Demand Aware Fair Resource Allocation in TDMA Wireless Networks

Xiaolong Huang and Soumya Das
QUALCOMM Inc., San Diego, California, USA
Email: {xhuang, soumyad}@qualcomm.com

***Abstract*—Meeting traffic demand and enforcing fairness are often times necessary but conflicting objectives for resource allocation in wireless networks. Due to the resource sharing nature of wireless networks, without a mechanism for enforcing fairness, simply assigning resources to meet traffic demand of some network flows can lead to resource starvation of other network flows. Balancing these two objectives is more complex multi-hop wireless networks, as the resource contention could be indirect. In this paper, an algorithm is introduced to allocate time slots in TDMA-based multi-hop wireless networks to achieve a designated a balance between meeting traffic demand and enforcing fairness. Numerical results show that the algorithm performs significantly better than other resource allocation algorithms. The introduced algorithm is well suited for distributed TDMA-based wireless networks, such as ECMA-368 based UWB networks.**

***Index Terms*—Congestion control, fairness, multi-hop wireless network, optimization, quality of service, resource allocation, TDMA.**

I. Introduction

Quality of Service (QoS) and fairness are both important yet often times mutually conflicting objectives for resource allocation and scheduling in wireless networks. Due to the resource sharing nature of a wireless environment, meeting the QoS of some flows without addressing the fairness issue may lead to resource starvation of other flows. The problem of balancing QoS and fairness becomes more complex when the network spans more than a single hop. This is evident in Fig. 1.

In Fig. 1, a flow contention graph is composed of three one hop flows $\{A, B, C\}$. Due to some underlying network topology, flow B contends with both flows A and C, while flows A and C do not contend with each other directly. Hence, a transmission of either flow A or flow C would block flow B. Two time slots are assumed available for the three flows to use. Each flow is assumed to require one time slot to meet its traffic demand. As we can see, the resource allocation strategy at the top leaves flow B no time resource to use, while the resource allocation strategy at the bottom can serve all traffic demands.

The problem of maximizing the time slot allocation efficiency in TDMA wireless networks by exploiting the spatial reuse is NP-complete [8]. Several algorithms [8], [10] have been introduced to probabilistically achieve the maximum resource allocation efficiency without considering QoS and fairness.

QoS and fairness of resource allocation in wireless networks have been studied in separate contexts extensively. Various fairness measures have been introduced. Some solutions are designed to achieve specific objectives, such as proportional fairness [1] and max-min fairness [2]. Algorithms to achieve those objectives are usually complex. Other resource allocation algorithms, such as DRAND [4], provide a level of fairness and spatial reuse in multi-hop ad hoc networks without specific objective functions but involve much less computational complexity. These algorithms enforce fairness in the absence of QoS requirement, hence can be applied when every flow impose infinite traffic demand. On the other end, various resource allocation algorithms are introduced to solely meet the QoS requirements. Schemes, such as [5], allocate time slots on a flow by flow basis to meet their traffic demands and can easily lead to unfair congestion situations.

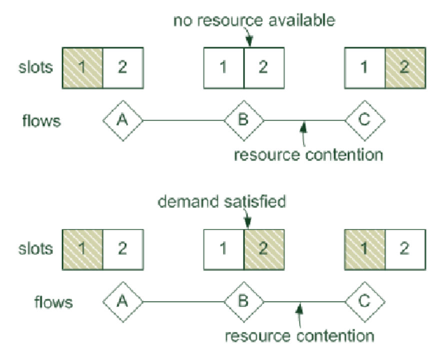

Fig. 1. Resource Allocation Problem in Multi-hop Wireless Networks

Several algorithms [9], [11] have been introduced to address the tradeoff between QoS and fairness by dynamically allocating time slots based on traffic loading and flow contention. In [3], a gradient method based resource allocation scheme is introduced to gradually regulate the data rate of end-to-end flows so that a utility function can be maximized across the network under the underlying flow contention constraint. Such schemes require adjusting allocated resources in a highly dynamic manner. However, demand assigned TDMA-based wireless networks, such as WiMedia networks [6], [7], expect such resource assignment to be static over a period of time. Hence, the aforementioned schemes are not suitable for such a deployment.

In this paper, a Demand Aware Fair resource allocation algorithm (DAF) is proposed to allocate time slots in TDMA-based multi-hop wireless networks. The DAF algorithm considers the requirement of QoS and fairness jointly, where QoS is measured by the amount of traffic demand that is being served. The number of assigned time slots remains static during flow holding (active) times. To meet the traffic demand and to preserve fairness among multiple flows, DAF is designed to achieve the following objectives in TDMA-based multi-hop wireless networks:

- A flow is guaranteed a minimal number of time slots, called fair share when it has infinite traffic demand. This imposes a basic standard of fairness.
- The traffic demand of a flow is met when it is lower than the fair share of the flow.
- Achieve a prescribed balance between serving the traffic demand and reducing congestion.

We show that the proposed DAF algorithm meets the traffic demand and enforces the predefined fairness. DAF is well suited for distributed TDMA-based wireless networks, such as a WiMedia network, in which nodes advertise their available time slots to their 2-hop neighbors and then exchange messages to reserve time slots for new flows.

