ENFORCING END-TO-END PROPORTIONAL FAIRNESS WITH BOUNDED BUFFER OVERFLOW PROBABILITIES

Nikhil Singh, Ramavarapu S. Sreenivas and Uday V. Shanbhag
In this paper we present a distributed flow-based access scheme for slotted-time protocols, that provides proportional fairness in ad hoc wireless networks under constraints on the buffer overflow probabilities at each node. The proposed scheme requires local information exchange at the link-layer and end-to-end information exchange at the transport-layer, and is cast in the framework of nonlinear optimization. We say a medium access control protocol is proportionally fair with respect to individual end-to-end flows in a network, if the product of the end-to-end rates of flows is maximized. A key contribution of this work lies in the construction of a distributed dual approach that comes with low computational overhead. We discuss the convergence properties of the proposed scheme and present simulation results to support our conclusions.
Enforcing End-to-End Proportional Fairness with Bounded Buffer Overflow Probabilities

Nikhil Singh, Student Member, IEEE, Ramavarapu S. Sreenivas*, Senior Member, IEEE, and Uday V Shanbhag*

*Industrial and Enterprise Systems Engineering, University of Illinois at Urbana-Champaign, Urbana, IL 61820

Abstract

In this paper, we present a distributed flow-based access scheme for slotted-time protocols, that provides proportional fairness in ad-hoc wireless networks under constraints on the buffer overflow probabilities at each node. The proposed scheme requires local information exchange at the link-layer and end-to-end information exchange at the transport-layer, and is cast as a nonlinear program. A medium access control protocol is said to be proportionally fair with respect to individual end-to-end flows in a network, if the product of the end-to-end flow rates is maximized. A key contribution of this work lies in the construction of a distributed dual approach that comes with low computational overhead. We discuss the convergence properties of the proposed scheme and present simulation results to support our conclusions.

Index Terms

Wireless LAN, Access protocols, Resource management.

In this paper we consider an ad-hoc wireless network [1] that carries several flows between various source-destination pairs under a slotted-time medium access control (MAC) protocol. Specifically, we are interested in a distributed scheme for the assignment of the network’s resources among flows, that is fair in terms of end-to-end flow rates. We assume that each node in the network has a finite buffer assigned to each flow routed through it. In addition to the objective of fairness, we are also interested in ensuring that the buffer overflow probabilities at each node does not exceed a pre-determined value.
The literature contains several references to fairness and its impact on the network performance. It has been observed by many researchers that the contention control mechanism used in 802.11-MAC [2] can be inefficient [3]. In [4], [5] a list of modifications is presented, that eliminates the unfairness commonly seen in the 802.11-MAC. The literature also contains a large volume of references (cf. [6], [7], [8], for example) where it is assumed that each network flow/link is associated with a concave utility function that could be maximized. In particular, for proportional fairness, it is assumed that the utility function has the form of \( \log x \), where \( x \) denotes the flow rate [6]. It is of interest to schedule individual transmissions on the links so as to maximize the sum of the utilities of the consumers. To achieve fairness, the schemes outlined in the above mentioned references use a penalty function which is updated by some form of feedback from the network. Using an appropriately defined cost that is implicitly dependent on the requested rates of each 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, given by the difference between the total utility and the penalty. An overview of network resource allocation through utility maximization is presented in [9].

In [10], the authors have addressed the problem of providing proportional fairness by considering joint optimization at both transport and link layers. Two algorithms are proposed for solving the problem in a distributed manner that converge to the globally optimal solutions. These results, generalized in [11], are based on the dual and the primal algorithms in convex optimization and need end-to-end feedback information to update variables maintained at the nodes. The algorithms presented in [10], [11] are oblivious of the queue dynamics of the network, which may increase delays and packet loss. Although our work is closely related to [10], [11], the problem formulation and the proposed solution differ significantly.

In [12], the solution approach uses a class of *queue back-pressure random access algorithms* (QBRA), where the actual queue-lengths of the flows are used to determine any node’s channel access probabilities. In this distributed algorithm, a node uses the queue-length information in a close neighborhood to determine its channel access probability to achieve proportionally fair rates and queue stability. This scheme has the advantage that no optimization needs to be performed and node’s can achieve proportional fairness just by exchanging the queue information in the local neighborhood. However, the frequency of exchange of this information plays a vital role in determining the performance of this algorithm. In optimization-based schemes, once the flow
rates have converged to the optimum, the frequency of information exchange does not play a significant role until the network topology, or the number of flows in the network, change.

In a different approach, several policies have recently been proposed for achieving rates close to the maximum throughput region through dynamic link scheduling [13], [14], [15], [16]. These scheduling algorithms use maximal matchings in every time slot using local contention algorithms and achieve near maximal schedules. Some policies also guarantee fairness of rate allocation among different sessions.

QoS (Quality of Service) is an important issue in ad-hoc wireless networks. Service guarantees can be provided for delays, packet loss, jitter and throughput based on the application requirements. Our approach in this work is to combine the QoS guarantee in addition to providing proportional fairness. Our main contributions are as follows:

1) We derive an expression for the buffer overflow probabilities for discrete-time queues. This derivation uses the fact that there cannot be simultaneous arrivals and departures at a node within the same slot in Aloha-type networks that do not have packet capture mechanisms.
2) Using the expression for buffer overflow probabilities mentioned above, we show that an upper bound on the buffer overflow probability translates to an upper bound on the utilization or load, which can then be used as constraints in an appropriately posed convex minimization problem under convex constraints. This is a reformulation of the proportionally fair end-to-end rate allocation problem. A distributed dual approach is then used to solve this convex minimization problem using an appropriate Lagrangian function. The dual problem is solved using a projected gradient method.
3) Finally, after making some observations about the distributed implementation of the above mentioned dual scheme, we present simulation results showing the satisfactory performance of our proposed algorithm in terms of fairness and QoS.

