Adaptive Channelization For High Data Rate Wireless Networks

Ritesh Maheshwari* Jing Cao† Anand Prabhu Subramanian‡ Fatima Zarinni* Samir R Das*

* Stony Brook University, USA † Beihang University, China ‡ Bell Labs, USA

Abstract

High data rate wireless networks (1 Gbps and up) are in the horizon and several standards are in the works. However, the task of designing multiple access protocols for such networks is fraught with new challenges as the bandwidth independent overheads dominate. We show that even when such overheads are kept at a minimum, the performance of multiple access protocols can be very poor. However, performance can be improved significantly by splitting the given bandwidth into multiple channels and running the multiple access protocol independently on these channels. Taking an 802.11-like CSMA/CA (DCF) protocol as an example we show via a modeling exercise how such channelization can improve performance and why it needs to be *adaptive* to traffic demand. We develop an Adaptive Multichannel (AMC) protocol and study its performance via simulations.

In addition, we also investigate a single-channel *Extended-Reservation* protocol in a high speed setting. Here, a sender, upon winning the contention, reserves the channel for multiple packet transmissions. We show that the Extended-Reservation protocol performs better than the single channel 802.11-like DCF protocol and is comparable to adaptive multichannel protocol (AMC), but it suffers from an inherent unfairness issue due to which many nodes starve for channel access. The multichannel protocol, on the other hand, is devoid of any fairness problems.

Finally, we develop a 'scaled down' prototype implementation using the USRP/GNURadio platform to demonstrate that adaptive channelization can be practical using appropriate programmable radio hardware and has tremendous performance potential. Taking our modeling, simulation and experimental results together, our work shows that a throughput gain of a factor of 2 is not unrealistic.

I. Introduction

At the beginning of this decade, many regulatory authorities worldwide (including FCC in US) set aside a large swath of spectrum (7 GHz wide) in the 60 GHz band for unlicensed use. This has promoted a large number of innovations in developing very high data rate (1 Gbps or above) wireless link technologies for the local area. For instance, the upcoming WirelessHD standard [5] can use up to 4 Gbps links over short distances (10m) for high-definition audio/video applications. The link technology actually allows up to 25 Gbps. The upcoming 802.11VHT standard [3] (VHT stands for very high throughput) is expected to provide at least 1 Gbps data rate and is expected to go over much longer distances.

As one goes for high data rates as above, the efficiency of the MAC protocol reduces. This happens due to the following reason. The per-packet MAC protocol overhead can be broadly classified in two parts – *bandwidth independent* and *bandwidth dependent* [26], [28]. The bandwidth independent part slowly becomes substantial as one improves the physical layer data rate. For the same packet length in bits, the transmission time of the packet and the bandwidth dependent overhead reduce proportionately with physical layer data rate; however, the bandwidth independent part stays the same. We will show later that the efficiency of a 802.11 like MAC protocol can easily be limited to only 50% with just 5 nodes if the data rate exceeds 1 Gbps. While this issue has apparently been noticed in simulation exercises in the 802.11VHT group [3], the research community is yet to undertake the challenge of developing efficient random access protocols for the very high speed regime. This issue is not limited to CSMA or 802.11-like protocols alone and can happen for TDMA protocols as well. However, we will limit discussions for CSMA only for its clear suitability for data networks.

The goal of our work is to find a mechanism to improve MAC layer efficiency in the very high speed regime. For this purpose, we investigate two potential solutions, namely, an adaptive multichannel approach and an extended-reservation-based approach. We describe briefly below, why these two approaches allow for a better channel utilization in high data rate networks:

A. Adaptive Multichannel approach:

The basic idea is to split the available single-channel bandwidth into multiple smaller channels. Each individual channel now has a smaller bandwidth supporting a proportionately slower data rate. This helps mask the bandwidth independent overhead. We will demonstrate this with analysis in Section III. However, such channel splitting carries its own overhead as guard bands must be used.

Also, such channelization must be adaptive to the traffic demand. For example, a smaller (larger) number of channels may be appropriate when a small (large) number of nodes are active or when traffic demand is low (high). The challenge is to adapt the system appropriately to ensure an optimum operating point at all times.

In some ways, adaptive channelization that we propose is reminiscent of the subcarrier allocation problem in OFDMA [30]. However, in OFDMA a centralized entity (base station) maps the available set of OFDM subcarriers into a set of sub-channels to be allocated to the active links. The mapping is done based on channel state information and traffic on the links to improve the overall spectral efficiency and is renewed periodically in a TDM fashion. In contrast, our goal is to develop an entirely distributed random access model agnostic to the physical layer. The goal is to optimize channel access efficiency in presence of significant bandwidth independent overhead.

B. Extended-Reservation approach:

The Extended-Reservation approach is similar to the IEEE 802.11e standard's 'Transmission Opportunity' (TxOp) protocol [1]. Here, a sender contends for the channel in the same way as 802.11 DCF, but after winning the channel, instead of transmitting only one packet, the sender can now transmit a maximum of L packets, back-to-back, with SIFS period separating the packets.

Here we refer to L as the *reservation limit*. The period of time that the back-to-back transmission of L packets will take is called the *reservation period*, and this period starts when a node gains access to the channel. A node cannot hold the channel longer than the reservation period. Once the reservation period is over, the sender reliquinshes control of the channel and goes into a random backoff before attempting to transmit the next packet.

This protocol can improve performance in high data rate networks, since once a node wins the channel, a burst of data can be sent, which has a similar effect of transmitting packets of a longer transmission time, but with the benefit of reduced bit error rate. Unlike the DCF protocol, where there is a random backoff before every packet transmission, the Extended-Reservation protocol (when $L > 1$) incurs only one backoff period before L packets, thus amortizing the bandwidth-independent overhead of backoff over multiple transmissions.

Though the Extended-Reservation protocol appears to be a good candidate for high data rate networks in terms of performance, it is obvious that for large values of L, fairness issues arise.

Our main contributions are as follows. First, we show via analytical modeling that single channel MAC protocol is very inefficient in high data rate networks and both channelization and the Extended-Reservation protocol can provide improvement in performance. (Sections III uptil III-E).

Second, we show via modeling that if we can adaptively channelize the spectrum based on traffic, then the Extended-Reservation protocol performs poorer than the adaptive channelization technique. We also show via simulations that for the cases where the Extended-Reservation protocol gives better performance, it suffers from serious fairness issues, however such a tradeoff between fairness and throughput does not exist with channelization. (Section III-F)

Third, we develop an adaptive channelization protocol and show via simulations that just channelization is not enough for better performance; channelization also needs to be adapted with varying traffic conditions (Sections IV, V, VI).

Finally, via a 'scaled down' prototype implementation on GNU Radio/USRP platform, we emulate the operation of a high-speed network and show such adaptive channelization can indeed be realized in practice (Sections VII and 9).

Related works and conclusions appear in Sections 2 and 10 respectively.

II. Related Works

Several approaches dynamically allocate variable amount of spectrum in cognitive radio based networks. For example, the KNOWS system [31] develops distributed allocation techniques for contiguous time-spectrum blocks to maximize the use of fragmented spectrum [32]. Spectrum is allocated based on pre-determined traffic demands, interference criterion, and bandwidth allocated to interfering transmissions. Similar spectrum allocation problems are also considered in [25], but in the context of cellular dynamic spectrum access networks, and are solved in centralized fashion.

Independent of cognitive radio or dynamic spectrum access, several works in current literature study adapting channel width dynamically to improve different performance measures. In [13], the authors make a case for adapting the channel width in wireless networks using 802.11 networks as a case study. However, they primarily study the impact of adapting channel width on data rate, power consumption and communication range on a single wireless link. In [19], the authors study a spectrum distribution problem in the context of 802.11 WLANs by providing wider channels to the more congested APs and smaller channels to less congested ones. The approach is aimed more towards load balancing than addressing MAC protocol overheads and can serve as complimentary to our approach.

A host of multichannel MAC protocols exist in literature [7], [9], [16], [17], [23], [24], [27]. But they are all geared towards networks with pre-configured or fixed channelization (such as in IEEE 802.11). In general, their goal is to efficiently utilize the channels by appropriately assigning interfering links to channels. In contrast, our goal here is to develop a basic MAC protocol which adaptively selects the number of channels itself. The task of assigning links to channels can be done by using any of the available protocols. In this work though, we have not directly used any of these protocols. There is also a significant amount of literature on channel assignments for multi-radio, multichannel networks (e.g., [21]), as well as channel selection for TDMA scheduling (e.g., [6]). Again, they are orthogonal to our work and we do not discuss them here.

The fact that channelization can improve MAC protocol efficiency was observed originally in [18] in the context of Ethernet. The authors in [26], [28] noticed the impact of bandwidth independent overheads on MAC protocol efficiency.

