# Maximizing Network Utilization with Max-Min Fairness in Wireless Sensor Networks

Avinash Sridharan and Bhaskar Krishnamachari
{asridhar,bkrishna}@usc.edu
Department of Electrical Engineering
University of Southern California

**Abstract**

The state of the art for optimal data-gathering in wireless sensor networks is to use additive increase algorithms to achieve max-min fair rate allocation ([1], [3]) while implicity trying to maximize network utilization. In this work we explicitly formulate the problem of maximizing the network utilization subject to a max-min fair rate allocation constraint in the form of two separate but dependent linear programs. We adopt a dual based approach to design an efficient distributed algorithm to achieve our objectives. The analysis of the dual proves the sub-optimality of previously proposed additive increase algorithms for this problem. We show through numerical evaluations that the proposed dual-based distributed algorithm can obtain solutions within 0.02% of the optimal in 99.65% of the cases (and within 20% in even the worst case) within just one iterative update of shadow prices through sub-gradient search, requiring a polynomial number of message exchanges (which appears to grow as $O(n^2)$ where $n$ is the number of sources in the network).

## I. Introduction

The problem we wish to address is as follows: there are $n$ sources in the network that are trying to send data to a single sink. Every source has a shortest path through one or more intermediate nodes to the sink. Every receiver in the network is bandwidth constrained. The objective of the problem is to maximize the utilization of the network capacity while maintaining a max-min fair rate allocation to all nodes. We define a rate allocation to be max-min fair if the minimum rate allocated to any flow is the maximum possible over all possible rate allocations.

There have been solutions proposed to the max-min fair rate allocation problem in the context of wireless sensor networks. In an earlier work [1], we presented a solution in the context of designing a scheduled access scheme that guarantees max-min fairness. Rangwala *et al.* [3] present a solution in the context of a congestion control problem. The common element between these two works is that they use additive increase algorithms to perform bandwidth allocation which guarantees max-min fair rate. However as we show in Section III such an approach will not guarantee the maximization of network utilization. Our approach of analyzing the dual to achieve a distributed solution to the problem follows the approach presented by Ye and Ordonez [4] where a distributed dual based gradient search algorithm is proposed for the problem of maximizing data extraction under energy constraints. In this work we formulate the maximization of network utilization while maintaining a max-min fair rate allocation as two separate but dependent LP's. We emphasize the shortcomings of additive increase algorithms by presenting their association with our primal. We then analyze the dual to obtain a distributed near optimal algorithm to achieve our objective.

## II. Modeling receiver bandwidth consumption in wireless networks

In this section we present a model that captures the bandwidth consumption at a receiver in a tree $\mathbf{T}$ rooted at the sink. The essence of the model is that it captures the interference observed by a receiver during reception of flows from its children. This model is identical to the one proposed by us in [1] and is similar to the one used by Rangwala *et al.* [3] to capture the effects of interference. We denote the set of all communication links in the network by the set $\mathbf{E}$, and the set of all nodes by the set $\mathbf{V}$. Every receiver in the network has a finite receiver bandwidth capacity given by the set $\mathbf{B}$. The routing tree rooted at the sink is denoted by $\mathbf{T} \subset \mathbf{G}$.

Due to the broadcast nature of wireless links, any flow from a child $i$ to its parent $j$ on the tree $T$ consumes bandwidth on all receivers that are neighbors of $i$ on the graph $G$ (we assume here that the neighbor set captures all interfering nodes, and therefore refer to the edges in $E$ that are not part of $T$ as noise edges). It is this feature that makes the problem of rate allocation in these networks very different from that observed on a wired network.

The bandwidth constraint at a receiver $i$ is modeled as:

$$\sum_{j \in C^i} r_{src}^{(j)} + \sum_{j \in N^i} \sum_{k \in C^j} r_{src}^{(i)} + \sum_{j \in N^i} r_{src}^{(j)} \leq B^{(i)} \tag{1}$$

where $r_{src}^{(j)}$ represents the total rate transmitted by a node $j$, $C^i$ is the set of all node $j$ that have $i$ in its path to the sink. $N^i$ is the set of all neighbors of $i$. We would like to make it explicit that we are modeling a half duplex radio. Hence a receiver cannot send and receive. This implies there is a noise link from receiver $i$ to itself, i.e. all diagonal elements of $\mathbf{N}$ are 1.