We introduce several important concepts in Section II. In Section III, the DAF algorithm and its objective is described. Numerical results are given in Section IV. Section V concludes this paper.

II. System Model

In this section, we present and develop several concepts for modeling the resource allocation problem in a TDMA-based multi-hop wireless network.

A. Maximal Common Slot Set

In this paper, a one-hop flow is simply referred to as a flow. A flow is always considered bidirectional so to take into account both data and acknowledgement transmissions.

In a TDMA-based wireless network, a flow f can only have transmissions in a time slot s during which its sender and receiver are not participating in transmissions for other flows. The time slot s is then said to be available to flow f.

Definition 1: A set of time slots S is said to be *commonly available* to a group of flows F, if $\forall s \in S$ and $\forall f \in F$, s is available to f. S is said be a *common slot set* of F. F is said to be a *common flow group* of S.

A time slots set S, a flows group F, and their relations can be modeled as a bipartite graph $G = (S + F, E)$. An edge exists between $s \in S$ and $f \in F$ if and only if s is available to f. A common slot set S and its common flow group F forms a complete bipartite graph.

Definition 2: A group of flows F is said to be the *maximal common flow group* of its common slot set S, if and only if S does not have another common flow group $\widetilde{F}$ so that $F \subset \widetilde{F}$.

Definition 3: A set of time slots S that has a maximal common flow group F is said to be a *maximal common slot set* (MCSS), if and only if 1) No other time slot set $\widetilde{S}$ has F as its maximal common flow group, and 2) No other time slot set $\widetilde{S}$, $\widetilde{S} \subset S$ has a maximal common flow group $\widetilde{F}$ such that $\widetilde{F} \supset F$.

The first requirement means that the complete bipartite graph formed by S and F includes all time slots that are available to F. The second requirement means that a maximal common slot set does not contain any common slot set that serves more flows than it does.

The significance of the maximal common slot set is that a TDMA frame can be partitioned into disjoint maximal common slot sets, where each MCSS set has a designated maximal flow group it can serve.

A simple way to obtain MCSSs is to start with individual time slots and their maximal common flow groups, and then group those time slots that have identical maximal common flow groups. The process stops when no two time slot sets have identical maximal flow groups. The maximum computational complexity is of $O(n^2)$, where n is the number of time slots. In this paper, we do not study the algorithm of obtaining maximal common slot sets.

B. Maximal Common Slot Set based Flow Contention Graph

In wireless networks, two transmissions may cause strong interference between each other if they overlap in time, frequency and space. In this paper, only single carrier TDMA-based multi-hop wireless networks are treated. Hence, only time and space domains can be explored, which manifests as TDMA operations and spatial reuse, respectively. To precisely capture the exploration of time and space, we introduce the concept of Maximal Common Slot Set based Flow Contention Graph (MCSS-FCG).

Definition 4: Two flows are said to be *contending flows* for each other if their simultaneous transmissions cause strong interference to each other and subsequently result in transmission failures. Under a protocol model, two flows are said to be contending for each other if the source or the destination of one flow is within the nominal communication range of that of the other flow.

A Flow Contention Graph (FCG) captures all flow contention information. The mapping from a nodal graph to a flow contention graph is well known [12] and illustrated in Fig. 2. In Fig. 2, all five flows are considered bidirectional. Each flow in the nodal graph is converted into a vertex in the flow contention graph. An edge exists between two vertices in a flow contention graph if and only if the corresponding two flows contend with each other.

Definition 5: A *maximal clique* is a set of vertices that induces a complete graph, and is not a sub-graph of any other complete graph. A *degree* of a maximal clique is defined as the number of vertices in that clique.

A flow contention graph can be decomposed into a set of maximal cliques as shown in Fig. 3. In Fig.3, the flow contention graph is composed of three maximal cliques, $\{A, B, C\}$, $\{B, C, D\}$ and $\{D, E\}$. Each clique is a complete graph and is not a sub-graph of any other complete graph in the flow contention graph.

Definition 6: A *Maximal Common Slot Set based Flow Contention Graph* (MCSS-FCG) is just a flow contention graph with respect to a maximal common slot set. All time

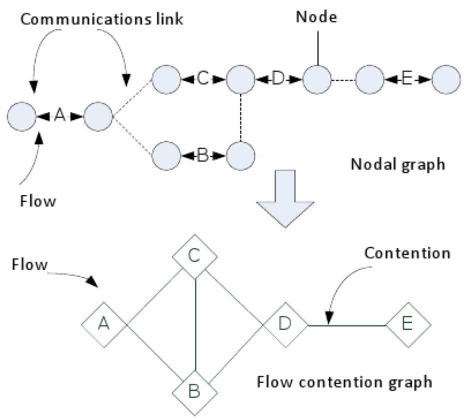

Fig. 2. Nodal Graph and its Flow Contention Graph

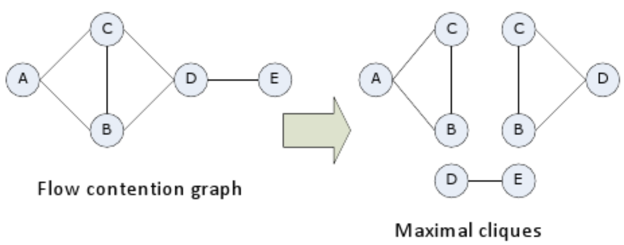

Fig. 3. Maximal Clique Decomposition of a Flow Contention Graph

slots in a maximal common slot set are available to all flows in the MCSS-FCG.

