction of packets to the subset of receivers on the same channel, resulting in further loss of efficiency.

For Common Hopping, each device cannot keep track of which other devices are busy. When $\delta = 1$, the receiver is always available when a sender successfully wins contention on the common hop since each receiver talks to only 1 sender. However, when δ is large, the probability that any particular receiver is available approaches $\frac{N-2k-1}{N-1}$ where k is the number of busy device pairs. This probability is small when the fraction of devices that are busy is large (i.e., high utilization and a small total number of devices compared to the number of channels). As a result, the last few channels are more difficult to be used efficiently.

For McMAC, the cause of degradation is similar to that of Common Hopping. However, since it allows parallel rendezvous, McMAC suffers less than Common Hopping when channel agreement traffic increases.

5.2.2 *Insensitivity of Dedicated Channel*

Fig. 6 (802.11b) shows Dedicated Control Channel is insensitive to the degree of communication because it knows the channel and device busy status perfectly by using a second radio. Since one of the 3 channels is set aside for control communication, the amount of rendezvous traffic is easily absorbed by the abundance of control channel capacity. As the number of available channels increases, the cost of using an extra channel for control purposes is relatively small, compared to the gain in knowing the channel and busy status of devices. Consequently, Dedicated Control Channel achieves the highest throughput except when $\delta = 1$.

5.2.3 *Delay Curves of Dedicated Channel*

The delay curves are not convex under the 802.11a settings and this requires some explanation. We also comment on the relationship between δ and the delay. The average delay in this case can be viewed as a sum of two stages. First, a pair of devices must rendezvous on the control channel. Second, they transfer one or more packets on the agreed data channel. The two stages saturates under different conditions.

The rendezvous process can generate at most $\frac{1}{e}$ or 0.3679 new agreement during every slot, on average. (We used $p_{succ} = e$.) In the 802.11a scenario, each 1024-byte packet lasts roughly 7 time slots. To fully utilize the 12 channels, the rendezvous process must generate $12/7 > 1$ new channel agreements per time slot which is impossible without sending multiple packets per rendezvous. Therefore, the control channel will saturate before the data channels do as the load increases.

When the control channel saturates at a load of roughly $\frac{1}{e}/\frac{12}{7}$ or 21.46%, the throughput is (0.2146×6 Mbps/channel $\times 12$ channels) or 15.45 Mbps. Therefore, at a load under 15.45 Mbps, the delay is very low because both stages are under-utilized. Once the first stage starts to saturate, the delay increases quickly for the first time. This explains the first jump in the delay at around 15 Mbps.

As the load increases, the number of queues waiting to be serviced in stage 1 remains constant due to the fixed traffic pattern, the average service delay in stage 1 is relatively insensitive to load. Since the total delay is dominated by the stage 1 delay before the second stage saturates, the delay rises very slowly. Finally, when the load approaches the capacity of the data channels, the second stage saturates, and hence the delay increases rapidly again.

Next, we note that when only the stage 1 has saturated, the delay at any particular load is roughly proportional to δ. For instance, the delays at 40 Mbps for $\delta = 1, 2, 5$ are in the ratios of roughly 1:2:5. Once the first stage saturates, one can view it as a random scheduler that serves the $N \times \delta$ always-backlogged queue pairs at a rate of $1/e$ in an i.i.d. fashion. The arrival rate of the second stage is equivalent to the output rate of the first stage. The service time of the second stage represents the time required to empty the sender's queue. The average amount of data found in a queue is proportional to the delay of stage 1 and inversely proportional to δ. Since the delay of stage 1 is proportional to δ, the service time of stage 2 is roughly independent of δ.

In summary, as δ increases, stage 1 slows down proportionally, while stage 2 remains at about the same rate, resulting in a delay that is proportional to δ.

5.3 Impact of Medium Occupancy Time Limit

In this subsection, we evaluate the delay performance using a mixture of long-lived CBR flows and some random file transfers. Allowing a device to transmit until the queue becomes empty can help achieve a higher throughput because more data can be transferred per channel agreement. However, the delay and/or jitter experienced by CBR flows might be worse.

We randomly pair up each device with another and add a CBR connection between them. The CBR connections have the same data rate, and sum up to 10% of the channel utilization. Then we start random i.i.d. file transfers with geometrically distributed lengths among every pair. The mean file size is 10KB. The arrival rate of the files is adjusted to give a total offered load between 10% (i.e., no file traffic) to 90% (i.e., 80% file traffic). MAC protocols send types of packets equally in a first-come-first-served order for the same destination, and randomly across different destinations.