III. CASE FOR CHANNELIZATION

In an 802.11-like CSMA/CA protocol, the major bandwidth independent overhead is the backoff time that is counted in terms of slots. The slot size must be at least the sum of maximum propagation time between two nodes, the carrier sense interval and the transmit-receive turnaround time. If there is a non-negligible time synchronization error, it must be accounted for in the slot size as well. Thus, the slot size has a lower bound that is independent of data rates. The analysis in the following subsection shows the impact of a constant slot size on MAC efficiency at high data rates.

A. Why Does Single Channel Work Poorly?

We start with the widely used model of 802.11 developed by Bianchi in [10]. It assumes a single collision domain (single hop network) and ideal channel conditions (perfect carrier sensing and no capture) with the network under saturated load. According to this model, in steady state, nodes transmit in an arbitrarily chosen time slot with probability τ. This probability τ can be computed by numerical means given the number of contenders (n), contention window and the maximum backoff stage in the binary exponential backoff scheme.

Assume as in [10], P_{tr} is the probability that there is at least one transmission in a slot. Thus,

$$P_{tr} = 1 - (1-\tau)^n. \tag{1}$$

The probability P_s that a transmission occurring on the channel is successful is given by,

$$P_s = \frac{n\tau(1-\tau)^{n-1}}{P_{tr}} = \frac{n\tau(1-\tau)^{n-1}}{1-(1-\tau)^n}. \tag{2}$$

Now, assume that the slot time is σ and the packet time is T_p. Ignore all interframe spacings and header overheads. Consider the basic access with no RTS/CTS or ACK[1]. By a straightforward application of renewal theory, the normalized throughput or the long run fraction of time spent in successful transmissions is given by,

$$S = \frac{P_{tr}P_sT_p}{(1-P_{tr})\sigma + P_{tr}T_p}. \tag{3}$$

Note that the packet time in slots (T_p/σ) is an influential determinant of throughput. Smaller values mean lower throughput. It is easily seen that for higher physical layer speeds T_p/σ will tend to be smaller. This is because slot size σ has a lower bound that is independent of speed, as we discussed before. For example, the propagation time of RF signals at 150m is 0.5μs. Carrier sense interval can easily add another 0.5μs, assuming that the fastest available A/D chips can digitize 512 Msamples/sec and 256 samples are used to sense carrier. Thus 1μs can serve as a lower bound on the slot size. For a 1000 byte packet, the packet time is 8μs for a 1GBps link ,and 0.8μs for a 10Gbps link. This gives us (T_p/σ) = 8 and 0.8, respectively. Note that these numbers are very conservative. We have assumed a larger than average packet size.

[1]These can be added but they generate distracting details

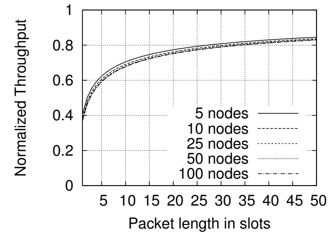


Fig. 1. Normalized throughput versus packet time (in slots) for a single channel 802.11-like network. Optimal contention window is assumed.

We plot throughput versus packet time for various number of nodes in Figure 1 as per Equation 3. For this plot, the minimum contention window size W is assumed to be optimal for the number of nodes and the optimal value is used to generate this plot. The optimal is computed via straightforward numerical techniques by computing the throughput for different values of W and choosing the optimal for presenting in the plot. For unoptimized W (such as in 802.11) the performance is likely to be worse. Still, we note very poor throughput for a very realistic range of packet times in our context. Packet times 1-5 slots mean efficiency between 0.4-0.6 even with an optimized window.[2]

Of course, efficiency improves with larger packet sizes. But packet sizes cannot be increased arbitrarily as packet error rates will increase for a given bit error rate in the underlying wireless channel, for a given SINR, modulation and coding. Also, from a more practical point of view packet coalescing to increase packet size may not be possible depending on packet generation/forwarding rates from the upper layer.

B. Modeling Multichannel Benefit

To see the benefit of splitting the channel up into multiple subchannels, assume that the given channel is divided into k smaller channels of the equal bandwidth. The above model can now be modified to compute the resulting throughput. Assume for simplicity that transmitters choose channels randomly for transmission and then contend on that chosen channel. Thus, on average there are now n/k nodes competing in each channel. The transmission probability, P_{tr}, becomes

$$P_{tr}(k) = 1 - (1-\tau)^{n/k}. \tag{4}$$

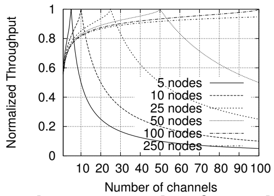
The successful transmission probability, P_s, becomes

$$P_s(k) = \frac{(n/k)\tau(1-\tau)^{(n/k)-1}}{P_{tr}(k)}. \tag{5}$$

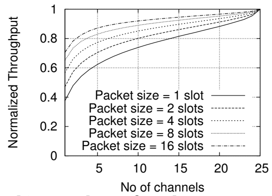
The packet time T_p is now longer as each channel now has a factor k smaller bandwidth. Thus, the packet time is kT_p, and the normalized throughput $S(k)$ of each channel is given by,

$$S(k) = \frac{P_{tr}(k)\ P_s(k)\ kT_p}{(1-P_{tr}(k))\sigma + P_{tr}(k)\ kT_p}. \tag{6}$$

All channels being identical, the aggregated normalized throughput in all k channels in also $S(k)$.



(a) Packet time = 4 slots for single channel.



(b) Number of nodes = 25.

Fig. 2. Normalized throughput of a 802.11-like network in a multichannel setting. Single collision domain and implicit ACK are assumed. Optimal contention window (for number of nodes per channel) is assumed for a fair comparison.

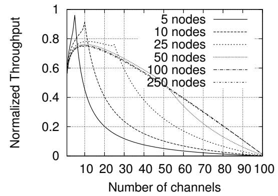
C. Results

In Figure 2 we present the throughput $S(k)$ as a function of the number of channels k, packet time in slots (T_p/σ) and number of nodes n. As before the range of packet time in slots has been been chosen carefully to reflect the realistic values possible in high data rate networks. We have also been careful with the choice of minimum contention window size W. *To clearly demonstrate the multichannel advantage we have used the optimal W for each choice of n and k pairs.* While the optimal may not be achievable in a real protocol or it may require complex estimation or adaptation that may be expensive [29], from the perspective of the analytical modeling this makes the fairest demonstration of the performance benefit of multichannel. To see this, assume that the single channel case is already sub-optimal because of a poor choice of W (assume, smaller than optimal). Now when we split the channel but not modify the contention window, we may get closer to the optimal as contention reduces due to channel splitting. Thus, it will be unclear how much benefit is due to channel splitting and how much due to a more suitable contention window with split channel as opposed to single channel. If we choose the optimal in all circumstances, the comparison is clearer.

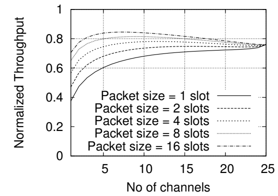
In Figure 2 we have plotted the normalized throughput $S(k)$ as per Equation 6 versus number of channels k. In Figure 2(a), the packet time is fixed at 4 slots and the number of nodes is varied. The throughput reaches optimal (100%) when the number of channels is equal to the number of nodes, as there is no contention and the optimum window size is 0. On the left of the optimal point, there are more nodes and less channels, thus throughput suffers due to contention. On the right of the optimal point, there are more channels and less nodes, so throughput suffers as many channels remain unused.

Note very poor efficiency for single channel even with an optimum contention window (about 0.56). In Figure 2(b), the number of nodes is constant at 50 and the packet time and the number of channels are varied. Note again that for small packet time, the single channel efficiency is very poor. For example, for small packets (1 slot), it is about 0.38. However, efficiency rapidly increases with increase in number of channels. For example, the efficiency increases by 50% with just 5 channels and doubles with 20 channels. The rate of increase in efficiency tapers off with larger number of channels. This means that just a handful of channels can make a significant performance impact.

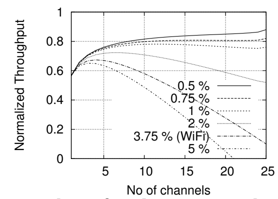
[2]In Figure 1 similar performance with different number of nodes is not surprising. It is an artifact of the use of optimal window. Figure 9 in [10] also has a similar observation.



(a) Packet time = 4 slots for single channel; guard band = 1%.



(b) Number of nodes = 25; guard band 1%.



(c) Number of nodes = 25; packet time = 4 slots for single channel.

Fig. 3. Throughput vs. number of channels and packet size for different guard bands.

D. Guard Bands

Channelization, however, comes with an overhead. When we divide a spectrum of bandwidth B in to k channels each of width b, there should be enough guard band separation between the channels, so that concurrent transmissions are possible on on each channel without interference. In practice, due to the non-linearity of power amplifiers, radio leakage occurs on each channel. The amount of this leakage determines the guard band separation needed between two adjacent channels.

