On Distributed Algorithms that Enforce Proportional Fairness in Ad-hoc Wireless Networks

Nikhil Singh
Electrical and Computer Engineering
University of Illinois at Urbana-Champaign
Urbana, IL 61820
Email: nsingh1@uiuc.edu

Ramavarapu S. Sreenivas
Industrial and Enterprise Systems Engineering & CSL
University of Illinois at Urbana-Champaign
Urbana, IL 61820
Email: rsree@uiuc.edu

In this paper we consider an Ad-hoc Wireless Network [3] that carries several flows between various source-destination pairs under the slotted-time MAC protocol – *STMAC* [15]. In specific we are interested in a distributed scheme for the assignment of the network's resources among flows that is fair in terms of *flow-success probabilities*.

The literature contains several references to fairness and its impact on the network performance. For example, references [14], [17] present a list of modifications that eliminate the unfairness that is commonly seen in the 802.11-MAC. The literature also contains a large volume of references (cf. [6], [7], [9], [10], for example) where it is supposed that each network flow/link is associated with a concave utility-function that could be maximized. Specifically, for proportional fairness, it is assumed that the utility function has the form of $\log x$, where x denotes the flow rates [6]. It is of interest to schedule individual transmissions on the links so as to maximize the sum of the utilities of the consumers. To achieve fairness, the schemes outlined in the above mentioned references use a penalty function which is updated by some form of feedback from the network. Using an appropriately defined cost that is implicitly dependent on the requested rates of each host within a neighborhood, the penalty is typically the total cost of all nodes in the network. A node maximizes (its view of) a common performance function that is the difference between the total utility and the penalty. The procedure presented in this paper does not requires any estimation of penalty functions on the flows, such as the loss rates or delays, and just works on the local two-hop information for the class of protocols identified in the next paragraph, where we address the issue of providing end-to-end flow based proportional fairness [6] where each node accesses the wireless medium with a probability so as to maximize the product of end-to-end *flow-success probabilities* of all flows. Since each node's access to the wireless channel is dependent on the underlying MAC protocol, the problem of providing proportional fairness must be considered in context of a specific protocol. Taking this into consideration, we define a family of protocols, and analyze their proportional fairness properties.

The rest of the paper is organized as follows. Section I presents an analysis of flow-based proportional fairness, where we derive the expression for floor access probabilities. Section II discusses a distributed implementation to achieve flow-based proportional fairness, when every host is allowed to retransmit packets.Section III presents some observations on the parallelization of the strictly sequential algorithm of section II. Section IV contains the formal specification of the ST-MAC protocol, and section V contains the details of the experimental comparisons of the ST-MAC protocol and the 802.11-MAC.

I. Notions and Definitions

We assume the following facts about the MAC protocol (1) Time is divided into slots of equal duration. (2) A successful transmission in a slot implies collision free data transmission in a time-slot. (3) The transmitting nodes always have data packets to transmit (i.e. we do not consider the arrival rates of packets for different flows, and assume that all flows have packets to transmit at all times). (4) All nodes in the network are assumed to have infinite-length buffers. (5) A data-packet is dropped following r-many attempts at retransmission. (6) We only consider unicast flows for our derivations. A wide variety of slotted-time protocols, like *ST-MAC* [15], etc., fit this description. These protocols typically involve a reservation slot that is tied to a specific data slot. Senders and receivers exchange appropriate messages in the reservation slot. The successful completion of the reservation process implies a collision free data transmission in the appropriate data slot under appropriate assumptions on the slot durations and the network topology (cf. [15]). We also assume there is a unique route for each flow within the network (which would be the case if we used AODV [12] as the routing protocol, for example). We denote this family of protocols as *protocol-family A*.

An ad hoc wireless network carrying a collection of CBR-flows, is represented as an undirected graph $G = (V, E)$, where V represents the set of *nodes*, and $E \subseteq V \times V$ is a symmetric[1] relationship that represents the set of bi-directional *links*. We assume all links of the network have the same capacity, which is normalized to unity. The 1-hop neighborhood of node $i \in V$ is represented by the symbol $N(i)$. When a node i communicates with a node $j \in N(i)$,

[1]That is, $(i, j) \in E \Leftrightarrow (j, i) \in E$.

we can represent it as an appropriate orientation of the link (i,j) in E, where i is the origin and j is the terminus. The context in which $(i,j) \in E$ is used should indicate if it is to be interpreted as a directed edge with i as origin and j as terminus. The set of CBR-flows that use a link $(i,j) \in E$ with i (j) as origin (terminus) is denoted by $\mathcal{F}(i,j)$.

When node i intends to transmit data to node $j \in N(i)$ for the l-th flow ($l \in \mathcal{F}(i,j)$) under a protocol from protocol-family A, it would transmit data in the appropriate time-slot with probability $p_{i,j,l}$. The expression $\mathcal{P}_{i,j} = \sum_{l \in \mathcal{F}(i,j)} p_{i,j,l}$ denotes the probability that node i transmits data to node j, and $\mathcal{P}_i = \sum_{j \in V} \mathcal{P}_{i,j}$, denotes the probability that node i will be transmitting to some node in its 1-hop neighborhood for some flow. The probabilities $p_{i,j,l}$'s should be chosen such that $\mathcal{P}_i$ is not greater than unity for any node $i \in V$. The probability of successful data transmission over link $(i,j) \in E$ for flow $l \in \mathcal{F}(i,j)$ with no retries, denoted by $\mathcal{S}_{i,j,l}$, is given by the expression

$$\mathcal{S}_{i,j,l} = p_{i,j,l} \left(1 - \sum_{(j,m) \in E, n \in \mathcal{F}(i,m)} p_{j,m,n} \right) \times \prod_{o \in N(j) - \{i\}} \left(1 - \sum_{(o,p) \in E, q \in \mathcal{F}(o,p)} p_{o,p,q} \right) \quad (1)$$