The rest of the paper is organized as follows. Section I presents the network model that is used in the rest of the paper. We then formulate the rate control problem as a convex optimization instance with bounds on the buffer overflow probabilities at each node. In section II, we discuss the dual-based solution approach and present a distributed implementation to achieve flow-based proportional fairness. The convergence of this algorithm to the unique global optimum is established. Section III contains the details of the experimental results verifying the optimality of the proposed scheme. Conclusions are provided in section IV.
I. Problem Formulation

A. Wireless Network Model

We assume the following:

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) Nodes cannot transmit and receive packets at the same time.
5) The receipt of more than one packet within the same time-slot will result in a collision.
6) Nodes in the network have a buffer of fixed size assigned to each flow routed through it.
7) We also assume there is a unique route for each flow within the network (which would be the case if we used DSDV [17] as the routing protocol, for example).

Additionally, we only consider unicast flows for our derivations.

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 (i.e., $(i, j) \in E \iff (j, i) \in E$), that represents the set of bi-directional links. We assume all links of the network have the same capacity, which is normalized to unity. The 1-hop neighborhood of node $i \in V$ is represented by the symbol $N(i)$. When a node $i$ communicates with a node $j \in N(i)$, 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 flows, using a link $(i, j) \in E$ with $i$ (j) as origin (terminus), is denoted by $F(i, j)$.

When node $i$ intends to transmit data to node $j \in N(i)$ for the $l$-th flow $(l \in F(i, j))$, it would transmit data in the appropriate time-slot with probability $p_{i,j,l}$. $P_{i,j} = \sum_{l \in F(i,j)} p_{i,j,l}$, denotes the probability that node $i$ transmits data to node $j$, and $P_i = \sum_{j \in V} P_{i,j}$, 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 $P_i$ is not greater than unity for any node $i \in V$. 

B. Link Success Probability Expression

The probability of successful data transmission over link \((i, j) \in E\) for flow \(l \in \mathcal{F}(i, j)\), denoted by \(S_{i,j,l}\), is given by the expression

\[
S_{i,j,l} = p_{i,j,l} \left( 1 - \sum_{(j,m) \in E, n \in \mathcal{F}(j,m)} p_{j,m,n} \prod_{o \in \mathcal{N}(j) - \{i\}} (1 - \sum_{(o,p) \in E, q \in \mathcal{F}(o,p)} p_{o,p,q}) \right).
\]

This is also the rate or the attainable throughput of flow \(l\) over link \((i, j)\).

C. Problem Statement

Consider an ad-hoc wireless network where there are \(r\) flows in the network. Each flow has a utility function associated with it, whose value is determined by the logarithm of the flow rate. The objective is to maximize the sum of the logarithms of the flow-rates under the operational constraints outlined below. We denote the logarithm of the rate of the \(l^{th}\) flow as \(f_l\). The end-to-end proportionally fair flow control problem can be stated as

\[
\max_{p_{i,j,l}} \sum_l f_l,
\]

where \((i, j) \in E\) and \(l \in \{1, 2, \ldots, r\}\), subject to additional constraints.

Let us assume that the \(l^{th}\) flow \((1 \leq l \leq r)\) spans over \(k_l\) links. We use the notation \(\langle l, q \rangle \in E\) to denote the \(l^{th}\)-flow’s \(q^{th}\)-link, where \(q \in \{1, 2, \ldots, k_l\}\) is indexed in ascending order starting from the source and terminating at the destination. Thus, \(\langle l, q \rangle = (i, j)\) implies the \(l^{th}\)-flow’s \(q^{th}\)-link from the source has \(i\) as the source node and \(j\) as the destination node. If \(\langle l, q \rangle = (i, j) \in E\) then we use the notation \(S_{\langle l, q \rangle, l}\) to denote \(S_{i,j,l}\). The logarithm of the rate of \(\langle l, q \rangle\) is represented as \(f_{l,q}\).

Let \(\mathbf{p} = (p_{i,j,l}, 1 \leq l \leq r, (i, j) \in E)\) be the vector of access probabilities of all the flows over each link in the network and \(\hat{f} = (f_{l,q}, 1 \leq l \leq r, 1 \leq q \leq k_l)\) the vector of the logarithm of link rates of all flows.

In the case of multi-hop wireless networks, the rate of any flow is the same as the rate of the bottleneck link in that flow. The logarithm of the rate of the \(l^{th}\)-flow is \(\min\{f_{l,q}: 1 \leq q \leq k_l\}\). Hence, the problem can be stated as \(\max_{p_{i,j,l}} \sum_l \min\{f_{l,q}, 1 \leq q \leq k_l\}\), subject to capacity constraints, and additional constraints on the buffer overflow probabilities which is addressed in the next subsection.
D. Buffer Overflow Probability of a Tandem of Discrete-Time Queues

As an illustration, consider a tandem of two discrete-time queues as shown in figure 1, where packet-arrivals and packet-departures occur at discrete-time instants (d-times). We assume

1) there cannot be a packet arrival and departure at the same d-time at either queue.
2) there is at most one packet-arrival or packet-departure from either queue at any d-time.
3) packets arrive into the first queue with a probability of \( p_a \) at any d-time.
4) if the first (second) queue is non-empty at a given d-time, the probability of a departure of a packet from the first (second) queue at that d-time is \( p_{d1} \) (\( p_{d2} \)).