The guard band size depends on a variety of factors including the physical layer technology, radio design, the transmit power spectral density, the channel width, the SNR threshold and the minimum separation between nodes in the network. Nonetheless, it is possible to design conservative guard bands such that their size does not depend on the data channel width. We use this conservative estimation scheme in the rest of the paper and express guard band size as a constant fraction of the total bandwidth. Specifically, each pair of channels of width b are separated by a guard band of size g. Thus, with k channels, $(k-1)g$ bandwidth is wasted in guard bands. Throughput $S(k)$ with guard bands can be computed by using Equation 6 except that now the packet time is slightly different. It is no longer kT_p, but $\frac{k \cdot B}{B-(k-1)g}T_p$.

In Figure 3, we show plots similar to Figure 2 except that a constant guard band (1% of the channel bandwidth B) is used throughout. An additional plot with varying guard band width is also presented. Note that the guard band indeed makes an impact on the performance with larger number of channels. For example, even with 1% guard band, 24% bandwidth is wasted for 25 channels. Unless guard band is too small, both too few and too many channels hurt performance. Thus, depending on the actual packet sizes the optimal number of channels are typically small (more in the order of 10 rather than 100). Note also from Figure 3(a) that the channel efficiency never reaches 100% due to the guard band wastage.

Note that, some radio technologies like OFDM [8] does not require guard bands between adjacent channels. In such cases, channelization will lead to much better performance benefit as shown in Section III.

E. Modeling the Extended-Reservation protocol

Next, we model another effective technique that provides better channel utilization than the 802.11 DCF in very high speed networks. The idea here is that once the sender node wins the channel, it can send a maximum of L packets back-to-back, (instead of just one packet), with SIFS period between the packets.

Note that this approach needs acknowledgement (ACK) packets to be sent by the receiver after every data packet sent by the sender. Without ACK packets after every data transmission, collisions cannot be detected and no remedial action can be taken. But if ACKs are used in the protocol, senders can detect collisions when ACKs do not arrive and they can then release the control of the channel to prevent any furthur wastage. Thus, we assume that ACKs are used in this protocol. More specifically, an ACK packet is sent by the receiver after every data packet reception.

The sender, after gaining access to the channel by winning a contention, reserves the channel for a maximum of L back-to-back Data/ACK handshakes. The sender sends the first packet, waits for SIFS to hear an ACK, if it receives an ACK, it sends the next packet, and this process continues until a maximum of L back-to-back Data/ACK handshakes have been accomplished. Here we define a *reservation period* as the time it takes for L back-to-back Data/ACK handshakes to be accomplished with SIFS between packets. The reservation period begins when the sender gains access to the channel.

If at any point within the reservation period, the sender, after sending a packet does not receive an ACK within SIFS period, then it assumes a collision and releases the channel immediately. The sender will also double its contention window, and contend for the channel again. Other nodes will be able to sense the channel idle for more than SIFS period (i.e.,DIFS), and will be able to continue counting down their backoff counters.

In order to model the Extended-Reservation protocol with ACKs enabled, we have again used a slight modification of the Bianchi's Model. The Saturated Normalized Throughput for this protocol, given a Reservation-Limit(L), is given by:

$$S_L = \frac{P_{tr}P_sLT_p}{(1-P_{tr})\sigma + P_{tr}P_sT_s + P_{tr}(1-P_s)T_c}. \quad (7)$$

Here, T_s is the average amount of time for which the channel is sensed busy after a successful transmisison starts, hence, $T_s = L(T_p + T_{Ack})$, where T_p is the transmission time for a packet and T_{Ack} is the time taken by an ACK. Note that $T_p + T_{Ack}$ is multiplied by L to get T_s , because, we are assuming ideal channel conditions and a single collision

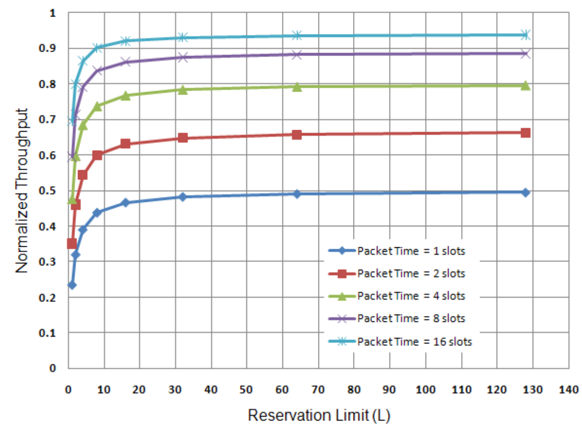


Fig. 4. Normalized throughput versus Reservation Limit for packets of different sizes. The number of nodes is 25 and optimal contention window is assumed.

domain. Hence, once a node starts a successful transmission it will be able to transmit all the L packets.

T_c stands for the the average amount of time that is wasted with collisions, and hence, $T_c = T_p + T_{Ack}$. Note that we are not multiplying L here to $T_p + T_{Ack}$. This is because, in a single collision domain, collisions can happen only at the first packet. Thus, after first packet gets collided, the collided senders will not send further packets in the reservation and release the channel. Only one Data/ACK exchange is wasted in collisions.

Note that P_{tr} and P_s are computed in the same way as before.

We use the above model to get some idea about the performance of the Extended-Reservation protocol. In all the results, the contention window has been optimized in order to find the true degree of performance benefit that is provided by the protocol. It is important to note that when the Reservation-Limit (L) is 1, then we have the same case as the ordinary 802.11 DCF protocol.

In Figure 4 we have evaluated the performance of the Extended-Reservation protocol for different Reservation-Limits and packet times. Here, the number nodes is fixed to 25, L is varied along the x-axis and the change in the normalized throughput, as L varies, is shown for packet transmission times of 1, 2, 4, 8 and 16 slots. Looking at the graph pertaining to packet time = 1 time slot, we can see that as L increases, the normalized throughput increases, and as L becomes very large, the normalized throughput reaches a limit of 0.5. In fact, for packet time of 1 slot, we have an upper bound of 50% channel utilization. This result is as expected, because as L becomes large, we will eventually have a single node reserving a channel and sending packets for a very long time, and since we have a packet time of 1 time slot, half of the long reservation period will be wasted in ACKs, and only half of the reservation period will be spent in useful transmissions. Note that here collisions are not going to cost much, since, if a collision happens then only two slots will be wasted - one for the packet transmission, the other for waiting to receive an ACK, (which the sender does not receive). As we increase L unboundedly, we will experience an approximately 28%

increase in performance for packet size of 1 time slot, when compared with the ordinary DCF protcol.

One very important thing to note here is that even though,the network throughput reaches the ideal throughput of the channel as L increases, the network suffers from serious fairness issues. Such high throughput is obtained by starving almost all the nodes except 1. (We will present results showing unfairness in the Extended-Reservation protocol in Section III-F2.)

Similarly, in Figure 4 with packet times of 2, 4, 8 and 16 we can see a similar behavior, and in all the cases, as L increases the throughput increases, and as L becomes very large, the throughput reaches the idealized throughput, however at the expense of fariness amongst nodes. Also it is observed that for a given L, as the packet size increases, the throughput also increases. This is because with larger packet sizes, the effect of the time spent in backoff is hidden by longer periods of useful transmssion after the backoff.

For all the packet sizes we can see that the Extended-Reservation protocol with L greater than 1, performs better than 802.11 DCF.

F. Extended-Reservation Protocol Vs. Adaptive Multichannel

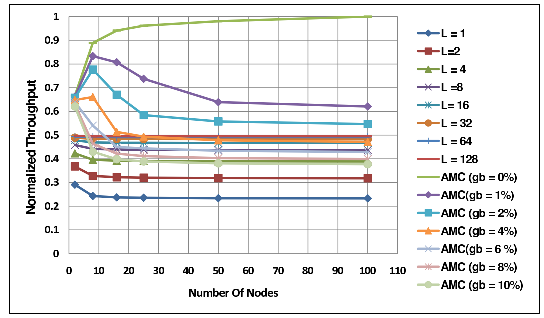
In this section we evaluate and compare the network performance of a protocol that will *adapt* the number of channels according to the traffic. With such a protocol, for a given number of nodes and packet size we will be able to split the single channel into that number of channels ($k*$) which will provide us the optimal throughput.

Recall that given a certain number of nodes, packet size and guardband, just splitting the channel into any number of channels does not necessarily give us the best possible throughput. If we split the avaliable bandwidth into less or more than $k*$ channels, then we will not be able to get the best possible throughput in the network. For example, from Figure 2, we can see that for 5 nodes, packet time of 4 time slots, and 0% guardband, we get the optimal throughput when we split the single channel in to 5 smaller channels of equal bandwith. However, 5 channels will not provide the optimal throughput if we have, for example, 10 nodes instead of 5. For 10 nodes and packet time of 4 slots, we get the optimal throughput if we split the channel into 10 smaller channels.