The probability of successful data transmission over link $(i,j) \in E$ for flow $l \in \mathcal{F}(i,j)$ with r-retries is

$$\left(1 - (1 - \mathcal{S}_{i,j,l})^{r+1} \right)$$

We adopt the the logarithm of the product of the success probabilities of each flow as the utility-function. The expression for the sum of the individual utility functions in an ad hoc network that uses a protocol from the protocol-family A is given by equation 2 below.

$$\begin{aligned} \Delta &= \prod_{(i,j) \in E, l \in \mathcal{F}(i,j)} \{1 - (1 - \mathcal{S}_{i,j,l})^{r+1}\}, \\ \Rightarrow \log \Delta &= \sum_{(i,j) \in E, l \in \mathcal{F}(i,j)} \log \{1 - (1 - \mathcal{S}_{i,j,l})^{r+1}\} \quad (2) \end{aligned}$$

The objective therefore is to maximize the term in equation 2 by appropriate assignments to the individual probabilities $p_{i,j,l}$ defined earlier.

Lemma 1: For any $(i,j) \in E$, any scheme that maximizes equation 2, subject to the constraint that $\sum_{l \in \mathcal{F}(i,j)} p_{i,j,l}$ is constant, will assign the same value to $p_{i,j,l}$ for all $l \in \mathcal{F}(i,j)$ (i.e. $p_{i,j,l_1} = p_{i,j,l_2}, \forall l_1, l_2 \in \mathcal{F}(i,j)$).

Proof: (Sketch) Equation 2 can be reorganized as

$$\sum_{l \in \mathcal{F}(i,j)} \log \{1 - (1 - \mathcal{S}_{i,j,l})^{r+1}\} + \sum_{(i_1,j_1) \in E - \{(i,j)\}, l_1 \in \mathcal{F}(i_1,j_1)} \log \{1 - (1 - \mathcal{S}_{i_1,j_1,l_1})^{r+1}\}$$

The second expression in the equation above remains unaffected as $p_{i,j,l}$ is varied such that $\sum_{l \in \mathcal{F}(i,j)} p_{i,j,l}$ is constant. The statement of the lemma is established using an exchange-argument on the first expression and equation 1. ■

From lemma 1 we note that all flows that use a link $(i,j) \in E$ should be treated alike. In effect, this permits us to drop the third-subscript (which denotes the flow) from $p_{i,j,l}$ and use the term $p_{i,j}$ in its place, where $p_{i,j}$ is the data transmission probability for any flow in $\mathcal{F}(i,j)$. *Mutatis Mutandis*, the flow-specific, third subscript from equations 1 and 2 can be eliminated, which results in

$$\Delta = \prod_{(i,j) \in E} \{1 - (1 - \mathcal{S}_{i,j})^{r+1}\}^{card(\mathcal{F}(i,j))},$$
$$\Rightarrow \log \Delta = \sum_{(i,j) \in E} card(\mathcal{F}(i,j)) \log \{1 - (1 - \mathcal{S}_{i,j})^{r+1}\} \quad (3)$$

The only reason a host $i \in V$ assigns a zero-value to $\mathcal{P}_{i,j}$ would be when it has no outgoing flows on $(i,j) \in E$. Similarly, a host $i \in V$ assigns a value of unity to $\mathcal{P}_{i,j}$ only if the flows in $\mathcal{F}(i,j)$ define the entire set of flows in $N(i)$ (i.e. no 1-hop neighbor of i other than j is involved with a flow). Essentially, these instances can be spotted a priori, and the variable $\mathcal{P}_{i,j}$'s that are assigned either the zero-value or the unity-value can be removed from the search process outlined in subsequent text. Following this, we can assume without loss in generality that the maximizer for equation 3 is strictly in the interior of the appropriately dimensioned unit-hypercube.

A. Zero Retries Case ($r = 0$)

There might be reasons to consider the case when no retries are permitted (i.e. $r = 0$). For instance, flows encoded using *digital fountain codes* [2], [8] are robust to erasures. These codes guarantee that the receipt of $K(1+\epsilon)$ coded-packets, sent over a channel with an erasure probability of ϵ, at the receiver is sufficient to decode K-many source-packets with high probability. If a particular packet transmission is unsuccessful at some link in the route, there is no need to attempt a retransmission. For this case Lemma 1 can be tightened to Lemma 2 as shown below.

Lemma 2: For any $i \in V$, any scheme that maximizes equation 2, subject to the constraint that $\sum_{j \in N(i), l \in \mathcal{F}(i,j)} p_{i,j,l}$ is constant, will assign the same value to $p_{i,j,l}$ for $j \in N(i)$ and $l \in \mathcal{F}(i,j)$, (i.e. $p_{i,j_1,l_2} = p_{i,j_3,l_4}, \forall j_1, j_3 \in N(i), \forall l_2 \in \mathcal{F}(i,j_1), l_4 \in \mathcal{F}(i,j_3)$).

Lemma 2 states that when no retries ($r = 0$) are permitted, any scheme that enforces proportional fairness should assign the same transmission-probability to each flow that either originates from i, or, is routed through it.