## III. FORMULATING THE CONSTRAINED OPTIMIZATION PROBLEM

We now formulate the maximization of the network capacity utilization while maintaining max-min fairness as two separate constrained optimization problems $\mathbf{P1}$ and $\mathbf{P2}$. The definition of variables used in our formulation are as follows: $\mathbf{B}$ is an $n \times 1$ vector representing the bandwidth available at each node $i \in \mathbf{V}$. $\mathbf{R_{src}}$ is an $n \times 1$ vector representing the rate allocated to each source $i \in \mathbf{V}$. $\mathbf{N}$ is an $n \times n$ matrix, that denotes the presence of a noise edge $n_{ij} \in \mathbf{n}$ between two nodes $i, j \in \mathbf{V}$ ($i$ can be equal to $j$). $Y$ is a scalar variable representing the minimum rate among all flows that acts as the objective function of $\mathbf{P2}$. $\mathbf{C}$ is an $n \times n$ matrix that gives the parent-child relationships on the data gathering tree, such that $c_{ij} \in C^i$ is 0 if node $j$ is not in node $i$ path to the sink and $c_{ij}$ is equal to 1 if j is in the path of node $i$ to the sink.$\mathbf{R_{in}}$ represents the total input rate incident on a receiver from all its child nodes. $\mathbf{R_{noise}}$ represents the total rate that is incident on each receiver from its neighbors. Note that the noise bandwidth $r_{noise}^{(i)} \in \mathbf{R_{noise}}$ represents the total outgoing bandwidth of node $i \in V$.

$$
\begin{array}{llll}
\mathbf{P1}: & & \mathbf{P2}: & \\
\max & \sum_{i\in\mathbf{T}} r_{src}^{(i)} \text{ s.t.} & \max & Y \text{ s.t.} \\
 & \mathbf{R_{in}} + \mathbf{N} \times \mathbf{R_{noise}} \preceq \mathbf{B} & & \mathbf{R_{in}} + \mathbf{N} \times \mathbf{R_{noise}} \preceq \mathbf{B} \\
 & \mathbf{R_{in}} = \mathbf{C} \times \mathbf{R_{src}} & & \mathbf{R_{in}} = \mathbf{C} \times \mathbf{R_{src}} \\
 & \mathbf{R_{noise}} = \mathbf{C} \times \mathbf{R_{src}} + \mathbf{R_{src}} & & \mathbf{R_{noise}} = \mathbf{C} \times \mathbf{R_{src}} + \mathbf{R_{src}} \\
 & r_{src}^{(i)} \geq Y^* \ \forall\, i \in \mathbf{T} & & r_{src}^{(i)} \geq Y \ \forall\, i \in \mathbf{T} \\
 & \mathbf{R_{src}} \preceq \mathbf{B} & & \mathbf{R_{src}} \preceq \mathbf{B}
\end{array}
$$

$Y^*$ in $\mathbf{P1}$ is the optimal solution of $\mathbf{P2}$. The constraints of our optimization problem come directly from our bandwidth consumption model presented in section II.

Also we would like to note that the optimal solution $Y^*$ of the primal $\mathbf{P2}$ is the $\min(B_{available}^{(i)}) \ \forall\, i \in \mathbf{V}$ where $B_{available}^{(i)} = \frac{B^{(i)}}{|C^{(i)}| + \sum_{j\in N^{(i)}} |C^{(j)}|}$. We omit the proof to this claim due to space constraints.We will from now on refer to node $i$ as the bottle neck node.

Solutions that achieve max-min fairness while implicitly trying to maximize network utilization( [1], [3]) use the following additive increase mechanism. Sources in the network are allowed to increase their rates equally by a small value $\epsilon$. When a receiver in the network is constrained, it constrains all its neighbors, its neighbors children and its own children. This process continues till the point when all nodes in the network are constrained. Since all nodes have equal increments and the first node to exhaust its bandwidth would be the bottle neck node, algorithms using additive increase technique would achieve the optimal solution to $\mathbf{P2}$. However we claim that although the additive increase algorithm consumes the network