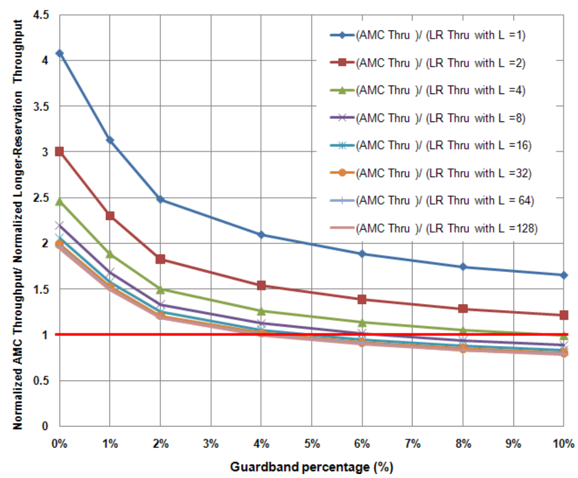
Note that in order to find the saturated normalized throughput of the AMC protocol, we use equation 6, however, now for a given number of nodes, guardband, and packet time in single channel, instead of k, we will use the corresponding optimal number of channels, $k*$. Moreover, we modify the equation slightly in order to account for ACKs and guardbands. Since the modification is trivial, it is not shown here.

In the discussion below, we will compare the AMC protocol and the Extended-Reservation protocol, both in terms of performance and in terms of fairness.

*1) **Throughput Comparision**:* In Figure 5(a), we are comparing the channel utilization of both the techniques. We vary the number of nodes along the x-axis, and we plot the throughput of the Extended-Reservation protocol with L being 1, 2, 4, 8, 16, 32, 64 and 128. We also plot the throughput of the the AMC protocol with guardbands of 0% uptil 10%. Here the packet time is 1 time slot, and optimal contention window is assumed in all cases.



(a) Normalized throughput for the Extended-Reservation and AMC protocols vs varying number of nodes.



(b) Ratio of normalized AMC throughput to normalized Extended-Reservation throughput vs varying guardbands. Number of nodes is fixed to 25.

Fig. 5. Performance Comparision between Extended-Reservation and AMC protocols. In both the figures the packet time is 1 time slot.

We see that the ordinary 802.11 DCF protocol (case of L =1), gives the worst normalized throughput, for all number of nodes, n. We can also see that as L increases, the throughput for the Extended-Reservation protocol increases,but reaches a limit of 0.5 for all n. (The reason for this has been explained before). Now, if we look at the AMC protocol, we can see that the AMC protocol with guardbands of 0 to 3 percent, even uptil 100 nodes, outperforms the Extended-Reservation protocol. (Even if the “Extended-Reservation” protocol is operating with very large Reservation-Limits (L), which causes the throughput obtained by this protocol to come close to the single channel’s ideal throughput).

The reason for this is that with the AMC protocol and small guardbands, we usually end *splitting* the channel into multiple channels, and it is done in such a way, that we not only have a smaller number of nodes competing in one subchannel, (which is a factor in the reduction of the backoff penalty), but also the number of channels is chosen in such a way, that the penalty incured by the guardbands is also kept at a minimum. Moreover, with small guardbands, the optimal number of channels for a given n is also more. If we compare this with the best case for the Extended-Reservation protocol, we are going to see that obviously, even for the case of two nodes, the Extended-Reservation protocol will do worse because with the Extended-Reservation protocol, every packet transmission is taking one time slot, and then we have 1 slot wasted due to an ACK. With the AMC protocol, and small guardbands, the packets will take a longer time to transmit, (proportionate to k*) and thus reducing the penalty incurred by ACKs. For example, with 0% guardband k* is usually equal to n, and thus, a separate channel is allocated for each sender. Since we are assuming optimal contention windows and saturated load, the senders always transmit, and they do not go into a random backoff. Note that for here we are not having any wastage due to guardbands. Here each packet is going to take n times longer time to transmit. However, we can see that we are not having a normalized throughput of 1.0 for all n, because we face ACKs overhead, but as n increases, we can see that the normalized throughput goes very close to 1.0, because larger n means more channels, and , hence proportionately longer packet times per channel, which mask the time taken by the ACKs.

We should also note that as the guardbands increase the AMC performance degrades, since now more portion of the bandwidth is not usable. Moreover, the optimal number of channels (k*), for a given number of contenders, decreases, as the size of the guardband increases. This will not only cause smaller packet times than network setting where k* is larger, but smaller k* would also cause more nodes to compete on a subchannel, which will inturn cause a larger contention window, and hence we will have more time spent in backoffs. Therefore, as the guardbands increase the performance of the AMC protocol degrades.

But still, we can see that the AMC protocol uptil guardbands of 4% performs better than the "Extended-Reservation" protocols with the Reservation Limits of uptil 16 packets. If the Reservation-Limits go beyond 16, then we can see that the Extended-Reservation protocol outperforms the AMC protocols of more than 4% guradbands. Moreover, with 10% guardbands, the AMC protocol performs better than the Longer Reservation protocol with the Rervation Limit of 1 and 2 data/Ack handshakes. However, with Reservation-Limits of 4 or higher, the Extended-Reservation protocol performs better than AMC protocol with guardbands of 10% or higher.

In Figure 5(b) the number of nodes have been fixed to 25, and the packet time in single channel is assumed to be 1. We find the ratio of the throughput of the AMC protocol, for a given guardband, to the throughput of the Extended-Reservation protocol, for a given L, in order to observe that uptil what point does the AMC protocol perform better than the Extended-Reservation protocol. Each of the curves is realated to a separate Reservation Limit. If the curve is above the horizontal line crossing at 1.0, then the AMC protocol is performing better, but if the curve goes below the red line, then the Extended-Reservation protocol with the associated L, is performing better. We can see that uptil a guardband of 4%, the AMC protocol is always performing better than the Extended-Reservation protocol, even when L is very large. We can see that when AMC protocol operates on a network with huge guardband (uptil 10%), then it still performs better than Extended-Reservation protocols with Reservation Limits of uptil 4, but performs worse than Reservation-Limits greater than 4.

Thus, we can see that the AMC protocol with small guardbands perform better than even the best case of the Extended-Reservation protocol. However, with guardbands of 4% and beyond, we can see that we can find Reservation-Limits, with which the Extended-Reservation protocol would perform better. However, now we are going to show that usually the Extended-Reservation protocol faces fairness issues, but the same type of problem is not seen in the AMC protocol.

*2) **Fairness Comparision**:* In the subsection III-F1, we saw that the Extended-Reservation protocol reaches the ideal single channel throughput as L grows large. We also showed that with guardbands of 4% , the Extended-Reservation protocol with Reservation-Limit of 16 or more, will outperform the AMC protocol. However, now we are going to show that this higher throughput of the "Extended-Reservation" protocol comes at the expense of reducing fairness in the network. If L becomes very large, then 1 node is going to occupy the channel for a very long period of time and starve all other nodes, despite the fact that other nodes have an equal priority to get a share of the network. Such a tradeoff between throughput and fairness is not experienced by the AMC protocol even with guardbands of uptil 10%.

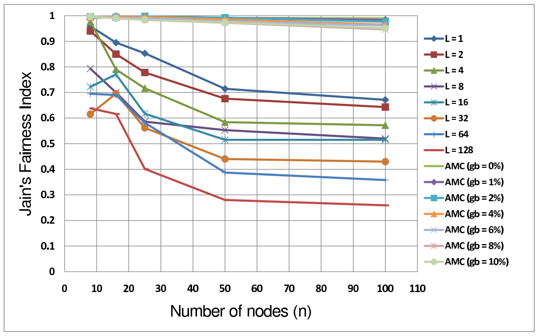
In order to study fairness, we developed a smiulator for the Extended-Reservation protcol. We also developed a simulator for the AMC protocol. (Please see Sections IV and V for details). We are assuming an optimal contention window and packet time of 1 time slot in the single channel setting. We run both the simulators for an equal amount of time and under saturated load.

In Figure 6(a), we vary the number of nodes (n) along the x-axis and we plot the Jain's Fairness Index for the Longer Reservation Protocol with the Reservation-Limit (L)being 1, 2, 4, 8, 16, 32 and 128. We also represent the Jain's Fairness Index for the AMC protocol with guradbands of 0,1,2,4,6,8 and 10 percent. The Jain's Fairness Index [20] is defined below:

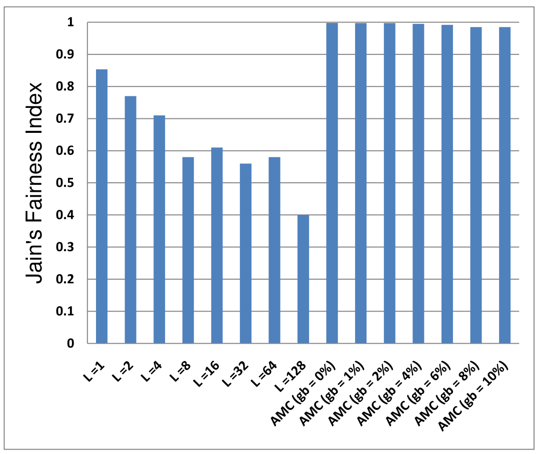
$$Jain'sFairnessIndex = \frac{(\sum_{alli} x_i)^2}{n \sum_{alli} x_i^2}. \tag{8}$$

Here, x_i stands for the throughput of node i, and n stands for the total number of the nodes in the network.