Before we write the expression for the product of the success probabilities of the k-many flows for the zero-retries case, it would be useful to identify (the not necessarily disjoint) subsets of these flows vis-á-vis an arbitrary node $i \in V$. As mentioned in the previous paragraph, let us suppose

1) m_i-many of the flows either originate, or, get routed through node $i \in V$,
2) n_i of them either have i as their final-destination, or, are routed through i, and,

3) q_i of them are flows that either terminate, or, get routed through some member in $N(i)$.

Theorem 3: When no retries are permitted, for protocol from protocol-class A to be proportionally fair, the probability with which a node $i \in V$ will transmit data for any flow that either originates, or, gets routed through it is given by $\mathcal{P}_i = \frac{m_i}{n_i+q_i}$.

Proof: The expression for the product of k-many flow success-probabilities would be

$$\Delta = \prod_{i \in V} (\frac{\mathcal{P}_i}{m_i})^{m_i} \times (1-\mathcal{P}_i)^{n_i} \times (1-\mathcal{P}_i)^{(q_i-m_i)}$$
$$\Rightarrow \log \Delta = \sum_{i \in V} (m_i \log \mathcal{P}_i - m_i \log m_i$$
$$+ n_i \log(1-\mathcal{P}_i) + (q_i - m_i) \log(1-\mathcal{P}_i)) \quad (4)$$

The above mentioned expression is strictly concave with respect to the $\mathcal{P}_i$'s ($i \in V$). The value of $\mathcal{P}_i$'s that maximize the above concave expression can be obtained by differentiating the expression for $\log \Delta$ with respect to $\mathcal{P}_i$'s and equating the result to zero. This results in

$$\frac{m_i}{\mathcal{P}_i} - \frac{n_i}{1-\mathcal{P}_i} - \frac{q_i - m_i}{1-\mathcal{P}_i} = 0$$
$$\Rightarrow \frac{m_i}{\mathcal{P}_i} = \frac{n_i + q_i - m_i}{1-\mathcal{P}_i}$$
$$\Rightarrow \mathcal{P}_i = \frac{m_i}{n_i+q_i} \Rightarrow \frac{\mathcal{P}_i}{m_i} = \frac{1}{n_i+q_i} \quad (5)$$

■

The expression shown above is the probability that $i \in V$ will send a data packet for any flow that either originates, or, gets routed through it. Note that the terms n_i and q_i can be inferred from exchanges between i and nodes in $N(i)$. A single-hop interpretation of equation (5) would result in the expression derived in the literature for proportional fairness in S-ALOHA networks [5]. Since every flow that originates, or, gets routed through a host $i \in V$ is also a flow that either terminates, or, gets routed through a member in $N(i)$, it follows that $m_i \leq q_i$, which in turn guarantees $\mathcal{P}_i \leq 1$. The choice of $p_{i,j,l}$'s derived from theorem 3 is not necessarily optimal when multiple retries are permitted (i.e. $r \neq 0$).

B. An Illustration

Consider the network shown in figure 1 with two flows. One of the flows, *flow 1*, originates from v_7 and terminates at v_4, and the route assigned to this flow is $v_7 \rightarrow v_1 \rightarrow v_2 \rightarrow v_3 \rightarrow v_4$. The other flow, *flow 2*, originates from v_0 and terminates at v_6, and the route assigned to this flow is $v_0 \rightarrow v_1 \rightarrow v_2 \rightarrow v_5 \rightarrow v_6$. Theorem 3 suggests the following assignments for the node probabilities: $\mathcal{P}_0 = \frac{1}{0+2} = 0.5, \mathcal{P}_1 = \frac{2}{2+2} = 0.5, \mathcal{P}_2 = \frac{2}{2+(2+1+1)} = 0.3333, \mathcal{P}_3 = \frac{1}{1+(1+2)} = 0.25, \mathcal{P}_4 = 0, \mathcal{P}_5 = \frac{1}{1+(1+2)} = 0.25, \mathcal{P}_6 = 0$ and $\mathcal{P}_7 = \frac{1}{0+2} = 0.5$, under the assumption that $r = 0$.

This in turn suggests $p_{0,1,1} = 0, p_{0,1,2} = \mathcal{P}_0 = 0.5, p_{7,1,1} = \mathcal{P}_7 = 0.5, p_{7,1,2} = 0, p_{1,2,1} = p_{1,2,2} = \frac{\mathcal{P}_1}{2} = 0.25, p_{2,3,1} = \frac{\mathcal{P}_2}{2} = 0.1667, p_{2,3,2} = 0, p_{2,5,1} = 0, p_{2,5,2} = \frac{\mathcal{P}_2}{2} = 0.1667, p_{3,4,1} = \mathcal{P}_3 = 0.25, p_{3,4,2} = 0, p_{5,6,1} = 0$ and $p_{5,6,2} = \mathcal{P}_5 = 0.25$. For the case when $r = 14$, the product of the transmission success probabilities by using the above floor access probability assignment is 0.2285.

The true values of floor success probabilities, which can be calculated by optimizing the function given by equation 2 where $\mathcal{P}_i \in [0.01, 0.99]$, are : $\mathcal{P}_0 = 0.5, \mathcal{P}_1 = 0.4575, \mathcal{P}_2 = 0.2942, \mathcal{P}_3 = 0.1645, \mathcal{P}_4 = 0.01, \mathcal{P}_5 = 0.1645, \mathcal{P}_6 = 0.01$ and $\mathcal{P}_7 = 0.5$. This yields a product of success probabilities of 0.2682, which is a 12.79% improvement over the zero-retry based probabilities obtained from theorem 3. Using a large value for the number of retries (e.g. $r = 14$ as with this illustration) can result in significant end-to-end delays. Typically, r is chosen to be around 4 (which is the *Default long-retry limit* in 802.11-MAC). For small values of r, the probability assignments obtained from theorem 3 are close to the optimal allocation, specifically for large random networks. In the next section we study distributed algorithms that can maximize the product of the transmission success probabilities.

