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 fair rate allocation while implicity trying to maximize network utilization. For the quantification of the problem we present a receiver capacity model to capture the interference existing in a wireless network. We also provide empirical evidence to motivate the applicability of this model to a real CSMA based wireless network. Using this model, we explicitly formulate the problem of maximizing the network utilization subject to a max-min fair rate allocation constraint in the form of two coupled linear programs. We first show how the max-min rate can be computed efficiently for a given network. We then adopt a dual-based approach to maximize the network utilization. The analysis of the dual shows the sub-optimality of previously proposed additive increase algorithms with respect to bandwidth efficiency. Although in theory a dual-based sub-gradient search algorithm can take a long time to converge, we find empirically that setting all shadow prices to an equal and small constant value, results in near-optimal solutions within one iteration (within 2% of the optimum in 99.65% of the cases). This results in a fast heuristic distributed algorithm that has a nice intuitive explanation — rates are allocated sequentially after rank ordering flows based on the number of downstream receivers whose bandwidth they consume. We also investigate the near optimal performance of this heuristic by comparing the rank ordering of the source rates obtained from the heuristic to the solutions obtained by solving the linear program.

I. Introduction

The applications envisioned for sensor networks are primarily data gathering applications. For such applications a common scenario would be multiple sensors sensing the environment and sending data over a shortest path tree to a central base station. Since the primary mode of communication for these devices is the wireless channel and the current standards (802.15.4) propose rates to the order of kilo bits per second ($\sim 250\ kbps$), bandwidth is a highly constrained resource in these networks. Also energy efficiency is a primary concern in these networks and communication cost is known to be the highest in terms of energy consumption. Hence it is imperative to maximize bandwidth utilization in these networks.

One of the primary objectives of a data gathering application is to present an accurate view of the sensed environment. This objective can be achieved only if we are able to obtain a fair amount of data from each of the sensors that are part of the network. This leads to the requirement of fair rate allocation amongst all sources in the network. Hence rate allocation amongst sources need not only be efficient (maximize utilization), but also fair([11], [12], [13]).

Additive increase-based mechanisms for rate control are popular in the context of wired networks. This is because they are optimal for lexicographic fairness [1], as well as for other notions of fairness such as proportional fairness [2]. The popularity of additive increase algorithms in wireline networks have also led to their adaptation to rate allocation in wireless sensor networks [9], [11].

The main difference between wireless networks and wireline networks is that flows in a wireless network not only consume bandwidth *usefully* on the links they are active on but also *wastefully* on links that they interfere with. Moreover, there is heterogeneity in the amount of interference (i.e., bandwidth wastage)

This work is supported in part by NSF grants numbered 0435505, 0347621, 0627028, 0430061 and 0325875

that each flow may cause. This fundamental difference between wired and wireless networks demands a fresh look at the problem of fair and efficient rate control algorithms for wireless networks in general and wireless sensor networks specifically.

In wireless settings, a fair rate allocation may treat equally flows that cause high interference as well as flows that cause less interference. On the contrary, a rate allocation that favors flows causing less interference may be able to provide higher network utilization (as measured by the total sum of the flow rates). Hence, there can be a fundamental tension between fairness and efficiency in wireless networks [3]. Consequently, the additive increase approaches that provide lexicographic fairness even in the context of wireless networks, are not well suited from the perspective of bandwidth efficiency. In this work, to address both fairness and efficiency goals, instead of looking at lexicographic fairness, we define the objective as maximizing the network utilization while ensuring that the rate allocations satisfy a slightly weaker notion of max-min fairness.

We model the problem as follows: There are n sources in the network that are trying to send data to a single sink over a given tree. Every source has a shortest path through one or more intermediate nodes to the sink. Every receiver in the network has limited bandwidth. The objective of the problem is to maximize the sum of the source rates subject to a constraint of max-min fair rate allocation. We define a rate allocation to be max-min fair if the minimum rate allocated to any flow is the maximum over all possible rate allocations.

One of the challenges of presenting a quantification to the above problem is to capture the effects of interference which is so unique to wireless networks. We achieve this by adding new links to the existing routing tree to represent interference between any two nodes. Further, instead of using a link-capacitated view where each link has a finite capacity we assume a node-capacitated view where each node has a finite capacity to receive data. This approach is critical to modeling wireless networks since a wireless network, unlike a wire line network, is composed of broadcast domains associated with each receiver instead of point to point links.

In order to motivate the applicability of the receiver capacity model in a real system, based on a CSMA based MAC, we also present empirical results on the Tmote Sky platforms where we measure the capacity region of the broadcast domain of a receiver. The empirical results for the two sender case and the receiver capacity values for the multiple sender (> 2) case suggests that the receiver capacity of the broadcast domain can be approximated by a linear relationship of the sender rates belonging to the specific broadcast domain. This observation corroborates the applicability of our receiver capacity model.

Using the receiver capacity model we formulate the above problem as two coupled linear programs — the first problem identifies the max-min rate allocation, while the second maximizes the sum-rate subject to the constraint determined by the solution of the first problem. We prove that the optimal solution to the first problem is the minimum of ratios of available bandwidths to upstream demands. This characterization allows for the efficient solution of the first problem via a tree-based aggregation and dissemination. We analyze the second problem using Lagrange duality. The analysis of the dual also presents us with an intuitive proof of the sub-optimality of additive increase mechanisms to achieve our objective of maximizing utilization while achieving a max-min fair rate allocation. Although solving the dual problem using sub-gradient search techniques can potentially result in slow convergence, we find empirically that initializing all shadow prices to a an equal, constant value of $\frac{1}{N+1}$ provides near-optimal results within one iteration. This gives a fast near-optimal distributed heuristic (which provides solutions within 2% of the optimum in 99.65% of the cases) that has an intuitive explanation — flows from sources are scheduled sequentially after rank ordering them on the number of downstream receivers whose bandwidth they consume (either directly or via interference).

The near optimal performance of the heuristic was quite surprising. On further investigating the results, obtained from the heuristic, and comparing the results with the optimal solution the following hypothesis was formulated; As long as the heuristic is able to generate a rank ordering of the sources 'similar' to the rank ordering of the optimal, the solutions obtained by the heuristic would be very close to the optimal. Further the structure of the problem itself lends to a solution where a majority of the sources are actually allocated the max-min rate in the final solution. Thus the rank ordering is limited to a small subset of sources, which helps the heuristic achieve a rank ordering similar to the optimal solution.

This paper is organized as follows: In section II we present our receiver bandwidth capacity model that will be essential in modeling the interference constraints in our optimization problem. In section III we empirically motivate our receiver capacity model in terms of its applicability to a real CSMA based wireless network. In section IV, using the bandwidth capacity model we formulate the problem of maximizing network utilization while allocating a max-min fair rate as two coupled linear programs. In section V we present a lemma that helps us calculate the max-min rate in a tree. In section VI we present an example to motivate our claim that additive increase algorithms while providing a max-min fair rate allocation but do not maximize network utilization. In section VII we take a dual based approach to design a near optimal heuristic for our problem. We also use the dual to present an intuitive proof for the sub-optimality of the additive increase algorithms. In section VIII we present simulation results to highlight the performance of our algorithm. In section IX we investigate the near optimal performance shown by our 1-step shadow pricing algorithm. In section X we present the related work pertinent to this problem. Finally in section XI we present our conclusions and the future for this work.

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. This model is identical to the one proposed by us in [9] and is similar to the one used by Rangwala *et al.* [11] to capture the effects of interference. We define a communication graph $\mathbf{G}$ as a set of nodes $\mathbf{V}$ and a set of communication links $\mathbf{E}$. To keep the analysis tractable we assume that the all communications links are perfect. A routing tree $\mathbf{T} \subset \mathbf{G}$ is created over the existing communication graph by selecting edges that would give the shortest hop count from a node to the root chosen randomly from the set of nodes $\mathbf{V}$. The assumption is that the tree $\mathbf{T}$ once selected remains fixed for the entire life time of flows existing in the network.