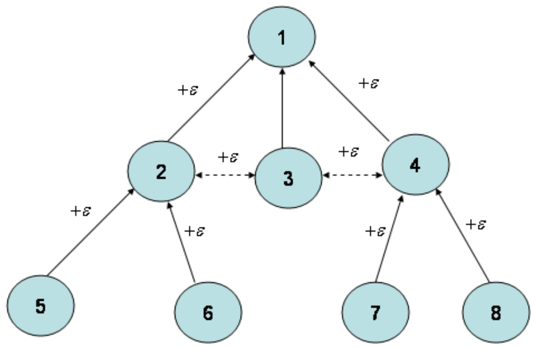



Fig. 1. An example depicting the sub-optimality of the additive increase technique to achieve the maximization of maximizing network utilization while maintaining a max-min fair rate allocation.

capacity it will not necessarily achieve a solution for **P1**. In this section we present insights into our claim through an example and present a more rigorous proof in Section IV. Assume all nodes, except node $1$ are sources in figure 1. In this topology if source rates of all nodes were incremented equally node $3$ would end up consuming bandwidth at $2$ and $4$ and hence the throughput achieved would be sub-optimal. A higher throughput could be achieved by simply allocating all nodes the max-min rate and than giving the remaining capacity to nodes $2$ and $4$. The example shows that there exist topologies where additive increase mechanisms might be sub-optimal.

Apart from the sub-optimality another draw back of additive increase algorithms is the estimation of the increment $\epsilon$. In real systems an accurate estimate of $\epsilon$ is critical to avoid oscillations [3]. Moreover the convergence of these algorithms is O($\frac{B}{\epsilon}$) where $B$ is the maximum receiver bandwidth, which implies a trade off between the speed of convergence and the accuracy of the solution depending on the choice of $\epsilon$.

## IV. A Dual Based Approach

The reason the primal based algorithms use an additive increment based approach is that the primal itself does not lend any information as to the size of the increments that would be required to achieve optimality. Nor does it give any information about the amount of decrements required by nodes when a constraint becomes active. We believe explicit knowledge of these two values would lead to an algorithm with a faster convergence time and robustness to network dynamics (addition and deletion of nodes).

In our approach, we will use the dual to come up with a distributed algorithm using the shadow price interpretation of the Lagrange multipliers [2].

### A. The Lagrange Dual

We consider the maximum bandwidth constraint as our domain and relax the constraints in the primal to obtain the Lagrange dual function. We will concern ourselves only with the dual of $\mathbf{P1}$ and assume that the optimal max-min rate will be calculated from the primal $\mathbf{P2}$. The Lagrange dual function of the primal $\mathbf{P1}$ is:

$$D(\lambda, \upsilon) = max_{\mathbf{R_{src}} \preceq \mathbf{B}} \left( -Y^*(\sum_{i \in T} \upsilon_i) + \sum_{i \in T} r_{src}^{(i)}(1 + \upsilon_i - (\sum_{j \in C^{(i)}} \lambda_j + \sum_{j \ \in N^{(i)}} \sum_{k \in C^{(j)}} \lambda_k + \sum_{i \in N^{(j)}} \lambda_j)) + \sum_{i \ \in T} \lambda_i B^{(i)} \right)$$

Since the original problem is a linear program the dual will also be an LP given by

$$\mathbf{D} : \min_{\lambda \preceq \ 0, \upsilon \preceq \ 0} D(\lambda, \upsilon)$$

Also since the solutions are feasible for both problems the duality gap would be zero [2]. Hence our objective would be to minimize the dual instead of maximizing the primal.

Let $\zeta^i(\mathbf{R^*_{src}}) = \sum_{j \in C^i} r_{src}^{(j)*} + \sum_{j \in N^{(i)}} \sum_{k \in C^{(j)}} r_{src}^{(k)*} + \sum_{j \in N^{(i)}} r_{src}^{(j)*}$. From the Lagrange dual function it can be seen that the sub-gradient w.r.t $\lambda_i$ and $\upsilon_i$ is:

$$\frac{\partial D}{\partial \lambda_i} = -(\zeta^i(R^*_{src}) - B^{(i)}) \tag{2}$$

$$\frac{\partial D}{\partial \upsilon_i} = r_{src}^{(i)*} - Y^* \tag{3}$$

Since the dual is a linear program, the objective of minimizing the Lagrange dual can be achieved by tracing the graph in the direction of the negative gradient. We will use the above fact to develop our distributed algorithm.

