A Fair Medium Access Control Protocol for Ad-hoc Networks with MIMO Links

Karthikeyan Sundaresan, Raghupathy Sivakumar and Mary Ann Ingram
School of Electrical and Computer Engineering
Georgia Institute of Technology

Abstract—

In this paper, we present a new medium access control (MAC) protocol for ad-hoc networks with multiple input multiple output (MIMO) links. MIMO links provide extremely high spectral efficiencies in multipath channels by simultaneously transmitting multiple independent data streams in the same channel. MAC protocols have been proposed in related work for ad-hoc networks with other classes of smart antennas such as switched beam antennas. However, as we substantiate in the paper, the unique characteristics of MIMO links coupled with several key optimization considerations, necessitate an entirely new MAC protocol. We present a centralized algorithm that has the key optimization considerations incorporated in its design. Finally, we present a distributed protocol called *stream-controlled medium access* (SCMA) that approximates the centralized algorithm, and compare its performance against that of baseline protocols that are CSMA/CA variants.

Keywords—**Simulations**

I. Introduction

Ad-hoc networks or multi-hop wireless networks have typically been considered for use in military and disaster relief environments, due to their capability to operate without any infrastructure support. In recent years, the use of the so called "smart antennas" in ad-hoc networks has gained consideration. The term "smart antennas" represents a broad variety of antennas that differ in their performance and transceiver complexity, such as the *switched beam* and the *fully adaptive array* antennas.

Switched beam antennas have a pre-determined radiation pattern. The ability of such antennas to concentrate power in a certain direction provides a *directive* gain that can be used for extending range or reducing power. However, due to their simple signal processing capabilities, they are incapable of *adaptively* nulling out interference. *Steered beam* antennas are similar to switched beam antennas, but for their ability to steer the beam to track a user. On the other hand, fully adaptive array antennas, because of their increased signal processing capability, can dynamically adapt their radiation pattern. This helps them adaptively maximize the gain for the desired signal, and at the same time attenuate the signal from an interference source (by nulling), thereby maximizing the signal to interference plus noise ratio (SINR). Further, the knowledge of channel state information (CSI) at the transmitter can help beamform to increase the SINR resulting in a gain referred to as the *antenna* or *array* gain. Even without CSI at the transmitter, the multiple elements can be used to transmit the same signal, increasing the reliability of the link leading to a gain known as the *diversity* gain.

Smart antennas, in the conventional sense, are typically employed at only one end of the communication link, mostly at the access point or base station. Recently, the use of MEAs at both ends of the communication link has gained consideration, resulting in a technology popularly referred to as the *multiple input multiple output* (MIMO) technology. *Ad-hoc networks with such MIMO links is the focus of this work.*

The presence of multiple elements at both ends of the link creates independent channels in the presence of multipath or rich scattering. Multiple independent data streams can be transmitted simultaneously on these different channels to provide extremely high spectral efficiencies (increase in capacity) that comes at the cost of no extra bandwidth or power [1]. This is referred to as *spatial multiplexing* and can be realized even without any channel state information (CSI) at the transmitter (e.g. BLAST [2]). Thus, while switched/steered beam antennas are ineffective in handling multipath [3], and fully adaptive array antennas merely mitigate the effect of multipath, MIMO links actually *exploit* multipath to provide the spatial multiplexing gain [4]. Furthermore, MIMO links are also capable of all the advantages provided by fully adaptive array antennas. We present more details on the specific advantages of MIMO in Section II.

While MIMO carries significant promise, and has been extensively researched in the physical layer research community [1], [5], [6], their flexibility and performance enhancement can be truly leveraged only by appropriately designed higher layer protocols. At the same time, the key differences[1] in the physical layer properties of MIMO and switched beam antennas necessitate protocols that are very different from those developed for ad-hoc networks with the latter class of antennas [7], [8]. Specifically, in this paper, we focus on the medium access control problem for ad-hoc networks with MIMO links, and consider the following questions:

- *What are the key optimization considerations that should be incorporated in the design of a MAC protocol designed for the target environment?*
- *How can the versatile properties of MIMO links be leveraged to effectively realize a practical distributed MAC protocol with the optimal design?*

While we systematically answer these questions later in the paper, briefly we use both results from related research at the physical layer, and detailed arguments to identify several optimization considerations. Based on these considerations, we first present a centralized MAC scheme, and then a distributed MAC protocol called **SCMA** (Stream-Controlled Medium Access) for ad-hoc networks with MIMO links. The centralized scheme serves both as a basis for the SCMA design, and as a benchmark for the latter's performance. Through packet level simulations, we show that SCMA approximates the performance of the centralized scheme quite reasonably, while outperforming simple extensions of the CSMA/CA protocol for the target environment.

The rest of the paper is organized as follows: Section II provides some background on MIMO links. Section III highlights the key optimization considerations that are essential for the de-

[1] We elaborate more on the differences in Sections II and III.

sign of a MAC protocol for the target environment. Section IV presents the centralized scheme. Section V describes the SCMA MAC protocol for ad-hoc networks with MIMO links. Section VI presents the simulation results comparing SCMA with two baseline protocols. Section VII discusses related work, and Section VIII concludes the paper.

II. MIMO BACKGROUND

A. Relevant PHY Layer Characteristics

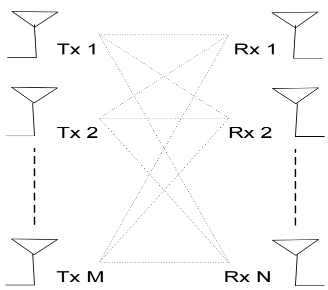


Fig. 1. MIMO Illustration

A MIMO link employs MEAs at both the transmitter (M elements) and the receiver (N elements) as shown in Figure 1. The presence of multiple elements at both ends of the communication link opens up independent channels (streams) for transmission in the presence of multipath or rich scattering. A transmitter has one of two options: it can either send dependent (correlated) signals on the different antenna elements or send independent signals.

If dependent signals are sent on the different streams, then this results in a *transmit-receive diversity* gain which is bound by the product of the number of elements at the transmitter and receiver (MN). This diversity gain can in turn be used to increase the transmission range[2]. The factor of range extension when compared to a single element case can approximately be given by,

$$d = (\frac{MN}{SNR_{thresh}})^{\frac{1}{p}} \quad (1)$$

where p is the path loss exponent, and SNR_{thresh} is the desired signal-to-noise ratio [9]. Thus for an antenna array with four elements, a range extension of two can be attained even when the path loss exponent is four (outdoor environments). For indoor environments, fewer number of elements will thus be required to achieve a similar range extension. However, the diversity gain provides diminishing returns as the number of elements becomes large [4].

On the other hand, when independent signals are transmitted on the different streams, as long as these streams fade independently, the receiver can isolate and decode the different signals. This results in a *spatial multiplexing gain* that provides a linear increase in capacity under conditions of independent and identically distributed (i.i.d) flat Rayleigh fading channel. This linear increase in capacity is with respect to the minimum of the number of elements at the transmitter and receiver. For M = N = *k*, the capacity is given by the following equation [4],

$$C \approx k \log_2(1+\rho) \quad (2)$$

where ρ represents the SNR.

On the other hand, if multiple elements are considered at only one of the ends, say M = 1, N = k, then the asymptotic relation is given by (see [4]),

$$C \approx log_2(1+k\rho) \quad (3)$$

The interesting point to note is that the capacity (with respect to *k*) increases *linearly* in the case of MIMO links, while it increases only *logarithmically* in the case of links employing multiple elements at only one of the ends [4]. *Our focus in this work is to exploit this spatial multiplexing gain to increase the capacity of the system*. However, we show later that the range extension possible through the diversity gain can be intelligently leveraged to address some key problems at the MAC layer.

Another degree of classification of MIMO links is based on whether the transmitter uses channel state information (CSI) with respect to the receiver. If the transmitter uses such information, the MIMO links are referred to as closed-loop MIMO (CL-MIMO), and an additional gain called *array or antenna gain* is exploited. Links in which the transmitter does not use CSI are referred to as open-loop MIMO (OL-MIMO). In this paper, we use OL-MIMO for any MAC layer control packet exchanges (e.g. Request-to-send and Clear-to-send messages in the CSMA/CA framework), but leverage these packet exchanges to exchange the CSI[3], thus enabling use of CL-MIMO for the actual data packet transmission.