If the Fairness Index is close to 1, then the network is in the best state in terms of fairness. Now, if we look at Figure 6(a) we can see that for the AMC protocol, for uptil 100 nodes, and for all guardband percentages, the Fairness Index is close to 1.



(a) Jain's Fairness Index for the Extended-Reservation protocol and AMC protocol Vs. varying number of nodes. The Extended-Reservation protocol is evaluated for L =1,2,4,8,16,32,64 and 128. The AMC protocol is evauated for guardbands of 1,2,4,6,8 and 10 percent



(b) Jain's fairness index for the Extended-Reservation protocol and the AMC protocol for the case of 25 nodes.

Fig. 6. Fairness Comparision between Extended-Reservation and AMC protocols. In both the figures the packet time is 1 time slot.

This means that even uptil 100 nodes and huge guardbands, every node is given an equal share of the network. If we look at the Extended-Reservation protocol, we can see that as L grows large, the Fairness Index drops. In fact, for any L greater than 1, the fairness in the network becomes poorer than the ordinary 802.11 DCF. This happens because, as L grows large, one node transmits packets for even a longer period of time, despite that there are other nodes in the network with the same priority that are waiting to transmit their packets. We can see that when L becomes very large, (and we acheive a high throughput close to the ideal single channel throughput as shown in Figure 5(a)), the fairness index comes closer to the worst case of $1/n$, for a given n. This happens because, as L grows very large, there is just one node that is transmitting, but other nodes starve.

Moreover, we can see that as the number of nodes increase, for a given 'L', the fairness decreases also. The reason for this is that now we have more nodes wanting a share in the network, but one node occupying the channel for 'L' data/Ack handshakes.

Such type of unfairness, cannot be seen with the AMC protocol even with large guradband sizes. If we look at the AMC protocol with 0% guardband, as the number of nodes increase, the fairness index remains approximately constant. The reason for this is that with 0% guardband the optimal number of channels is equal to n, and in average one node is assigned a seperate channel. Hence, all nodes get an equal share of the bandwidth. With guardbands greater than 0%, we can see that as n increases the Fairness Index decreases by a slight amount. The reason for this is that, due to the limitation imposed by the guardbands, we cannot have a channel per sender as n grows large. Therefore, with large n, we have several nodes assigned to the same channel, which causes contention, and we get a slight reduction in fairness. Despite this, we can see that uptil 100 nodes and 10% guradband the AMC protocol maintains a fairness index of more than approximately 0.95.

In Figure 6(b) we have fixed the number of nodes to 25, and we have bar graphs that represent the Jain's Fairness Index for the Extended-Reservation protocol and AMC protocol, with different Reservation-Limits and guardband sizes, respectively. As, we can see for all the AMC cases, the fairness index is close to 1.0. Which means that the AMC protocol provides an equal share of the bandwidth to all nodes. However, with the Extended-Reservation protocol, the index value drops as L increses. For example, when L = 4, we see that approximately 30% of the nodes suffer from unfairness, and when L = 128 we see that 60% of the nodes suffer from unfairness.

From Figure 5(b) we can see that for the case of 25 nodes, L needs to be at least greater than 4, in order to provide a better throughput than the AMC protocol with greater than 4% guardband. However, it is undesirable to use the Extended-Reservation protocol, with L greater than or equal to 4 here, since as shown in Figure 6(b), at least 30% of the nodes will suffer from unfairness.

Going back to figure 6(a), we can see that when the number of nodes is 50 or more, then even the Extended-Reservation protocol, with small Reservation-Limits become unusable if we are concerned about fairness. We can see that with L =1, and the 50 nodes case, we will have 30% of the nodes facing unfairness, and with L=4 we have 41%, and with L = 128 we can see that 72% of the network face unfairness. We can see that L=4 and more becomes undesirable if n is equal to 20 nodes or beyond.

Thus, we can conclude here, that for the saturated case, and packet size of 1 time slot, the AMC protocol performs better than the Extended-Reservation protocol, for small guardbands (0 to 3 percent), even if we have a large network and the Extended-Reservation protocol is operating with large Reservation-Limits. We can see that in order for the Extended-Reservation protocol to outperform the AMC protocol with 4% guardband, the Extended-Reservation protocol must operate with L greater than or equal to 16. However, we can see from 6(a), that with L= 16 or greater the fairness in the network reduces undesirably, espicially with larger number of nodes. A similar argument holds for AMC protocols of guradbands uptil 10%. In fact, with more nodes in the network, we can see that even the case when L=1, becomes untolerable, in terms of fairness.

The results above show that there exists a tradeoff between throughput and fairness in the Extended-Reservation protocol. If fairness is not important to some extent then we can use the Extended-Reservation protocol, with appropriate Reservation-Limits instead of AMC with guardbands greater than or equal to 4% in order to get a higher throughput. But, if fairness amongst nodes must be practiced at all times, even at the cost of throughput, then, the AMC protocol is the right protocol to use.

In the rest of the paper, we assume fairness should be maintained at all times, and hence, we will disregard the Extended-Reservation protocol, from this point onwards.

IV. Adaptive Multichannel MAC Protocols: Background

A. Need for Adaptation

The preceding analysis demonstrates that channelization is effective in improving efficiency for high data rate wireless networks. The analysis shows that an optimum number of channels exists depending on the operational parameters. If there is no guard band wastage, the optimum number is equal to the number of contending nodes. The optimum number is smaller when guard band is non-zero. Thus, a mechanism to adapt the number of channels with the number of contending nodes is useful. We will describe a simple protocol for channel adaptation in Section V. In spirit, the idea of channel adaptation is somewhat similar to the adaptation of contention window in CSMA protocols [29]. It is a hard problem as it requires sophisticated estimation mechanisms to estimate the number of contenders. Fortunately, some degree of success has been reported in literature [11], [12]. In the simplest form these techniques track the collision probability p by continuously measuring idle times, successful transmit times, busy times and unsuccessful transmit times. Using the estimated value of p, number of nodes n can be estimated from [10]. We will use a similar methodology in estimating the number of contending nodes when evaluating our protocol in Section VI.

B. Protocol Design Background

As discussed in Section II, there is a host of multichannel MAC protocols in current literature. But they are all geared for networks with pre-configured or fixed channelization (such as in IEEE 802.11) and thus most of them are not suitable for adaptation. Negotiation-based protocols such DCA [27], MMAC [24], LCM-MAC [17] that utilize an out of band mechanism (e.g., a control channel or a synchronized time period for control) to negotiate the channel to be used for transmission can be used, but need to be suitably modified to support channel adaptation. Also, these protocols suffer from a severe bottleneck problems in their out of band mechanism (control channel or control period) as they do per-packet or per-cycle negotiations, requiring them to transmit many control packets. This issue has been studied in depth [27].

We propose an Adaptive Multi-Channel (AMC) MAC protocol that mimics our modeling of multichannel operation in the previous sections. In particular, the protocol has the following properties:

1) The available frequency band B is adaptively divided into a near-optimal number of channels. To enable this the radio interface on each node is capable of changing center frequencies and channel bandwidths on the fly, within the given frequency band.
2) Each node has only one *half-duplex* physical radio interface for data packets. This makes interface costs reasonable. However, we do assume a high degree of capability on the part of interface (see below). When transmitting, the node can only transmit in one channel. The node also has a half-duplex control interface and a low bandwidth control channel in the simplest implementation of the protocol. However, as will be discussed in Section V-B these are not strictly required.
3) The data interface has a very general reception ability. Let us refer to the ordered set of tuples $<$center frequency, bandwidth$>$ of each channel as a channel configuration. Assuming that the maximum possible split is k-way, there are k possible channel configurations, with $1, \ldots, k$ channels. Thus, there can be $k(k+1)/2$ possible channels in the system. We assume that the interface is able to receive on all possible $k(k+1)/2$ channels at the same time.[3] This ability (i) removes the need for informing the receiver about the channel to be used before transmission, and thus eliminates a significant control overhead; (ii) allows for correct packet reception always (barring collisions and channel errors) by reducing the deafness and multichannel hidden terminal problems [17];[4] and (iii) allows different nodes to have different channel configurations for transmissions and still packets may be received correctly.
4) The onus of channel selection is purely on the sender. For the purpose of our study here, we assume a simple scheme where the sender selects a channel randomly and then contends and transmits only on that channel. This happens for each packet transmission.
5) When not transmitting, the interface always listens on all channels and keeps a population estimate on a continuous basis. As indicated before, techniques available in literature to build such population estimates can be used [11], [12].