A MCSS-FCG is only meaningful with respect to its maximal common slot set. MCSS-FCGs associated with different maximal common slot sets are distinct. Flows that are assigned time slots in different maximal common slot sets do not contend with each other. We note that, when all time slots are available to all flows in the network, there exists a unique MCSS-FCG in the network, which is the overall flow contention graph itself.

An example of maximal common slot sets of their MCSS-FCG is illustrated in Fig. 4. There are 7 flows, $\{A, B, ..., G\}$, in the overall flow contention graph. Each flow has some particular time slots available in the frame for its use. Three maximal common slots are assumed to be identified. Each maximal common slot has a flow contention graph that is formed by its common flow group. A flow may reside in multiple MCSS-FCGs. For example, all three maximal common slot sets are available for flow A. Hence, flow A resides in all three MCSS-FCGs.

C. Fair Share in a MCSS-FCG

Definition 7: A *fair share* is defined as a number of time slots that shall be assigned to a flow when every flow in the network imposes infinite traffic demand that can saturate the network.

Assigning fair shares to flows when flows contend among one another in a traffic overloaded network enforces a standard of fairness. Since flows that are assigned with time slots in

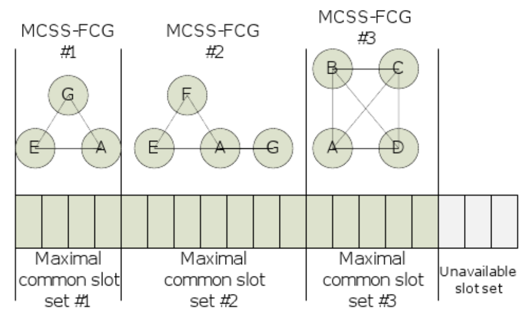

Fig. 4. Maximal Common Slot Set and its MCSS-FCG

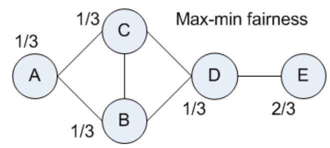

Fig. 5. Max-min Fair Share Assignment

different maximal common slot sets do not contend with each other, the fair share is only meaningful with respect to a specific MCSS-FCG. When a flow resides in multiple MCSS-FCGs, this flow has a fair share value in each MCSS-FCG.

Fair shares can be assigned based on a specific objective or assigned arbitrarily based on service agreement. To give an intuitive idea of fairness, the normalized max-min fair share assignment in a perfect graph studied in [2] is shown in Fig. 5. It is feasible to allocate and schedule $\frac{1}{3}$ time slots to Flows A, B, C, and D and allocate $\frac{1}{3}$ time slots to Flow E. The fairness nature of this allocation is that no flow can increase the assigned amount of time slots without reducing the time slots of other flows that are already assigned with less or equal amount of time slots.

In our resource allocation algorithm study, a simple rule of assigning fair shares is used. The fair share for a flow in a MCSS-FCG that has L time slots in the associated maximal common slot set is set to $\frac{L}{d}$, where d is the highest degree of all maximal cliques the flow resides in.

III. Resource Allocation Algorithm

At any point in time, a TDMA-based multi-hop wireless network may be serving existing flows while a new set of flows may be initiated in the network. The traffic demand of these new flows can be expressed in bits per frame. To negotiate the traffic demand to be served, a signaling process can compute and then reserve time slots for these flows on selected routes. Our DAF resource allocation algorithm used by the signaling process strives to achieve the following objectives:

- The portion of traffic demand within the fair share of a flow should be fully met.
- Minimize the cost associated with inadequate serving of traffic demand and the cost associated with allocating time slots above fair shares.

DAF is executed over a set of MCSS-FCGs. DAF comprises two processes, namely the *inter-graph process* and the *intra-graph process*. The inter-graph process selects, in each iteration step, a maximal common slot set and its associated MCSS-FCG to execute the intra-graph process. Its selection of the maximal common slot set is critical to meet the traffic demand. The intra-graph process assigns time slots in the selected MCSS-FCG with an objective to balance between QoS and fairness.

We note that, for a real implementation in a distributed manner, the computational complexity of obtaining a complete MCSS-FCG can be large. To reduce the control overhead and computational complexity, a MCSS-FCG used for computation may cover only the maximal cliques where the underlying flow resides. Once the computation is finished, the source/destination of the underlying flow can advertise its assignment results to the source/destination of a contending flow. To ensure a feasible time slot assignment, the final time slot assignment of a flow can be set to the minimum among all assignments of this flow suggested by the sources/destinations of this flow and contending flows. Such an assignment computed in a distributed manner could be sub-optimal. Nevertheless, we proceed in the following to describe the algorithm in its centralized form.