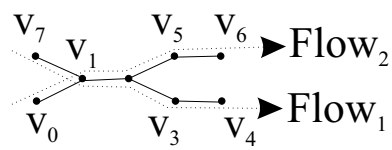

Fig. 1. An illustrative example.

II. Distributed Algorithms that Enforce Proportional Fairness

We now present a result from section 2.7 of [1] about the convergence of *Block Coordinate Descent* schemes. This result will be used to establish the existence of distributed procedure for the enforcement of Proportional Fairness for the *protocol family A* defined earlier.

Let $\mathcal{R}$ denote the set of reals, and let us suppose we intend to maximize $f : \mathcal{R}^n \rightarrow \mathcal{R}$ subject to some constraints expressed in terms of ranges of values that each component of the argument of $f(\bullet)$ can take. More specifically, we intend to maximize $f(x_1, x_2, \ldots, x_n)$ subject to the constraint $x_i \in \mathbf{X}_i$, for an appropriately defined closed, convex set $\mathbf{X}_i \subseteq \mathcal{R}$. Additionally, let us suppose that for each $i \in \{1, 2, \ldots, n\}$, the problem

$$\min f(x_1, \ldots, x_{i-1}, \xi, x_{i+1}, \ldots, x_n)$$
$$\text{subject to: } \xi \in \mathbf{X}_i$$

has at least one solution. The *block coordinate descent* (or, equivalently, a *nonlinear Gauss-Seidel*) method is an iterative scheme where the next iterate $(x_1^{k+1}, x_2^{k+1}, \ldots, x_n^{k+1})$ is obtained from the previous iterate $(x_1^k, x_2^k, \ldots, x_n^k)$ as follows for $i = 1, 2, \ldots, n$,

$$x_i^{k+1} \in arg \max_{\xi \in \mathbf{X}_i} f(x_1^{k+1}, \ldots, x_{i-1}^{k+1}, \xi, x_{i+1}^k, \ldots, x_n^k).$$

The following result presents sufficient conditions for the convergence of this iterative process.

Theorem 4: [1] Suppose f is continuously differentiable over $\mathbf{X} = \prod_{i=1}^{n} \mathbf{X}_i$. Furthermore, suppose that for each i and $\mathbf{x} \in \mathbf{X}$, the maximum

$$\max_{\xi \in \mathbf{X}_i} f(x_1, \ldots, x_{i-1}, \xi, x_{i+1}, \ldots, x_n)$$

is uniquely attained. Let $\{\mathbf{x}^k\}$ be the sequence generated by the block coordinate descent method outline above. Then, every limit point of $\{\mathbf{x}^k\}$ is a stationary point.

Specifically, if the function f is strictly concave over $\mathbf{X}$, the stationary point that is attained in the limit is the maximizer of f.

Since $p_{i,j,l} = \frac{\mathcal{P}_{i,j}}{card(\mathcal{F}(i,j))}, j \in N(i), l \in \mathcal{F}(i,j)$, we might as well suppose that each host $i \in V$, has associated with it a collection of values for $\mathcal{P}_{i,j}, j \in N(i)$. These values can be modified using the block coordinate descent approach outlined above. That is, hosts update $\mathcal{P}_{i,j}$-terms instead of $p_{i,j,l}$-terms ($l \in \mathcal{F}(i,j)$), this reduces the dimensionality of the problem in most cases. To establish the convergence of this procedure we first show that the expression shown in equation 3 is strictly concave with respect to $\mathcal{P}_{i,j}$-terms.

Theorem 5: The expression

$$\log \Delta = \sum_{(i,j) \in E} card(\mathcal{F}(i,j)) \log \{1 - (1 - \mathcal{S}_{i,j})^{r+1}\}$$

is strictly concave with respect to $\mathcal{P}_{i,j}$-terms.

Proof: (Sketch) The routes assigned to the flows are unique. So, the $card(\mathcal{F}(i,j))$ term that multiplies each summand term in equation 3 can be viewed as a constant. We can show the Hessian of the term

$$\log \{1 - (1 - \mathcal{S}_{i,j})^{r+1}\} \tag{6}$$

with respect to the $\mathcal{P}_{i,j}$-terms is negative definite (The final version of the paper will contain the relevant details). This establishes the strict concavity of each summand in equation 3. Since a finite sum of strictly concave functions is also strictly concave, the result follows. ■

It is relatively straightforward to show that the expression in equation 6 is strictly concave with respect to a specific $\mathcal{P}_{i,j}$-term (while the others are held constant). This in turn would imply the expression in 3 is also strictly concave with respect to a specific $\mathcal{P}_{i,j}$-term (while the others are held constant). These observations and theorem 4 suggests that the block coordinate descent approach will yield the values of $\mathcal{P}_{i,j}$ for each $(i,j) \in E$ that maximize the expression in equation 3.

The iterative process proceeds as follows – host i computes the value of $\mathcal{P}_{i,j}$ that maximizes the expression in equation 3, where all variables other than $\mathcal{P}_{i,j}$ are assumed to be constant. This value is then communicated to all (relevant) hosts in the network, which is then followed by a repetition of the above process at a different host possibly. In the next section we show that this "strict synchronization" among the hosts during iterative process, where at most one host modifies its probability-value at a given time can be relaxed significantly.