V. Adaptive Multi-Channel Protocol: Operation

In this paper we will only study a distributed protocol that works in a single collision domain (single cell). While single collision domain is a limitation of our current study, our goal

[3] A prototype implmentation with k=5 using software radios has been shown possible in [14]

[4] These problems arise in several multi-channel protocools because nodes can listen to only one channel at any instant [17].

in this work is to explore the opportunities in channelization for high data rate networks – rather than promoting specific protocols, scenarios or architectures.

The simplest version of the Adaptive Multichannel (AMC) protocol that we will study here uses a low bandwidth control channel and a separate control interface. The protocol uses minimal control traffic to communicate certain status information (see below). As will be discussed in Section V-B, both the control channel and control interface are not strictly necessary.

A. Protocol Operation

The protocol is centered on two basic operations: 'split' and 'merge.' Given a k channel configuration, the 'split' operation moves the system to the $k+1$ channel configuration. The 'merge' operation does the opposite. Since the available bandwidth B is known and channels are all equal, knowledge of k is sufficient for the nodes to learn what channels are in use for communication.

The split and merge operations are implemented by broadcast SPLIT and MERGE control packets in the control channel. These broadcasts can be initiated by any node when the optimal channelization according to the current population estimate (aggregate in all channels) does not match with the current channelization. If the population estimate is n, then the analysis in Sections III and III-D can provide the corresponding optimal number of channels, $k^*(n)$. This becomes the threshold for 'merge' and 'split' operations. If the current channelization, k, is less than $k^*(n)$, then the node should 'split'; if k is greater than $k^*(n)$, the node should 'merge'.

The SPLIT and MERGE packets ensure that all nodes can keep track of the current number of channels used. To provide a degree of fault tolerance against lost control packets, each node also periodically broadcasts the current number of channels (according to its own information) through BEACON packets on the control channel. Upon receiving such BEACON packets a node changes its understanding of the current channel configuration to the minimum of its own information and the value contained in the BEACON. This minimizes the 'period of vulnerability' only to the interval until the next successful BEACON reception. Note, however, while during this period different nodes can use different channel configurations for transmission, packet reception is still possible because of the assumption that nodes can receive in any channel in all possible configurations. Also see the discussion on multicell operation in Section V-B.

Use of randomness can prevent synchronous behavior and thrashing. For example, more than one node can otherwise broadcast SPLIT messages almost back to back based on the same estimate causing the channels to be split more than necessary only to merge back momentarily. Much of these are matters of details and can be fine-tuned for a given architecture.

B. Discussions

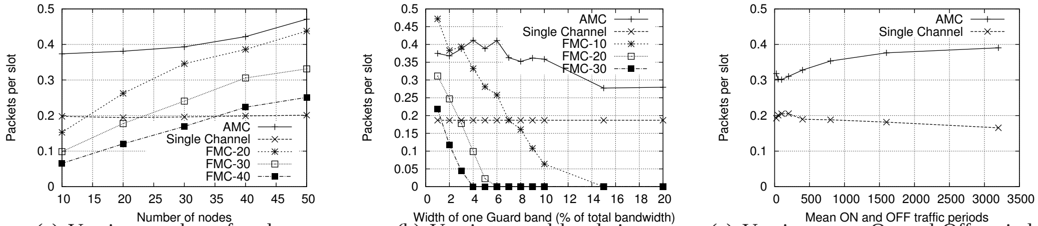
Control Channel Bottleneck: As observed in prior work [27], the control channel has the potential to become a bottleneck when used in multichannel operations. While this is true for negotiation-based protocols like DCA [27], bSMART [32] etc, AMC does not send explicit control messages for negotiation or coordination between senders and receivers for transmissions. Control packets are used only for merging, splitting and beaconing actions that are not frequent.

Deafness and Multichannel Hidden Terminal: Deafness occurs in a multichannel protocol when a sender does not know the state of its receiver (transmitting or ready to receive) [17]. In AMC, all nodes are always listening to all channels (except when actually transmitting). This helps address deafness. Multichannel hidden terminal arises when a node switches its channel to transmit on another channel, but it does not know the 'state' of that new channel, i.e., possible ongoing transmissions in the interference neighborhood [17]. But in AMC, a node always knows the state of all channels (except for periods when it is transmitting).

Improvements and Extensions: Several alternative approaches are possible around the same basic idea presented above. For example, control channel and control interface are not strictly needed. It is possible to send the control packets in the data channels instead. The downside is that nodes which are busy transmitting will miss these packets (due to the half-duplex assumption). Also, we have studied the AMC protocol only for a single collision domain operation. For multiple collision domains – either for single- or multi-hop operation – it is possible that different parts of the network use different channel configurations for optimal throughput. Thus, it is possible that some nodes might 'see' different channel configurations being used by different nodes in the neighborhood. This is not necessarily a problem if nodes are assumed capable of receiving packets simultaneously in all possible channel configurations as indicated before in Section IV. It is, however, possible that overlapped channels are created due to different channel configurations used by different nodes. This may introduce unintended interference. A control channel based protocol with suitable optimizations can alleviate this effect and lead the system to its optimal performance goal. We leave this design as an open question and a topic of our future work.

VI. Simulation Results

We have developed a slotted-time discrete event simulator to simulate the AMC and fixed channel multichannel protocols. While the analysis in Sections III and III-D shows the theoretical limits of the possible improvement due to channelization, the simulation results can show the real improvements possible when adaptation overheads are also taken into account. We compare our AMC protocol to a simple fixed multichannel (FMC) and single channel CSMA protocols. The fixed multichannel protocol is similar in most aspects to our AMC protocol except that it does not adaptively change



(a) Varying number of nodes (b) Varying guard band size (c) Varying mean On and Off periods

Fig. 7. Simulation comparison of the AMC protocol with fixed multichannel (FMC) and single channel protocols (25 nodes, packet time = 1 slot, guard band width (g) = 1% of total bandwidth B).

the channelization, and all nodes channelize the spectrum into the same number of channels. A comparison with fixed multichannel protocol helps illustrate the need for adaptive channelization with varying traffic. The FMC protocol has a parameter k, which denotes the number of channels.

We use an 'on-off' traffic model for each node indicating bursty (on) periods alternating with silence (off) periods. The periods are exponentially distributed with chosen means which are set equal in the all results reported here. Note that the analysis in Sections III and III-D has used saturated traffic. Thus, the simulation results always do not directly correspond to the analysis results.

A single collision domain is assumed and no channel error is modeled. Every node generates packets of constant sizes for its neighbors during its on period. Exponential backoff mechanism is used by every node for control as well as data transmissions, with a maximum of 6 backoff stages (i.e., $m = 6$). The minimum contention window W for each channel is optimized for the population estimate of that channel (as discussed in Section IV). A simple table (pre-computed) lookup achieves this. We vary the number of nodes, guard band width, and the mean on and off periods. The following parameters are used whenever otherwise not specified: packet time in slots (T_p/σ) is 1, the guard band size is 1% of the single channel bandwidth B, both mean on and off periods are 1000 slots and the number of nodes are 25.

Figure 7 shows the aggregate throughput in packets/slot from fairly long simulation runs with varying parameters. FMC-k denotes the fixed multichannel protocol with k channels. Evidently, AMC beats any FMC protocol or the single channel protocol almost always. On average, the improvement over single channel is about 100%. While FMC protocols tend to do better than single channel, they are almost always poorer than AMC.

In Figure 7(a) FMC protocols have higher throughput with increasing number of nodes because of the increase in effective offered load. AMC also has higher throughputs with increasing number of nodes, but offers relatively stable behavior. Single channel performance does not change as it is always under saturation. In Figure 7(b), we note that with larger guard bands AMC is even more preferable. FMC protocols can offer very poor performance if the guard band wastage is large. AMC is always able to choose the appropriate channelization for the best throughput.

Figure 7(c) exposes the weakness of the AMC protocol in that it relies on estimation of number of nodes and adapts relatively poorly when on-off periods are small while the aggregate offered load is the same. The performance differential is seen to be almost 25%. However, its performance relative to single channel remains high.

VII. Software Radio Implementation

In this section, we demonstrate the advantage to be gained from adaptive channelization using a prototype implementation on a 6 node software radio-based network. We use the GNURadio/USRP platform [2], [4]. While this platform is by no means high data rate, our 'scaled-down in speed' implementation still demonstrates the following.

(i) We show that in the USRP/GNURadio platform the slot time must be in the order of tens of milliseconds for an effective implementation of a CSMA protocol. This means that with the highest feasible data rate in this platform (1Mbps), packet time in slots (T_p/σ) is quite small even for reasonably large packets (e.g., in the order of a few KBs or smaller). This opens up the possibility of a performance boost via channelization.