We ran the simulation experiments with two sets of network parameters. However, due to space constraints and the similarity between the two, we present only the results pertaining to the 802.11a scenario. Fig. 8 shows the delay experienced by CBR traffic under 802.11a with occupancy

Parameter Investigated	Dedicated Control Channel	Split Phase	Common Hopping	McMAC
Ability to Use Many Channels	Good for long pkts Limited for short pkts	Limited	Limited	Good
Sensitivity to Packet Len (Lower is better)	High if many avail. channels Low if few avail. channels	High	High	Medium
Sensitivity to Channel Switching Time (Lower is better)	Low	Low	Very High	High
Sensitivity to Receiver Contention (Lower is better)	Low	Medium	High	Medium

Table 1: Summary of comparison of four representative protocols under different operating conditions.

limits of 3.3 ms (top) and 10.5 ms (bottom) respectively. The jitter curves follow the same trend as the delay ones, and are omitted.

In all cases, the increase in the medium occupancy limit reduces the delay for the CBR traffic. Furthermore, the standard deviation of delay (not shown) is almost identical in value to the delay. In short, both delay and jitter are lower as one increases the medium occupancy limit. Allowing devices to occupy the medium longer not only benefits the file transfer traffic, but also the CBR traffic by reducing delay and jitter for all.

Simulation results show that Split Phase does not benefit from a longer medium occupancy limit. As we saw in Fig. 6 and Fig. 7, when the traffic is non-disjoint, the performance of Split Phase is severely impacted because a device can only communicate with a small subset of others during each data phase. Due to the small amount of traffic sent to each destination, a longer occupancy limit is not useful.

Common Hopping improves tremendously as one increases the occupancy limits indicating that the rendezvous process is indeed a bottleneck. McMAC improves moderately but not drastically because the reduction in rendezvous traffic has less effect on parallel rendezvous protocols where rendezvous is less of a bottleneck.

The throughput of Dedicated Control Channel increases dramatically as the medium occupancy limit increases under the 802.11a settings with 12 channels because the rendezvous is indeed the bottleneck. It improves only a little in the 802.11b case with 3 channels (not shown) because rendezvous is not the bottleneck.

6. CONCLUSIONS

We classified various multi-channel MAC protocols into 4 generalized categories: Dedicated Control Channel, Common Hopping, Split Phase, and McMAC. We then compared them using analysis and simulation. We developed analytical models for the four protocols using Markov chains and simulated them in a time slotted simulator under a variety of operating conditions. Tab.1 summarizes our findings.

Overall, when channels are numerous and packets are short, multiple rendezvous protocols perform better than single rendezvous protocols by eliminating the control channel bottleneck. When traffic is non-disjoint, all protocols except Dedicated Control Channel, which use only 1 radio cannot monitor channel and neighbor busy status perfectly, thereby reducing achievable throughput. Also note that separating control packets from data packets does not improve performance because when the data channels are congested, generating more successful rendezvous is useless. Finally, we found that when MAC protocols treat various packet types (e.g., real-time video and file transfers) equally, allowing a sender to transmit multiple packets to the same destination after each rendezvous is highly beneficial. The throughput, delay, and jitter improve for all types of packets, using any of the 4 protocols.

7. REFERENCES

[1] P. Bahl and R. Chandra and J. Dunagan, "SSCH: Slotted Seeded Channel Hopping for Capacity Improvement in IEEE 802.11 Ad-Hoc Wireless Networks", In Proc. of Mobicom 2004.

[2] J. Chen and S. Sheu and C. Yang, "A new multichannel access protocol for IEEE 802.11 ad hoc wireless LANs," In Proc. of PIMRC 2003, Vol. 3, pp. 2291 - 2296

[3] F. Herzel, G. Fischer, and H. Gustat, "An Integrated CMOS RF Synthesizer for 802.11a Wireless LAN", IEEE Journal of Solid-state Circuits, 18(10), October 2003. IETF RFC 3561, July 2003.

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

[5] H. W. So, J. Walrand, "McMAC: A Multi-Channel MAC Proposal for Ad-Hoc Wireless Networks", Technical Report, April 2005.

[6] Jungmin So and Nitin Vaidya, "Multi-Channel MAC for Ad Hoc Networks: Handling Multi-Channel Hidden Terminals Using A Single Transceiver," ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc), May 2004.