Every receiver in the network is considered to have a constant finite receiver capacity B (this receiver capacity could be different for different nodes in the network). 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 a wireless network very different from that observed on a wired network.

We illustrate our model, which we refer to as the "Receiver Capacity Model" for the remainder of this work, with an example. Figure 1 shows a 6 node topology. The solid lines indicate a parent child relationship in the tree. The dashed line represent noise links. For each source, any rate consumed by the source on the link with its parent would result in consumption of an equal rate on the noise links. Thus when node 2 sends its data to node 1, node 2 not only consumes capacity at node 1 but also at node 3, since the same flow exists over link $2 \rightarrow 1$ and noise link $2 \rightarrow 3$.

The radios are assumed to be half duplex. The half duplex nature of the radio forces flows to be received at a particular rate in a particular slot and then forwarded at the same rate in the next available slot. This results in flows, originating from the child, consuming twice the allocated rate at the parent.

Based on our model the constraint on the rates at node 3 would be as follows:

$$r_{noise}^{(2)} + r_{noise}^{(3)} + r_{src}^{(6)} \leq B^{(3)} \tag{1}$$

where $B^{(3)}$ is the receiver capacity of node 3 and $r_{src}^{(6)}$ is the source rate of node 6. $r_{noise}^{(2)}$ and $r_{noise}^{(3)}$ are the output rates at node 2 and node 3 respectively and are given by:

$$r_{noise}^{(2)} = r_{src}^{(2)} + r_{src}^{(4)} + r_{src}^{(5)}$$

and

$$r_{noise}^{(3)} = r_{src}^{(3)} + r_{src}^{(6)}$$

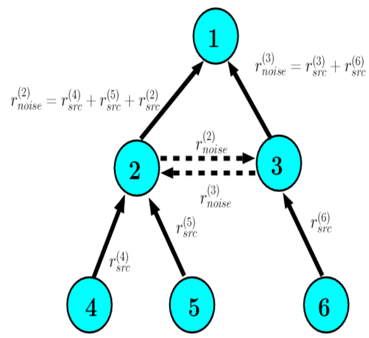


Fig. 1. A 6 node topology: An illustrative example of the receiver capacity model

The half duplex assumption for the radios forces the term $r_{src}^{(6)}$ to appear twice in equation 1. Once independently to account for the consumption of bandwidth during reception at node 3 and once as part of the term $r_{noise}^{(3)}$ to account for the forwarding of the flow originating at node 6.

In general the receiver capacity constraint at a node i can be given as follows:

$$\sum_{j \in C^{(i)}} r_{src}^{(j)} + \sum_{j \in N^{(i)}} \sum_{k \in C^{(j)}} r_{src}^{(k)} + \sum_{j \in N^{(i)}} r_{src}^{(j)} \leq B^{(i)} \tag{2}$$

Where $N^{(i)}$ is the set of all neighbors of i. The half duplex assumption implies that $i \in N^{(i)}$. $C^{(i)}$ is the set of all nodes j that have i in its path to the sink. $r_{src}^{(j)}$ represents the rate at which data generated at node j is being transmitted.

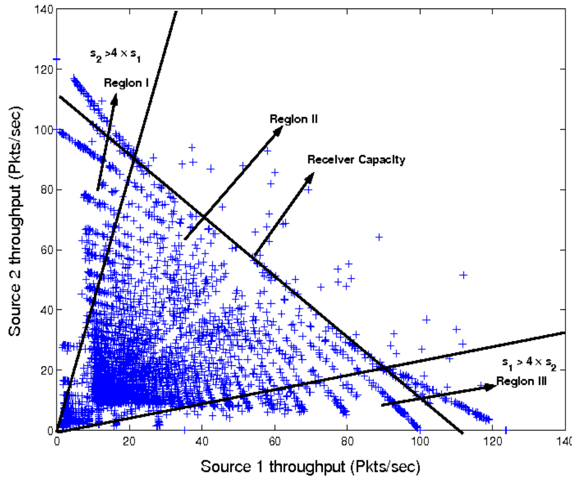
III. Empirical Validation of the Receiver Capacity Model

In order to ascertain the efficacy of the above model in a real system we performed an experiment using 3 Tmote Sky's in a 2 sender, single receiver configuration. The Tmote Sky's have a CC2420 radio which is compatible with the IEEE 802.15.4 MAC protocol. The Tmote Sky's were running TinyOS-2.x which has a default CSMA stack available for the CC2420 radio's. By making the sender motes transmit at different combination of source rates, we were able to plot the capacity region of the CC2420 CSMA stack for the TinyOS-2.x platform shown in figure 2(a). The x and y axis represent the good put achieved by each of the sources. Thus a cross in figure 2(a) represents a combination of source rates that was achieved by each of the sources.

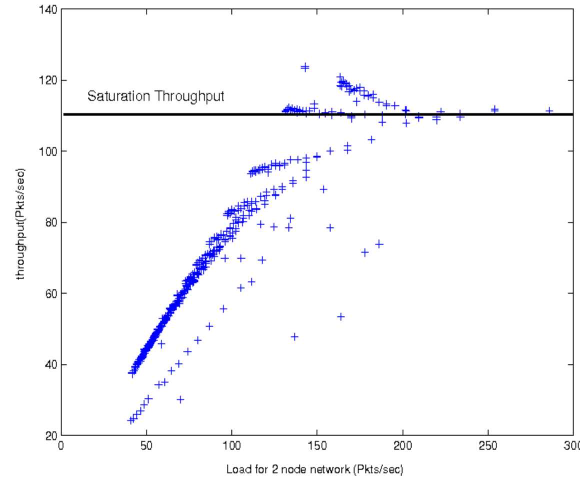
The receiver capacity is determined by the boundary of the capacity region (set of achievable rate vectors) of the MAC protocol. This plot shows that the capacity region can be divided into three regions. When s_1 and s_2 are comparable to each other (Region II), the capacity region is $s_1 + s_2 \leq 110$ pkts/sec, where 110 pkts/sec is the receiver capacity in this region. Region I/III corresponds to rate vectors (s_1, s_2) where $s_2 > 4s_1$ ($s_1 > 4s_2$).

An interesting observation that can be made from figure 2(a) is that although the receiver capacity in regions I and III is greater then the receiver capacity in region II, if we simply extend the boundary of region II into regions I and III the loss in capacity would be small. Thus we could represent the boundary of the capacity region by a linear combination of the source rates of sources 1 and 2.

The empirical evidence presented in figure 2(a) justifies our linear approximation of the receiver capacity (currently for the two sender single receiver case). With the simple two sender experiments we can further show that the receiver capacity is equal to the saturation throughput of the CSMA MAC. The saturation throughput of the MAC, is the overall throughput seen by the receiver when the system is overloaded, i.e. each source in the broadcast domain always has a packet to transmit. Figure 2(b) presents the load vs throughput curves for the two sender, single receiver case. The x-axis plots the sum load on the system



(a) Capacity region for two senders single receiver



(b) Saturation throughput for the two sender single receiver case

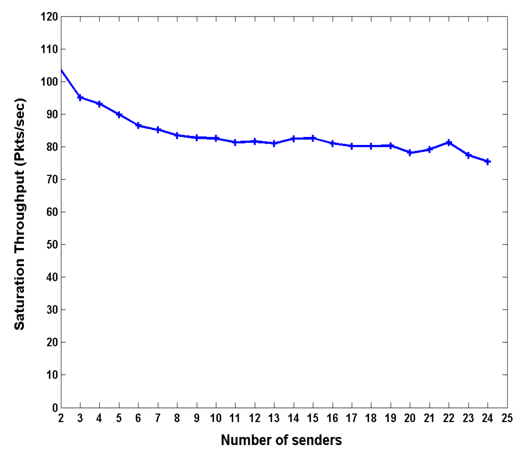


Fig. 2. Receiver capacity for senders > 2

(the sum rates of the two sources) and the y-axis plots the overall throughput observed at the receiver. As can be seen the saturation throughput coincides with the observed receiver capacity. We were able to equate the points that correspond to the saturation throughput in figure 2(b) to the the points on the boundary of the capacity region II in figure 2(b), validating our observation. Thus the receiver capacity of a broadcast domain can be equated to the saturation throughput of the MAC protocol.