(ii) We show that adaptive channelization is feasible and effective on this platform with some careful engineering.

In the following we describe the relevant details of the platform and experimental results.

A. Prototype Platform

GNURadio [2] is an open source software development platform that provides several signal processing blocks necessary to implement software defined radios using low-cost RF hardware and general purpose computers. The Universal Software Radio Peripheral (USRP) [4] is the most commonly used RF hardware along with GNURadio. The USRP motherboard has 4 high-speed analog to digital converters (ADCs) and 4 high-speed digital to analog converters (DACs) which are connected to an FPGA. The FPGA, in turn, connects to a USB2 interface and thereby to a host computer. The baseband samples are transferred between the the USRP motherboard and the host computer using the USB2 interface.

Daughter boards implementing the RF front end can be plugged in on the motherboard. Daughter boards have direct access to the ADC and DAC converters. For our prototype implementation, we have used the RFX2400 daughter board [4] that operates in the 2.3 – 2.9 GHz band, though the methods described below are general for any frequency band.

Receive and transmit channel widths can be changed in gnuradio by *decimation* and *interpolation*. In the USRP, a user-controlled parameter called the decimation rate can be changed from 4 to 256 (in multiple of 2), allowing us to tune to channels of width in the range of 62.5 KHz – 8 MHz when using GMSK modulation (that we use in our study). Similarly, the interpolation rate parameter can be changed from 4 to 512 (in multiples of 2), allowing us to send data in channels of width in the range of 62.5 KHz – 16 MHz, when using GMSK modulation.

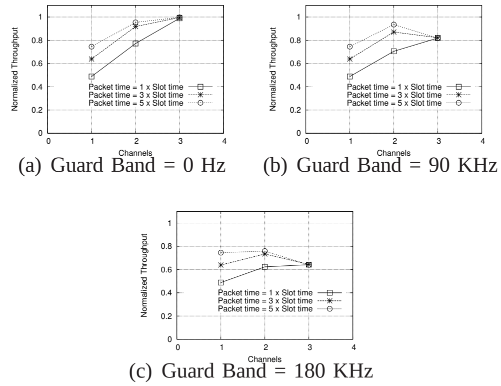
B. CSMA Protocol Implementation

In our implementation, carrier sensing is done in software on the host computer using raw I and Q samples from the USRP. The I and Q samples are magnitude-squared and a moving average of I^2+Q^2 is compared with a carrier sense threshold to detect the presence of carrier in any given channel. Carrier sense threshold is tuned for each channel used to provide a 100% accuracy in carrier sensing in our testbed. Using this mechanism, we developed an 802.11-like CSMA based MAC protocol without RTS/CTS and ACK.

The time domain I and Q samples from the ADC need to be transferred to the host machine from USRP through the USB interface on the receive side, and vice versa on the sender side. This transfer delay is dependent on the decimation rate in USRP (which determines the channel width). USB block size and number of USB blocks in the buffer also introduce delays. These delays present ‘blind spots’ for carrier sensing [15], [22], when a potential interferer is transmitting, however a potential sender cannot sense the carrier. The slot time must be carefully chosen so that it exceeds the ‘blind spot’ delay. In our implementation, we use channel widths varying from 200 KHz to 1 MHz and our measurements show corresponding delay range from 30 ms to 8 ms. We choose a slot time of 32 ms – slightly higher than this maximum time to ensure that samples are available for carrier sensing decision.

Backoffs are counted in slots. The minimum contention window size is always chosen as the optimal using a table look-up following the model in Section III. Exponential backoffs are used as before.

Limitations: Before we go forward we caution the reader about the limitation of the experiments: (i) The number of contending nodes are either statically fixed or told by an oracle in our experiments. Population estimation procedure are not implemented. (ii) The Ethernet interface serves as the control channel. It is relatively fast and effectively error-free. (iii) Receivers are told by the oracle when and which channel to receive on. Thus, any cost due to multichannel reception is not evaluated. However, any overhead of channel switching on the sender or receiver is captured. (iv) Of course, the slot time (32ms) is several orders of magnitude larger than 802.11a/b (9μs and 20μs, respectively), which in turn again orders of magnitude larger than the our target slot size (approximately of 1μs). However, this is purely due to the hardware limitation.



(a) Guard Band = 0 Hz (b) Guard Band = 90 KHz

(c) Guard Band = 180 KHz

Fig. 8. Throughput vs. number of channels and packet size for different guard bands.

Thus, our experiments should be viewed as ‘scaled-down in speed,’ but still with a real limitation of slot size, albeit due to a different artifact.

VIII. Experimental Evaluation

The 6 GNU Radio/USRP nodes in our testbed are grouped into 3 sender-receiver pairs. The nodes are deployed in such a way that for any given channel configuration, (i) a single collision domain is created on every channel, but (ii) there is no adjacent channel interference even with a zero guard band. This topology gives us an opportunity to evaluate the benefit of channelization for different guard band widths. Due to processing limitations in the host machine used in our setup, the maximum usable bandwidth B without any underrun or overrun in USRP is 1 MHz. We choose the center frequency at 2.5 GHz to avoid interference from other wireless devices operating in the unlicensed 2.4 GHz band. Three different channel configurations are used by dividing the 1 MHz channel into 1, 2 and 3 subchannels. (Further division is meaningless as our testbed can have at most 3 transmitters.) The actual width of the channels (b) depend on the guard band size (g) to be used. As noted before, GMSK modulation is used with spectral efficiency of 1 bps/Hz providing 1 Mbps nominal throughput in the entire band.

In the benchmarking experiments reported below, each sender transmits back-to-back UDP packets (indicating saturated load) for 60 seconds and the throughput is measured at the corresponding receiver. The throughput is normalized to the nominal channel bit rate of 1 Mbps for presentation. Repeating the experiments at different times showed little variation and thus confidence intervals are not shown.

A. Fixed Channelization

Here we study the benefit of fixed channelization in different emulated high speed networks by varying the packet length. As before, packet length is presented in terms of packet time (when using single channel) counted in slot time, i.e., T_p/σ. Figure 8 shows the aggregated normalized throughput when all three senders transmit simultaneously using 1, 2 or 3 channels. With 1 channel, all senders are on the same channel. With 2 channels, one sender is on one channel and the other 2 are on

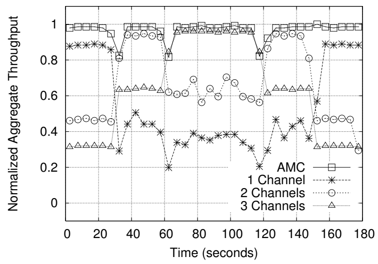


Fig. 9. Benfits of adaptive channelization compared to fixed channelization. (Packet time = 1 slot time. Zero guard band.)

the other channel. With 3 channels, they are all on separate channels. We show results for 0 Hz guard band, a moderate size guard-band of 90 KHz and a large guard band of 180 KHz.

For each channel configuration, throughput improves as expected with increase in packet length. Since the contention windows are optimized based on the number of contending nodes, there is zero overhead leading to a very high efficiency. For any specific packet length, there is a sizable improvement in channel efficiency due to channelization. With zero guard band, there is monotonic increase in throughput when more channels are used. Figure 8(b) and 8(c) show however that the rate of improvement decreases with guard band size and finally throughput goes down with more channels when guard band wastage become significant. The experience here qualitatively follows the modeling experience in Sections III and III-D.

B. Adaptive Channelization

Now we study the benefits of adaptive channelization compared to fixed channelization as well as using a single channel. We have implemented the AMC protocol using the Ethernet as the control channel. Due to the limitation of the current hardware, the receivers tune to only one channel where the sender will transmit, as told by an oracle. Due to the above limitation, population size is also not estimated, but the nodes learn about the number of contenders via an oracle. These oracles are implemented by a combination of scripting and broadcast communication on the Ethernet.

We use specific traffic pattern to demonstrate the power of adaptive channelization. The first sender starts transmitting at 0s and ends at 180s. Similarly, the second transmitter transmits from 30s to 150s and the third transmitter transmits from 60s to 120s. When transmitting, all senders transmit back-to-back UDP packets as before. Note the above pattern means that traffic ramps up at intervals of 30s, from 1 to 2 to 3 senders, and then ramps down similarly again. Figure 9 shows the aggregate normalized throughput computed every 5s along a timeline, when using three different fixed channel configurations and also using the AMC protocol. AMC almost always gives close to 100% throughput as nodes try to occupy individual channels leading to zero backoff (recall again contention window is optimized) and no bandwidth independent overhead. There is indeed some degradation in throughput when channels are split and merged (at 30s,60s,120s and 150s). This happens exactly when the number of contenders change and there is a change in number of channels. Change in channel configuration takes as much as 500ms in our testbed. Synchronization via the Ethernet (for implementing the oracle and control messages) also adds to this latency. But overall AMC performs significantly better than any fixed channel configuration as it matches the number of contenders to the number of channels.