Assumptions 1 and 2 listed above follow directly from the fact that a node cannot transmit and receive packets at the same time. These assumptions ensure that the state-transitions in the discrete-time Markov chain that describes the example at hand, involves at most an unit-change in the number of packets in any queue. In subsequent text, we assume \( p_a < p_{d1} \), and \( p_a < p_{d2} \). These assumptions are required for specific infinite summations that are implicit in subsequent discussion, to take finite values.

![Fig. 1. A tandem of two discrete-time queues.](image)

The discrete-time Markov chain for the tandem of two queues is shown in figure 2(a). The state \((i, j)\) denotes the presence of \(i\) packets in the first (second) queue. We denote the probability of the system to be in state \((i, j)\) with \(p(i, j)\). A state-transition \((i, j) \xrightarrow{p_a} (i + 1, j)\) denotes an packet-arrival into the first queue, with a probability of \(p_a\). Similarly, state-transition \((i, j) \xrightarrow{p_{d1}} (i - 1, j + 1)\) denotes a packet-departure from the first queue (and an packet-arrival into the second queue), with probability \(p_{d1}\). Finally, state transition \((i, j) \xrightarrow{p_{d2}} (i, j - 1)\) denotes a packet-departure from the second queue, with probability \(p_{d2}\).

Using the discrete-time Markov chain that describes the dynamics of the first queue (which is independent of the second queue), it can be shown that the probability of seeing \(i\) packets at any discrete-time instant in the first queue is given by the expression

\[
\left( \frac{p_a}{p_{d1}} \right)^i \left( 1 - \frac{p_a}{p_{d1}} \right).
\]
(a) The discrete-time Markov Chain for the tandem of two queues shown in figure 1. The state \((i, j)\) denotes the fact that there are \(i\) customers in the first queue and \(j\) customers in the second.

(b) The general structure of the incoming and outgoing arcs for state \((i, j)\).

Fig. 2. Discrete-time Markov Chain

This will serve as the flow-balance requirement of the discrete-time Markov chain shown in figure 2(b), given by:

\[
p(i + 1, j - 1) p_{d1} + p(i - 1, j) p_a + p(i, j + 1) p_{d2} = p(i, j) \left( p_{d1} + p_{d2} + p_a \right),
\]

where \(p(i, j)\) is the probability of seeing \(i\) (\(j\)) packets in the first (second) queue at any time instant. It is relatively straightforward to show that as long as \(p_a < \min\{p_{d1}, p_{d2}\}\), the solution to (4) subject to the constraint in (3) has a “product-form” given by

\[
p(i, j) = \left( 1 - \frac{p_a}{p_{d1}} \right) \left( \frac{p_a}{p_{d1}} \right)^i \left( 1 - \frac{p_a}{p_{d2}} \right) \left( \frac{p_a}{p_{d2}} \right)^j.
\]

The probability of seeing \(j\) packets in the second queue at any discrete-time instant is therefore given by, \(\sum_{i=0}^{\infty} p(i, j) = \left( 1 - \frac{p_a}{p_{d2}} \right) \left( \frac{p_a}{p_{d2}} \right)^j\). Assuming the first queue has an unlimited buffer-size, the probability of a packet-arrival into the second queue is given by \(p_{d1} \left( \frac{p_a}{p_{d1}} \right) = p_a\). That is, as long as \(p_a < \min\{p_{d1}, p_{d2}\}\) and the first buffer has an unlimited-size, the second queue can be treated independently of the first, where the probability of arrival to second queue is \(p_a\) and the probability of a departure from a non-empty queue is \(p_{d2}\). The following theorem presents a generalization of this observation.

**Theorem 1.1:** Consider a tandem of \(n\) discrete-time queues with unlimited buffer-size, where at any discrete-time instant, the probability of a packet-arrival into the first queue is \(p_a\), and
the probability of a packet-departure from the $i$-th, non-empty queue is $p_{di}$, ($i = 1, 2, \ldots, n$). If $p_a < \min\{p_{di}, i = 1, 2, \ldots, n\}$, the probability of seeing $j$-many packets at any discrete-time instant in the $i$-th queue is given by $\left(1 - \frac{p_a}{p_{di}}\right) \left(\frac{p_a}{p_{di}}\right)^j$.

**Proof:** This can be established by an induction argument over $n$. The base-case is established for $n = 1$ along with the observation that the probability of a departure from the first queue at any discrete-time instant, under the assumed conditions, is $p_a$. As the induction hypothesis, we suppose the claim made in the theorem is true for a $n$-many discrete-time queues connected in tandem, together with the observation that the probability of a packet-departure from the last discrete-time queue is indeed $p_a$. The induction step is established by adding a discrete-time queue to the last queue in a tandem to form a tandem of $(n+1)$ queues. By the induction hypothesis, the probability of a packet-arrival into the $(n+1)$-th queue at any discrete-time instant is $p_a$, and the probability of a packet-departure, when the $(n+1)$-th queue is non-empty is $p_{d(n+1)}$. The analysis of the discrete-time Markov chain for this queue will yield the probability of seeing $j$-many packets at any time instant in the $n+1$-th queue to be

$$\left(1 - \frac{p_a}{p_{d(n+1)}}\right) \left(\frac{p_a}{p_{d(n+1)}}\right)^j.$$ 

The probability of a packet-departure from the $(n+1)$-th queue is given by $p_{d(n+1)} \times \frac{p_a}{p_{d(n+1)}} = p_a$, which completes the induction step. 