Although the above relation, between the receiver capacity and the saturation throughput, is inferred from an experiment where only two senders were present, we claim that this relationship holds for $n > 2$ senders as well. To justify this claim we measured the saturation throughput for multiple senders (> 2), for the TinyOS-2.x CSMA stack in figure 2. Figure 2 shows that the degradation in saturation throughput with increasing number of senders is relatively small. For e.g. the drop in saturation throughput when 8 to 20 senders are present in a broadcast domain as compared to when only 3 senders are present is just 20%. This suggests, that at the edges of the capacity region, when one sender's source rate is negligible as compared to the source rate of all the other senders, the receiver capacity is not much larger than the receiver capacity at the middle of the capacity region where the source rates are comparable (This is similar to the comparison of region I and III with region II in the two sender case). Thus the capacity region for a multi sender case could be approximated by a plane with boundary of the capacity region represented by a linear combination of the source rates. Hence, the motivation to use the equivalence

TABLE I
LIST OF VARIABLES USED IN FORMULATING PROBLEMS $\mathbf{P1}$ AND $\mathbf{P2}$

$\mathbf{R_{src}}$:	An $N \times 1$ vector representing the rate allocated to each source $i \in \mathbf{V}$
$\mathbf{N}$:	An $N \times N$ matrix representing the presence of a noise edge $n_{ij} \in N$ between two nodes $i, j \in \mathbf{V}$
$\mathbf{C}$:	An $N \times N$ matrix that gives the parent-child relationships on the data gathering tree. $c_{ij} \in C^{(i)}$ is 0 if node i is not in node $j^{'s}$ path to the sink and $c_{ij} = 1$ otherwise.
$\mathbf{R_{in}}$:	An $N \times 1$ vector, representing the total input rate arriving at each node.
$\mathbf{R_{noise}}$:	An $N \times 1$ vector, representing to total output rate exiting from a node.
Y:	A scalar, representing the minimum rate among all flows.

between the saturation throughput and the receiver capacity holds for the multi-sender case as well.

IV. Problem Formulation

Using the receiver capacity model we can now formulate the maximization of the network capacity utilization while maintaining max-min fairness as two coupled constrained optimization problems $\mathbf{P1}$ and $\mathbf{P2}$. The variables used in our formulation are presented in table I:

The optimization problem is formulated as follows:

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

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

The constraints of our optimization problem come directly from our bandwidth consumption model that we had presented in section II. The problem $\mathbf{P1}$ is the max-min rate problem. The optimal solution Y^* to $\mathbf{P1}$ gives the highest possible minimum rate achievable amongst all possible rate allocation vectors. The problem $\mathbf{P2}$ uses Y^* as a constraint in order to guarantee the best possible minimum rate to all its sources and presents a rate allocation vector that will maximize the sum rate, thus maximizing utilization. In the following section we present a lemma showing that the solution to $\mathbf{P1}$ can be found by taking the minimum of the ratios of the available bandwidths to upstream demands. The algorithm itself can be implemented by using a tree-based aggregation and dissemination mechanism.

V. Calculating the Max-Min source rate on a tree

The max-min rate is the optimal solution to the problem $\mathbf{P1}$ denoted by Y^*. In order to calculate the max-min rate for a given tree we define the term available bandwidth at a receiver ($B_{available}^{(i)}$) as follows:

$$B_{available}^{(i)} = \frac{B^{(i)}}{\Gamma^{(i)}} \tag{3}$$

Where $\Gamma^{(i)}$ is defined as:

$$\Gamma^{(i)} = \sum_{j \in C^{(i)}, i \neq j} c_{ij} + \sum_{j \in N^{(i)}} \sum_{k \in C^{(j)}, k \neq j} c_{jk} + \sum_{j \in N^{(i)}} n_{ij}$$

$\Gamma^{(i)}$ is the sum of the total number of immediate children of node i, the total number of neighbors of node i and the total number of children of each of node i's neighbors.

The optimal solution of $\mathbf{P1}$ could be found by observing the available bandwidth $B^{(i)}_{available}$ at each receiver in the network and selecting the minimum of these. The following lemma justifies our claim.

Lemma 5.1: The optimal solution Y^* of the primal $\mathbf{P1}$ is the $\min(B^{(i)}_{available})\ \forall\ i \in \mathbf{V}$.

Proof: We define a node k as a bottle neck node if:

$$k = \text{argmin}(B^{(i)}_{available})\ \forall\ i \in \mathbf{V}$$

- **Case 1:** Assume:

$$Y^* < \min(B^{(i)}_{available})\ \forall\ i \in \mathbf{V}$$

  We can do a rate allocation for all sources j that are children of the bottle neck node k or the children of the neighbor of the bottle neck node k such that $r^{(j)}_{src} = B^{(k)}_{available}$ without violating the bandwidth constraint on node k. Since k is the bottle neck node, r^{j}_{src} will be the minimum of all rates allocated to all sources. This implies that we have a rate allocation where

$$\min(r^{i}_{src}),\ \forall\ i\ \in\ V > Y^*$$

  Thus we have a contradiction.
- **Case 2:** Assume $Y^* > \min(B^{(i)}_{available})\ \forall\ i \in V$. Then for node k,

$$\sum_{j \in C^{(k)}} r^{(j)}_{src} + \sum_{g \in N^{(k)}} \sum_{z \in C^{(g)}} r^{(z)}_{src} + \sum_{j \in N^{(k)}} r^{(j)}_{src} > B^{(k)}$$

  Thus the bandwidth capacity constraint is violated for node k.

Hence $Y^* = \min(B_{available}(i))\ \forall\ i\ \in\ \mathbf{V}$. ■

Based on this lemma a simple algorithm can be developed to calculate the max-min rate on a tree. In order to find the max-min rate every child calculates its available bandwidth ($B^{(i)}_{available}$) and forwards it to the parent. The parent computes the minimum of these and compares it with its own available bandwidth. It then forwards the minimum of these two quantities to its parent. The parent thus performs an aggregation on the available bandwidth in its sub-tree and forwards the minimum to its own parent. Since a parent does not calculate its available bandwidth and forward the aggregated minimum available bandwidth to its parent, till it receives information from all its children, the aggregation process proceeds sequentially. The algorithm to calculate the max-min rate terminates at the root. The minimum available bandwidth calculated by the root would then be the minimum of all available bandwidths and hence would quantify the max-min rate of the tree. The root can now disseminate this information to every node in the tree by sending it downstream over the tree.

VI. ADDITIVE INCREASE ALGORITHMS AND MAX-MIN FAIRNESS

Currently proposed solutions that achieve max-min fairness while implicitly trying to maximize network utilization([9], [11]) use the following additive increase mechanism. Sources in the network are allowed to increase their rates equally by a small value ϵ. 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 $\mathbf{P1}$. Even though additive increase algorithms can achieve a solution to $\mathbf{P1}$, while consuming the network capacity, we claim that it will not necessarily achieve a solution for $\mathbf{P2}$. In this section we present insights into our claim through an example and present a more rigorous proof in Section VII. Assume all nodes except node 1 are sources in figure 3. Let node 5 be the bottleneck node. For this topology any increment in the rate of node 3 will consume bandwidth at node 2 and node 4. For e.g. if we increment the rate at node 3 by ϵ we will be consuming a bandwidth ϵ at receiver 2, a

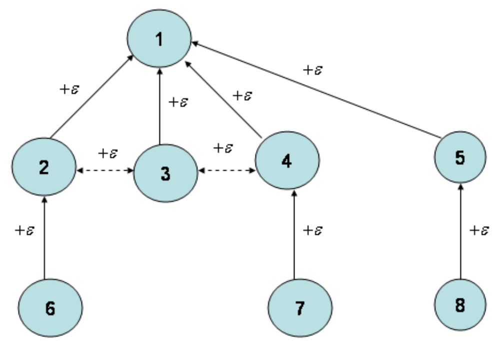


Fig. 3. An example depicting the sub-optimality of the additive increase technique for maximizing network utilization while maintaining a max-min fair rate allocation.

bandwidth ϵ at receiver 4 and a bandwidth ϵ at receiver 1. Thus an increment ϵ in source rate of node 3 will result in wastage of network capacity equal to 2ϵ. A higher throughput could be achieved by simply allocating all nodes the max-min rate and then giving the remaining capacity to nodes 2 and 4. It is easy to see that this allocation would ensure that for increment ϵ in source rates of 2 and 4 they would not be wasting any bandwidth. 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 ϵ. In real systems an accurate estimate of ϵ is critical to avoid oscillations [11]. 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 ϵ.