### B. The Distributed Algorithm

The Lagrange dual function can be rewritten as:

$$\mathbf{D} : \min_{\lambda \preceq 0, \upsilon \preceq 0} \left( \max \left( \sum_{\forall \ i} r_{src}^{(i)} \mu_i \right) + \sum_{\forall \ i} \lambda_i B^i - Y^* \sum_{\forall \ i} \upsilon_i \right)$$

Where $\mu_i$ is given by:

$$\mu_i = 1 + \upsilon_i - (\sum_{i \in C^{(j)}} \lambda_j + \sum_{j \in N^{(i)}} \sum_{k \in C^{(j)}} \lambda_k + \sum_{i \in N^{(j)}} \lambda_j) \tag{4}$$

To solve the dual $\mathbf{D}$ we could use sub gradient techniques. Sub gradient techniques are iterative, where at each step $t$ we increment the shadow prices $\lambda_i$ and $\upsilon_i$ in the direction of the negative gradient as follows:

$$\lambda_i(t+1) = [\lambda_i(t) + \alpha_t(\zeta^i(R^*_{src}) - B^i)]^+ \tag{5}$$

$$\upsilon_i(t+1) = [\upsilon_i(t) - \alpha_t(r_{src}^{(i)*} - Y^*)]^+ \quad (6)$$

Where $\mathbf{R}^*_{\mathbf{src}}$ are the optimal source rates that solves:

$$\max(\sum_{\forall\ i} r_{src}^{(i)}\mu_i) \quad (7)$$

In order to find a feasible solution to our dual $\mathbf{D}$ we will have to allocate all $r_{src}^{(i)}$ at least the max-min rate $Y^*$. For a fixed $\lambda$ and $\upsilon$, given that all sources are allocated at least the max-min rate in order to find a solution to equation 7 we would require to allocate the maximum available bandwidth to the source having the highest $\mu_i$. We would than allocate the remaining bandwidth to the next highest $\mu_i$ till will exhaust the network capacity. The coefficients $\mu_i$ thus provide an ordering for bandwidth update amongst all sources. The ordering of bandwidth increments at each step can be achieved by simply giving a weight equal to $w_i = \frac{1}{\sum \lambda_j^{(t)}}$ where $j$ belongs to the set of receivers that node $i$ interferes with and $\lambda_j^{(t)}$ is the value of $\lambda_j$ at step $t$. Note that this maximization needs to be performed at every step $t$. This order also clearly shows the failure of **additive increase algorithms** which update bandwidths of all sources equally without taking into consideration the ordering that is implicit from their interference sets.

To achieve the optimal $\mathbf{D}$ we should be running the sub gradient algorithm for multiple iterations till the shadow prices converge. Fortunately through simulations we can show that in our specific problem $99.65\%$ of the time we achieve the optimal in the very first iteration. The details of the simulation and its performance with respect to the optimal are presented in Section V. Our algorithm for maximizing network utilization with max-min fair rate allocation consists of optimizing equation 7.

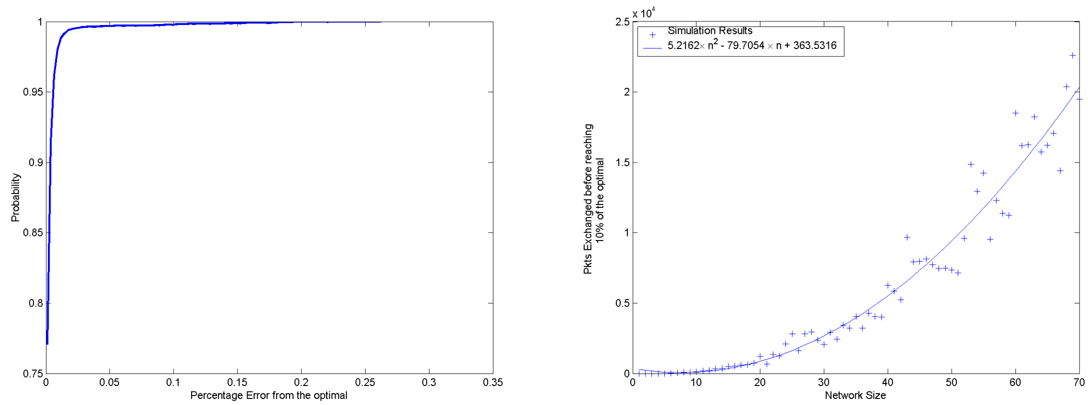
We now present an algorithm for the maximization of equation 7. The algorithm *Maximization of network utilization* proceeds as follows; In **step 1**, each node calculates its per node available bandwidth. The bottle neck bandwidth is than the minimum of all the available bandwidths. This bottle neck bandwidth is allocated to every source in the network. In **step 2**, we calculate the pending bandwidth at each node in the network. For any receiver if the pending bandwidth is negative or zero it constraints all its neighbors their children and its own children. A constrained node can no longer increment its source rate. In **step 3**, for every node in the network we look at the pending bandwidth at every node that is on the path from the source to the sink, and nodes that are neighbors to these intermediate nodes, and set the pending available bandwidth to the minimum of these. In case the pending available bandwidth is positive, we compare its weight with every other source that is not constrained and increment its bandwidth only if it has the maximum weight. In **step 4**, we check the constrained flag for all nodes in the network and if all nodes have been constrained the program terminates, else we repeat the algorithm from **step 2**.