B. Abstraction

We make the following abstraction and assumptions for the PHY layer in our work. We assume that all the nodes in the network employ antenna arrays with the same number of elements (M = N = *k*), and operate in the TDD (time division duplex) mode. The signals sent on the different channels represent the different streams transmitted. The total number of elements at a node correspond to the total available resources or *degrees of freedom* (DOFs) at the node. In a MIMO link, a receiver can isolate and decode all the incoming streams successfully as long as the total number of incoming streams (*n*) is less than or equal to its DOFs ($n \leq k$). On the other hand, if the incoming streams overwhelm the DOFs at the receiver ($n > k$), it will not be possible to decode any of the desired signal streams, if the excess ($n - k$) streams degrade the *k* streams below their receive threshold. However, if the strength of the excess (say, interfering) streams is far weaker than that of the desired (*k*) streams such that the desired streams can still be received with atleast the receive threshold, then it will be possible to decode the desired streams. In terms of transmission, a transmitter can use up all the available DOFs (taking into account DOFs available at nodes in the neighborhood after they have suppressed any interference) to spatially multiplex signals.

C. PHY Layer Flexibility

We now briefly outline the characteristics that are unique to MIMO and potentially relate to designing and realizing an efficient MAC protocol.

[2]Range extension will not be the same in all directions. In fact, we do not assume a circular or pi-shaped radiation pattern. Further discussions can be found in [9].

[3]The actual information can be achieved using several conventional PHY layer training mechanisms [10].

- *Adaptive Resource Usage:* In the case of MIMO, any resource not spent in suppressing interference can be dedicated to increasing the gain for the desired transmission as long as the same number of resources is available at both the transmitter and receiver. However, switched beam antennas cannot gain from spatial multiplexing if the angular spread of the desired signal path is larger than the beam width. Hence, typically, switched beam antennas can be left to use only a subset of their elements (1 in the worst case) for an ongoing transmission even in the absence of any interfering streams.
- *Tx Range vs. Capacity Trade-off:* While splitting the data stream into *k* parallel independent streams and transmitting them simultaneously on *k* elements provides us with multiplexing gain, MIMO also allows for leveraging diversity gain by transmitting the same (dependent) stream on multiple elements. Note that diversity gain does not require multiple elements to be present at both the transmitter and receiver. This diversity gain can provide us with *range extension* (a larger transmission range) or *power minimization*, or *better link reliability* as desired.
- *Flexible Interference Suppression:* Irrespective of the location of the interference sources, receivers in a multipath environment with MIMO links can suppress interfering streams as long as they have sufficient number of DOFs to do so. In the worst case, even in the presence of *k*-1 interfering streams, a receiver can still receive desired data transmission transferred on a single stream (provided the presence of multipath causes the signals to fade independently). This is in contrast to switched beam antennas, where interference sources in the same beam as the desired signal can simply not be tolerated.
- *Robustness to Multipath Fading:* MIMO does not require *line of sight (LOS)*, and can leverage multipath productively. Hence, it can be applied to rich scattering and multipath environments which are very common indoors. In contrast, for effective operation switched beam antennas require an *LOS* path between the transmitter and receiver because they are not optimized for multipath effects. This presents a major obstacle in using these antennas in multipath environments where the desired signal can arrive from multiple desired directions. On the other hand, any channel gain possible through the use of multiple elements is degraded if the angular spread of the desired signal multipath is larger than the beam-width.

The advantages outlined above demonstrate the potential benefits that can be gained by employing MIMO. In the next section, we outline the key optimization considerations that need to be taken into account in order to realize an efficient MAC protocol for the target environment.

III. Motivation

CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance) is the de-facto MAC protocol considered for use in ad-hoc network environments. Interestingly, a simple extension of CSMA/CA for MIMO links can be realized that can provide a k fold improvement in throughput performance through spatial multiplexing (k is the number of elements at each node), compared to a pure omni-directional environment.

We refer to the simple extension to CSMA/CA as CSMA/CA(k). Essentially, CSMA/CA(k) works in the exact same fashion as CSMA/CA except that *all* transmissions are performed using k streams to tap the spatial multiplexing gain. Such a protocol, when compared to default CSMA/CA operating in the same network topology, but with omni-directional antennas, will achieve k times the throughput performance as the latter[4]. While a k fold improvement[5] is indeed quite attractive, the question that we answer in this section is: *Is it possible for a more intelligent MAC scheme to realize better performance?* More importantly, does the degree of improvement justify the development of a new MAC scheme, instead of using a protocol such as CSMA/CA(k). We argue that the answers to both the questions is *yes*. We discuss why CSMA/CA(k) does not truly leverage the capabilities of MIMO using simple toy topologies. More importantly, we show in Section VI that the difference in performance between the "unaware" CSMA/CA(k) scheme and MIMO "aware" MAC scheme increases with increasing number of antenna elements.

In the rest of the section, we outline the key optimization considerations that need to be accounted for in the MAC design in order to effectively utilize the capabilities of MIMO.

A. Stream Control Gains

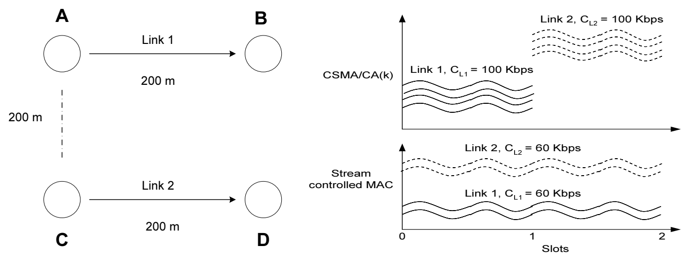


Fig. 2. Stream control topology

When a link is allowed to use only a subset of the maximum possible number of streams (say m out of k), *it can distribute its transmit power over just the m strongest channel modes (streams)*. Thus, when compared to two interfering links operating using TDMA at the maximum number of streams k, letting the links operate simultaneously but with $\frac{k}{2}$ streams will result in improving the overall utilization in the network. We term the gains achievable through such simultaneous operation of interfering but stream controlled links as *stream control gains*.

In the simple toy topology shown in Figure 2 where the nodes have four-element MEAs each, consider transmissions from node A to node B, and from node C to node D. CSMA/CA(k) allows only one transmission to take place in a given time slot but the transmission proceeds with all the four streams. On the other hand, consider a stream-controlled MIMO MAC where the two transmissions proceed simultaneously but the number of streams transmitted by each node is optimized (in this case to two streams) to give the maximum overall network throughput (Figure 2). For this simple two link topology, an improvement of 20% can be obtained in capacity over that of a TDMA scheme [11]. In general, as the number of mutually interfering links (l) increases, the subset of streams used by each of the links decreases ($\frac{k}{l}$), which in turn increases the gain obtained from performing stream control.

In a CL-MIMO system, there is a one-to-one mapping between streams and transmit array weight vectors; with the help of CSI

[4] Some tuning of the constant intervals used by CSMA/CA is essential.

[5] The *k* fold improvement represents only the theoretical bound for the capacity of a link and not the theoretical bound for multi-hop ad-hoc networks in general.

each antenna element transmits a super-position of all (weighted) data streams. In the receiving node, there will be a different array weight vector for each stream. Therefore, there will be a channel gain for each stream, which is the stream gain. These *stream gains are not equal*, and for moderate-to-low SNR, they can *have quite large disparities* even in the presence of interference [11]. This in turn motivates the need for performing stream control in order to increase the network utilization, wherein the best possible channel modes (two in the above example) are selected for transmission. In the above example, the normalized gains of 1, 0.9, 0.7, and 0.6 were assumed on the four streams. Hence, during stream control, the two best streams with gains of 1 and 0.9 were chosen by the two links to provide an improvement of around 20%. We summarize the above discussions by the following observation:

OBSERVATION 1: Multiple interfering links operating simultaneously using stream control achieve better overall throughput performance when compared to a scenario in which they operate using TDMA and k streams each.

B. Partial Interference Suppression

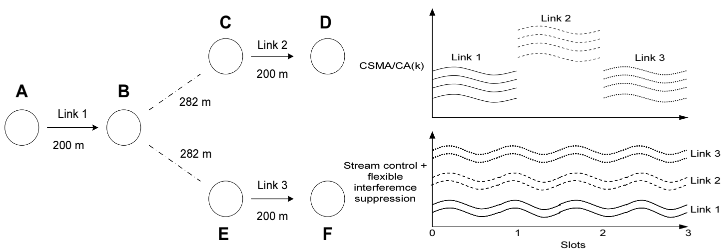


Fig. 3. Flexible interference suppression topology

When two interfering links are positioned such that the interference signals traverse a longer distance (R) than the desired signals (D) with $1 \leq \frac{R}{D} \leq 2$, the improvement of MIMO with stream control over CSMA/CA(k) can increase significantly. This is due to the flexible interference suppression capabilities provided by MEAs. The amount of resources that need to be sacrificed (expended) at a node to suppress an interference depends on the actual strength of the interference [4], [11].