III. Parallelization of nonlinear Gauss-Seidel iterations

Two hosts that are not within a 2-hop neighborhood can simultaneously update probability values associated with them. That is, during the period when a host $i \in V$ is in the process of updating $\mathcal{P}_{i,j}, j \in N(i)$, all values of $\mathcal{P}_{k,l}, k \in \{N(i) \cup N(N(i))\}, (k,l) \in E$, have to remain unchanged for the block coordinate descent algorithm outlined in the previous section to converge to the global optimum.

Theorem 6: In the $(k+1)$-th step of the Gauss-Seidel iterations of the coordinated block descent algorithm of section II for any node i to update values of $\mathcal{P}_{i,j}$ it only requires the values of $\mathcal{P}_{k,l}, k \in \{N(i) \cup N(N(i))\}$ and $(k,l) \in E$.

Proof: (Sketch) A synchronous implementation of the algorithm requires that any node m does not carry out its $(k+1)$-th update without first receiving the results of k-th or $k+1$-th update from the nodes whose access probability appears in the function f_m, where $f_m = f_{m_1} + f_{m_2} + f_{m_3}$ and can be recognized in equation 3 as follows

$$\begin{aligned}
\log \Delta = & \underbrace{\sum_{(m,j) \in E} card(\mathcal{F}(m,j)) \log \{1 - (1 - \mathcal{S}_{m,j})^{r+1}\}}_{f_{m_1}} \\
& + \underbrace{\sum_{(i,m) \in E} card(\mathcal{F}(i,m)) \log \{1 - (1 - \mathcal{S}_{i,m})^{r+1}\}}_{f_{m_2}} \\
& + \underbrace{\sum_{\underbrace{(i,j) \in E}_{i \in \{N(N(m)) - \{N(m), m\}\}}} card(\mathcal{F}(i,j)) \log \{1 - (1 - \mathcal{S}_{i,j})^{r+1}\}}_{f_{m_3}} \\
& + \underbrace{\sum_{(i,j) \in E} card(\mathcal{F}(i,j)) \log \{1 - (1 - \mathcal{S}_{i,j})^{r+1}\}}_{remaining\ terms}
\end{aligned}$$

In any iteration, for node m to maximize the function $\log \Delta$, it only needs to maximize f_m as all the other terms are independent of $\mathcal{P}_{m,j}, (m,j) \in E$ and hence while keeping the other variables in f_m as constant, maximizing $f_m \Rightarrow$ maximizing $\log \Delta$ with respect to $\mathcal{P}_{m,j}$'s ,$(m,j) \in E$.

Looking at the above expression we can see that f_m only depends on the $\mathcal{P}_{i,j}$'s, $(i,j) \in E$ of the two-hop neighbors of node m as $\mathcal{S}_{i,j}, (i,j) \in E$ is the link success probability and depends only on two-hop $\mathcal{P}_j$'s, $j \in V$, hence the result follows.

■

The result of the parallel distributed Gauss-Seidel algorithm applied to the example given in section I-B is shown in table I

Cycles	Access Probability Values								Δ
	$\mathcal{P}_0$	$\mathcal{P}_1$	$\mathcal{P}_2$	$\mathcal{P}_3$	$\mathcal{P}_4$	$\mathcal{P}_5$	$\mathcal{P}_6$	$\mathcal{P}_7$	
Initial values same as zero-retry floor access probabilities	0.5	0.5	0.3334	0.25	0.01	0.25	0.01	0.5	0.2285
Values After first cycle of iterations	0.5	0.4866	0.2969	0.1628	0.01	0.1671	0.01	0.5	0.2667
Values After second cycle of iterations	0.5	0.4578	0.2942	0.1644	0.01	0.1645	0.01	0.5	0.2682
Values After third cycle of iterations	0.5	0.4575	0.2942	0.1645	0.01	0.1645	0.01	0.5	0.2682
Optimal values for example in fig 1	0.5	0.4575	0.2942	0.1645	0.01	0.1645	0.01	0.5	0.2682

TABLE I

DISTRIBUTED GAUSS-SIEDEL ALGORITHM APPLIED TO THE EXAMPLE OF SECTION I-B.

IV. IMPLEMENTATION DETAILS AND PERFORMANCE ANALYSIS

A. ST-MAC Protocol Specification

In this section we present a brief overview of the *ST-MAC* protocol, which is a subset of HRMA protocol [16]. Additional details can be found in [15]. The proper execution of this protocol requires the following: (1) when multiple packets arrive at a host, the host hears noise (i.e. there is no packet-capture mechanism in the physical-layer); (2) all links are bi-directional; (3) host cannot transmit and receive packets simultaneously; and (4) all nodes in the network are synchronized.

A single super-frame of the *Slotted-Time MAC* (ST-MAC) protocol consists of a fixed-number of sub-frames and a SYNC frame consisting of three SYNC-slots as shown in Figure 2 (these are design parameters which can be modified depending on the network topology). Each sub-frame consists of a *Request-to-send* slot (RTS-slot), a *Clear-to-send* slot (CTS-slot) and a *data* slot (DATA-slot). The RTS- and CTS-slot within a sub-frame correspond to the DATA-slot within the same sub-frame. The structures of the RTS-, CTS- and DATA-packets are identical to that of 802.11-MAC [11]. The SYNC-packets, are used to broadcast the time for the start of new super-frame. A new node only joins the network at the beginning of new super-frame after it has this information of when the new super-frame is going to begin. Let us suppose