A. Inter-Graph Process

The following notations are used:

- Denote by $\{G_n\}$, $n = 1, 2, ..., N$, the set of MCSS-FCGs. Denote by $\{cl_n^k\}$, $k = 1, 2, ..., K$, the set of maximal cliques in G_n. Denote by d_n^k the degree of cl_n^k.
- Denote by $\{s_l\}$, $l = 1, 2, ..., L$, the set of flows in $\{G_n\}$. Denote by q_l the outstanding traffic demand of s_l.

Algorithm 1: Inter-Graph Process

For each G_n, set the maximal degree $\hat{d_n}$ of G_n to be $\hat{d_n} = \max_{k=1}^{K} d_n^k$;
Sort $\{G_n\}$ in an ascending order of $\{\hat{d_n}\}$;
Denote by n_i the original index of the i^{th} MCSS-FCG in the sorted set;
for $i = 1, 2, ..., N$ **do**
    Execute the intra-graph process based on G_{n_i}. Denote by $q_l^{n_i}$ the traffic demand served in G_{n_i} for s_l after the intra-graph process;
    for $l = 1, 2, ..., L$ **do**
        $q_l := q_l - q_l^{n_i}$;
        Remove s_l from $\{G_n\}$ if $q_l == 0$;
    end
    Update $\{\hat{d_n}\}$. Sort $\{G_n\}$ in an ascending order of $\{\hat{d_n}\}$;
end

The inter-graph process is specified by Algorithm 1. The process utilizes the time slots of a MCSS-FCG ahead of other MCSS-FCGs, if its flow contention level is the lowest (i.e., the highest degree of all of its maximal cliques is the lowest compared to other MCSS-FCGs). By doing this, the traffic demand of a flow can be served as much as possible before this flow enters the competition for the precious time slot resources with other contending flows in those high contention level MCSS-FCGs. Once time slots are assigned to a flow in an intra-graph process, the demand of this flow is reduced accordingly. If the demand of a flow is fully met, the flow is removed from the set of MCSS-FCGs. Hence, the inter-graph process aggressively reduces the flow contention levels of MCSS-FCGs in each iteration.

The computational complexity of the inter-graph process is identified to be $O\left(NW + N^2L + N^3\right)$, where W denotes the complexity of intra-graph process, N denotes the number of MCSS-FCGs, and L denotes the number of flows in the network. The result comes with the worst case assumption for the sorting complexity known as $O\left(k^2\right)$, where k is the number of elements for sorting. The derivation of the complexity is straightforward and hence omitted in this paper due to the page limit.

B. Intra-Graph Process

The intra-graph process iterates over maximal cliques in a MCSS-FCG. For each maximal clique, an intra-graph resource allocation algorithm calculates the number of slots to be assigned to each flow within the maximal clique. If a flow resides in multiple maximal cliques, the number of time slots assigned to the flow in this MCSS-FCG is set to the minimum of all values assigned to the flow.

The trade-off between meeting the traffic demand and preserving fairness takes the center stage of the resource allocation algorithm. The intra-graph resource allocation algorithm executed over a maximal clique strives to achieve all objectives listed at the beginning of Section III. The algorithm minimizes the total cost incurred from inadequate serving of traffic demand and allocating time slots beyond a fair share.

The cost functions associated with inadequate serving of traffic demand and allocating time slots beyond a fair share can be quite general as long as they have the following properties.

Denote by $u(z)$ the cost function induced by allocating z time slots above the fair share of a flow. With respect to a maximal clique, denote by x_i and f_i the actual number of time slots allocated to flow i and the fair share of flow i, respectively. We need

- $u(z_i) = u(x_i - f_i)$ and $u(x_i - f_i) = 0, \forall x_i \leq f_i$.
- $u(z)$ is a strictly convex and strictly increasing function w.r.t. z.

Hence, the total cost induced by allocating time slots above fair shares is $\sum_i u(x_i - f_i) I_{x_i \geq f_i}$, where I is an indicator function. $I_{x_i \geq f_i} = 1$ if $x_i \geq f_i$, otherwise $I_{x_i \geq f_i} = 0$.

Denote by $v(z)$ the cost function induced by inadequate serving of traffic demand of a flow, where z denotes the traffic demand that is not served after the allocation. Denote by R_i the data rate (in bits per slot) that can be achieved for the transmission of flow i. Denote by q_i the traffic demand (in bits per frame) from flow i. We need

- $v(z_i) = v(q_i - R_i x_i)$ and $v(q_i - R_i x_i) = 0, \forall \frac{q_i}{R_i} \leq x_i$.
- $v(z)$ is a strictly convex and strictly increasing function w.r.t. z.

Hence, the total cost induced by inadequate serving of traffic demand is $\sum_i v(q_i - R_i x_i) I_{\frac{q_i}{R_i} \geq x_i}$.

The optimization problem for achieving all objectives specified in Section III is described below:

$$\min_{x_1,x_2,\ldots,x_L} w_u \sum_{i=1}^{L} u(x_i - f_i) I_{x_i \geq f_i} + w_v \sum_{i=1}^{L} v(q_i - R_i x_i) I_{\frac{q_i}{R_i} \geq x_i}. \quad (1)$$

The problem is subject to the following constraints:

- The portion of traffic demand within the fair share of a flow should be fully met.

$$R_i x_i = q_i, \forall \frac{q_i}{R_i} \leq f_i. \quad (2)$$

- The actual assigned resource is no greater than q_i.

$$R_i f_i \leq R_i x_i \leq q_i, \forall \frac{q_i}{R_i} \geq f_i. \quad (3)$$

- The total number of slots assigned in a maximal clique is no larger than s_n, where s_n is the total number time slots in MCSS-FCG G_n.

$$\sum_{i}^{L} x_i \leq s_n. \quad (4)$$

The cost functions are weighted by w_u and w_v, which can be used as preferences given to QoS and fairness, respectively.