This is better illustrated through the toy topology in Figure 3, where the nodes have a four-element array each. Consider three transmissions, from node A to node B, from node C to node D, and from node E to node F. CSMA/CA(k) allows only one of the transmissions to take place in any time slot but on all four streams. Consider a stream-controlled MIMO MAC where the nodes operate with two streams each. Now the transmitters C and E are outside the receive range but within the carrier-sensing range of receiver B. Hence, the number of DOFs required to suppress interference at node B in this case would only be a fraction of the total number of interfering streams, which in turn depends on the strength of the interference. Assuming this fraction to be half, this allows the three transmissions to take place simultaneously on two streams each (Figure 3). Hence, fewer resources are required to suppress interference when the interfering signals are from far away than when they are from close by. This, in turn results in more of the resources at a node being available for improving the performance of the desired transmissions/receptions.

But it must be noted that additional resources can be made available at any node due to flexible interference suppression, only as long as the node operates on a subset of the maximum number of streams possible. This is because, if the node operates on all available streams then it will have to expend all its resources to receive desired signal streams from its intended receiver. Hence, no additional resources will be made available in this case. Thus, *the gain of flexible interference suppression can be obtained only in conjunction with stream control*. This explains why CSMA/CA(k) cannot exploit the advantage of flexible interference suppression, even if its mechanism of silencing nodes in the two hop neighborhood of any transmission, is extended to incorporate flexible interference suppression.

In the above example, the average number of streams/slot is six for a stream-controlled MAC, while it is only four for CSMA/CA(k). Also it has been shown in related work that the performance gain over a TDMA based approach, obtained by employing flexible interference suppression can be as high as 50% for a simple two link topology [11]. Based on the above discussions, we make our second observation:

OBSERVATION 2: The flexible interference suppression capabilities of MEAs helps create additional resources at a node that can be used in conjunction with stream control for additional transmissions (receptions) to provide additional gain.

C. Receiver Overloading

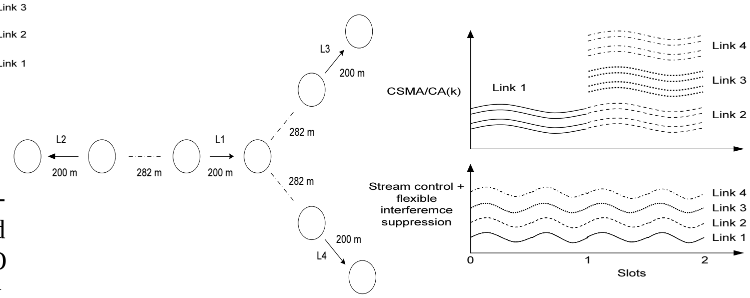


Fig. 4. Receiver overloading topology

While the above two factors directly help a MIMO aware MAC achieve improved performance over CSMA/CA(k), there exists one facet of MIMO that can potentially *degrade* its performance when compared to CSMA/CA(k). In CSMA/CA(k), since there can be only one active transmitter in any *contention region*, the other passive receivers in the same region can be *overloaded* with more streams than they can receive. This paves the way for spatial reuse. Hence, if a passive receiver belongs to more than one otherwise non-overlapping contention regions, then there can be an active transmitter in each of those contention regions.

On the other hand in a MIMO MAC employing stream control, all the transmitters within a contention region use the best subset of their streams such that no receiver in the region is overloaded. But if any of the receiver nodes also belong to other contention regions, then this prevents the nodes of those other contention regions from transmitting since this will overload the active receiver. This in turn reduces the advantage of spatial reuse and could potentially degrade performance.

For example consider the simple topology in Figure 4. There are four links, namely L1, L2, L3 and L4. The link L1 interferes with L2, L3, and L4, but the latter three links do not interfere with each other. If four element MEAs are used, CSMA/CA(k) can schedule L1 during one slot with four streams, and L2, L3, and L4, during the next slot with four streams each. Thus, the average throughput in the network in terms of streams per slot is eight. However, if a stream controlled MIMO MAC is used, all four links will operate exactly with one stream each, as any more streams will overload the receiver of link L1. Thus, the average throughput obtained is just four which is smaller than that of CSMA/CA(k) (Figure 4). Further, this degradation would increase as the number of passive receivers (belonging to multiple contention regions) increases, and also as the number of contention regions that a passive receiver belongs to increases. We attribute the above advantage of CSMA/CA(k) to its ability to perform *receiver overloading*, i.e. a passive receiver can be exposed to more than the maximum number of interfering streams. Thus, our final observation is:

OBSERVATION 3: The inability to overload a passive receiver because of performing pure stream control could result in a performance degradation that outweighs the gains from stream control.

IV. Centralized SCMA

In this section, we present the centralized *stream-controlled medium access* (SCMA) protocol for ad-hoc networks with MIMO links. The design of a centralized algorithm has two potential benefits. (i) It provides a basis for the design of the distributed algorithm, and (ii) It serves as a benchmark against which the distributed algorithm can be compared.

The centralized algorithm has the objective of maximizing the network utilization subject to a given fairness model. The fairness model that we employ is the proportional fairness model. A good exposition on the motivation for the fairness model can be found in [12]. While the primary goal is to come up with a channel allocation vector that is proportionally fair, the maximization of the network utilization can be achieved only by realizing the optimization considerations identified in Section III. Thus, the centralized algorithm attempts to leverage the benefits of stream control and partial interference suppression, while at the same time enabling the receiver overloading possible in CSMA/CA.

A. Insights and Overview

The basis of the centralized SCMA algorithm rests on an observation about the (lack of) receiver overloading problem: there exist a specific subset of links in the network that contribute to the lack of receiver overloading when performing pure stream control. An example of such a link is Link 1 in Figure 4. We refer to such links as *bottleneck links*. An alternative description for bottleneck links is that they belong to *multiple contention regions* in the network. It can be observed in Figure 4 that Link 1 belongs to three contention regions with links 2, 3, and 4 respectively.

If such bottleneck links are scheduled in the non-stream controlled fashion (operating on all *k* streams), such links can essentially be removed from further scheduling considerations, leaving the scheduling algorithm with only independent contention regions within which pure stream control can be employed. Upon closer look, it can further be observed that the bottleneck links can be identified by identifying vertices in the *flow-contention graph*[6] of the underlying network that belong to multiple maximal cliques.

The centralized SCMA algorithm is designed based on the above insights, and has the following key elements: (i) identification of bottleneck links (link classification) - referred to as *red* links in the algorithm; (ii) scheduling of bottleneck (red) links in a non-stream controlled manner; and (iii) scheduling of the non-bottleneck (white) links in the network based on pure stream control.

B. Centralized Algorithm

$INPUT$: Network Topology graph G = (V,E), and K
    V = nodes in the network
    E = pair of nodes within reception range of each other
    K = number of antenna elements at each node
Step1: First generate the Flow Contention Graph G' = (V',E')
    from G based on neighborhood properties
Step 2: Color the vertices in G': COLOR(G')
Step 3: Obtain the schedule : SCHEDULE(G")
COLOR(G')
1 Color all the nodes WHITE
2 Rank the nodes based on the tuple (d,s)
3 White nodes all have a rank of ∞
4 Choose the highest rank node and color it RED
5 Remove this node and all edges emanating for further coloring
6 Re-rank the remaining nodes after updating (d,s) values
7 If there are no nodes with rank $< \infty$ exit, else goto step 4
SCHEDULE(G")
8 $\forall\ i\ \epsilon\ V'$,$service_i$ = 0, $resource_i$ = K, $allocation_i$ = 0
9 $slot\_index$ = 0
10 $While\ ((min(service_i) \leq min(service_j))$
    $i\ \epsilon$ RED, && $j\ \epsilon$ WHITE
11   Find I $\subseteq$ RED, such that, $\forall\ i\ \epsilon$ I,
        $resource_i\ \geq\ K$ &&
        $(resource_j\ \geq\ w_{ij} * K,\ \forall\ j\ Neighbor(i)$ && $allocation_j\ >\ 0)$
12   If I = $\emptyset$, break
13   else Choose the node $i\ \epsilon$ I, such that, $service_i = min(service$(I))
14        $service_i = service_i + K,\ allocation_i = 1$
15        $\forall\ j\ \epsilon\ Neighbor(i),\ resource_j = resource_j - w_{ij} * K$
16   Find I, If I $\neq\ \emptyset$ goto 6
17   else Find J $\subseteq$ WHITE, such that, $\forall\ p\ \epsilon$ J,
            $resource_p\ \geq\ 1$ &&
            $(resource_q\ \geq\ w_{pq},\ \forall\ q\ Neighbor(p)\ allocation_q\ >\ 0)$
18   If J $\neq\ \emptyset$, do $Schedule\_white$, goto 3
19   else $slot\_index + +,\ \forall\ i\ \epsilon\ V'$,
                $allocation_i = 0,\ resource_i = K$ goto 3
20 $While\ (min(service_j)\ <\ max(service_i),\ i\ \epsilon$ RED, && $j\ \epsilon$ WHITE)
21   Find J, If J = $\emptyset$ break
22   else do $Schedule\_white$, goto 13
Schedule_white()
23 Choose the node $j\ \epsilon$ J, such that $service_j = min(service$(J))
24 $service_j = service_j + +,\ allocation_j = 1,$
25 $\forall\ p\ \epsilon\ Neighbor(j),\ resource_p = resource_p - w_{jp}$
26 Find J, If J $\neq\ \emptyset$ goto 16
27   else $slot\_index + +,\ \forall\ i\ \epsilon\ V',\ allocation_i = 0,$
        $resource_i = K$, return