We now turn our attention to the case when the buffer-size of these queues are limited to $M$. The probability of seeing $i$ packets in the first queue at any discrete-time instant is given by

$$\left(1 - \frac{p_a}{p_{d1}}\right) \left(\frac{p_a}{p_{d1}}\right)^i \left(1 - \left(\frac{p_a}{p_{d1}}\right)^{M+1}\right).$$

The buffer overflow probability is obtained by setting $i = M$ in the above expression. Let $\beta$ be an acceptable upper-bound on the buffer overflow probability. It can be shown that if $\frac{p_a}{p_{d1}} < \left[\frac{\beta}{1+\beta}\right]^{1/M}$, the buffer overflow probability at the first queue is indeed less than $\beta$.

Unlike the unlimited buffer-size situation discussed earlier, the presence of finite buffer-sizes in the queue introduces a dependence between queues. For instance, when the first queue has an unlimited buffer-size the probability of a packet-departure from this queue (which is also the probability of packet-arrival to the second queue) can be shown to be $p_a$. However, when the buffer-size is limited to $M$, we have the probability of a packet-departure from the first queue
is given by
\[
p_{d1} \left(1 - \frac{1 - \frac{p_a}{p_{d1}}}{1 - \left(\frac{p_a}{p_{d1}}\right)^{M+1}}\right) = p_a \left(1 - \left(\frac{p_a}{p_{d1}}\right)^{M+1}\right) \leq p_a,
\]
under the assumptions made earlier. That is, the packet-arrival probability into the second queue is smaller than the packet-arrival probability to the first queue \(p_a\). The following theorem takes the observations made above and presents a condition under which the buffer overflow probabilities at any queue is no larger than a prescribed upper bound. This is used in the convex optimization framework outlined in the next section.

**Theorem 1.2:** Consider the tandem of discrete-time queues introduced in the statement of theorem 1.1, where each queue has a buffer-size of \(M\). Suppose \(p_{dj} = \min_{i=1,...,n} \{p_{di}\}\), and \(\frac{p_a}{p_{dj}} < \frac{M}{M+1}\). Then, the probability of seeing \(M\) packets in the \(i\)-th queue \((i = 1 \ldots n)\) is no greater than
\[
\left(\frac{1 - \frac{p_a}{p_{dj}}}{1 - \left(\frac{p_a}{p_{dj}}\right)^{M+1}}\right) \left(\frac{p_a}{p_{dj}}\right)^M = \alpha(\leq 1).
\]

**Proof:** Suppose \(\rho = \frac{p_a}{p_{dj}}\), we first note that the expression \(\left(\frac{1 - \rho}{1 - \rho^{M+1}}\right)\rho^M\), increases monotonically with respect to \(\rho\) if \(\rho \leq \frac{M}{M+1}\). Let \(p_{ai}\) be the probability of a packet arrival into the \(i\)-th queue, we know \(p_{ai} \leq p_a\). If \(\rho_i = \frac{p_{ai}}{p_{di}}\), since \(p_{di} \geq p_{dj}\), it follows that \(\rho_i \leq \rho < \frac{M}{M+1}\). The observation follows directly from the monotonicity property mentioned above.

A direct consequence of theorem 1.2 is that if we are able to pick a \(p_a\) such that
\[
\frac{p_a}{p_{dj}} < \left[\frac{\beta}{1 + \beta}\right]^{1/M}, \tag{5}
\]
then the buffer overflow probability at the \(i\)-th queue in the tandem of discrete-time queues will be no higher than \(\beta\) at all queues. In the next section, this observation is used in a convex programming solution to the problem of enforcing proportional fairness in the presence of constraints on the buffer overflow probabilities.

**E. Problem Formulation with Buffer Overflow and Capacity Constraints**

Let us assume the loss rate bounds for the \(l\)-th-flow translates to each node along the flow sustaining a traffic intensity (ratio of arrival probability and departure probability at a node) no more than \(\rho_l(= \frac{p_a}{p_{dj}})\).
Also, each link-rate in the network cannot exceed the capacity of that link given by (1). Since the logarithmic function is strictly increasing, each link constraint can be re-written as

\[ f_{l,q} \leq \log(S_{\langle l,q \rangle,l}) \]  

(6)

Each constraint in (6) forms a convex set of \((f_{l,q}, p)\). We also assume that there is a minimum achievable data-rate for each flow, i.e., \( \exists \epsilon \), s.t. \( \epsilon \leq f_{l,q} \), \( \forall l, q \) \((1 \leq l \leq r, 1 \leq q \leq k_l)\). We define the feasible set of access probabilities as,

\[ \tilde{P} = \{ p : \sum_{j \in N(i), l \in \mathcal{F}(i,j)} p_{i,j,l} \leq 1, \ 0 \leq p_{i,j,l} \leq 1, \ (i, j) \in \mathcal{E}, \ l \in \mathcal{F}(i, j) \} \].