[7] Z. Tang and J. Garcia-Luna-Aceves, "Hop Reservation Multiple Access (HRMA) for Multichannel Packet Radio Networks," In Proc. of the IEEE IC3N'98, Seventh International Conference on Computer Communications and Networks, 1998.

[8] A. Tzamaloukas and J.J. Garcia-Luna-Aceves, "Channel-Hopping Multiple Access ", Proc. IEEE ICC 2000, New Orleans, Louisiana, June 18-22, 2000.

[9] A. Tzamaloukas and J. Garcia-Luna-Aceves, "Channel-hopping multiple access with packet trains for ad hoc networks," In In Proc. IEEE Device Multimedia Communications (MoMuC '00), Tokyo, 2000.

[10] Shih-Lin Wu, Chih-Yu Lin, Yu-Chee Tseng, and Jang-Ping Sheu. "A 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 (ISPAN '00), page 232, Dallas/Richardson, Texas, USA, December 2000.

[11] Shih-Lin Wu, Chih-Yu Lin, Yu-Chee Tseng, Chih-Yu Lin and Jang-Ping Sheu. "A Multi-Channel MAC protocol with Power Control for Multi-hop Mobile Ad-hoc Networks," The Computer Journal, Vol. 45(1), page 101-110, 2002.

[12] Wing-Chung Hung, K.L. Eddie Law, and A. Leon-Garcia, "A Dynamic Multi-Channel MAC for Ad-Hoc LAN," In Proc. 21st Biennial Symposium on Communications, pages 31-35, Kingston, Canada, June 2002.

[13] Markus Zannoth, Thomas Rhlicke, and Bernd-Ulrich Klepser, "A Highly Integrated Dual-Band Multimode Wireless LAN Transceiver," IEEE Journal of Solid-State Circuits, vol. 39, no. 7, July 2004, pp. 1191 - 1195.

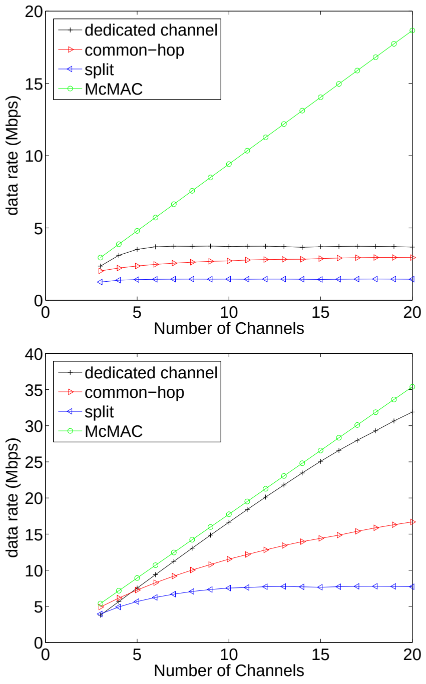


Figure 3: Throughput vs. no. of channels. Avg. packet size is 1Kbyte (top) or 10Kbytes (bottom).

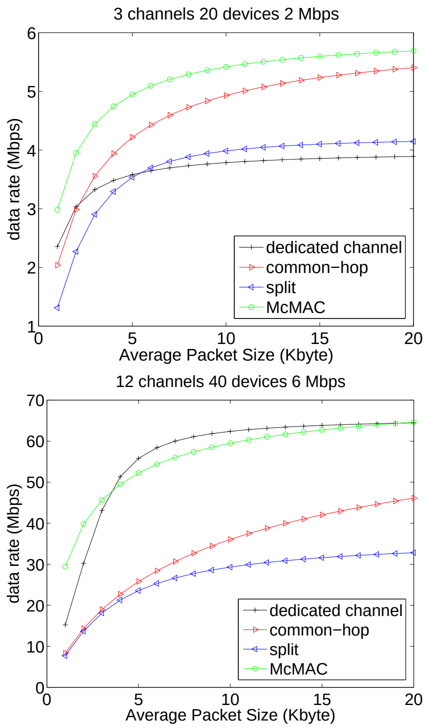


Figure 4: Throughput vs. packet size; Slot Time = 812μs (Top) and 200μs (Bottom); Switching Time = 100μs.