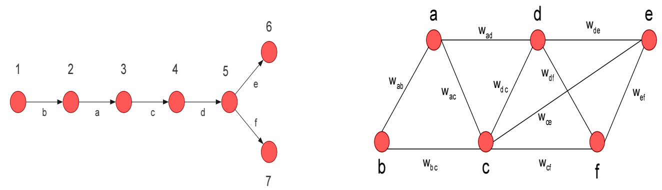
Fig. 5. Pseudo Code for Centralized Algorithm

We now present the centralized algorithm, the pseudo code for which is presented in Figure 5. We also use a running example of the network topology in Figure 6 (a) to illustrate the different stages of the algorithm.

[6] A graph with vertices representing links in the underlying network, and an edge between two vertices existing if the two corresponding links contend with each other in the underlying network [12].

B.1 Graph Generation

Given the network topology, the *flow contention graph* G' = (V',E',W) is generated (Figure 6 (b)), where V' represents the set of links in the network (Step 1). E' represents the edges between any two vertices in G', whose links contend with each other in the underlying network, and the weight of the edge (ϵ W) represents the amount of interference caused by one link on the other. Before being able to classify the vertices, it is necessary to identify all the non-overlapping contention regions in the link contention graph. This is equivalent to the problem of identifying all the maximal cliques in G'. We next explain how this can be achieved.



a) Node Graph

b) Flow Contention Graph

Fig. 6. Graphs

B.2 Clique Identification and Ranking

Identifying all the maximal cliques in a graph is known to be an NP-Hard problem. Hence the centralized algorithm makes use of another algorithm that determines all the maximal cliques in chordal graphs (having less than 4 cycles). It first determines the *perfect elimination ordering* (PEO) using LexBFS (Lexicographic Breadth First Search) [13] for the chordal graph and then applies a *linear* algorithm that detects all the maximal cliques given the PEO using a theorem by Fulkerson and Gross [14].

Though the algorithm works for only graphs having cycles of size less than four, note that the graph in our case corresponds to the flow contention graph. Hence for a cycle of size four to be present in the flow contention graph, a cycle of atleast size eight must be present in the node graph with no nodes being present inside the cycle. Given the typical size of ad-hoc networks to be of the order of 1000m in either dimension, the probability of finding such a scenario is very low. Hence, we use this algorithm to determine all the maximal cliques in the flow contention graph.

Once all the maximal cliques have been obtained, the vertices (in G') are then ranked. Every vertex has two attributes (*d*,*s*) : *clique degree d* (number of maximal cliques that the vertex belongs to) and maximum size *s* of all possible cliques that it belongs to. The vertices are ranked lexicographically based on the tuple (*d*,*s*) with the vertex having the highest degree ranked first, and the maximum size *s* is used to break ties. However, it is not necessary to rank vertices that have a degree of one. The vertices are then colored (Step 2).

In the example in Figure 6, the different maximal cliques in Figure 6 (b) are *cdef*, *abc* and *acd*. Vertex *c* obtains the highest rank with a degree of 3, followed by *d* that as a degree of 2. The rest of the vertices all have a degree of one.

B.3 Coloring

Initially all the vertices are colored white[7] (line 1, see Figure 5). Based on the tuple information (d,s) for each vertex, the vertices are ranked lexicographically as described before (lines 2-3). Then the vertex with the highest rank is recursively chosen, and colored red[8], following which, the particular red colored vertex and edges emanating from it are removed from G' (lines 4-5). The tuple (d,s) of the remaining vertices in G' are updated and the remaining vertices are re-ranked once again (line 6). The process repeats until no more vertices can be colored red (line 7). Once this is done the schedule for channel allocation is obtained (Step 3).

In the example, vertex *c* is colored red first, followed by vertex *d*. The rest of the vertices, with a degree of one, are colored white.

B.4 Red Vertex Allocation

The allocation begins with the red vertices which are the bottleneck links in the underlying network. A red vertex can be scheduled in any slot[9] only if it can operate on all *k* (= number of elements) streams. The red vertices are scheduled based on their rank, starting with the highest ranked vertex. At every slot, the algorithm also attempts to maximize the utilization, i.e. after scheduling a red vertex with *k* streams, the algorithm checks to see if any other red vertices can be scheduled in the same slot with *k* streams (line 15). If yes, all such red vertices are scheduled in the same slot. In addition to these red vertices, all the white vertices that can use at least one stream in that slot are also scheduled (lines 16-17). Then the algorithm attempts to increase the number of streams that these scheduled white vertices can use in the same slot. Note that while the red vertices can be scheduled only if they can use *k* streams, the white vertices can be scheduled even if they can use 1 stream. But among the white vertices that can be scheduled, the algorithm performs a fair allocation of streams (lines 23-27). The scheduling of white vertices in the same slot as that of red vertices (if possible) exploits spatial reuse and thereby helps maximize utilization. The scheduling of the red vertices is stopped when the red vertex with the minimum service (allocation) has received greater service than that of the white vertex with the minimum service (line 10).

In the example, the switching happens when both the red vertices *c* and *d* have obtained a service of *k* streams each. In one slot, vertex *c* can alone be scheduled with *k* streams. But in the other slot when vertex *d* is scheduled with *k* streams, vertex *b* can also be scheduled with *k* streams.

B.5 White Vertex Allocation

Once the scheduling switches to the white vertices, the allocation is done on a stream by stream basis to all the white vertices that can be scheduled in the same slot (lines 20-22). This results in a fair allocation of streams to all the white vertices that can be scheduled in the same slot. At a high level, the red vertices being the bottleneck links (responsible for utilization degradation) follow a schedule similar to that of the links in CSMA/CA(k) to avoid the degradation, while the white vertices

[7] A white vertex in G' corresponds to a white link in the network.

[8] A red vertex in G' corresponds to a red link in the network.

[9] Slot corresponds to the time for a packet transmission in the centralized approach.

perform stream control to exploit the advantage of MIMO. The scheduling switches back to the red vertices once the white vertex with minimum service has an allocation greater than or equal to the red vertex with maximum service (line 20). This condition to switch the schedule between the red and white vertices ensures that all the vertices obtain an allocation of atleast *k* streams at the end of every *l* (= number of vertices) slots.

In the example, white vertices *a*, *b* belong to one clique, while *e*, *f* belong to another independent clique. Hence, these vertices perform stream control operating on $\frac{k}{2}$ streams each in their independent cliques simultaneously. The switching would occur after two slots when each of these vertices would have obtained an allocation of *k* streams.

V. DISTRIBUTED SCMA

In this section we present the distributed SCMA scheme that aims to approximate the centralized algorithm. The distributed scheme achieves this goal in a purely localized manner without the requirement of any large scale coordination in the network.

