This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE INFOCOM 2008 proceedings.

On A Non-Linear Optimization Approach for 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

***Abstract*—In this paper we present a partially asynchronous, fully distributed flow-based access scheme for slotted-time protocols, that guarantees proportional-fairness in ad hoc wireless networks. This problem of providing fairness in wireless networks is considered in the framework of non-linear optimization. We say a medium access control algorithm is *proportionally fair* with respect to individual end-to-end flows in a network, if the product of the end-to-end *flow-success probabilities* is maximized.**

I. Introduction

In this paper we consider an Ad-hoc Wireless Network [1] that carries several flows between various source-destination pairs under the slotted-time MAC protocol In specific we are interested in a fully distributed scheme for the assignment of the network's resources among flows that is fair in terms of *flow-success probabilities*.

The literature contains a large volume of references (cf. [2], [3], [4], [5], 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 [2]. 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 node 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 require 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 section II. The algorithm presented in this paper addresses the issue of providing end-to-end flow based proportional fairness [2] where each node accesses the wireless medium with an access probability so as to maximize the product of end-to-end *flow-success probabilities* of all flows. In [6] the authors have addressed the problem of providing proportional fairness by considering joint optimization at both transport layer and link layer, whereas in this paper resource allocation problem is considered only at the link layer. 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 utility function to be maximized in this paper, is the product of of end-to-end *flow-success probabilities* of all flows in the network, where nodes in the network are allowed a fixed number of retries in case of unsuccessful transmissions. We assume that the number of retries permitted in the network is a predetermined fixed value. If the network topology or the number of flows in the network change over time, the nodes in the network adjust the utility function accordingly, in a distributed way, by using the local information. Thus, the proposed scheme will work well in mobile ad hoc wireless networks too.

The *flow-success probability* is equivalent to the packet-delivery ratio, which is the ratio of the total number of data packets received by the destination to those transmitted by the source. In this model, the network traffic is modeled as data-packets and time is divided into slots of sufficient duration to allow the successful transmission and reception of one data packet. At any time there are several flows in the network, and for proportional fairness we maximize the product of end-to-end packet-delivery ratio of all the flows. The packet delivery ratio reports the effectiveness of the source nodes in delivering the data-packets to the destination nodes. This metric plays a very significant role in determining the quality of service for multi-hop wireless networks.

In [7], we presented an iterative synchronous algorithm for achieving proportional fairness in ad hoc wireless networks. The algorithm requires the nodes in the network to have some form of ordering. In this paper we show that no fixed ordering of nodes is needed for the stations to converge to the optimal value of access probabilities, and compare the result with that of the synchronous implementation of distributed Gauss-Seidel algorithm.

The rest of the paper is organized as follows. Section II

The work of N. Singh and R. Sreenivas was supported in parts by grants NSF CNS-0437415 and NSF ECCS-0426831.

presents an analysis of flow-based proportional fairness, where we derive the expression for the product of the end-to-end *flow success probabilities* where every node in the network would attempt r retries before discarding a data packet. Section III discusses a distributed implementation to achieve flow-based proportional fairness using a block coordinate descent algorithm. The convergence of this algorithm to the unique global optimum is established using just the link access probabilities of the two-hop neighbors of each node. Section IV presents the partially asynchronous parallel implementation of the strictly sequential algorithm of section III.

II. 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 time-slot implies collision free data transmission in that 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 attempts at retransmission. (6) We only consider unicast flows for our derivations. A wide variety of slotted-time protocols, like *ST-MAC* [8], etc., can 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. We also assume there is a unique route for each flow within the network (which would be the case if we used AODV [9] 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 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 relationship that represents the set of bi-directional *links* (i.e. $(i,j) \in E \Leftrightarrow (j,i) \in E$). 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 $j \in N(i)$, 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

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

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

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

The product of the end-to-end *flow success probabilities* (Δ) in an ad hoc network that uses a protocol from the protocol-family A is given by the expression

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

We adopt the the logarithm of Δ as the utility-function, which is given by equation 2 below.