VII. A Dual Based Approach

In order to gain insights into the dynamics of the problem we plan to adopt a dual based approach. The shadow price interpretation of the Lagrange multipliers [10] from the dual will present us with mechanisms to design distributed algorithms that maximize the network utilization while guaranteeing a max-min fair rate to all sources.

A. The Lagrange Dual

We introduce Lagrange multipliers in order to relax constraints in the primal $\mathbf{P2}$ to obtain the Lagrange dual function. We will concern ourselves only with the dual of $\mathbf{P2}$ and assume that the optimal max-min rate will be calculated from the primal $\mathbf{P1}$ using lemma 5.1. The Lagrange dual function of the primal $\mathbf{P2}$ is:

$$D(\boldsymbol{\lambda}) = \max_{\mathbf{R_{src}} \succeq \mathbf{Y}^*} \left(\sum_{i \in T} r_{src}^{(i)} - \boldsymbol{\lambda}^{\mathbf{T}} \times ((\mathbf{N} \times (\mathbf{C} + \mathbf{1}) + \mathbf{C}) \times \mathbf{R_{src}} - \mathbf{B}) \right)$$

On expanding the matrix notation we get:

$$\begin{aligned} D(\boldsymbol{\lambda}) \;\; &= \max_{\mathbf{R_{src}} \succeq \mathbf{Y}^*} (\textstyle\sum_{i \in T} r_{src}^{(i)} \\ &- \textstyle\sum_{i \in T} \lambda_i (\sum_{j \in C^{(i)}} r_{src}^{(j)} \\ &+ \textstyle\sum_{j \in N^{(i)}} \sum_{k \in C^{(j)}} r_{src}^{(k)} \\ &+ \textstyle\sum_{j \in N^{(i)}} r_{src}^{(j)} - B^{(i)})) \end{aligned} \tag{4}$$

We can rearrange equation 4 to obtain:

$$\begin{aligned} D(\lambda) \;\; &= \max_{\mathbf{R_{src}} \succeq \mathbf{Y}^*} (\textstyle\sum_{i \in T} r_{src}^{(i)} (1 \\ &-(\textstyle\sum_{i \in C^{(j)}} \lambda_j \\ &+ \textstyle\sum_{i \ \in C^{(j)}} \sum_{k \in N^{(j)}} \lambda_k \\ &+ \textstyle\sum_{i \in N^{(j)}} \lambda_j)) + \sum_{i \in T} \lambda_i B^{(i)}) \end{aligned} \tag{5}$$

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

$$\mathbf{D} : \min_{\boldsymbol{\lambda} \succeq \ 0,} D(\boldsymbol{\lambda})$$

Also since the solutions are feasible for both problems the duality gap would be zero [10]. 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 λ_i is:

$$\frac{\partial D}{\partial \lambda_i} = -(\zeta^i(R^*_{src}) - B^{(i)}) \tag{6}$$

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. Analyzing the Dual to Design a Distributed Algorithm

The Lagrange dual function can be rewritten as:

$$\mathbf{D} : \min_{\lambda \succeq 0} \left(\max_{\mathbf{R_{src}} \succeq \mathbf{Y}^*} \left(\sum_{\forall \ i} r_{src}^{(i)} \mu_i \right) + \sum_{\forall \ i} \lambda_i B^i \right)$$

Where μ_i is given by:

$$\mu_i = 1 - (\sum_{i \in C^{(j)}} \lambda_j + \sum_{i \in C^{(j)}} \sum_{k \in N^{(j)}} \lambda_k + \sum_{i \in N^{(j)}} \lambda_j) \tag{7}$$

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 λ_i in the direction of the negative gradient as follows:

$$\lambda_i(t+1) = [\lambda_i(t) + \alpha_t(\zeta^i(R^*_{src}) - B^i)]^+ \tag{8}$$

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

$$\max_{\mathbf{R_{src}} \succeq \mathbf{Y}^*} (\sum_{\forall \ i} r_{src}^{(i)} \mu_i) \tag{9}$$

At every step t we are required to find the $\mathbf{R^*_{src}}$ that solves equation 9. The resulting $\zeta^i(R^*_{src})$ would then be used to calculate a new value of λ using equation 8. If the μ_i in equation 9 are negative the optimal value of $\mathbf{R}^*_{\mathbf{src}} = Y^*$. This would result in a decrement of $\lambda_i(t)$, since $\zeta^i(R^*_{src}) < B^i$. If all μ_i are positive, in order to optimize equation 9, we could set $R^*_{src} = \inf$. This however would result in a large increment in the value of $\lambda_i(t)$ since $\zeta^i(R^*_{src}) >> B^i$. Thus when all μ_i we should choose an R^*_{src} such that $\zeta^i(R^*_{src}) \leq B^i$. In other words, although we could simply run the iterative algorithm and allow λ_i to oscillate around the optimum value and converge over some number of iterations (which could be potentially large), we can be more intelligent and never allow $\zeta^i(R^*_{src})$ to exceed to B^i. Further given a fixed λ_i and a positive μ_i, in order to find a feasible solution for equation 9, assuming λ is fixed, we will require to allocate all $r^{(i)}_{src}$ at least the max-min rate Y^*. Given that all sources are allocated at least the max-min rate, we would require to allocate the maximum available bandwidth to the source i having the highest μ_i. We would then proceed, allocating the remaining bandwidth to the source with the second highest μ_i. We continue allocating bandwidths to sources till all sources have been constrained. Thus bandwidth allocation is based on an ordering of the sources based on their coefficients μ_i. Also, instead of looking at μ_i for each source i we could assign each source i a weight w_i given by:

$$w_i = \frac{1}{\sum_{i \in C^{(j)}} \lambda_j + \sum_{i \in C^{(j)}} \sum_{k \in N^{(j)}} \lambda_k + \sum_{i \in N^{(j)}} \lambda_j} \tag{10}$$

The ordering, and the prioritization of rate allocation, in order to maximize equation 9, can now be done based on the weights w_i for each source i.

To achieve the optimal $\mathbf{D}$ we should be running the sub-gradient algorithm for multiple iterations ($t > 1$), solving the maximization problem in equation 9, until the shadow prices converge. Fortunately through simulations we can show that by setting $\lambda_i = \frac{1}{N+1}, \forall\ i$ in our specific problem, 99.65% of the time we achieve close to 2% of the the optimal in the very first iteration. The details of the simulation and its performance with respect to the optimal are presented in section VIII. Thus instead of running the sub-gradient algorithm for multiple iterations, we set the shadow prices $\lambda_i = \frac{1}{N+1}, \forall\ i$ and perform only the first iteration of the sub-gradient algorithm. Our algorithm for maximizing network utilization with max-min fair rate allocation therefore simply consists of optimizing equation 9 by setting the shadow prices to 1. The specifics of the algorithm have been provided in section VII-D.

Setting the shadow prices $\lambda_i = \frac{1}{N+1}, \forall\ i$, presents an intuitive explanation to the algorithm. When we set all shadow prices to an equal constant (say $\lambda_i = \frac{1}{N+1}, \forall\ i$) , the weight w_i is inversely proportional to the number of receivers node i interferes with during its data transmission to the sink. Thus the ordering suggests that we allocate the maximum bandwidth to nodes that cause the least amount of interference.