A. Overview

The basic components inherent in the design of SCMA are outlined below:

- SCMA performs carrier sensing and retains the control packet exchanges (RTS/CTS handshake) employed in CSMA/CA for collision avoidance. In SCMA, carrier sensing helps a node sense the channel in order to determine the number of resources (streams) that it has to sacrifice to suppress the interference. Collision avoidance, though motivated by the hidden terminal problem, also helps the nodes obtain control information of available resources at the receiver, and other neighbors, and thereby make a decision on the number of resources to be used for transmission.
- Although SCMA performs collision avoidance, the contention resolution no longer happens in the backoff domain. Instead SCMA, performs contention resolution in the persistence domain similar to [12], which makes it easier to achieve a proportional fairness model. Further, it has been shown in [12] that if the persistence parameters (x_i) of the flows are adapted according to

$$\dot{x_i} = \alpha - \beta p_i x_i \tag{4}$$

then the system converges to the optimal point of maximizing the aggregate network utilization for a proportional fairness model. α and β are system parameters, p_i is the loss probability experienced by the flow, and $\dot{x_i}$ is the rate of change of persistence.
- However, the above adaptation assumes a single level scheduling. Hence, to extend the adaptation to the dual scheduling (red and white links) employed in SCMA, the persistence value (P_{old}) obtained from the basic adaptation is translated into a new persistence value, P_{new}. This P_{new} is the same as that of P_{old} for the red links, while it is scaled up for the white links. Since the white links in a clique operate simultaneously, using only a subset of the streams ($K_{new} < k$), their P_{new} value will be a scaled version of P_{old} as in,

$$P_{new} = (P_{old} * K_{old})/K_{new}, \tag{5}$$

The entire adaptation process still happens on the P_{old} values, but the links thereafter appropriately identify their P_{new} value based on their color and use it to determine access to the channel. This ensures that the resulting channel allocation vector is still a proportionally fair one (formally proved in [9]). Further details on the adaptation mechanism are provided in Section V-B.2.

B. Distributed Algorithm

The distributed algorithm has to address the following challenges in being able to approximate the centralized algorithm:

- The links must identify whether they belong to multiple contention regions or not and hence color themselves in a distributed fashion,
- Since the channel access mechanism for the white links (involving stream control) is different from that of the red links, the adaptation of the persistence parameter for the white links must be appropriately tuned such that proportional fairness is still ensured, and
- For the white links to be able to perform stream control, it is essential for the transmitting nodes of the white links to estimate the fair share of operation (streams) in their contention region, in a distributed manner.

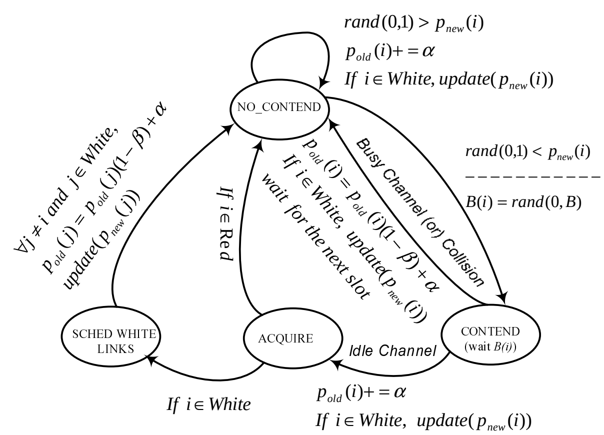


Fig. 7. SCMA State Diagram

We now present the components of the distributed algorithm that address these challenges in the process of approximating the centralized algorithm. In addition, the design of these components also helps leverage the advantages provided by the optimization considerations. The state diagram and the pseudo code for the algorithm are presented in Figures 7 and 8 respectively.

B.1 Coloring

Coloring is necessary to distinguish between the red and white links, in order to leverage the optimization considerations.

We relax the requirement that it is sufficient if the links are able to identify whether they belong to multiple contention regions or not, instead of actually identifying the number of contention regions they belong to[10]. Further, only a red link will be overwhelmed in terms of resources due to members of the different cliques that it belongs to, transmitting at the same time. This fact is exploited in aiding the transmitting and receiving nodes determine the color of their respective links.

Every transmitter initially starts transmitting on all the streams (with $P_{new} = P_{old}$) like CSMA/CA(k) until it determines its

[10]This relaxation makes the distributed algorithm only an approximation of the centralized approach.

CONTEND( )
1 $\forall$ $Slot\ S,\ node\ i$
2 $state$ = NO_CONTEND
3 If $uniform(0,1) \leq P_{new,i}$
4 $state$ = CONTEND
5 $B_i = uniform(0,B)$
6 Defer(B_i)
7 If $(Check\_resources(\ ) == Available)$
8 Acquire_channel()
9 If $(Acquire\_status(\ ) == Collision)$
10 $P_{old,i} = P_{old,i} * (1-\beta)$
11 If $i \in$ WHITE
12 $P_{new,i} = \frac{P_{old,i} * K_{old,i}}{K_{new,i}}$
13 else $state$ = ACQUIRE
14 $\forall\ j \in Neighbor(i)$
15 Update Resources
16 If $(resources_j < 0)$, $color(j)$ = RED
17 Recolor(i)
18 If ($i \in$ WHITE), Co-ordinate_Schedule(i)
19 If (Remaining_resources(i))
20 $K_{new,i} = K_{new,i} + \frac{Remaining\_resources(i)}{white\_count}$
21 else $P_{old,i} = P_{old,i} * (1-\beta)$
22 If $i \in$ WHITE
23 $P_{new,i} = \frac{P_{old,i} * K_{old,i}}{K_{new,i}}$
24 $P_{old,i} = P_{old,i} + \alpha$
25 If $i \in$ WHITE
26 $P_{new,i} = \frac{P_{old,i} * K_{old,i}}{K_{new,i}}$

Co-ordinated_Schedule(i)
27 $\forall\ j \in Neighbor(i)$ && $color(j)$ == WHITE
28 $P_{old,j} = P_{old,j} * (1-\beta)$
29 $P_{new,j} = \frac{P_{old,j} * K_{old,j}}{K_{new,j}}$
Recolor(i)
30 Check slot history from previous service slot in conjunction with receiver to color
31 If $i \in$ WHITE, $K_{new,i} = \frac{K_{old,i}}{white\_count}$

Fig. 8. Pseudo Code for Distributed Algorithm

color. It locally determines the color of its link in conjunction with its receiver. Specifically, the transmitter observes the usage of resources between every two slots that it has gained access to the channel. If the transmitter or the receiver observe more than *k* streams during any of the slots[11] that it has not obtained access to the channel, then the link is automatically colored red (line 16). It is possible for a link to be red even if the transmitter and the receiver individually do not observe more than *k* streams. This is because there could be an active link within the interference range of the transmitter but not within that of the receiver or vice versa. The transmitter and receiver in this case would not be able to "independently" identify the correct color. Hence the transmitter during its RTS/CTS exchanges with the receiver compares its version of the winner list (IDs of nodes that have obtained channel access in its neighborhood) with that of the receiver for the slots in between their successive channel accesses. If the list happens to be different in at least one of the slots, then the link is colored red, since this effectively means that this link as a whole is exposed to more than *k* stream transmissions at the same time (line 17). Otherwise the link is colored white.

[11] Slot corresponds to the duration of a RTS/CTS/DATA/ACK exchange in the distributed approach.

B.2 Contention and Channel Access

Once the nodes have successfully colored their respective links, they adopt a channel contention mechanism that is tuned to their color. There are four possible states in which a node can be, namely *Contend*, *No_Contend*, *Acquire* and *Sched_White_Links* (Figure 7). Every node having a packet to transmit, first decides to contend for the channel with a probability of P_{new}. This persistence probability P_{new} is the same as P_{old} for the red links, while it is scaled for the white links (line 12). If the node succeeds, it moves from the *No_Contend* state to the *Contend* state (lines 2-4), where it chooses a waiting time uniformly distributed from the interval (0,B). B is a constant and set to 32 in the simulations as advocated in [12]. The node then waits for the backoff period (in slots), after which it tries to access the channel to see if the channel is busy (lines 5-7). The busy state of the channel in our case corresponds to a lack of sufficient amount of resources at the transmitter or the receiver or the two-hop neighbors.

If the node finds the channel to be busy, it gives up the slot and decrements its persistence by $\beta * P_{old}$. Similarly if the channel is idle but if the node faces or detects collision, it decrements its persistence by a factor of β. In addition to decrementing the value of P_{old}, if the node belongs to a white link then it also has to update its P_{new} value (lines 8-12). On the other hand if the node finds the channel to be idle, and does not experience any collision then it moves to the *Acquire* state where it transmits. Every node in the two-hop neighborhood of this transmission would automatically expend the appropriate number of resources to suppress this transmission (line 15). At the end of the slot, all the nodes having a packet to transmit in the next slot increase their persistence P_{old} by α with the white links also updating their P_{new} value (lines 24-26). The values of α and β are chosen to be 0.1 and 0.5 based on the rationale provided in [12].