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

The objective therefore is to maximize 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: 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 summand in the above 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 $\mathcal{F}(i,j)$ (i.e. $p_{i,j} = \mathcal{P}_{i,j}/card(\mathcal{F}_{i,j})$, where $card(.)$ denotes the cardinality). *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 node $i \in V$ assigns a zero-value to $\mathcal{P}_{i,j}$ would be when it has no outgoing flows on $(i,j) \in E$. Similarly, a node $i \in V$ assigns a value of unity to $\mathcal{P}_{i,j}$ only if the flows in $\mathcal{F}(i,j)$ define the entire set of incoming flows in $N(i)$ (i.e. no 1-hop neighbor of i other than j is involved with an incoming 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.

III. Distributed Algorithms that Enforce Proportional Fairness

We now present a result from section 2.7 of [10] about the convergence of *Block Coordinate Descent* schemes. This result will be used to establish the existence of distributed procedures 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

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

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 2: [10] 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 node $i \in V$, 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, nodes 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.

Lemma 3: 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: 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 that the Hessian of the term

$$\log \{1 - (1 - \mathcal{S}_{i,j})^{r+1}\} \tag{4}$$

is negative definite [7]. 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 4 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 2 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 expression of the derivative of the product of the end-to-end *flow-success probabilities*, Δ, with respect to a $\mathcal{P}_{i,j}$-term ($i \in V, j \in N(i)$), only involves link access probabilities of nodes within a two-hop neighborhood of i (cf. section IV). The iterative process proceeds as follows:

1) Node 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 held constant.
2) This value is then communicated to all nodes in the two-hop neighborhood of node i, which is then followed by a repetition of the above step at the next node in the prescribed sequence.

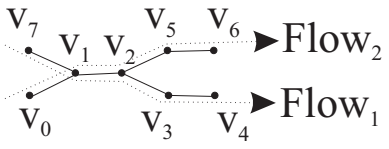

Fig. 1. An illustrative example.

The results of the distributed Gauss-Seidel algorithm applied to the example given in figure 1, is shown in table I. Here in each cycle, the nodes follow a strict order for updating the access probability values, i.e. in one cycle the order in which the nodes update is $\{V_0, V_1, V_2, V_3, V_4, V_5, V_6, V_7\}$.

In the next section we present the algorithm which requires no ordering of the nodes.

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
SYNCHRONOUS DISTRIBUTED GAUSS-SEIDEL ALGORITHM APPLIED TO THE EXAMPLE OF FIGURE 1.

Iterates	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	0.228519
After Nodes 0 and 3 simultaneously update on initial values	0.5	0.5	0.3334	0.1640	0.01	0.25	0.01	0.5	0.245172
After Nodes 7 and 5 simultaneously update on previous iterate	0.5	0.5	0.3334	0.1640	0.01	0.1684	0.01	0.5	0.260437
After Node 1 updates on previous iterate	0.5	0.4596	0.3334	0.1640	0.01	0.1684	0.01	0.5	0.263391
After Node 7 updates on previous iterate	0.5	0.4596	0.3334	0.1640	0.01	0.1684	0.01	0.5	0.263391
After Node 2 updates on previous iterate	0.5	0.4596	0.2943	0.1640	0.01	0.1684	0.01	0.5	0.268182
After Node 1 updates	0.5	0.4581	0.2943	0.1640	0.01	0.1684	0.01	0.5	0.268186
After Nodes 0 and 3 simultaneously update on previous iterate	0.5	0.4581	0.2943	0.1644	0.01	0.1684	0.01	0.5	0.268187
After Nodes 0 and 5 simultaneously update on previous iterate	0.5	0.4581	0.2943	0.1644	0.01	0.1646	0.01	0.5	0.268239
After Node 1 updates on previous iterate	0.5	0.4575	0.2943	0.1644	0.01	0.1646	0.01	0.5	0.268240
After Node 2 updates on previous iterate	0.5	0.4575	0.2942	0.1644	0.01	0.1646	0.01	0.5	0.268240
After Node 5 updates on previous iterate	0.5	0.4575	0.2942	0.1645	0.01	0.1646	0.01	0.5	0.268240
After Node 3 updates on previous iterate	0.5	0.4575	0.2942	0.1644	0.01	0.1645	0.01	0.5	0.268240
Optimal values for example in fig 1	0.5	0.4575	0.2942	0.1645	0.01	0.1645	0.01	0.5	0.268240

TABLE II
PARTIALLY ASYNCHRONOUS PARALLEL DISTRIBUTED GAUSS-SEIDEL ALGORITHM APPLIED TO THE EXAMPLE OF FIGURE 1.