C. Sub-Optimality of the Additive Increase Algorithms

In section VI we presented a motivating example for the sub-optimality of additive increase algorithms. Our analysis of the dual in the previous section provides a more quantitative argument for this claim. Primarily it suggests that the rate allocations in the network need to follow an ordering based on the amount of interference that each source generates while transmitting data to the sink. On the contrary, in additive increase algorithms no such ordering exists since all sources are allowed to increment by the same amount. The lack of prioritization in rate allocation is the primary cause for the sub-optimality of additive increase algorithms.

D. The Algorithm

We now present an algorithm for the maximization of equation 9. The algorithm '*Maximization of Network Utilization'* presented in figure 4 proceeds as follows; In the **initialization** phase all sources in the network set their 'CONSTRAINED' flag to 'FALSE'. Every node i calculates its weight w_i using equation 10 and setting the shadow price $\lambda_i = \frac{1}{N+1},\ \forall\ i$. In order to calculate the weight w_i, the node i requires information about the number of parents it has (the number of nodes between itself and the sink), and the number of neighbors of each of its parents and the total number of its neighbors. Each of the three quantities can be obtained by the node during the process of tree formation itself. In effect, every node during the tree formation process, needs to forward the total number of neighbors it possesses and the number of hops to the sink. These two quantities can be used by the nodes to calculate the quantities mentioned above for calculating the weight w_i at the end of the tree formation.

In **step 1**, each node calculates its per node available bandwidth. The bottle neck bandwidth is then the minimum of all the available bandwidths. From lemma 5.1, this bottle neck bandwidth is the max-min rate and hence is allocated to every source in the network. An algorithm to calculate the minimum available bandwidth on a tree using a tree-based aggregation and dissemination mechanism is presented in section V.

In **step 2**, we calculate the pending bandwidth at each node in the network. To calculate its pending bandwidth every receiver notes the total output rate from each of its children and the total output rate from each of its neighbors. The pending bandwidth is then the difference between the bandwidth capacity of the receiver and the sum of the output rates from all its children and its neighbors. 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. From an implementation perspective, for this step we require that every node has information about the maximum weight currently active in the network. This can be done by pushing the information about the weights to the root and the root then disseminating the maximum weight to all its children. The calculation of the minimum pending bandwidth, described above, can also be done using a tree-based implementation. Every node starting from the root needs to gather its own pending bandwidth and its neighbors pending bandwidth and pass on to its children the minimum of these quantities.

Once a node has incremented its bandwidth (since it was the node with the highest weight), it would become constrained since it would have consumed the maximum available bandwidth in its path. Therefore it would require to remove itself from the list of active sources allowing some other node to become the source with the highest weight. It can perform this operation by informing the root and allowing the root to disseminate this information over the tree.

In **step 4**, we check the constrained flag for all nodes in the network and if all nodes have been constrained the algorithm terminates, else we repeat the algorithm from **step 2**.

While describing the various steps of the algorithm we have presented an implementation perspective to these steps as well. The implementation description gives an operational picture of the algorithm in a real system. This description suggests that although the algorithm is not completely distributed (the decision making is not completely local, it relies on information exchange with the root) it would be more scalable than an implementation where all the computation is done centrally — maintaining the complete topology information centrally and running an LP solver to compute the optimum. By allowing information exchange between the root and the various nodes we have made most of the computation distributed (the pending bandwidth and the weights are calculated locally at the nodes) and have reduced the complexity of the computation at the root. Our asymptotic analysis of the algorithm suggests that the over head of this information exchange is not high, giving us an acceptable polynomial bound on the number of messages exchanged.

E. Asymptotic Bounds for the Dual based Algorithm

The asymptotic bounds on the dual based algorithm can be calculated as follows: **Step 1** of the algorithm would take $O(n)$ transmissions to calculate the max-min rate. **Step 2** of the algorithm would take $O(n)$ transmissions to calculate the pending bandwidth at each of the intermediate nodes. In **Step 3** of the algorithm once a source node is constrained it needs to populate this information to all nodes in the tree in order to remove itself from the list of source nodes. In order achieve this goal a simple mechanism

Algorithm *Maximization of Network Utilization:*

Initialization
$contrained_i$ = FALSE $\forall\ i$
$\lambda_i = \frac{1}{N+1}\ \forall\ i$
$w_i = \frac{1}{\sum_{i \in C^{(j)}} \lambda_j + \sum_{i \in C^{(j)}} \sum_{k \in N^{(j)}} \lambda_k + \sum_{i \in N^{(j)}} \lambda_j} \forall\ i,\ w_i\ \in\ \mathbf{W}$
[Step 1] max-min Rate:
$B^{(i)}_{available} = \frac{B^{(i)}}{\sum_{j \in C^{(i)}, i \neq j} c_{ij} + \sum_{j \in N^{(i)}} \sum_{k \in C^{(j)}, k \neq j} c_{jk} + \sum_{j \in N^{(i)}} n_{ij}} \forall\ i$
$r^{(i)}_{src} = \min(B^{(i)}_{available})\ \forall\ i$
[Step 2] Pending Bandwidth:
for $\forall\ i$ **if** $\sum\ c_{ij} \neq 0$ such that $j \in\ C^{(i)}, j \neq i$
    do $B^{(i)}_{pending} = B^i - \zeta^i(B_{src})$
        if $B^{(i)}_{pending} \leq 0$ **then**
            do Constrain all children, neighbors, and neighbors children.
                Remove constrained nodes from list $\mathbf{W}$.
[Step 3] Updating Source bandwidth:
for $\forall\ i$
    do $pend\_bw = \min(B^{(j)}_{pending}), \forall\ j$ **such that** $i \in C^j$ **or** $i \in N^j$ **or** $k \in N^j,\ i \in C^k$
        if ($w_i == max(\mathbf{W})$ **and** $constrained_i == FALSE$) **then**
            do $r^{(i)}_{src} = r^{(i)}_{src} +\ pend\_bw$
[Step 4] Checking termination condition:
if ($contrained_i$ = TRUE) $\forall\ i$ **then end**
else goto **Step 2**

Fig. 4. Algorithm for the maximization of network utilization

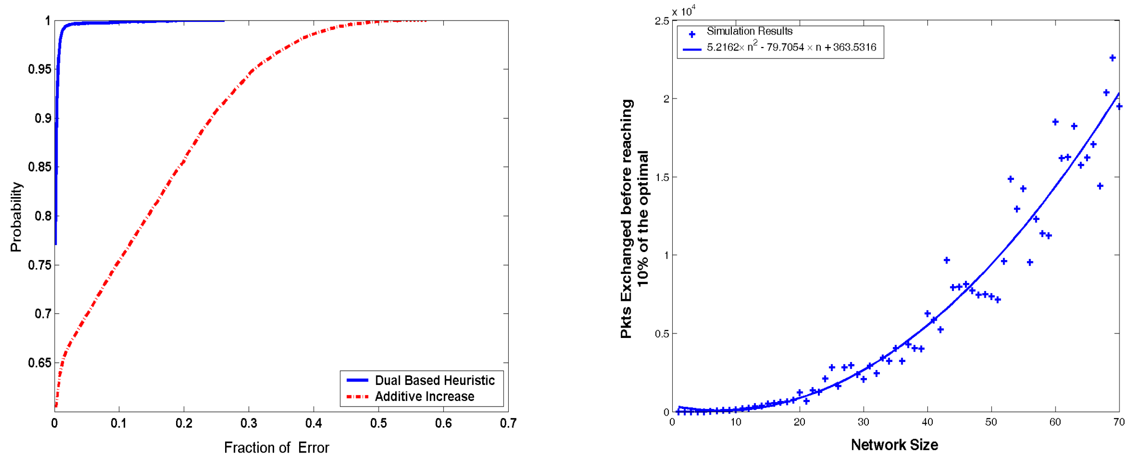
would be to propagate this information to the root which will collate this information into a new list of sources that are capable of incrementing their bandwidth. This new list could than be propagated to all sources in the tree. Since the total edges in a tree having n nodes are $n-1$, the total number of transmissions to accomplish **Step 3** would be $O(n)$.