In being able to determine if the channel is busy, a node needs to know about the resource availability at nodes in its two hop neighborhood. This can be achieved by piggybacking the amount of resources remaining at a node in its control packet transmissions. However, to make these control packets decode-able in the two hop neighborhood, the reception range needs to be extended by a factor of two. This in turn can be achieved by transmitting the control (RTS/CTS) packets as multiple copies on atleast four streams. In this case, we exploit the diversity gain of MIMO to provide us with this range extension factor of two (with 4 element MEAs and a path loss exponent of 4), instead of its spatial multiplexing gain. Note that this range extension will not be the same in all directions and will depend on the radiation pattern currently used by the transmitting node[12]. This range extension mechanism has the additional benefit of aiding the white links in their stream control process, which we explain subsequently. In related work [9] we have profiled the overhead of transmitting the control packets on multiple streams, and showed that as long as more than four elements are employed, multiplexing gain can still be leveraged. Hence, the overhead of exploiting diversity gain for a range extension factor of two will tend to decrease as the number of elements increases.

[12] Detailed exposition on how this range extension can be achieved can be found in [9].

B.3 White Link Adaptation

For the white links to be able to perform stream control and hence determine the appropriate persistence (P_{new}) with which to contend for the channel, they need to estimate the fair share of resources to use in their contention region. While the computation of fair share of the red links is relatively easy (*k* streams), the fair share estimation for the white links is non-trivial.

Every node advertises the color of its link (if colored) in its transmissions. During the initial phase, when a white link may not be aware of the other white links in its contention region, it will not be able to arrive at the correct fair share. Hence for this purpose, every node transmits for one more slot on all *k* streams (even after it has colored itself white) along with its color information to inform the other members of the clique about its newly colored link. Since the control packets are decode-able within a range that is twice the normal reception range due to the diversity gain, the other members of the clique will receive this information. This helps the white links keep track of the number of white links in the same clique (say *w*) and hence help them arrive at the fair share in the clique k_{new} (=$\frac{k}{w}$) (line 31).

B.4 Co-ordinated Scheduling

Distributed execution of the stream control mechanism by the white links is a challenge that has to be accomplished. This requires that the white links operate simultaneously on a subset of the maximum allowable number of streams. Since persistence is used to ensure proportional fairness, it is possible that only some of the white links in a clique actually contend for a slot. Hence, even if one of the white links gains access to the channel it must be ensured that all the other white links in the same clique are also scheduled in the same slot, failing which the advantage of stream control cannot be leveraged.

Accordingly, when the first white link in a clique gains access to the channel, it also co-ordinates the other white links in the clique to transmit in that slot using their own estimated fair share (line 18). This corresponds to the node entering the *Sched_White_Links* state in Figure 7. Specifically this link's RTS/CTS messages will contain a flag ordering the schedule of all the white links in the clique. Since the control messages can be decoded within the two-hop neighborhood (owing to the range extension factor of two), all the white links in the clique will be able to listen to the command of this initiating link and thereby schedule themselves in the same slot, irrespective of whether they contended for channel access in that slot or not. However, the contending white links that were not the initiator of the co-ordinated scheduling will still have their P_{old} values decremented by the factor β to be in conformance with the normal adaptation algorithm (lines 27-29).

In addition, to be able to leverage the advantages of flexible interference suppression, the nodes belonging to white links observe if their fair share can be increased based on the remaining resources available at the end of their transmission (line 20). If so, the node increases its fair share only by a fraction of the remaining resources to allow other white links in the clique to increase their fair share as well. Thus the resources are fully utilized in the clique, thereby leveraging the advantage of flexible interference suppression. We also discuss the stability of the protocol in the presence of network dynamics in Section VI.

VI. PEFORMANCE EVALUATION

In this section, we present performance results for SCMA obtained through simulation studies. We use an event-driven packet level simulator for recording the results. We use UDP as the transport layer protocol and CBR as the traffic generator. The packets are generated at a rate of 100 packtes/sec and are of size 1Kbyte. The number of flows and elements in the antenna array are parameters that vary from one experiment to another. We extend the distributed proportional fair contention resolution (PFCR) mechanism [12] to PFCR(k) and CSMA/CA to CSMA/CA(k) for ad-hoc networks with MIMO links and consider these as baseline protocols in our simulation study. CSMA/CA(k) and PFCR(k) are variants of CSMA/CA and PFCR respectively that use *k* streams for their transmissions and receptions. Further, we use a simple model that conservatively[13] assumes only about 20% gain from performing stream control. We compare the performance of SCMA, CSMA/CA(k) and PFCR(k) with that of the centralized protocol and thereby present results to highlight the benefits of the mechanisms involved in our distributed protocol.

The metrics we use for comparing the various schemes are *throughout* and *relative standard deviation*. Since the algorithm provides proportional fairness that is location dependent, standard deviation or normalized standard deviation would not be able to capture well the degree of fairness provided by the scheme. Hence we have chosen to use relative standard deviation as the fairness metric. Specifically, the distribution of throughput of the various schemes is compared against that of the centralized scheme and the standard deviation normalized to the mean is obtained.

A. Macroscopic Results

We now present a set of representative topologies to highlight the advantages of stream control and flexible resource usage.

A.1 Toy Topologies

We discuss the results with respect to the throughput metric first, followed by the fairness metric.

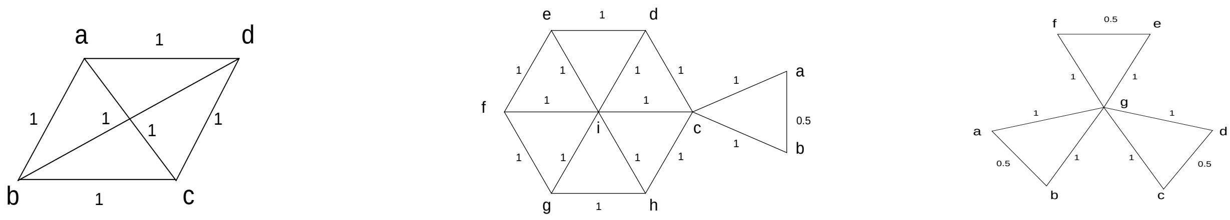
- *Scenario 1*

This scenario is used to highlight the gains from performing pure stream control. We consider a simple four-clique flow contention graph as shown in Figure 9(a). Each node has a four-element array. The comparative results for the different schemes are presented in Figure 10(a). All the links are white in this case and since all the link weights are 1, there is no performance gain due to flexible interference suppression. The gain of SCMA over PFCR(k) and CSMA/CA(k) (10-20%) is solely contributed by stream control since each link in the clique transmits in every slot on a single stream. Further, CSMA/CA(k) performs better than PFCR(k). But as we shall show later, the degree of fairness provided by PFCR(k) is much higher than that of CSMA/CA(k).

- *Scenario 2*

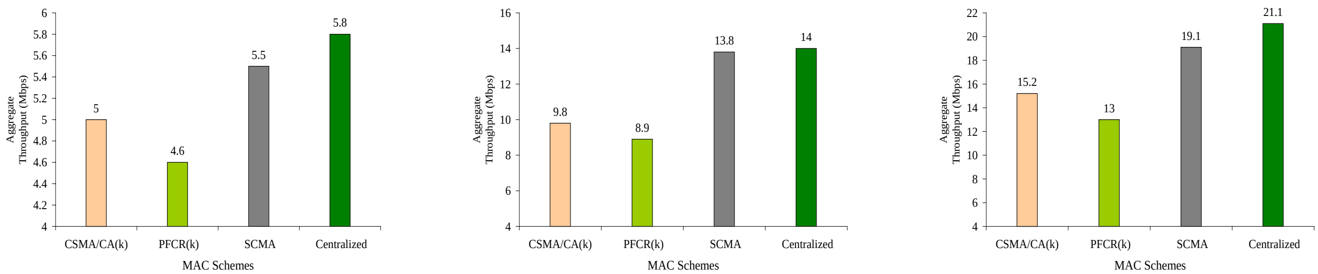
This scenario is used to highlight the gains from performing both stream control and flexible interference suppression, with stream control being the dominating contributor. We consider a

[13]The statistical gains on the different streams could have significant disparities resulting in a much larger gain [11].