IV. PARTIALLY ASYNCHRONOUS PARALLEL GAUSS-SEIDEL ITERATIONS

Lemma 4: In any step of the Gauss-Seidel iterations of the coordinated block descent algorithm of section III, for any node i to update values of $\mathcal{P}_{i,j}$ it only requires the values of $\mathcal{P}_{k,l}, k \in \{N(i) \cup (N(N(i)) - \{i\})\}$ and $(k,l) \in E$.

Proof: A synchronous implementation of the algorithm requires that any node m does not carry out its $n+1$-th update without first receiving the results of n-th or $n+1$-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

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

In any iteration, for node m to maximize the function $\log\Delta$, it only needs to maximize f_m, as the remaining terms are independent of $\mathcal{P}_{m,j}, (m,j) \in E$. This is because the link success probability $\mathcal{S}_{i,j}, (i,j) \in E$ depends only on two-hop $\mathcal{P}_j$'s, $j \in V$. 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, and the result follows. ■

It is interesting to note that the f_m term can be constructed by node m, just by the exchanging information with its two-hop neighbors. This implies that the cost function to be maximized is also constructed in a distributed way, and if the ad hoc wireless network's topology/number of flows in the

network changes over time, the cost function will be modified accordingly by the nodes in the network.

A. Modified Gauss-Seidel Iterations

Let us define the modified *non-linear Gauss-Seidel Algorithm* as follows:

Define $\mathbf{P}$ as the set of all access probabilities, $\mathcal{P}_{i,j}$'s, $(i,j) \in E$ of n elements. In this iterative scheme, for moving to the next iterate, we define, at each iterate k, $\mathbf{x}$ ($\mathbf{x} \in \mathbf{X}$ for an appropriately define closed, convex set $\mathbf{X} \subseteq \mathcal{R}$) as any subset of $\mathbf{P}$ which contains m elements $(x_1, x_2, \ldots, x_m)$, which are access probabilities associated with nodes not in the two-hop neighborhood of each other and have successfully exchanged proper messages to silence their two hop neighborhoods. By silencing the neighborhood, we imply that no nodes in the two hop neighborhood will update the access probability values associated with them, while the nodes whose access probability values are in set $\mathbf{x}$ are doing the updates. The next iterate $(\mathbf{x^{k+1}}, (\mathbf{P}-\mathbf{x})^{\mathbf{k+1}})$ (where $(\mathbf{P}-\mathbf{x})$ has $n-m$ elements) is obtained from the previous iterate $(\mathbf{x^k}, (\mathbf{P}-\mathbf{x})^{\mathbf{k}})$ as follows,

$$\mathbf{x}^{k+1} \in arg\max_{\xi \in \mathbf{X}} f(\xi, (\mathbf{P}-\mathbf{x})^k) \tag{5}$$
$$(\mathbf{P}-\mathbf{x})^{k+1} = (\mathbf{P}-\mathbf{x})^k$$

Let $z_i{}^k = ({x_1}^{k+1}, \ldots, x_i^{k+1}, x_{i+1}^k, \ldots, x_m^k, (\mathbf{P}-\mathbf{x})^k)$, $1 \le i \le m$.