The algorithm terminates when all sources are constrained. Thus **Step 3** and **Step 2** will be executed $O(n)$ times. Hence the algorithm would converge to the solution within $O(n^2)$ transmissions.

VIII. 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 10 and 250. We choose 20 such bandwidth distributions for each tree. Thus for each network size we have 9 different trees, for each tree there are 20 different instances (each with a different bandwidth distribution) giving a total of 180 instances for each network size. Since our network size ranges from 6 to 70, we have a total of $180 \times 65 = 11700$ different trees for our evaluation.

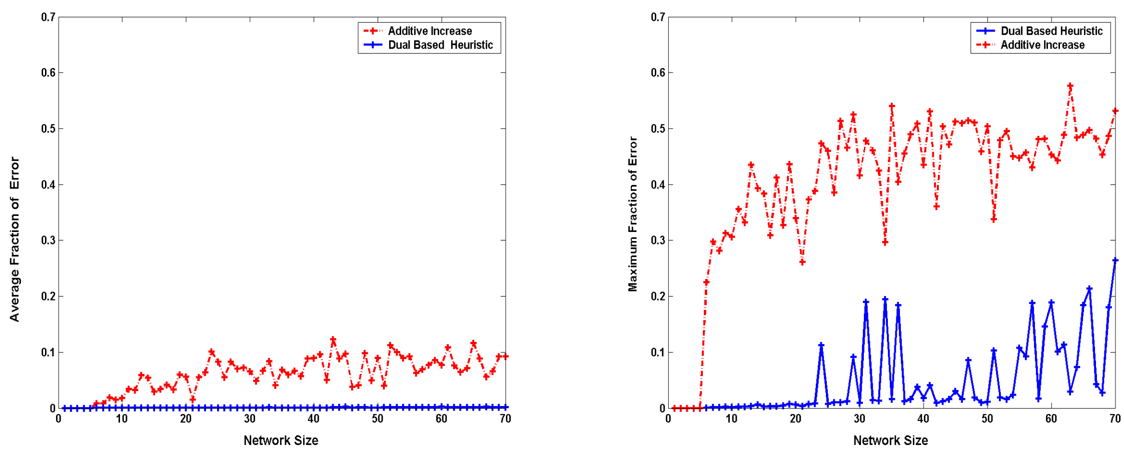
In order to evaluate the performance of our algorithm for each of the 11700 instances we generated an LP for the problem $\mathbf{P2}$. Using a centralized solver LP SOLVE [21] we obtained the optimal solution for the max-min rate Y^* and the solution to the problem $\mathbf{P2}$. We then ran our dual based algorithm and an additive increase algorithm, described in section VI, on each of the 11700 instances to solve the problem $\mathbf{P2}$ in a distributed manner. Figure 5(a) shows the CDF of the error between the optimal solution from a centralized solver and the solutions obtained from our dual based algorithm, and the additive increase algorithm. For the dual based algorithm the CDF in figure 5(a) shows that for 99.65% of the instances we are able to achieve close to 2% of the optimal throughput. Of the instances that had greater than or



(a) CDF of the error observed between the optimal throughput achievable and the throughput achieved using dual based algorithm and the additive increase algorithm.

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

Fig. 5. Performance evaluation of the dual based algorithm and the additive increase algorithm.



(a) Average percentage error vs network size

(b) Maximum percentage error vs network size

Fig. 6. Performance analysis of the dual based algorithm and the additive increase algorithm on the basis of the percentage error generated for different network sizes.

equal to 10% error, we were close to 10% of the optimal throughput in 18 instances and close to 20% in two of the instances. Figure 5(a) also highlights the sub-optimality of the additive increase algorithms. It shows that in more than 15% of the runs we experienced an error greater than 20%, there were 10% of the runs which experienced an error greater than 30% and 5% of the runs experienced an error greater than 40%. As highlighted in section VI the sub-optimality of the additive increase algorithm is due to the lack of prioritization of the sources during rate allocation.

In figure 5(b) we plot the number of packets transmitted before the dual based algorithm converges to a solution. Based on a regression fit, we estimate that it grows as O(n^2). These bounds match the asymptotic bounds that were obtained analytically in section VII-E.

We define the fraction of error as follows:

$$fraction\ of\ error\ =\ \frac{|optimal\ sum\ rate\ -\ heuristic\ sum\ rate|}{optimal\ sum\ rate}$$

Figures 6(a) and 6(b) show the average fraction of error and maximum fraction of error observed while running the dual based algorithm and the additive increase algorithms across different network sizes. For the dual based algorithm figure 6(a) reiterates the results of figure 5(a) showing that across different network sizes the average percentage error remains very close to zero. The additive increase algorithm however becomes progressively worse as the size of the network is increased. Although the average error exhibited by the additive increase algorithms is not large ($\sim\ 10-12\%$), the maximum error exhibited is quite large ($\sim\ 40-55\%$). The performance of the additive increase algorithm depends on the placement of the bottleneck node in the topology. If the bottleneck node is very close to the root, in most cases all sources would not be able to get more than the max-min rate even in the optimal solution. In these scenarios the additive increase algorithm would be able to achieve the optimal. However as the bottleneck node starts moving away from the root, the rate distribution among sources would change with a few sources getting very high rates in the optimal solution. Under such a scenario the additive increase algorithm seems to fail. For small networks, since the average diameter of the network is also small, the bottleneck node would be close to the root. However for large networks since the diameter is large, chances of the bottleneck node being farther away from the root are higher leading to an uneven distribution of source rates. This reasoning thus throws light on the performance of the additive increase algorithm as the network size is increased.

IX. Investigating the efficacy of 1-step shadow pricing

Our performance analysis of the heuristic shows that even though we have assigned equal shadow prices to all sources in the network, we are still able to achieve near optimal results.Ideally, if we had run the sub-gradient algorithm, at every step of the algorithm each source would have achieved a new shadow price. Hence, when the algorithm converges the optimal shadow price for sources need not be equal. This in turn implies that the rank ordering of the sources in the optimal solution and the heuristic could be different depending on the network topology and receiver bandwidth distributions for the specific network topology.

The above observation encourages us to formulate a hypothesis that we would be verifying through comparison of our results with the optimal solution. The observation suggests that the end solution is insensitive to the actual rank ordering of the sources, as long as we are able to achieve a relative ordering that is 'similar' to the optimal rank ordering we can achieve a utilization that is very close to the optimal.

The ability of the heuristic to achieve a 'similar' rank ordering comes from the structure of the optimization problem. The objective of the problem is to maximize the sum rate while ensuring that every source in the network gets at least the max-min rate. To achieve this objective, the sources with the highest rank would be given the max-min rate (lowest source rate) and the remaining bandwidth would be distributed amongst the remaining sources(starting with the source having the lowest rank or highest source rate). In the optimal solution a majority of the sources belong to the lowest rank. Therefore, as long as the rank order of the remaining sources obtained by assigning equal shadow prices are similar to the rank order obtained in the optimal solution, the heuristic will present near optimal solutions.

In order to validate the above hypothesis, and present a credible argument for the performance of the heuristic, we compared different metrics related to rank order of the sources obtained from the heuristic algorithm and the optimal solutions.

Figure 7 shows the cumulative percentage bandwidth allocated to sources versus the number of sources when arranged in sorted order. In order to present this comparison only the optimal values were used, which in turn were obtained by solving the LP for each of the 11700 network instances. As can be seen even if we consider 10 sources these sources account for approximately 50% of the total allocated bandwidth. This clearly implies that in the final solutions a small percentage of sources are allocated majority of bandwidth and the selection of these sources would determine the final solution.