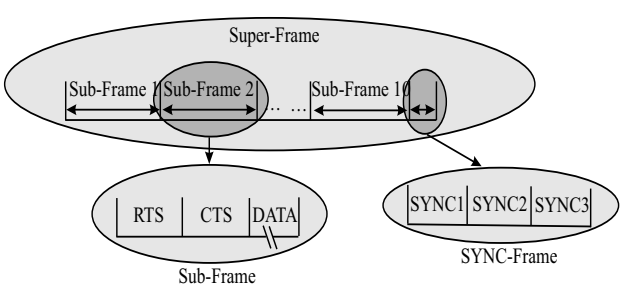

Fig. 2. A Super-Frame in the ST-MAC Protocol.

the maximum single-hop transmission delay of the network is w; the transmission/reception time for a packet of size m is $x(m)$; the maximum turnaround time (i.e. the time it takes for a host to switch between transmitting and receiving modes) is y; and the processing time for a packet of size m is $z(m)$. We set the slot-time for the packet of size m to $(2x(m) + y + w + z(m))$. The ST-MAC protocol assumes the network satisfies the inequality $y + w + z(m) < x(m)$. That is, the sum of the turnaround time, transmission delay and processing time is strictly less than the transmission/reception time of a packet. All data packets are assumed to be of the same size. The slot-lengths of the RTS-, CTS-, DATA- and SYNC-slots are $(2x(m) + y + w + z(m))$, for an appropriate value of m.

A host that is in receipt of a clear CTS-signal that is not meant for it, or noise, during the CTS-slot of a sub-frame will remain silent for the duration of the DATA-slot in the same sub-frame. If a source intends to multicast data during a specific sub-frame, it sends an appropriately structured multicast RTS signal with the same probability as when it sends an unicast RTS, and proceeds with the transmission of data in the DATA-slot of the frame, without paying heed to the signals it might receive in the CTS-slot. Additionally, a host that is in receipt of a multicast RTS-signal does not respond with a CTS-signal in the CTS-slot of the sub-frame. While there can be collisions at any given receiver on multicast transmissions, we can show that for unicast transmissions, following the RTS-CTS exchange there can be no collisions at the receiver [16].

A source i intending to send unicast data to any destination j for flow $l \in \mathcal{F}(i,j)$ in the DATA-slot on a sub-frame issues an RTS-signal in the corresponding RTS-slot with a probability $p_{i,j,l}$. After successful reception of the RTS-signal, the destination responds with a CTS-signal in the CTS-slot of the sub-frame. On hearing a clear CTS-signal from the destination, the source proceeds with the transmission of data in the DATA-slot of the sub-frame. Specifically, the probability that an RTS-signal is followed by the receipt of a CTS-signal is equivalent to the probability of data-transmission alluded to earlier as $\mathcal{S}_{i,j}$. If it is of interest to improve the link-layer reliability of the network, the destination can be made to acknowledge the receipt of data on the previous data-transmission, the next time it sends a CTS-signal to the source. That is, there is scope for a delayed link-layer acknowledgment within the CTS-slot, if necessary. By eliminating the immediate acknowledgement of proper receipt of data, we are in effect eliminating the exposed-terminal problem that is found in the 802.11-MAC.

B. Implementation of the access-probability update algorithm

The implementation of the iterative procedures of the earlier sections require each host being updated of the relevant probability values and number of flows in its 2-hop neighborhood. For this we suppose that the probability values and number of flows associated with 1-hop neighbors of a host can be inserted into the RTS- and CTS-packets. If the set of 1-hop probabilities is too large to be included into a single RTS- or CTS-packet, we might have to segment the set into smaller parts, and use multiple RTS- and CTS-packets to achieve the same end result. Additionally, we need to ensure that none of the hosts in a 2-hop neighborhood simultaneously update their probability values. This is accomplished as follows.

1) We order the nodes in the network lexicographically using their (unique) IP-addresses.
2) Every node in the network computes the zero-retry floor-access probability of section I-A and takes it as the initial value for $\mathcal{P}_{i,j}$. Using the RTS- and CTS-packets, nodes inform each other of their and their one-hop neighborhood's current values of the probabilities and number of flows. These values have a sequence number associated with them. For example, the probability value at the k-th step of the iteration has a sequence value of k associated with it. The sequence number is included alongside the probability values in the RTS- and CTS-packets.
3) After the probabilities associated with the set of hosts in the 2-hop neighborhood of i that are lexicographically lower than i attain a sequence number of $k+1$, the i host proceeds to update its $k+1$-th update to its probability values.
4) Also, for the sake of robustness to erasures, all nodes maintain a list of past iteration values in case it cannot update its own value. It can be shown that not more than two past values for each probability is necessary.

V. Performance Evaluation

In this section we describe the experiments that show how ST-MAC protocol performs when the RTS-signal in the corresponding RTS-slot is transmitted with a probability as determined by equation 5. For this we compare the performance of the ST-MAC protocol against 802.11-MAC protocol. These experiments involved the implementation of the ST-MAC protocol within the NS2 network simulator [13]. A detailed implementation of the 802.11-MAC protocol [11] already exists as a part of the simulation model. As there is no notion of proportional fairness in the 802.11-MAC, it is unfair to compare ST-MAC with 802.11-MAC in terms of fairness. However, the results presented in this section give a clear picture of how the zero-retry fairness scheme combined with no exposed terminal problem, performs in a wireless network involving random hosts. The final version of the paper will include the comparison between the zero-retry case and the optimal floor-access probability in terms of average network throughput, end-to-end delays and packet delivery ratio.

A. Traffic and mobility models used in the experiments

A constant-bit-rate (CBR) traffic was generated for each source-destination pair. We chose AODV [12] as the ad hoc routing protocol as it requires less information to be exchanged and does not require enumerating paths within the packets.