From lemma 4, we know that for any node to update the access probability values associated with it, the node only requires the two-hop neighbors access probability values. Thus, while the elements of $(\mathbf{P}-\mathbf{x})^k$ are held constant, the update equation given by equation 5 is same as the an iterative scheme where the next iterate $(x_1^{k+1}, x_2^{k+1}, \ldots, x_m^{k+1})$ is obtained from the previous iterate $(x_1^k, x_2^k, \ldots, x_m^k)$, $x_i \in \mathbf{X}_i$, for an appropriately defined closed, convex set $\mathbf{X}_i \subseteq \mathcal{R}$, as follows. For $i = 1, 2, \ldots, m$,

$$x_i^{k+1} \in arg\max_{\xi \in \mathbf{X}_i} f(x_1^{k+1}, .. \tag{6}$$
$$\ldots, x_{i-1}^{k+1}, \xi, x_{i+1}^k, \ldots, x_m^k, (\mathbf{P}-\mathbf{x})^k)$$

This observation is based on the fact that the update of any of the elements of the set $\mathbf{x}$, is independent of all the other elements of set $\mathbf{x}$. Using this result and the definition of the Gauss-Seidel iterations (equation 6), we obtain

$$f(\mathbf{x}^k, (\mathbf{P}-\mathbf{x})^k) \le f(z_1^k, (\mathbf{P}-\mathbf{x})^k) \le f(z_2^k, (\mathbf{P}-\mathbf{x})^k) \tag{7}$$
$$\ldots \le f(x^{k+1}, (\mathbf{P}-\mathbf{x})^k) \le f(\mathbf{x}^{k+1}, (\mathbf{P}-\mathbf{x})^{k+1}) \quad \forall k$$

This observation and the proof of convergence of Gauss Seidel given in proposition 3.9 in chapter three of reference [11], guarantees that this procedure will converge to the global maxima, as long as each node keeps updating the access probabilities associated with it within a finite time period. This modified algorithm is partially asynchronous as we don't require any ordering of the nodes, however we will still need to prevent the nodes from using the old data (access probabilities), when new data is available.

The results of the modified Gauss-Seidel algorithm applied to the example given in figure 1, are shown in table II. In the result presented, the order in which the nodes do update is random. Each time any node does the update, it silences the nodes in its two-hop neighborhood. When compared with the synchronous Gauss-Seidel algorithm, this algorithm takes more number of cycles to converge to the optimal solution, but the advantage is that we don't need to order the nodes of the network.

V. Conclusion

In this paper we presented an algorithm to optimize proportional fairness in terms of the product of end-to-end *flow-success probabilities* of all flows in slotted-time MAC-protocols for ad-hoc wireless networks. We formulate and optimize the product of end-to-end *flow-success probabilities* for the case where nodes in the wireless network are allowed r retransmissions. We show that a unique global optimum can be achieved, by using a partially asynchronous fully distributed iterative algorithm by using the link access probabilities of the two-hop neighbors of each node. Simulation results for small networks verify our analytical observations.

References

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

[2] 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*, vol. 49, 1998.

[3] S. Kunniyur and R. Srikant, "End-to-End Congestion Control Schemes: Utility Functions, Random Losses and ECN Marks," in *INFOCOM (3)*, 2000, pp. 1323–1332.

[4] J. Mo and J. Walrand, "Fair end-to-end window-based congestion control," *IEEE/ACM Transactions on Networking*, vol. 8, no. 5, pp. 556–567, 2000. [Online]. Available: citeseer.ist.psu.edu/mo95fair.html

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

[6] X. Wang and K. Kar, "Cross-layer rate control for end-to-end proportional fairness in wireless networks with random access," in *MobiHoc '05: Proceedings of the 6th ACM international symposium on Mobile ad hoc networking and computing*. New York, NY, USA: ACM Press, 2005, pp. 157–168.

[7] N. Singh and R. S. Sreenivas, "On Distributed Algorithms that Enforce Proportional Fairness in Ad-hoc Wireless Networks," in *5th Intl. Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks*, 2007.

[8] 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.

[9] C. Perkins and P. Bhagwat, "Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers," in *ACM SIGCOMM'94 Conference on Communications Architectures, Protocols and Applications*, 1994, pp. 234–244. [Online]. Available: citeseer.ist.psu.edu/perkins94highly.html

[10] D. Bertsekas, *Nonlinear Programming*, Second Edition ed. Athena Scientific, 1999.

[11] D. Bertsekas and J. N. Tsitsiklis, *Parallel and Distributed Computation:Numerical Methods*. Athena Scientific, 1997. [Online]. Available: http://web.mit.edu/dimitrib/www/pdc.html