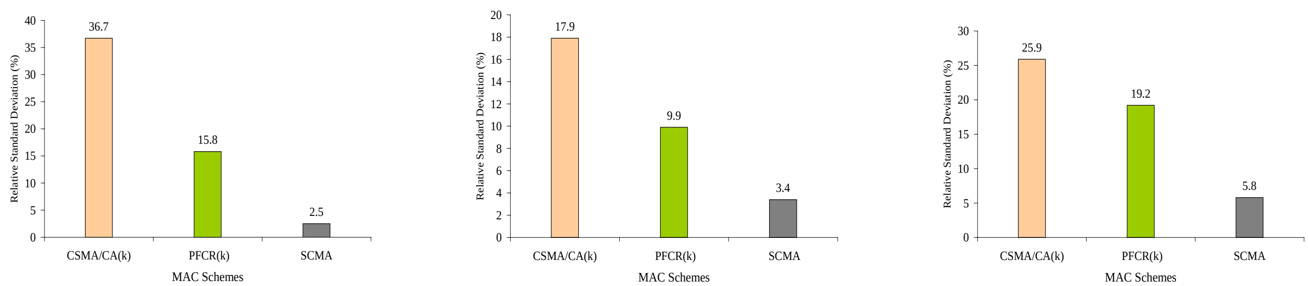
a) Scenario 1 b) Scenario 2 c) Scenario 3

Fig. 9. Flow Contention Topologies



a) Scenario 1 b) Scenario 2 c) Scenario 3

Fig. 10. Throughput



a) Scenario 1 b) Scenario 2 c) Scenario 3

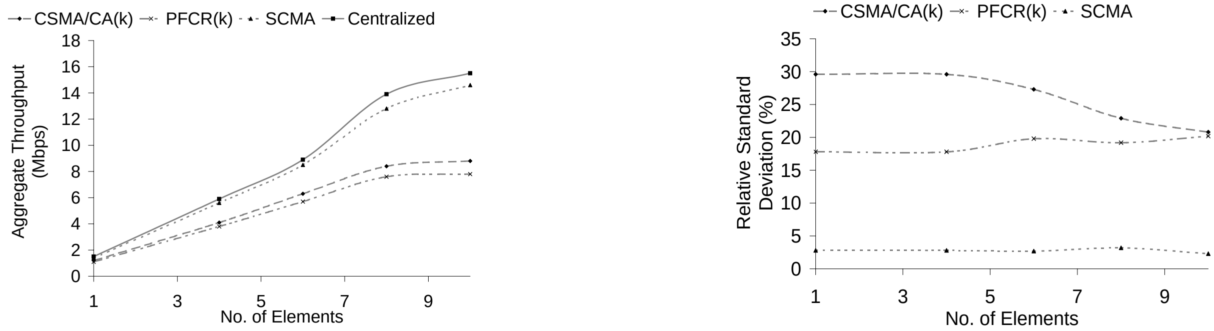
Fig. 11. Unfairness (%)

flow contention topology made up of both red and white links as shown in Figure 9(b). The link *c* belongs to two cliques and is the only red link in this topology. Every node in the network has an array of five elements. The edge weight between the white links *a* and *b* is 0.5. This indicates that the links *a* and *b* can potentially use twice their fair share in the clique since they will require to sacrifice only one stream for every two streams used by the other white link. The result in Figure 10(b) indicates that SCMA achieves a net gain of about 40% over PFCR(k) of which about 15% is contributed solely by the additional resources made available at the links *b* and *c* due to flexible interference suppression. Since we have a clique of size six of which five links are white, this represents the case of maximum stream control gain that can be achieved with five elements. All the five links operate simultaneously on a single stream each to provide a gain of 25%.

- *Scenario 3*

This scenario is also used to highlight the gains from performing both stream control and flexible interference suppression, but with flexible interference suppression being the dominating contributor. In Figure 9(c) we consider a single red link that is a part of three cliques. Every node has an array of six elements each. To highlight the significant improvement that can be obtained by performing flexible interference suppression, we consider three sets of weakly interfering links with edge weights of 0.5 each. The result in Figure 10(c) indicates that SCMA achieves a gain of about 46% of which a significant portion of around 30% is contributed purely by the additional resources made available at the outer links of the topology. Though the outer links have a six-element array, they end up using only three streams since the cliques are all of size three with two white links in each. Hence the maximum gain of stream control for a six element case (when six links use one stream each) cannot be obtained in this case. However, note that CSMA/CA(k) or PFCR(k) cannot leverage the gain of flexible interference suppression because they do not perform stream control.

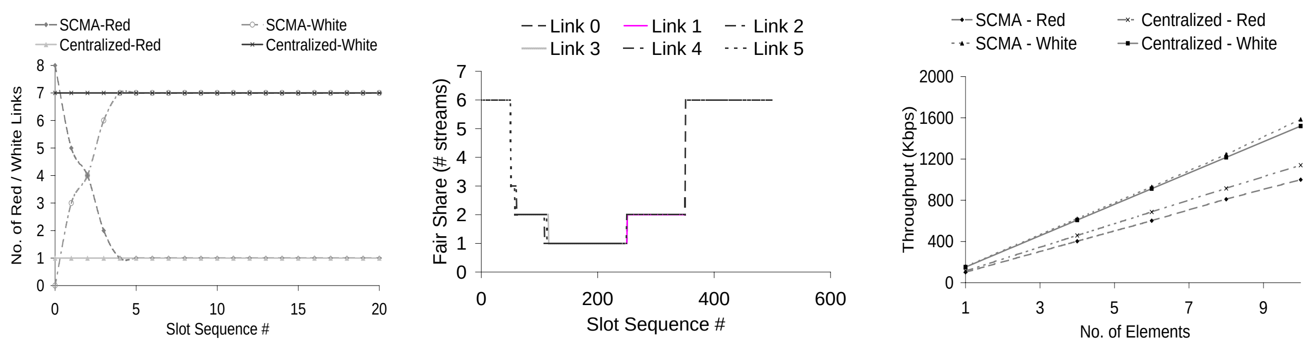
In terms of fairness, Figure 11 presents the relative standard deviation for the representative topologies considered thus far. Since the centralized algorithm is ideally fair we present the relative standard deviation of the throughput distributions obtained by the various schemes with respect to the centralized scheme. It can be observed that PFCR(k) reduces the degree of unfairness by over 50% as compared to CSMA/CA(k). Moreover SCMA further provides a better degree of fairness over PFCR(k) by over 50%. This could be attributed to the fact that SCMA, by virtue



(a) Throughput

(b) Unfairness

Fig. 12. Typical Network Topologies



(a) Coloring Convergence (slot sequences)

(b) Fair Share Convergence (slots)

(c) Throughput [Red/White]

Fig. 13. Microscopic Results

of performing stream control, is able to allocate streams to the links on a much finer granularity than CSMA/CA(k) or PFCR(k) wherein the granularity of stream allocation is always *k* streams.

A.2 Typical Network Topologies

For more generic topologies, we consider a random network topology consisting of 50 nodes distributed uniformly over an area of 750m by 750m. The random scenarios are generated using the *setdest* tool and the results are averaged over several seeds and also across different number of elements present in the antenna array. Mobility is not considered in these scenarios. Figures 12(a) and (b) present the results for the various schemes in comparison with the centralized scheme. The utilization in Figure 12(a) shows an increasing trend with the number of elements for all the schemes. However, for CSMA/CA(k) and PFCR(k) the improvement in utilization obtained for every extra element employed starts to decrease due to the absence of stream control. Hence the scalability of SCMA in terms of utilization is better when compared to CSMA/CA(k) and PFCR(k). The fairness results are presented in Figure 12(b). Although we are not able to conclude any relation about the trend in fairness with respect to the number of elements, the result clearly shows that SCMA provides an improvement of around 15% to 25% when compared to CSMA/CA(k) and PFCR(k).

B. Microscopic results

• *Convergence of Coloring:* In order to leverage the advantage of stream control effectively, a node must first color it's link correctly. Hence, we investigate the efficiency of the coloring mechanism employed by SCMA. We consider the topology shown in Figure 9(b) for this purpose. The topology consists of one red link and seven white links. The result presented in Figure 13(a) indicates the amount of time it takes for SCMA to complete the coloring. The x-axis is measured in slot sequences, where each slot sequence corresponds on an average to the time taken for all the members of the largest clique to get access to the channel once. Note that every link takes atleast two slot sequences in SCMA to color itself and accordingly start using the appropriate number of streams. Hence we measure the convergence time in terms of slot sequences. The result shows that the coloring done by SCMA converges to that of the centralized scheme in only about three to four slot sequences.