To help solving the optimization problem 1, the following derived functions are introduced.

- Denote by $U_i(x)$ the rate of cost increase induced by allocating time slots beyond fair share for flow i. Precisely, we define

$$U_i(x) = w_u \frac{\partial u(x - f_i)}{\partial x}. \quad (5)$$

  Note that, U_i increases with respect to x, since $\frac{\partial U_i(x)}{\partial x} = w_u \frac{\partial^2 u(x-f_i)}{\partial x^2} > 0$.
- Denote by $V_i(x)$ the rate of cost decrease induced by inadequately serving traffic demand for flow i. Precisely, we define

$$V_i(x) = -w_v \frac{\partial v(q_i - R_i x)}{\partial x}. \quad (6)$$

  Note that, V_i decreases with respect to x, since $\frac{\partial V_i(x)}{\partial x} = -w_v \frac{\partial^2 v(q_i - R_i x)}{\partial x^2} < 0$.

Hence, $V_i(x) - U_i(x)$ is a non-increasing function with respect to x. We call $V_i(x) - U_i(x)$ the characteristic function of flow i.

The intra-graph resource allocation process is specified by Algorithm 2. The algorithm essentially does the following: 1) when the rate of cost decrease in inadequately serving traffic demand is less than the rate of cost increase in allocating extra time slots beyond fair share, the assignment moves towards the fair share; 2) when the rate of cost decrease in inadequately serving traffic demand is more than the rate of cost increase in allocating extra time slots beyond fair share, the assignment moves towards meeting the traffic demand of the flow; 3) the number of time slots assigned to a flow is set to a value between the fair share and the traffic demand such that the assignment balances these two conflicting costs.

Algorithm 2: Intra-Graph Algorithm

for *any flow i that has $q_i \leq R_i f_i$* **do**
  Set $\hat{x_i} = \frac{q_i}{R_i}$ as the number of time slots to be assigned;
end
for *any flow i that has $q_i > R_i f_i$* **do**
  Calculate the max value and the min value of $V_i(x) - U_i(x)$. In fact, we have $V_i(f_i) - U_i(f_i) \geq V_i(x) - U_i(x) \geq V_i(\frac{q_i}{R_i}) - U_i(\frac{q_i}{R_i})$;
end
Put all max and min values of $V_i(x) - U_i(x)$ into one set Φ. Sort the elements of Φ in an increasing order. Denote the sequence by $\phi_j, j = 1, 2, ..., J$;
for $j = 1, 2, ..., J$ **do**
  for *all flow $i = 1, 2, ...\widetilde{L}$ that do not have final slot assignments* **do**
    Calculate x_i as follows:

$$x_i = \begin{cases} f_i & \phi_j \geq V_i(f_i) - U_i(f_i) \\ \frac{q_i}{R_i} & \phi_j \leq V_i(\frac{q_i}{R_i}) - U_i(\frac{q_i}{R_i}) \\ x | V_i(x) - U_i(x) = \phi_j & \text{o.w.} \end{cases}$$

  end
  if $\sum_{i}^{L} x_i \leq s_n$ **then**
    Set $\phi_H = \phi_j$ and $\phi_L = \phi_{j-1}$;
    Break;
  end
end
for *all flow $\Omega = \{i\}$ that have $\phi_L \leq V_i(x_i) - U_i(x_i) \leq \phi_H$* **do**
  Solve the equation array $V_i(x_i) - U_i(x_i) = V_j(x_j) - U_j(x_j)$, $i \in \Omega$ and $\sum_{l=1}^{L} x_l = s_n$, L is the total number of flows;
end
Set $\hat{x_i} = x_i$;