Also, we define the QoS region as a set of vectors as defined by

\[ \mathcal{G} = \{ \hat{f} : \epsilon \leq f_{l,1}, \ f_{l,1} = f_{l,q} + \delta_l, 2 \leq q \leq k_l \} \],

where \( \delta_l = \log \rho_l \). The overall optimization problem can now be stated as:

\[ \mathbf{V} : \max_{p_{i,j,l}} \sum_l \min_{f_{l,q}} f_{l,q} \begin{array}{c} \leq \log(S_{\langle l,q \rangle,l}) \\ \langle l, q \rangle \in \mathcal{E}, f_l \in \tilde{P} \end{array} \]  

(7)

Without loss of generality, we assume that all the flows in the network span at least two links.

From the constraint imposed by the QoS region, we observe that for any feasible solution to \( \mathbf{V} \), the first link will always have the lowest rate and hence it will be the bottleneck. Therefore for any feasible solution, the rate of any flow \( l \), is same as \( f_{l,1} \). If we replace \( f_{l,1} \) by \( f_l \), and define the feasible set of flow rates as \( \tilde{F} = \{ f : \epsilon \leq f_l \leq \delta_l, \forall l \} \), where, \( f = (f_l, 1 \leq l \leq r) \), we can rewrite \( \mathbf{V} \) as the following convex optimization problem,

\[ \mathbf{U} : \min_{p_{i,j,l}} \sum_l -f_l \]  

(8)

\[ \begin{array}{c} f_l \leq \log(S_{\langle l,1 \rangle,l}), \ \langle l, 1 \rangle \in \mathcal{E}, \\ f_l \leq \log(S_{\langle l,q \rangle,l}) + \delta_l, \ \langle l, q \rangle \in \mathcal{E}, 2 \leq q \leq k_l, \\ p \in \tilde{P}, \ f \in \tilde{F}. \end{array} \]
II. Solution Approach

A. Dual-based Algorithm

We can write the Lagrangian function for the problem stated in (8) as,

\[ L(f, p, \lambda) = \sum_t -f_t + \sum_{l,1} \lambda_{l,1}(f_l - \log(S_{l,1}, l)) + \sum_{l,q(2 \leq q \leq k_l)} \lambda_{l,q}(f_l - \log(S_{l,q}, l) - \delta_l). \]  

(9)

Let us denote \( \Lambda = (\lambda_{l,q} : \forall l, \ 1 \leq q \leq k_l) \) as a vector of Lagrange multipliers. As the Slater constraint qualification is satisfied by the convex program given by (8), convex duality implies that at the optimum \( \Lambda^* \), the corresponding \( f, p \) are the solutions to the primal problem [18]. The dual problem can be solved using the gradient projection method similar to the scheme used in [19]. Note that the Lagrangian is separable in terms of the probabilities \( p \) and the logarithm of the rates \( f \). The dual function can be stated as:

\[ Q(\Lambda) = \inf_{f \in \tilde{F}, p \in \tilde{P}} L(f, p, \Lambda) \]

The following proposition is significant for obtaining the distributed solution for the non-linear program given by (8).

**Proposition 2.1:** For a given \( \Lambda \), solution to \( \inf_{f \in \tilde{F}, p \in \tilde{P}} L(f, p, \Lambda) \) is given by:

\[ p_{i,j,l} = \frac{\lambda_{l,q,(l,q)=(i,j)}}{\sum_{(y,z)=(i,k), k \in N(i)} \lambda_{y,z} + \sum_{(y,z)=(k,i), k \in N(i)} \lambda_{y,z} + \sum_{v \in N(i) \setminus \{i\}, \delta_l} \lambda_{y,z}} \]  

(10)

and,

\[ f_l = \begin{cases} \epsilon & \text{if } \sum_q \lambda_{l,q} > 1 \\ \delta_l & \text{if } \sum_q \lambda_{l,q} < 1 \\ \min\{\log(S_{l,1}, l), \min_{2 \leq q \leq k_l} (\log(S_{l,q}, l) - \delta_l)\} & \text{if } \sum_q \lambda_{l,q} = 1 \end{cases} \]  

(11)

**Proof:** For a given \( \Lambda \), the Lagrangian is convex with respect to \( p \), and taking the derivative with respect to \( p \) gives the infimum of the Lagrangian when \( p_{i,j,l} \) equals the expression in (10).

The denominator of (10) is essentially the sum of three terms. The first term is the sum of the Lagrange multipliers associated with all outgoing flows from node \( i \). The second term is the sum of the Lagrange multipliers associated with all incoming flows to node \( i \). Finally, the third term is the sum of the Lagrange multipliers associated with all incoming flows to nodes in the
one-hop neighborhood of node $i$ (excluding the flows incoming from node $i$). It is not hard to show that this expression satisfies the constraint $0 \leq p_{i,j,l} \leq 1$ and $0 \leq P_i \leq 1$.

Once we have the $p$ that minimizes the Lagrangian, $S_{(l,q),l}$ is parameterized (assumed to be a given constant). We then solve the following convex optimization problem to get $f$, which minimizes the Lagrangian for the given $\Lambda$, as specified by

$$
\min_{f} \left( \sum_{l} -f_l + \sum_{l,1} \lambda_{l,1}(f_l - \log(S_{(l,1),l})) + \sum_{l,q(2\leq q\leq k_l)} \lambda_{l,q}(f_l - \log(S_{(l,q),l}) - \delta_l) \right). \tag{12}
$$

Since for any $l$, the coefficient of $f_i$ in (12), is given by $(-1 + \sum_q \lambda_{l,q})$, the value of $f_i$ is either the lower or the upper bound of $f_i$, depending on the sign of its coefficient, which gives (11). When $\sum_q \lambda_{l,q} = 1$, the value of $f_i$ is set to satisfy the complimentary slackness condition.