The pause-time is kept constant and equal to the simulation time (i.e. there is no mobility) in all our experiments. Lucent's WaveLAN with 2Mbps bit rates and 250m-transmission range is used for the radio model. An omnidirectional antenna is used by the mobile/wireless nodes.The carrier sense range is the same as the transmission range in our simulation experiments, i.e. the packets are assumed to interfere with each other only when a receiver is within the transmission range of two sources that are transmitting simultaneously. Simulations are run for 500 simulated seconds.

In the simulations the retry limit for data packets is set to 4 for both the protocols which is the default 802.11-MAC long-retry limit. In our derivation we assumed infinite buffer space, however, in simulations, Drop Tail Queue of size 50 packets was selected as an interface queue for both these protocols, unless mentioned otherwise. It is interesting to note that even though we are restricting the buffer space, approximately-proportional fairness scheme performs quite well.

B. Performance Metrics

The following metrics are used in the various scenarios to evaluate the two protocols:

1) *Average Network throughput*: This is defined as the total data received at traffic destinations over the simulation period divided by the simulation time (in Mbps).
2) *Packet Delivery Ratio*: This is defined as the ratio of the total number of data packets received by the destinations to those sent by the CBR sources over the simulation period.
3) *Average End-to-End delay of data packets*: This is defined as the average delay between the time at which the data packet was originated at the source and the time it reaches the destination. Data packets that are lost en route are not considered. Delays due to route discovery, retransmissions and queuing are included in the delay metric.

C. NS2 Implementations of Equation 5

Using the IP-addresses of the packets, a host is made aware of the number of CBR-flows that either originate, terminate, or, are routed through it. For these simulations, these numbers are counted over the past ten super-frames (cf. figure 2) and presented as an estimate for the next ten super-frames. The choice of optimal number of super-frames to be counted to present an estimate of number of CBR-flows depends on the traffic and mobility pattern of any network. However we are not considering these factors for the reported simulations and fixed this number to be ten at random.

This information is then made available to the 1-hop neighbors of the host by an appropriate field in the RTS/CTS

Time	802.11-MAC Throughput(Mbps) Flow 1, Flow 2	ST-MAC Throughput(Mbps) Flow 1, Flow 2
Flow 1 starts first	0.044, 0.329	0.135, 0.238
Flow 2 starts first	0.454, 0.124	0.440, 0.136

TABLE II

COMPARISON OF CBR-FLOW THROUGHPUTS, WHERE FLOWS START AT A DIFFERENCE OF 0.2 SEC FROM EACH OTHER.

packets. Each host $i \in V$ estimates its value for $\mathcal{P}_i$ as per equation 5.

D. Simulation Experiments

To compare the performance of 802.11-MAC protocol and ST-MAC protocol we first considered a scenario involving only two hosts and a single CBR-flow. The maximum average throughput (i.e. when the source has packets to transmit all the time) obtained by 802.11-MAC protocol is 0.699 Mbps. In comparison, ST-MAC protocol achieved a maximum throughput of 0.580 Mbps.

We then compare 802.11-MAC protocol with ST-MAC protocol in a scenario with three nodes and two flows, to show that ST-MAC protocol does better than 802.11-MAC even without eliminating exposed terminal problem (as there is no exposed terminal problem in a network with three nodes). In this scenario, three nodes A, B and C are used where nodes A and C are one hop neighbors of node B and nodes A and C are not one hop neighbors of each other($A \leftrightarrow B \leftrightarrow C$). Flow 1 is from A to B and flow 2 is from A to C. Every host has a buffer space sufficient to store all the packets it receives and A has data to transmit all the time for both flows. Each CBR-flow generates 12.5 data packets (of 512 bytes) to keep the link saturated (The notion of fairness comes into play only when the network resources are scarce with respect to the demand). The maximum throughput obtained in this experiment by ST-MAC for flow 1 ($A \rightarrow B$) is 0.290 Mbps and for flow 2 ($A \rightarrow B \rightarrow C$) it is 0.238 Mbps. 802.11-MAC obtains average throughput of 0.234 Mbps for both flows. The average delay for both protocols is very high due to the large buffers. We observe that even for a simple 3-host network, ST-MAC is performing better than 802.11-MAC.

If we fix the queue length to be 50 for the above mentioned 3-host experiment, then the average throughput for each link varies as a function of start time of flows, as the queue overflows and depending on the start times either packets of flow 1 or flow 2 get dropped more frequently using Drop Tail Queue. Table II summarizes the observation. The delays are comparable for both these protocols. If we implement a more fair queuing scheme in place of Drop Tail Queue, the average throughput in case of ST-MAC protocol will be similar to the one obtained when every host has a buffer space sufficient to store all the packets. Ongoing work includes implementing approximation algorithm together with a fair queueing scheme.

Finally, we considered randomly generated networks consisting of 100 hosts within a $1000m \times 1000m$ field. Each source generates 4 data packets per second of 512 bytes each. The CBR-connections start and end at random times during the simulation period. The number of CBR-flows is varied from 10 to 50 in intervals of 10 for these randomly distributed hosts. All results presented are averaged over 20 random simulations, each run involving random CBR-connections over randomly distributed nodes. Figures 3, 4 and 5 show the plots of the average network throughput, packet-delivery ratio and end-to-end delay as a function of the number of connections within the above mentioned scenarios. On an average, the ST-MAC protocol provided an 26.6% improvement in average network throughput and a 14.5% improvement in the packet-delivery-ratio, with a comparable end-to-end delay over 802.11-MAC.