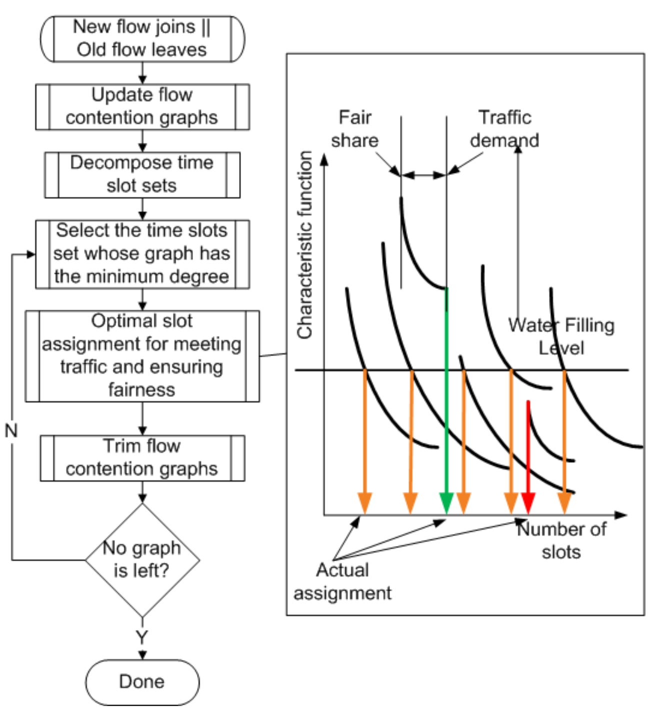

Fig. 6. DAF algorithm

 ISBN: 978-1-61208-203-5

Solving the equation array at Step 15 in Algorithm 2 gives the precise optimal solution, but can be computationally expensive, as $V(.)$ and $U(.)$ can be nonlinear polynomials as defined earlier. Hence, the bisection method can be used to find a solution arbitrarily close to the optimal solution with much less complexity. At Step 15 in Algorithm 2, the following bisectional iteration should be executed. In each iteration step, set $\hat{\phi} = \frac{\phi_L+\phi_H}{2}$ and set $x_i = \{x|V_i(x) - U_i(x) = \hat{\phi}\}$. If $\sum_{l=1}^{L} x_l < s_n$, set $\phi_H = \hat{\phi}$. If $\sum_{l=1}^{L} x_l > s_n$, set $\phi_L = \hat{\phi}$. And then go to the next iteration. The iteration ends when the difference between $\sum_{l=1}^{L} x_l$ and s_n is less than a negligible error margin.

A graphic representation of the resource allocation results of the intra-graph algorithm is shown in Fig. 6. The intra-graph algorithm is shown as one step in the inter-graph process. The characteristic function of each flow is evaluated against a common water filling level, which is fundamentally determined by the network scenario. A flow should be served exactly an amount so that its associated characteristic function level is closet to the common water filling level.

In the following, we prove that Algorithm 2 solves the optimization problem in Eq.1.

Theorem 1: For any functions $u(z)$ and $v(z)$ that possess the general properties described above, the optimization problem 1 has the following optimal solution:

$$\hat{x_i} = \begin{cases} \frac{q_i}{R_i} & \text{if } q_i \le R_i f_i \\ f_i & \text{if } q_i > R_i f_i \text{ and } \gamma > V_i(f_i) - U_i(f_i) \\ \in \left(f_i, \frac{q_i}{R_i}\right) & \text{if } q_i > R_i f_i \text{ and } \gamma = V_i(\hat{x_i}) - U_i(\hat{x_i}) \\ \frac{q_i}{R_i} & \text{if } q_i > R_i f_i \text{ and } \gamma < V_i(\frac{q_i}{R_i}) - U_i(\frac{q_i}{R_i}) \end{cases} \tag{7}$$

where $\gamma > 0$ is selected so that $\sum_{i}^{L} x_i \le s_n$.

Proof: From Eq. 2, we have $\hat{x_i} = \frac{q_i}{R_i}, \forall q_i \le R_i f_i$. This indicates that we only need to solve the problem for $q_i > R_i f_i$, so that $\frac{q_i}{R_i} > x_i$ and $x_i > f_i$. Hence, identity functions in Eqn. 1 can be subsequently removed and the problem is reduced to the following:

$$\min_{x_1, x_2, \ldots, x_{\widetilde{L}}} w_u \sum_{i=1}^{\widetilde{L}} u(x_i - f_i) + w_v \sum_{i=1}^{\widetilde{L}} v(q_i - R_i x_i) \tag{8}$$

$$\text{s.t.} \begin{cases} f_i \le x_i \le \frac{q_i}{R_i} \\ \sum_{i=1}^{\widetilde{L}} x_i \le s_n - \sum_{j} \frac{q_j}{R_j}, q_j \le R_j f_j \end{cases} \tag{9}$$

Let $Q_t = \sum_{j} \frac{q_j}{R_j}, q_j \le R_j f_j$. We have Q_t to represent the total number of time slots assigned to those flows whose traffic demands are less than their fair shares.

Based on the Karush-Kuhn-Tucker conditions, we have

$$\begin{cases} \frac{\partial\left(w_u \sum_{i=1}^{\widetilde{L}} u(\hat{x_i}-f_i) + w_v \sum_{i=1}^{\widetilde{L}} v(q_i - R_i\hat{x_i})\right)}{\partial \hat{x_i}} \\ + \frac{\partial \beta_i (R_i\hat{x_i} - q_i)}{\partial \hat{x_i}} + \frac{\partial \sigma_i (f_i - \hat{x_i})}{\partial \hat{x_i}} + \frac{\partial \gamma \left(\sum_{i=1}^{\widetilde{L}} \hat{x_i} - (s_n - Q_t)\right)}{\partial \hat{x_i}} = 0 \\ \beta_i(R_i\hat{x_i} - q_i) = 0 \\ \sigma_i(f_i - \hat{x_i}) = 0 \\ \gamma\left(\sum_{i=1}^{\widetilde{L}} \hat{x_i} - (s_n - Q_t)\right) = 0 \\ \beta_i \ge 0, \sigma_i \ge 0, \gamma \ge 0 \end{cases}$$

$$\Leftrightarrow \begin{cases} U_i(\hat{x_i}) - V_i(\hat{x_i}) + R_i\beta_i - \sigma_i + \gamma = 0 \\ \beta_i(R_i\hat{x_i} - q_i) = 0 \\ \sigma_i(f_i - \hat{x_i}) = 0 \\ \gamma\left(\sum_{i=1}^{\widetilde{L}} \hat{x_i} - (s_n - Q_t)\right) = 0 \\ \beta_i \ge 0, \sigma_i \ge 0, \gamma \ge 0 \end{cases}$$

For the case $\sigma_i > 0$, we have $\hat{x_i} = f_i$. Consider the following sub-cases.