IX. Conclusions and Future Work

In wireless networking literature, use of multiple channels to improve throughput performance is not new. However, in this paper we have offered a refreshing viewpoint. In regimes where the bandwidth independent overheads dominate, single channel performance suffers, with efficiency often falling below 50% even with an optimal contention window. Splitting the available channel into multiple smaller channels has the potential to improve performance considerably. We have shown using realistic numbers that high data rate wireless networks (1 Gbps and up) definitely falls in this regime. However, the number of channels to use depends on the number of contending nodes. Thus, the channelization must be adaptive. This issue is further complicated by use of guard bands.

We have developed an Adaptive Multichannel (AMC) protocol that adapts the number of channels to use based on an estimation of the number of contenders. Simulation results show an often factor of 2 performance improvement relative to using a single channel. We have further demonstrated the viability of the AMC approach using a software radio testbed using the GNU Radio/USRP platform. Experiments using 3 links and upto 3 channels show a similar scale of performance improvement without any additional spectrum use.

Note that multichannel protocols are not the only solution to this problem. One other potential solution we investigated is the Extended-Reservation protocol. Here, a sender, upon winning the contention, will send k data packets back to back, instead of one data packet. This will amortize the cost of bandwidth independent overheads over multiple packet transmissions. We have modeled this protocol and compared it against the adaptive multichannel technique. The results show that as the guard band size increases, the performance of multichannel approach as compared to the reservation approach becomes worse. However, using the Extended-Reservation protocol comes at a cost of losing fairness amongst nodes. We show that generally even if 4 packets are sent back to back, then at least 30% of the nodes suffer from unfairness. As the number of packets that are sent back to back increase, the network suffers from greater unfairness, and eventually becomes untolerable. But unlike the Extended-Reservation protocol, with the adaptive multichannel protocol the jain's fairness index remains very close to 1.0, even with large number of nodes, and huge guardbands. We have shown that with guardbands between 0% and 3%, even if we have uptil 100 nodes, the adaptive multichannel case wins over the Extended-Reservation protocol, in terms of both throughput and fairness. Since, the Extended-Reservation protocol suffers from serious fairness issues, as L grows large we have looked at the multichannel approach in more detail to understand its potential instead of the reservation-based approach.

Our future work will also consider extending the AMC protocol for multihop/multicell operation and more realistic evaluations on higher speed platforms, specifically focusing on addressing the receiver capability issues. Finally, the current work considers only uniform channel splitting. Non-uniform channels may provide better load balancing [19] and is worth exploring.

References

[1] "802.11e standard," http://standards.ieee.org/getieee802/download/802.11e-2005.pdf.

[2] "GNU Radio - The GNU Software Radio," http://www.gnu.org/software/gnuradio/.

[3] "Project IEEE 802.11 VHT study group," http://www.ieee802.org/11/Reports/vht_update.htm.

[4] "USRP - The Universal Software Radio Peripheral," http://www.ettus.com/.

[5] "Wireless HD technology," http://www.wirelesshd.org/technology.html.

[6] M. Alichery, R. Bhatia, and L. Li, "Joint Channel Assignment and Routing for throughput optimization in Multi-Radio wireless mesh networks," in *Proc. ACM MobiCom*, 2005.

[7] E. Aryafar, O. Gurewitz, and E. Knightly, "Distance-1 Constrained Channel Assignment in Single Radio Wireless Mesh Networks," in *Proc. IEEE INFOCOM*, 2008.

[8] A. R. S. Bahai, B. R. Saltzberg, and M. Ergen, *Multi Carrier Digital Communications: Theory and Applications of OFDM*. Springer, 2004.

[9] 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*, 2004.

[10] G. Bianchi, "Performance analysis of the IEEE 802.11 Distributed Coordination Function," *IEEE Journal on Selected Areas in Communication*, vol. 18, no. 3, pp. 535–547, 2000.

[11] G. Bianchi and I. Tinnirello, "Kalman filter estimation of the number of competing terminals in an IEEE 802.11 network," in *Proc. IEEE Infocom Conference*, 2003.

[12] F. Cali, M. Conti, and E. Gregori, "Dynamic tuning of the IEEE 802.11 protocol to achieve a theoretical throughput limit," *IEEE/ACM Transactions on Networking*, vol. 8, pp. 785–799, 2000.

[13] R. Chandra, R. Mahajan, T. Moscibroda, R. Raghavendra, and P. Bahl, "A case for adapting channel width in wireless networks," *SIGCOMM Comput. Commun. Rev.*, vol. 38, no. 4, pp. 135–146, 2008.

[14] L. Choong, "Multi-Channel IEEE 802.15.4 Packet Capture Using Software Dened Radio," Master's thesis, UCLA, 2009.

[15] R. Dhar, G. George, A. Malani, and P. Steenkiste, "Supporting Integrated MAC and PHY Software Development for the USRP SDR," *Networking Technologies for Software Defined Radio Networks, 2006. SDR '06.1st IEEE Workshop on*, pp. 68–77, Sept. 2006.

[16] N. Jain, S. R. Das, and A. Nasipuri, "A multichannel mac protocol with receiver-based channel selection for multihop wireless networks," in *Proc. of the IEEE International Conference on Computer Communication and Networks (ICCCN)*, 2001.

[17] R. Maheshwari, H. Gupta, and S. R. Das, "Mutichannel MAC Protocols for Wireless Networks," in *Proc. IEEE SECON*, 2006.

[18] M. A. Marsan and D. Roffinella, "Multichannel local area network protocols," *IEEE J. Selected Areas in Communications*, pp. 885–897, Nov. 1983.

[19] T. Moscibroda, R. Chandra, Y. Wu, S. Sengupta, P. Bahl, and Y. Yuan, "Load-aware spectrum distribution in wireless LANs," in *Proc. IEEE ICNP Conference*, 2008.

[20] J. R., C. D.M., and H. W., "A quantitative measure of fairness and discrimination for resource allocation in shared systems." Eastern Lab, Washington University, Tech. Rep., 1984.

[21] A. Raniwala and T. Chiueh, "Architechture and algorithms for an IEEE 802.11-based multi-channel wireless mesh network," in *Proc. IEEE INFOCOM*, 2005.

[22] T. Schmid, O. Sekkat, and M. B. Srivastava, "An Experimental Study of Network Performance Impact of Increased Latency in Software Defined Radios," in *Proc. Intl. Symposium on Computer Architecture*, 2006.

[23] J. Shi, T. Salonidis, and E. Knightly, "Starvation Mitigation Through Multi-Channel Coordination in CSMA Multi-hop Wireless Networks," in *Proc. ACM MobiHoc*, 2006.

[24] J. So and N. H. Vaidya, "Multi-channel mac for ad hoc networks: handling multi-channel hidden terminals using a single transceiver," in *Proc. ACM MobiHoc*, 2004.

[25] A. P. Subramanian, H. Gupta, S. R. Das, and M. M. Buddhikot, "Fast spectrum allocation in coordinated dynamic spectrum access based cellular networks," in *Proc. IEEE DySPAN Symposium*, 2007.

[26] T. Todd and J. Mark, "Capacity Allocation in Multiple Access Networks," *IEEE Transactions on Communications*, vol. 33, no. 11, 1985.

[27] S.-L. Wu, C.-Y. Lin, Y.-C. Tseng, and J.-P. Sheu, "A New Multi-Channel MAC Protocol with On-Demand Channel Assignment for Multi-Hop Mobile Ad Hoc Networks," in *Proc. International Symposium on Parallel Architectures, Algorithms, and Networks (ISPAN)*, 2000.

[28] X. Yang, N. Vaidya, and P. Ravichandran, "Split-Channel Pipelined Packet Scheduling for Wireless Networks," *IEEE Transactions on Mobile Computing*, vol. 5, no. 3, 2006.

[29] Y. Yang, J. Wang, and R. Kravets, "Distributed optimal contention window control for elastic traffic in single-cell wireless LANs," *IEEE/ACM Transactions on Networking*, vol. 15, no. 6, pp. 1373–1386, 2007.

[30] H. Yin and S. Alamouti, "OFDMA: A Broadband Wireless Access Technology," in *Proc. IEEE Sarnoff Symposium*, 2006.

[31] Y. Yuan, P. Bahl, R. Chandra, P. A. Chou, I. Ferrell, T. Moscibroda, S. Narlanka, and Y. Wu, "Kognitiv networking over white spaces," in *Proc. IEEE DySPAN*, 2007.

[32] Y. Yuan, P. Bahl, R. Chandra, T. Moscibroda, and Y. Wu, "Allocating dynamic time-spectrum blocks for cognitive radio networks," in *Proc. ACM MobiHoc Symposium*, 2007.