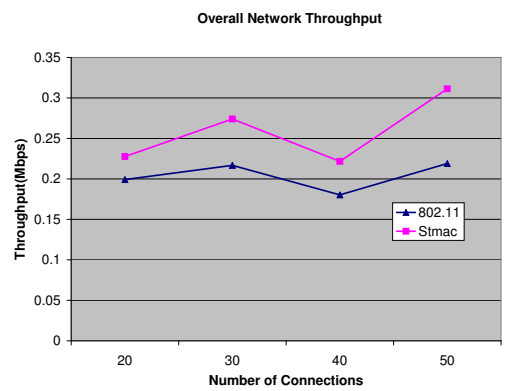

Fig. 3. Network throughput as a function of the number of connections.

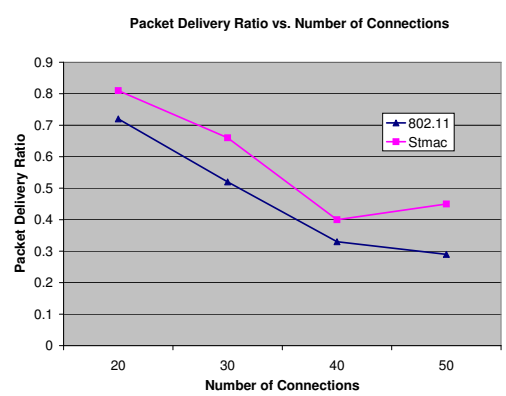

Fig. 4. Packet-delivery-ratio as a function of the number of connections.

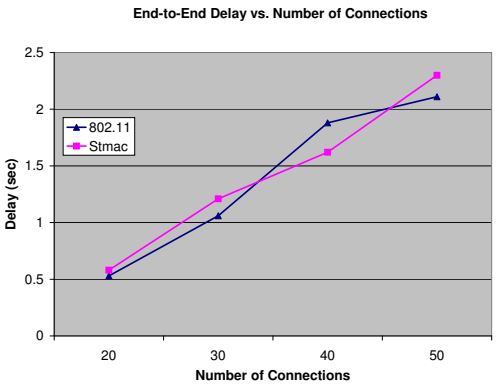

Fig. 5. End-to-end-delay as a function of the number of connections.

REFERENCES

[1] D. Bertsekas. *Nonlinear Programming*. Athena Scientific, Second Edition edition, 1999.

[2] John W. Byers, Michael Luby, Michael Mitzenmacher, and Ashutosh Rege. A Digital Fountain Approach to Reliable Distribution of Bulk Data. In *SIGCOMM*, pages 56–67, 1998.

[3] M.S. Gast. *802.11 Wireless Networks: The Definitive Guide*. O'Reilly & Associates, Sebastapol, CA, 2002.

[4] L. Jiang and SC Liew. Proportional Fairness in Wireless LANs and Ad Hoc Networks. In *IEEE Wireless Communications and Networking Conference (WCNC)*, volume 3, pages 1551– 1556, 2005.

[5] K. Kar, S. Sarkar, and L. Tassiulas. Achieving Proportional Fairness Using Local Information in Aloha Networks. *IEEE Transactions on Automatic Control*, 49(10):1858–1862, October 2004.

[6] F. Kelly, A. Maulloo, and D. Tan. Rate control in communication networks: shadow prices, proportional fairness and stability. In *Journal of the Operational Research Society*, volume 49, 1998.

[7] Srisankar Kunniyur and R. Srikant. End-to-End Congestion Control Schemes: Utility Functions, Random Losses and ECN Marks. In *INFOCOM (3)*, pages 1323–1332, 2000.

[8] Michael Luby. LT Codes. In *The 43rd Annual IEEE Symposium on Foundations of Computer Science*, pages 271–282, 2002.

[9] Jeonghoon Mo and Jean Walrand. Fair end-to-end window-based congestion control. *IEEE/ACM Transactions on Networking*, 8(5):556–567, 2000.

[10] T. Nandagopal, T.-E. Kim, X. Gao, , and V. Bharghavan. Achieving MAC Layer Fairness in Wireless Packet Networks. In *ACM Mobicom*, pages 87–98, 2000.

[11] O'Hara and A. Petrick. *IEEE 802.11 Handbook: A Designer's Companion*. Standards Information Network, IEEE Press, 1999.

[12] C.E. Perkins and E.M. Royer. Ad Hoc On-Demand Distance Vector Routing. *IEEE Workshop on Mobile Computing Systems and Applications (WMCSA)*, 1999.

[13] The VINT Project. The Network Simulator - NS2. Report, 2004.

[14] S. Sharma, K. Gopalan, N. Zhu, P. De, G. Peng, and T. Chiueh. Implementation Experiences of Bandwidth Guarantees on a Wireless LAN. In *ACM/SPIE Multimedia Computing and Networking (MMCN 2002)*, 2002.

[15] N. Singh. The Design, Analysis and Performance Evaluation of a Slotted-Time MAC-protocol for Ad-hoc Wireless Networks. MS Thesis, Department of General Engineering, University of Illinois at Urbana-Champaign, Urbana, IL, 2004.

[16] Z. Tang and J.J. Garcia-Luna-Aceves. Hop-Reservation Multiple Access (HRMA) for Ad-Hoc Networks. In *IEEE INFOCOM*, 1999. New York, New York.

[17] N.H. Vaidya, P. Bahl, and S. Gupta. Distributed fair scheduling in a wireless LAN. In *6th Annual International Conference on Mobile Computing and Networking*, 2000.