1) For $\beta_i = 0$, $\hat{x_i} < \frac{q_i}{R_i}$, we have $U_i(f_i) - V_i(f_i) + \gamma = \sigma_i > 0$. Hence, $\gamma > V_i(f_i) - U_i(f_i)$.
2) For $\beta_i > 0$, we have $\hat{x_i} = \frac{q_i}{R_i}$. Since we have $q_i > R_i f_i$, and $\hat{x_i} = f_i$ is contradictory to $\hat{x_i} = \frac{q_i}{R_i}$, this case is abandoned.

For the case $\sigma_i = 0$, $\hat{x_i} > f_i$, consider the following sub-cases.

1) For $\beta_i > 0$, we have $\hat{x_i} = \frac{q_i}{R_i}$ and $U_i(\frac{q_i}{R_i}) - V_i(\frac{q_i}{R_i}) + \gamma = -R_i\beta_i < 0$. Hence, we have $\gamma < V_i(\frac{q_i}{R_i}) - U_i(\frac{q_i}{R_i})$.
2) For $\beta_i = 0$, $\hat{x_i} < \frac{q_i}{R_i}$, we have $U_i(\hat{x_i}) - V_i(\hat{x_i}) + \gamma = 0$. Hence, $\gamma = V_i(\hat{x_i}) - U_i(\hat{x_i})$ and $f_i \le \hat{x_i} \le \frac{q_i}{R_i}$.

■

Theorem 2: Algorithm 2 achieves the optimal solution for the optimization problem in Eq.1.

Proof: Assume $\sum_{l=1}^{L} \frac{q_l}{R_l} > s_n$. Otherwise, the optimal solution is $\hat{x_l} = \frac{q_l}{R_l}$.

At the end of step 13 in Algorithm 2, we have identified two bounds ϕ_L and ϕ_H. Due to the execution of Step 10 and 11, setting $\gamma = \phi_L$ and setting $\hat{x_l}$ according to Eq. 7, we will have $\sum_{l=1}^{L} x_l > s_n$. Setting $\gamma = \phi_H$ and setting $\hat{x_l}$ according to Eq. 7, we will have $\sum_{l=1}^{L} x_l \le s_n$. Hence, the optimal solution exists for $\gamma \in [\phi_L, \phi_H]$.

Consider those flows $\{i\}$ where $V_i(f_i) - U_i(f_i) < \phi_L$. Since $\phi_L \le \gamma$, we have $V_i(f_i) - U_i(f_i) < \gamma$. At the end of Step 13, we already set $\hat{x_i} = f_i$, which is the optimal solution of these flows according to Theorem 1. Consider those flows where $V_j(\frac{q_j}{R_j}) - U_j(\frac{q_j}{R_j}) > \phi_H$. Since $\phi_H \ge \gamma$, we have $V_j(\frac{q_j}{R_j}) - U_j(\frac{q_j}{R_j}) > \gamma$. At the end of Step 13, we already set $\hat{x_j} = \frac{q_j}{R_j}$, which is the optimal solution of these flows according to Theorem 1. The remaining flows have $V_k(x_k) -$

$U_k(x_k) = \gamma$, which is the optimal solution for them according to Theorem 1.

■

Assume the bisection method to be used at Step 15. The computational complexity of the intra-graph process is $O\left(L^2 + L\log_2 \frac{\phi_H - \phi_L}{\Delta}\right)$, where L denotes the number of flows in the MCSS-FCG, and Δ denotes the negligible error margin. The derivation of the complexity is straightforward and hence omitted in this paper due to the page limit.

IV. NUMERICAL RESULTS

In this section, the performance of DAF is demonstrated through numerical experiments. The benefit of the inter-graph process is demonstrated by Scenario 1, where MCSS-FCGs are simple so that the algorithm execution order over the list of MCSS-FCGs has a more significant impact on the time slot assignment than resolving the flow contention in each MCSS-FCG. The benefit of the intra-graph process is demonstrated by Scenario 2, where MCSS-FCGs are complex so that it is critical to address the trade-off between meeting traffic demands and preserving fairness within each MCSS-FCG.

We compare the time slot allocation results obtained by DAF to those obtained by DRAND [4]. Furthermore, we extend DRAND to an enhanced version that employs our inter-graph process. We refer to the enhanced version of DRAND as E-DRAND. E-DRAND selects, in each iteration step, a maximal common slot set and its associated MCSS-FCG to execute the generic DRAND algorithm. E-DRAND helps identify the benefit of the inter-graph process. Numerical results show that DAF performs significantly better than DRAND and E-DRAND.