In figures 8(a), 8(b), and 8(c) we dig a bit deeper and highlight the above trend on a per source basis. The X-axis in figure 8 represents the rank allocated to each source based on the source rate allocated to each source in the final solution. A lower rank represents a higher source rate. On the Y-axis we plot

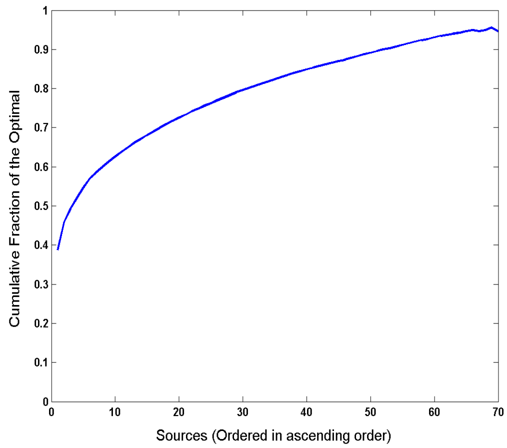
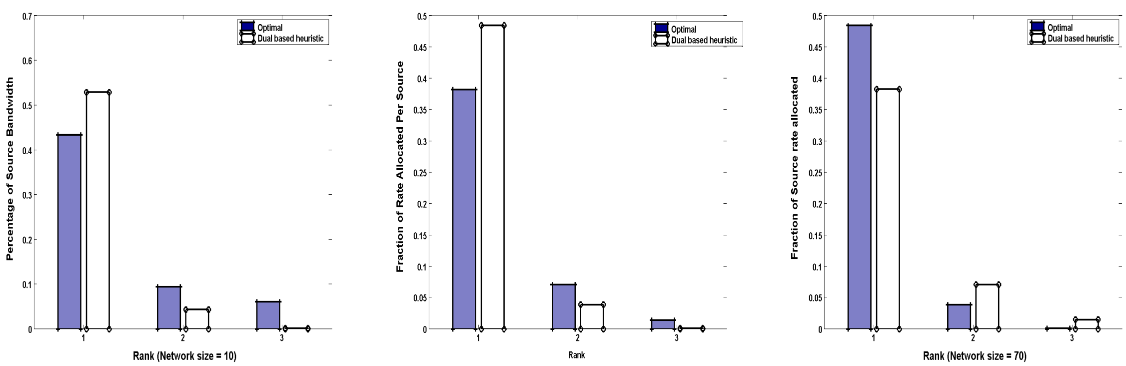


Fig. 7. The cumulative percentage of source bandwidth.



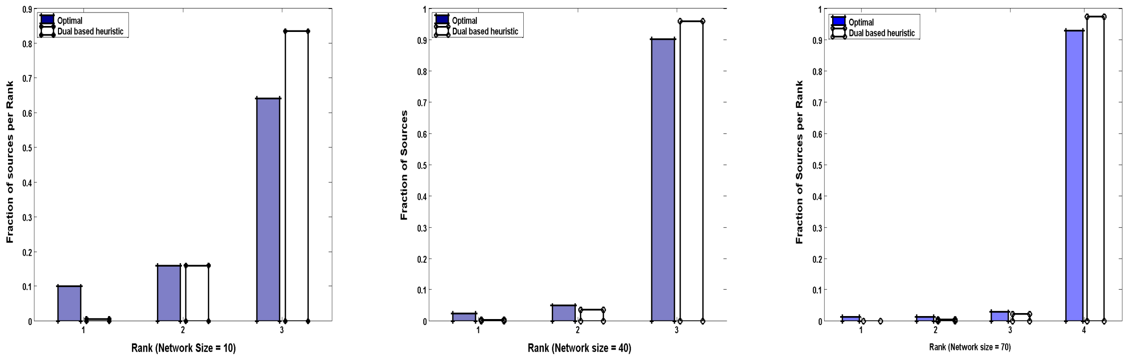
(a) Network Size = 10 (b) Network Size = 40 (c) Network Size = 70

Fig. 8. The distributions of the percentage of the total rate that was allocated to a source belonging to a particular rank for different network sizes. The skewed distributions highlight that a few sources are allocated a large percentage of the available bandwidth in order to maximize the utilization.

the average percentage of the total allocated bandwidth that was given to a single source belonging to a particular rank. We perform this analysis for networks of different sizes ranging from 10 to 70. We plot the values for the optimal solutions as well as the solutions obtained by running the heuristic. The data presented strengthens our claim that the higher rank sources account for a majority of the allocated bandwidth. Also the trend in the bandwidth distribution amongst ranks, for different network sizes, remains the same.

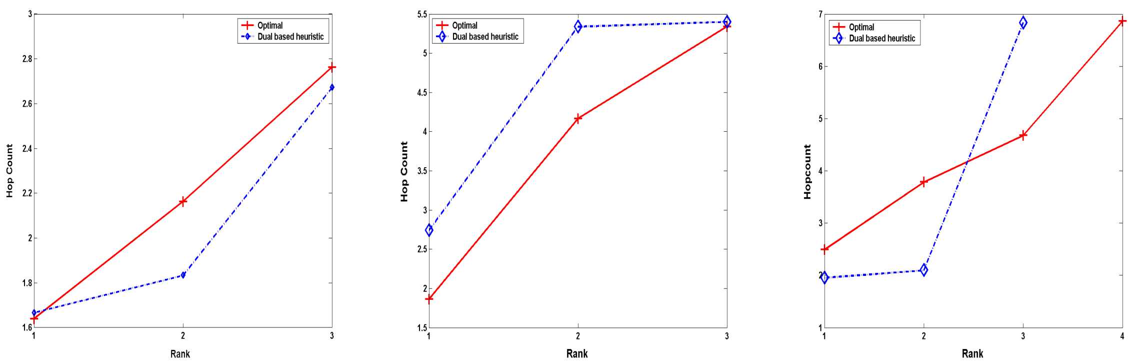
Another question we would like to ask is, what is the distribution of the number of sources belonging to different ranks. Figures 9(a), 9(b) and 9(c) answer this question. In contrast to the bandwidth distribution it can be observed that a majority of the sources (on an average 85% of the sources) belong to the highest rank. As in figure 8 the trend remains the same across different network sizes. Since in the optimal as well as the heuristic all the sources are assured at least the max min rate, figure 9 implies that a majority of the sources in the network are allocated the max-min rate. There are only $\sim 15\%$ of the sources who have a rate different then the max-min rate whose ordering matters in the final solution. Further as can be seen from Figures 8 and 9 since the distribution of the heuristic and the optimal are quite similar it would appear the optimal values are quite insensitive to exact ordering of the sources.

Finally in order to quantify our investigation we require to know the identity of the sources that are actually belonging to the higher ranks. To answer this we plot the average number of hops versus the rank of the sources in figures 10(a), 10(b), 10(c) . Figure 10 suggests that higher rank nodes have a much



(a) Network Size = 10 (b) Network Size = 40 (c) Network Size = 70

Fig. 9. The distributions of the percentage of the total sources that belong to a particular rank for different network sizes. The skewed distributions highlight that most of the sources belong to the highest rank, while only a few belong to the lowest.



(a) Network Size = 10 (b) Network Size = 40 (c) Network Size = 70

Fig. 10. The dependance of rank on hop count. The largest ranks are much farther away from the root as compared to the hop count of the smaller ranks.

larger hop count as compared to sources belonging to other ranks. This makes intuitive sense as well since a larger hop count implies a larger amount of interference, which in turn implies allocating the minimum possible (max-min) rate to sources having the highest rank.

The above comparison of various metrics related to rank ordering, between the heuristic and the optimal solutions, justifies our claim that the accuracy of the solution obtained by the heuristic depends solely on the 'similarity' of the ordering obtained by the heuristic as compared to the optimal source rate ordering. Figures 9 and 10 show that, for the network sizes under consideration, since the number of sources that affect the solution are small and are primarily within a few hops from the sink, the heuristic is able to achieve an ordering that is similar to the optimal solution resulting in near optimal performance.

X. Related Work

Application of optimization theory to the design and analysis rate control algorithms was first introduced in the wireline context in the seminal paper by Kelly *et al.* [2]. This seminal work established that distributed additive increase-multiplicative decrease rate control protocols can be derived as solutions to an appropriately formulated optimization problem. The application of duality and the sub-gradient approach to solve the same problem was then introduced in the classic work by Low and Lapsley [4]. Since these two works there has been considerable research primarily in the wired context in understanding not only rate control algorithms but network protocols in general and their interaction across multiple layers from the perspective of the optimization problem they aim to solve. A detailed survey of this literature is presented in an article by Chiang *et al.* [5].