The dual problem

maximize $Q(\Lambda)$

subject to $\Lambda \geq 0$

can now be solved using the gradient projection method, where the Lagrange multipliers are adjusted in the direction of the gradient $\nabla Q(\Lambda)$:

$$
\lambda_{l,q}^{n+1} = \left[ \lambda_{l,q}^n + \alpha^n \frac{\partial Q(\Lambda^n)}{\partial \lambda_{l,q}} \right]^+ \tag{13}
$$

where $[z]^+ = \max\{0, z\}$. The variable $\alpha^n$ is the step size at the $n^{th}$ iteration which can either be a constant, or, a step size that satisfies the requirements

$$
\sum_{n=1}^{\infty} \alpha^n = \infty, \sum_{n=1}^{\infty} (\alpha^n)^2 < \infty,
$$

and the slope is given by,

$$
\frac{\partial Q(\Lambda^n)}{\partial \lambda_{l,1}} = (f_{i1}^n - \log(S_{(l,1),l}^n)), \tag{14}
$$

$$
\frac{\partial Q(\Lambda^n)}{\partial \lambda_{l,q}} = (f_{iq}^n - \log(S_{(l,q),l}^n) - \delta_l), \quad 2 \leq q \leq k_l. \tag{15}
$$

The choice of the step-size in (13) presents two variants of a dual-based algorithm that are further discussed in the following section.
B. Convergence of the Proposed Scheme

The minimization of the Lagrangian function in our case is separable in terms of the probabilities $p$ and the logarithm of the rates $f$. The minimization with respect to $f$ is done following the minimization with respect to $p$. The resulting solution after these two steps yields the dual function $Q(\Lambda)$. Reference [20], presents a proof of convergence of approaches that use a projected sub gradient method involving step sizes that are not summable, but are square summable. Since the only sub-gradient of a differentiable function is its gradient, this proof is equally applicable to an approach that uses the projected gradient method. The proof of convergence of projected gradient method with constant step size, under assumption of Lipschitz continuity, to a neighborhood around the optimum, is presented in the appendix.

From the first-order KKT conditions, we note that the optimal value of the Lagrange multipliers, $\lambda_{l,q}^*$, that solve the dual-problem satisfy the requirement $\sum_q \lambda_{l,q}^* = 1$. Following the convergence of (13), it follows that $\sum_q \lambda_{l,q}^n = 1$.

C. Implementation of the Dual-Based Algorithm

The dual-based algorithm for end-to-end proportionally fair rate allocation under buffer over-flow constraints in random access wireless networks can be summarized as follows:

1) Initialize the iteration count $n$ to zero. If $\langle l, q \rangle = (i, j)$ for some flow $l$, node $i$ chooses an initial value of $\lambda_{l,q}^0$ such that $0 < \lambda_{l,q}^0 < 1$.

2) Node $i$ passes the value of $\lambda_{l,q}^n$ to the source of the $l$th-flow. The logarithm of the rates $(f_l)$ are then computed by sources using (11) in $O(1)$ time.

3) Every node that the $l$th-flow is routed through, obtains the value of $f_l^n$ from the source.

4) After obtaining the $\lambda_{l,q}^n$-values from nodes within a 2-hop neighborhood, each node com-

putes the access probability values $(p_{i,j,l})$ according to (10).

5) Each node increments the value of $n$ and computes $\lambda_{l,q}^{n+1}$ by the gradient projection algorithm given by (13) in $O(1)$ time.

6) Steps 2, 3, 4 and 5 are repeated till an appropriate stopping condition (discussed below) is satisfied.

In case of networks with fixed flows (i.e. no arrival and departure of flows in the network), step sizes that are not summable, but are square summable can be used and the stopping criteria is when $\sum_q \lambda_{l,q} = 1$, $\forall l$. 

When flows can arrive and depart in the network, constant step size is the preferred option and in this case there is no stopping criteria, i.e. the nodes continue to run the optimization algorithm without termination. The access probabilities are updated periodically while maintaining the flow rates in the QoS region. If these probabilities yield flow rates that are not in the QoS region during the any phase of optimization, nodes along a flow can reduce their access probabilities in such a way that the resulting flow rates are in the QoS region. For instance, in the case of the reservation-based, slotted-time, random-access protocol STMAC [21], nodes along the flows starting from the destination to the source can decrease their transmission rates (by reserving the slots but not transmitting the packets so that the overall $P_i$ for any node $i$ is unchanged) to bring the flow rates in the QoS region.

### III. Performance Evaluation

![An ad-hoc wireless network.](image)

For our simulation comparisons, we consider the example shown in figure 3, from references [10], [12]. The nodes are labeled from 1 to 6. The interference model is that each node interferes with the reception at its one hop neighbors. For example nodes 1 and 3 cause interference at node 2; nodes 6, 5, 2 and 4 cause interference at node 3. Three end-to-end flows, namely, $flow_1$, $flow_2$, $flow_3$ are setup in this network. The source, the sinks, and the path of three flows are shown in table I.

We suppose each flow can tolerate a loss of 45 in every 100,000 packets. Additionally, we suppose each node has a buffer that can store 50 packets for each flow that is routed through it. This translates to a value of $\rho = 0.86$. For $\rho_1 = \rho_2 = \rho_3 = 0.86$, the globally optimal solutions to the problem defined in (8) was computed using the fmincon function in MATLAB. This solution and the solutions given by the dual-based algorithm are presented in table II.