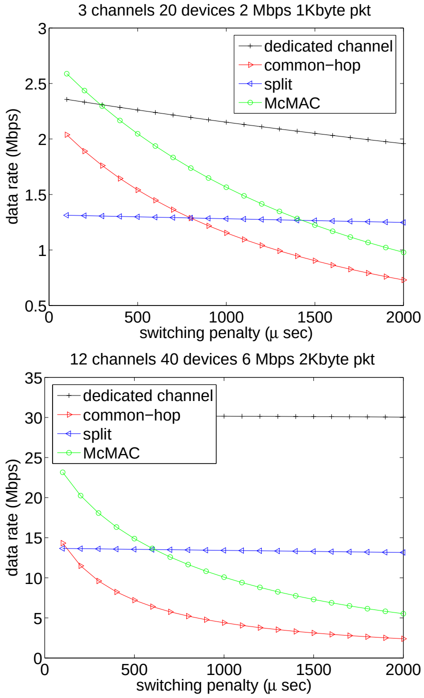


Figure 5: Throughput vs. Switching Penalty: Slot Time = 812μs (Top) and 200μs (Bottom).

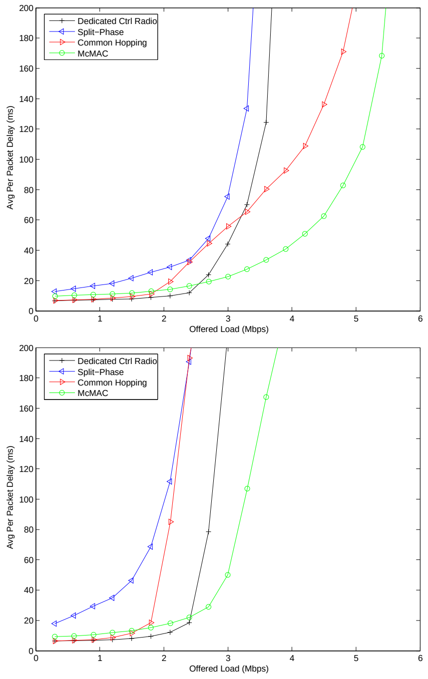


Figure 6: Avg per packet delay vs. CBR traffic load. Top: $\delta = 1$; Bottom: $\delta = 5$; 802.11b settings.

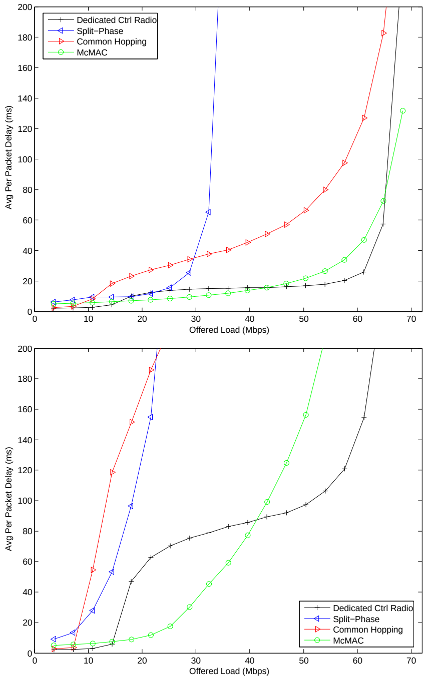


Figure 7: Avg. per packet delay vs. CBR traffic load. Top: $\delta = 1$; Bottom: $\delta = 5$; 802.11a settings.

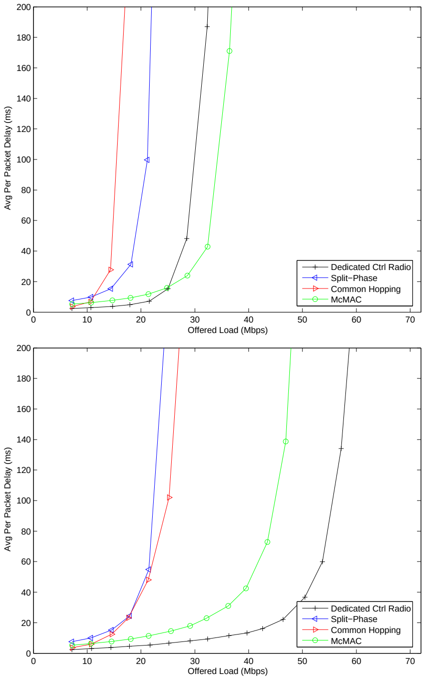


Figure 8: Effects of medium occupancy limit under the 802.11a scenario. The senders have to relinquish the medium after 3.3ms (top) vs 10.5ms (bottom).