**Algorithm** *Maximization of network utilization*

```
Initialization
contrained_i = FALSE ∀ i
[Step 1] max-min Rate:
B^i_available = B^i / (|C^i| + Σ_{j∈N^i} |C^j|) ∀ i
r^(i)_src = min(B^(i)_available) ∀ i
[Step 2] Pending Bandwidth:
for ∀ i ∋ |C^(i)| ≠ 0
    do B^(i)_pending = ζ^i(B_src) − B^i
        if B^(i)_pending ≤ 0 then constrain all neighbors, their children and your own children
[Step 3] Updating Source bandwidth:
for ∀i
    do pend_bw = min(B^(j)_pending), i ∈ C^j | i ∈ N^j | j ∈ N^k, i ∈ C^k
        if (pend_bw < 0)
            then r^(i)_src = Y*
                constrained_i = TRUE
            else
                if (w_i = max(W) ∀ j, ∋ constrained_j = FALSE & constrained_i = FALSE)
                    r^(i)_src = r^(i)_src + pend_bw
[Step 4] Checking termination condition:
if (contrained_i = TRUE) ∀ i then end
else goto Step 2
```

## V. Performance Evaluation

In order to evaluate the performance of our algorithm we choose network sizes ranging from 6 to 70. For each network size we choose 9 instances of trees obtained by running a shortest path algorithm on a random deployment. For each instance of a tree we give every receiver in the tree a bandwidth uniformly chosen between 100 and 250. We choose 10 such bandwidth distributions for each tree. For our evaluation we initialize $\lambda_i = 1, \forall\ i$ and $\mu_i = 1, \forall\ i$. Figure 2(a) shows the CDF of the error between the optimal solution from a centralize solver and the solution obtained from our distributed algorithm. The




(a) CDF of the error observed between the optimal throughput achievable and the throughput achieved using our distributed algorithm.

(b) Performance of the algorithm in terms of the number of packets exchanged before achieving 10% of the optimal.

CDF clearly shows that for 99.65% of the instances we are able to achieve close to 0.02% of the optimal throughput. For the instances where we were not able to achieve the optimal, we were close to 10% of the optimal throughput in 18 instances and close to 20% in two of the instances. In Figure 2(b) we plot the number of packets required to achieve 10% of the optimal. Based on a regression fit we estimate that it grows as O($n^2$). In the full version of the paper we plan to justify this asymptotic performance analytically.

## References


[1] Avinash Sridharan and Bhaskar Krishnamachari, “Max-Min Far collision-Free Scheduling for Wireless Sensor Networks”, Workshop Multihop Wireless Networks (MWN’04), IPCCC 2004.

[2] Stephen Boyd and Lieven Vandenberghe, “Convex Optimization”, Cambridge University Press, ISBN:0521833787.

[3] Sumit Rangwala, Ramakrishna Gummadi, Ramesh Govindan and Konstantinos Psounis. Interference-Aware Fair Rate Control in Wireless Sensor Networks, In Proceedings of ACM SIGCOMM Symposium on Network Architectures and Protocols, Pisa, Italy, September 2006

[4] W. Ye and F. Ordonez, ”A sub-gradient algorithm for maximal data extraction in energy-limited wireless sensor networks,” Proceedings of the International Conference on Wireless Networks, Communications and Mobile Computing, June 2005