We have used two approaches to the dual-based algorithm outlined earlier. In the first approach we use a constant step size $\alpha^n = 5 \times 10^{-4}$ (cf. (13)), and the logarithm of the minimal achievable
Flow Links (Source, Sink on the path) Observed flow rates in Matlab

<table>
<thead>
<tr>
<th>Flow</th>
<th>Links</th>
<th>Observed flow rates</th>
</tr>
</thead>
<tbody>
<tr>
<td>flow1</td>
<td>(1, 1) = (6, 5), (1, 2) = (5, 3), (1, 3) = (3, 2), (1, 4) = (2, 1)</td>
<td>0.0462</td>
</tr>
<tr>
<td>fflow2</td>
<td>(2, 1) = (6, 3), (2, 2) = (3, 4)</td>
<td>0.1123</td>
</tr>
<tr>
<td>fflow3</td>
<td>(3, 1) = (1, 2), (3, 2) = (2, 3), (3, 3) = (3, 4)</td>
<td>0.0799</td>
</tr>
</tbody>
</table>

**TABLE I**
Path of the flows and observed flow rates in a MATLAB simulation of the network shown in figure 3.

<table>
<thead>
<tr>
<th>Variables</th>
<th>( p_{6,5,1} )</th>
<th>( p_{5,3,1} )</th>
<th>( p_{3,2,1} )</th>
<th>( p_{2,1,1} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>optimum solutions</td>
<td>0.0881</td>
<td>0.2185</td>
<td>0.1028</td>
<td>0.0657</td>
</tr>
<tr>
<td>dual-based solutions with constant step sizes</td>
<td>0.0892</td>
<td>0.2165</td>
<td>0.1078</td>
<td>0.0675</td>
</tr>
<tr>
<td>dual-based solutions with diminishing step sizes</td>
<td>0.0882</td>
<td>0.2191</td>
<td>0.1032</td>
<td>0.0661</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Variables</th>
<th>( p_{6,3,2} )</th>
<th>( p_{3,4,2} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>optimum solutions</td>
<td>0.3388</td>
<td>0.1329</td>
</tr>
<tr>
<td>dual-based solutions with constant step sizes</td>
<td>0.3353</td>
<td>0.1419</td>
</tr>
<tr>
<td>dual-based solutions with diminishing step sizes</td>
<td>0.3377</td>
<td>0.1333</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Variables</th>
<th>( p_{1,2,3} )</th>
<th>( p_{2,3,3} )</th>
<th>( p_{3,4,3} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>optimum solutions</td>
<td>0.1776</td>
<td>0.2949</td>
<td>0.0892</td>
</tr>
<tr>
<td>dual-based solutions with constant step sizes</td>
<td>0.1875</td>
<td>0.2929</td>
<td>0.0903</td>
</tr>
<tr>
<td>dual-based solutions with diminishing step sizes</td>
<td>0.1761</td>
<td>0.2949</td>
<td>0.0893</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Variables</th>
<th>( f_{1} )</th>
<th>( f_{2} )</th>
<th>( f_{3} )</th>
<th>( U^{*} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>optimum solutions</td>
<td>0.0465</td>
<td>0.1143</td>
<td>0.0767</td>
<td>-7.8051</td>
</tr>
<tr>
<td>dual-based solutions with constant step sizes</td>
<td>0.0461</td>
<td>0.1109</td>
<td>0.0792</td>
<td>-7.8118</td>
</tr>
<tr>
<td>dual-based solutions with diminishing step sizes</td>
<td>0.0464</td>
<td>0.1136</td>
<td>0.0759</td>
<td>-7.8239</td>
</tr>
</tbody>
</table>

**TABLE II**
The optimal results and the solution given by the distributed algorithm

rate, \( \epsilon \) was set to be -10. Figure 4 shows how the aggregate utility, access probabilities and flow rates converge when the dual-based algorithm with fixed step size is used.

The second approach involves the use of a square summable, but not summable step sizes. In this case the step size at the \( n^{th} \) iteration \( \alpha^{n} = \frac{1}{n} \). The value of \( \epsilon \) is set to be -10. Figure 5 illustrates the convergence of the aggregate utility, access probabilities and flow rates when the dual-based algorithm with diminishing step sizes is used.

Note that in figure 4(a) there is a thickening of the aggregate utility function, indicating that
the computed values do not exactly converge to the optimal value but instead they fluctuate around it. This happens as the step-size is a constant and hence our algorithm achieves solution that is close to optimal.

If $\rho_1 = \rho_2 = \rho_3 = 1$, we get the optimal solution of $U^* = -7.4897$ using MATLAB’s fmincon function. This is higher than when $\rho_1 = \rho_2 = \rho_3 = 0.86$, but the buffer overflow will be significantly higher. To demonstrate this, we simulated the network in figure 3 in MATLAB, using access probabilities obtained for $\rho_1 = \rho_2 = \rho_3 = 1$ and $\rho_1 = \rho_2 = \rho_3 = 0.86$, and ran the simulation for a duration of $5 \times 10^4$ time slots, where each node in the network has infinite length buffers (i.e. no packets are dropped in the simulation). For a random instance of the simulation, we plot the queue-lengths as a function of time, for every flow at each node. In the plots, unit of time is a single time-slot of fixed duration. The flow rates observed for $\rho_1 = \rho_2 = \rho_3 = 0.86$
Fig. 6. Case I: $\rho_1 = \rho_2 = \rho_3 = 1$

Fig. 7. Case II: $\rho_1 = \rho_2 = \rho_3 = 0.86$

are presented in table I.