For wireless networks, the problem of cross layer optimization has been addressed in the works by Chiang *et al.* [6], Johansson *et al.* [7] and Wang *et al.* [8]. These works introduce the dual decomposition technique to address the problem of cross layer optimization in wireless networks and present algorithms for performing joint transport and power control [6] or joint transport and MAC layer design [8].

The problem of rate allocation in particular has been looked at from different perspectives in the domain of wireless networks. Liao *et al.* [18] look at the max-min fair rate allocation problem for packet based wireless access networks. They achieve their goal by assigning the flows at the access point a concave utility function and applying a max-min fair criteria on these flows. However [18] addresses the problem for fixed infrastructure based wireless access networks which is different from our scenario of a data gathering tree in wireless sensor network. Moreover their objective is to maximize the minimum utility of all flows and not to maximize the network utilization.

In [19] Calin *et al.* propose a routing scheme that can optimize resource allocation in a wireless ad-hoc network in order to maximize the aggregate utility of all flows in the network. They use the shadow price interpretation of the dual to present a bidding scheme that allows for a combined routing and rate control heuristic. This work assumes a concave utility function resulting in a notion of proportional fairness. We differ from [19] since in this work we explicitly try to maximize the network utilization while maintaining a weaker notion of the max-min fairness criterion.

Kun *et al.* propose EWCCP [20], a congestion control algorithm for wireless ad-hoc networks designed to provide proportional fairness to flows in the network. The similarity between EWCCP and the algorithm proposed in this work is that congestion signaling in EWCCP explicitly takes into account the interference set of a node while generating a congestion signal for flows traversing that specific node. This is similar to our ordering of the sources for rate allocation based on the amount of interference generated by each source. The difference between the two works is that EWCCP is designed to work within the context of an AIMD protocol, namely TCP, whereas we show in our work that additive increase algorithms are sub-optimal to the joint problem of maximization of network utilization and achieving the highest minimum rate possible. Moreover the notion of fairness achieved by EWCCP is proportional fairness as opposed to max-min fairness.

Wang *et al.* [15], [16] present algorithms for achieving max-min fairness and lexicographic max-min fairness [1] for Aloha random access networks. However the objective of [15] and [16] is to ensure fairness of link rates and not end-to-end flows. In [17] Tassuilas *et al.* present a centralized algorithm for achieving lexicographic max-min fairness in wireless ad hoc networks. Our objective in this work differs from that of [17], since the objective here is to maximize network utilization while maximizing the minimum rate. Moreover our aim is to present a distributed solution as compared to the centralized solution presented in [17].

The problem of max-min fair rate control has been looked at in the context of wireless sensor networks. In an earlier work [9], we presented an additive increase-based rate allocation scheme that guarantees a weaker notion of max-min fairness. In [9] we present a TDMA-based MAC which guarantees a max-min rate allocation by assigning slots to various sources. The number of slots correspond to a source rate that is calculated using an additive increase scheme. Rangwala *et al.* [11] also present an additive increase-multiplicative decrease solution for fair congestion control. The source rates in IFRC are allowed to increase using an additive increase algorithm similar to the one described in section VI. Both these works try to achieve a max-min fair rate allocation while trying to implicitly maximize network utilization. As shown in section VI these techniques are sub-optimal when the dual objective of maximization of network utilization and fairness are taken into consideration.

In the field of wireless sensor networks duality based approaches are not limited to designing and analyzing rate control algorithms. Our approach of analyzing the dual to achieve a distributed solution follows the approach presented by Ye and Ordonez [14], where a distributed dual based gradient search algorithm is proposed for the problem of maximizing data extraction under energy constraints.

XI. Conclusion and Future Work

We have formulated the problem of maximizing network utilization while guaranteeing the best possible minimum rate to sources in a wireless sensor network. We model the problem as two coupled linear programs. By analyzing the dual we are able to show that existing additive increase techniques are provably sub-optimal. Moreover our analysis of the dual results in a heuristic that presents near optimal performance.

There are several directions in which we could extend this work. One of our goals is the implementation of our dual based algorithm on a real sensor network test bed. The objective of such an implementation would be to do a performance comparison with existing rate control mechanisms, such as IFRC [11], that have additive increase algorithms at the core of their design. A real test bed implementation would also help validate the assumptions we have made while modeling the constraints in our problem. In our modeling we have made an implicit assumption that every receiver can hear every interferer that is consuming bandwidth at the receiver. In a real deployment this assumption might be weak since transmitters can cause interference in receivers that are not within range. A real test bed environment would help us ascertain the affects of such phenomenon on the results obtained from our algorithm.

In the current problem formulation we have focused on transport layer optimization alone. Interesting extensions to this work include joint transport/routing/MAC cross-layer design.

References

[1] D. Bertsekas and R. Gallagher, Data Networks, Prentice Hall, 1992

[2] F. P. Kelly, A. K. Maulloo D. K. H. Tan, Rate Control for Communication Networks:Shadow Prices, Proportional Fairness and Stability, Journal of the Operational Research Society, 1998.

[3] A. Tang, J. Wang and S. Low. Is Fair Allocation Always Inefficient. Proceedings of IEEE Infocom 2004

[4] S.H. Low and D. E. Lapsley, Optimization Flow Control, I: Basic Algorithm and Convergence, IEEE/ACM Transactions on Networking, 7(6):861-75, December 1999

[5] M. Chiang, S. H. Low, A. R. Calderbank, and J. C. Doyle, Layering as optimization decomposition: A mathematical theory of network architectures, Proceedings of IEEE, January 2007.

[6] M. Chiang, Balancing Transport and Physical Layers in Wireless Multihop Networks: Jointly Optimal Congestion Control and Power Control, IEEE Journal of Selected Areas in Communications, vol. 23, no. 1, pp. 104-116, January 2005

[7] Bjrn Johansson, Pablo Soldatti, Mikael Johansson, Mathematical Decomposition Techniques for Distributed Cross-Layer Optimization of Data Networks, IEEE JSAC, vol. 24, no. 8, August 2006.

[8] X. Wang and K. Kar, Cross-Layer Rate Optimization for Proportional Fairness in Multi-Hop Wireless Networks with Random Access, IEEE JSAC, vol. 24, no. 8, August 2006.

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

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

[11] Sumit Rangwala, Ramakrishna Gummadi, Ramesh Govindan and Konstantinos Psounis. Interference-Aware Fair Rate Control in Wireless Sensor Networks, In Proceedings of ACM SIGCOMM, Pisa, September 2006

[12] A. Woo and D.Culler. A transmission control scheme for media access in sensor networks. Mobicom, 2001.

[13] C.T. Ee, R. Bajcsy, Congestion control and fairness for many-to-one routing in sensor networks, in: ACM SenSys, 2004

[14] 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

[15] X. Wang and K. Kar, Distributed Algorithms for Max-min Fair Rate Allocation in Aloha Networks, In Proceedings of Annual Allerton Conference, Urbana-Champaign, October 2004.

[16] X. Wang, K. Kar and J.-S. Pang, Lexicographic Max-min Fairness in a Wireless Ad-hoc Network with Random Access,In Proceedings of IEEE Conference on Decision and Control (CDC),San Diego, December 2006

[17] L. Tassiulas, S. Sarkar, Maxmin Fair Scheduling in Wireless Networks, in Proceedings of Infocom 2002, pp. 763-772, New York, USA, June 2002

[18] Raymond R.-F. Liao, Andrew T. Campbell, A Utility-Based Approach for Quantitative Adaptation in Wireless Packet Networks, Wireless Networks, 2001.

[19] Calin Curescu, Simin Nadjm-Tehrani,Price/utility-based optimization of resource allocation in ad hoc networks, IEEE Secon, 2005.

[20] Kun Tan , Feng Jiang , Qian Zhang , Sherman Shen, Congestion Control in Multi-hop Wireless Networks, IEEE 2005.

[21] http://lpsolve.sourceforge.net/5.1/