• *Convergence to Fair Share - Network Dynamics:* The second important aspect of the distributed mechanism after coloring is to ensure that the colored links use the appropriate number of streams. This is not a problem for the red links which operate on all the streams. Hence, we study the impact of dynamics in the network on the estimation of fair share by the white links. We consider the dynamic topology of a single clique whose clique size basically varies with time as new links come in and existing links leave the clique. All the nodes possess a six element array each. The result is presented in Figure 13(b). Essentially, the topology is created as a single-link clique, builds to a three-link clique at the end of 50 slots and finally to a six-link clique at the end of 150 slots. The single link initially operates on all six streams. With the entry of two more links at the 50^{th} slot, the fair share converges to two streams within 5 slots. Then with the entry of three more links at the 100^{th} slot, the fair share further reduces and converges to one stream within about ten to fifteen slots ($<$ 3 slot sequences). We also consider the case of existing links leaving the clique wherein three flows (links) leave at the end of 250 slots and two more at the end of 350 slots. In both these cases, the remaining links are able to utilize the resources left over by the departing links almost instantaneously. Although time is measured directly in terms of slots here as opposed to slot sequences, we can easily translate the result for slot sequences. Note that the convergence in the case of white links leaving the clique happens much faster than the case of new white links coming in. This is because when new white links come in, they first color themselves before estimating their fair share and the existing white links in the clique then converge to this new fair share. On the other hand, when white links leave the clique, the remaining white links locally estimate the new fair share in the clique, based on the free resources remaining at the nodes after accounting for the transmissions in the clique, without the need for any coloring.

- *Red-White Deviation:* Earlier in the section we presented aggregate throughput results for the random scenarios. We now present the throughput obtained by the red and white links in SCMA, averaged across the seeds and compare it with that of the centralized algorithm. This is to ensure that SCMA provides a fair distribution of throughput not just on an aggregate basis but also individually with respect to the red and white links. This is captured in Figure 13(c). It can be seen that the throughput obtained by the red and white links in SCMA are close enough to those obtained in the centralized algorithm. However, the white links in SCMA obtain a slightly higher throughput than the white links in the centralized scheme. We believe that the co-ordinated scheduling of all the white links in the same clique is responsible for this.

VII. RELATED WORK

To the best of our knowledge we are not aware of any existing MAC protocols for ad-hoc networks with MIMO links in the literature. However, there has been significant contribution in the related area of directional (switched beam) antennas. Hence we provide a short synopsis of the related work in the area of MAC protocol for ad-hoc networks with directional antennas.

[15] considers a cellular scenario in which the base-station is equipped with a multi-beamforming antenna, and discusses the improvement in static SDMA/TDMA system capacity on performing dynamic slot assignment. However, the scope of the work does not include ad-hoc networks. [7], [8], [16], [17], [18] propose MAC protocols for ad-hoc networks with directional (switched beam) antennas. While directional antennas offer more spatial flexibility when compared to omni-directional antennas, they are more restrictive than the MIMO links we consider in this paper. [7] and [19] use the directive gain provided by directional antennas for the purpose of range extension and minimization of power consumption respectively. However the focus is not on MIMO links. [12] presents the proportional fairness model for the problem of channel allocation in wireless ad-hoc networks, which we have augmented with several design optimizations that are specific to the MIMO environment. Finally, [20] presents a polling based media access protocol for the environment of cellular networks with smart antennas being employed at the base stations. But the scheme cannot be extended to work for ad-hoc networks due to its centralized nature.

VIII. CONCLUSIONS

We have identified the potential advantages of MIMO links in wireless ad-hoc networks. The problem of fair channel allocation for the target environment has been presented and the key optimization considerations for the design of an ideal MAC protocol for such an environment have been discussed. We have presented a centralized algorithm that incorporates the propitious characteristics of MIMO links as well as the optimizations possible in relation to MIMO environments. We have also proposed a distributed MAC protocol called the SCMA that approximates the centralized version. The proposed SCMA scheme clearly outperforms CSMA/CA(k) and PFCR(k) protocols and performs nearly as well as the centralized algorithm suggesting that it is beneficial to employ SCMA as the MAC layer protocol for MIMO environments. We are also currently investigating the possibility of moving the contention resolution mechanism of SCMA to the backoff domain.

REFERENCES

[1] G. J. Foschini and M. J. Gans, “On Limits of Wireless Communications in a Fading Environment When Using Multiple Antennas,” *Wireless Personal Communications*, vol. 6, pp. 311–335, 1998.

[2] G. J. Foschini, “Layered Space-Time Architecture for Wireless Communication,” *Bell Labs Technical Journal*, vol. 6, pp. 311–335, Mar. 1998.

[3] J. C. Liberti and T. S. Rappaport, *Smart Antennas for Wireless Communications: IS-95 and Third Generation CDMA Applications*, Prentice Hall, 1999.

[4] D. Gesbert, M. Shafi, D. Shiu, P. J. Smith, and A. Naguib, “From Theory to Practice: An Overview of MIMO Space-Time Coded Wireless Systems,” *IEEE JSAC*, vol. 21, no. 3, pp. 281–301, Apr 2003.

[5] J. H. Winters, J. Salz, and R. D. Gitlin, “The Impact of Antenna Diversity on the Capacity of Wireless Communication Systems,” *IEEE Transactions on Communications*, vol. 42, no. 2, 1994.

[6] J. H. Winters, “Optimum Combining in Digital Mobile Radio with Cochannel Interference,” *IEEE JSAC*, vol. 2, pp. 528–539, July 1984.

[7] R. R. Choudhury, X. Yang, R. Ramanathan, and N. H. Vaidya, “Using Directional Antennas for Medium Access Control in Ad Hoc Networks,” in *Proceedings of ACM MOBICOM*, Atlanta, 2002.

[8] A. Nasipuri, S. Ye, and R.E. Hiromoto, “A MAC Protocol for Mobile Ad Hoc Networks Using Directional Antennas,” in *IEEE Wireless Communication and Networking Conference(WCNC)*, 2000.

[9] K. Sundaresan, R. Sivakumar, and M. A. Ingram, “Medium Access Control Protocol for Ad Hoc Networks With MIMO Links,” in *Technical Report*, GNAN Research Group, May 2003.

[10] A. Hottinen, R. Wichman, and O. Tirkkonen, *Multiantenna Transceiver Techniques for 3G and Beyond*, Wiley, John and Sons, Feb. 2003.

[11] M. F. Demirkol and M. A. Ingram, “Control Using Capacity Constraints for Interfering MIMO Links,” *Proc. of the Int. Symp. on Personal, Indoor and Mobile Radio Communications*, vol. 3, pp. 1032–1036, Sep. 2002.

[12] T. Nandagopal, T-E. Kim, X. Gao, and V. Bhargavan, “Achieving MAC Layer Fairness in Wireless Packet Networks,” Proceedings of ACM MOBICOM 2000, Aug 2000.

[13] D. Rose, R. E. Tarjan, and G. Lueker, “Algorithmic Aspects of Vertex Elimination on Graphs,” *SIAM Journal*, pp. 5:146–160, 1976.

[14] D. R. Fulkerson and O. A. Gross, “Incidence Matrices and Interval Graphs,” *Pac. J. Math*, vol. 15, pp. 835–855, 1965.

[15] F. Shad, T. Todd, V. Kezys, and J. Litva, “Indoor SDMA Capacity Using A Smart Antenna Basestation,” in *IEEE ICUPC*, 1997, pp. 868–872.

[16] Y.B. Ko, V. Shankarkumar, and N.H. Vaidya, “Medium Access Control Protocols Using Directional Antennas in Ad Hoc Networks,” in *Proceedings of IEEE INFOCOM*, Mar. 2000.

[17] R. Ramanathan, “On the Performance of Ad Hoc Networks with Beamforming Antennas,” in *Proceedings of ACM MOBIHOC*, Oct. 2001.

[18] M. Sanchez, T. Giles, and Jens Zander, “CSMA/CA with Beam Forming Antennas in Multi-hop Packet Radio Networks,” in *Swedish Workshop on Wireless Ad-hoc Networks*, Stockholm, Mar. 2001.

[19] A. Nasipuri, K. Li, and U. R. Sappidi, “Power Consumption and Throughput in Mobile Ad Hoc Networks Using Directional Antennas,” in *Proceedings of IEEE IC3N*, Oct. 2002.

[20] S. Krishnamurthy, A. Acampora, and M. Zorzi, “Polling based Media Access Protocols for Use With Smart Adaptive Array Antennas,” in *IEEE Transactions on Networking*, 2001.