Figures 6 and 7 demonstrate the queueing performance of our algorithm. Case I is the plot of queue-lengths as a function of time when we use the optimal access probabilities without considering buffer overflow constraints i.e. $\rho_1 = \rho_2 = \rho_3 = 1$. Case II shows how the queue-length varies as a function of time, when the optimal access probabilities obtained by setting $\rho_1 = \rho_2 = \rho_3 = 0.86$ is used. We can observer from the plots, that if a buffer size of 50 was used, then the fraction of packets transmitted, that are lost, in case I will be much higher than as compared to case II.

IV. CONCLUSION

In this paper, we proposed a distributed scheme for providing end-to-end proportionally fair flow rates in a slotted-time, multi-hop, random access network with a general network topology, with bounds on the buffer overflow probabilities at each node. After noting that each flow in
the network can be viewed as a tandem of discrete-time queues, we converted the constraints on buffer overflow probabilities into appropriate constraints on the link rates, which permitted the reformulation of the original problem into an appropriately posed convex minimization problem under convex constraints. We solved this problem using an appropriately constructed Lagrange function, and showed that the dual-problem converges under square-summable, but not summable, step-sizes. The optimal values of the Lagrange multipliers of the dual-problem were then used to arrive at the optimal values of the attempt probabilities. After presenting aspects of distributed implementation of this dual-based approach, we verified the correctness of the approach using an example from the literature.

APPENDIX

PROOF OF CONVERGENCE OF PROJECTED GRADIENT METHODS WITH CONSTANT STEP SIZE

Suppose $f : \mathbb{R}^n \to \mathbb{R}$ is a convex function defined over a convex set $C$, having a non empty set of minimum points $M^*$. To minimize $f$, the projected gradient method uses the iteration $x_{k+1} = [x_k - h_{k+1}g(x_k)]^+$, where $x_k$ is the $k$-th iterate, $g(x_k)$ is the gradient of $f$ at $x_k$ and $h_{k+1}(x_k)$ is the step size, and for constant step size we have $h_k = h \forall k$. We assume that $f$ satisfies the Lipschitz continuity condition and therefore, $||g(x_k)||_2 \leq G, \forall k$.

**Theorem A.1:** For any $x^* \in M^*$, as $k \to \infty$ one can find a $x$, such that $f(x) = \lim_{k \to \infty} f(x_k)$ and $f(x^*) - f(x) \leq G^2h/2$.

**Proof:** If $g(x_{k^*}) = 0$ for some $k^*$, then $f(x_k) = f(x^*) \forall k \geq k^*$ and we may take $x = x^*$. If $g(x_k) \neq 0 \forall k$, then $x_{k+1} = [x_k - h_{k}g(x_k)]^+$. Let $z_{k+1} = x_k - h_{k+1}g(x_k)$ (without projection). Observe that

$$||x_{k+1} - x^*||_2 \leq ||z_{k+1} - x^*||_2.$$  \hfill (16)

This is true as when we project a point onto $C$, we move closer to every point in $C$. Now,

$$||z_{k+1} - x^*||_2^2 = ||x_k - h_{k}g(x_k) - x^*||_2^2 = ||x_k - x^*||_2^2 - 2h_{k}g(x_k)^T(x_k - x^*) + h_{k}^2||g(x_k)||_2^2.$$  

From (16), we have

$$||x_{k+1} - x^*||_2^2 \leq ||x_k - x^*||_2^2 - 2h_{k}g(x_k)^T(x_k - x^*) + h_{k}^2||g(x_k)||_2^2$$ \hfill (17)

From the definition of the gradient we have,

$$f(x^*) \geq f(x_k) + g(x_k)^T(x^* - x_k).$$ \hfill (18)
From 17 and 18, we get the following inequality

$$||x_{k+1} - x^*||_2^2 \leq ||x_k - x^*||_2^2 - 2h(f(x_k) - f(x^*)) + h^2||g(x_k)||_2^2.$$ (19)

Recursively from (19), we get,

$$||x_{k+1} - x^*||_2^2 \leq ||x_0 - x^*||_2^2 - 2h \sum_{i=0}^{k}(f(x_i) - f(x^*)) + h^2 \sum_{i=0}^{k}||g(x_i)||_2^2.$$ (20)

Using $$||x_{k+1} - x^*||_2^2 \geq 0$$, we have,

$$2h \sum_{i=0}^{k}(f(x_i) - f(x^*)) \leq ||x_0 - x^*||_2^2 + h^2 \sum_{i=0}^{k}||g(x_i)||_2^2.$$ (21)

Combining this with $$\sum_{i=0}^{k}(f(x_i) - f(x^*)) \geq (k+1)(f(x_k) - f(x^*))$$, we get the inequality

$$2h(k+1)(f(x_k) - f(x^*)) \leq ||x_0 - x^*||_2^2 + h^2 \sum_{i=0}^{k}||g(x_i)||_2^2.$$ (22)

Given that $$||g(x_i)|| \leq G$$, for all $$i$$, we have, $$f(x_k) - f(x^*) \leq \frac{||x_0 - x^*||_2^2 + h^2(k+1)G^2}{2h(k+1)}.$$ (23)

The right hand side converges to $$G^2h/2$$ as $$k \to \infty$$.

REFERENCES