A. Simulation Model

The inputs of our program include flows, their traffic demands, and MCSS-FCGs, so that the program does not implement control protocols or graph computation algorithms to prepare these input parameters. We did not conduct our simulation using a full fledged network simulator, such as NS-2, since a generic program can provide us rich scenarios with flexible input parameters, without the involvement of inessential network events in a full fledged network simulator.

The example function $u(z)$ that represents the cost induced by allocating extra time slots above the fair share is set to $u(z) = z$, $z \geq 0$, equivalent to $u(x-f) = x - f, x \geq f$. Hence, we have $U_i(x) = w_u$. The example function $v(z)$ that represents the cost induced by inadequately serving traffic demand is set to $v(z) = \frac{1}{q-z} - \frac{1}{q}$, $0 \leq z \leq q$, equivalent to $v(q - Rx) = \frac{1}{Rx} - \frac{1}{q}$, $0 \leq x \leq \frac{q}{R}$. Hence, we have $V(x) = \frac{w_v}{Rx^2}$. Under this setting, the cost induced by inadequately serving traffic demand increases drastically when more traffic demand is not served.

We set the weights for these cost functions so that their rate of changes are about the same. Assume a typical flow requires 50 time slots among 256 available time slots. We set $\frac{w_u}{w_v} = \frac{1}{2500R}$. For a concise presentation, we assume a common data rate $R = 1$ bit per slot for all flows.

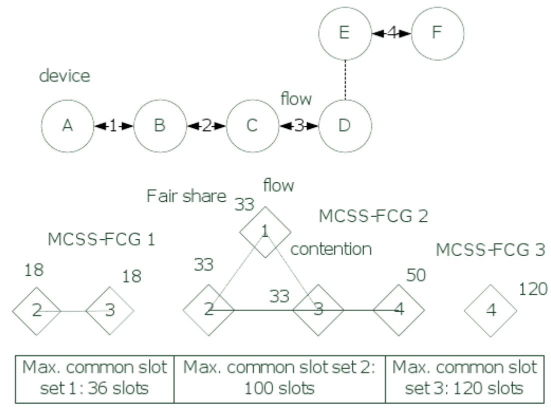

Fig. 7. Network Scenario 1

The criterion for fairness is defined with low computational complexity. The fair share for a flow i, in a MCSS-FCG, G_n, is set to be equal to $\frac{s_n}{d_i^n}$, where d_i^n is the maximum degree of all maximal cliques flow i resides within G_n, s_n is the number of time slots available in G_n. An assignment is said to be fair for flows in the MCSS-FCG, if every flow in the MCSS-FCG is assigned an amount of slots that is larger than or equal to its fair share.

To quantitatively evaluate the performance of the algorithms, we compare the values of the objective function specified in Eq. 1 calculated on the overall contention graph. The objective function specified in Eq. 1 represents the total cost incurred from inadequate serving of traffic demand and allocating time slots beyond a fair share.

B. Network Scenario 1

A network scenario with a relatively low flow contention level is studied to highlight the benefit of the inter-graph process. In this setup, fairness is a lesser issue, since traffic demand can be well served by using a particular order in which the inter-graph process selects MCSS-FCGs. The scenario is illustrated in Fig. 7.

There are 4 new flow arrivals. Each flow has an opportunity to utilize some time slots. Three maximal common slot sets and their corresponding MCSS-FCGs are identified. Flow 3 can utilize 136 time slots in total (36 from MCSS 1 and 100 from MCSS 2), however it has to contend with many other flows in two maximal common slot sets. Flow 4 can utilize 120 time slots without facing any contention from other flows. Two traffic demand patterns are simulated. They are specified by their corresponding traffic demand vectors in Table I and Table II. Traffic pattern 2 has slightly higher traffic demand than traffic pattern 1. The results under traffic pattern 1 and traffic pattern 2 are shown in Table I and Table II, respectively.

DAF is shown to meet the joint QoS and fairness requirement much better than DRAND and perform about the same as E-DRAND does, as it achieves the lowest objective function value. This result highlights the significant benefit of using our inter-graph process.

TABLE I
SLOT ASSIGNMENT COMPARISON UNDER TRAFFIC PATTERN 1

Flow ID	Demand (slots)	DAF	DRAND	E-DRAND
#1	35	35	28	35
#2	70	52	46	65
#3	35	35	35	35
#4	100	100	100	100
	Objective Value	-70.6374	-59.5093	-67.2527

TABLE II
SLOT ASSIGNMENT COMPARISON UNDER TRAFFIC PATTERN 2

Flow ID	Demand (slots)	DAF	DRAND	E-DRAND
#1	50	36	24	36
#2	50	50	42	50
#3	50	50	46	50
#4	100	100	100	100
	Objective Value	-49.5556	-24.9617	-49.5556

TABLE III
SLOT ASSIGNMENT DETAILS UNDER TRAFFIC PATTERN 1

Flow ID	FCG #1	FCG #2	FCG #3
#1		35	
#2	18	35	
#3	18	17	
#4			100

The time slot assignment in each MCSS-FCG under traffic pattern 1 is shown in Table III. The following observations can be drawn from traffic pattern 1's results:

- The advantage of the inter-graph process can be clearly seen. Flow 4 is assigned as many slots as possible in G_3 due to the effect of the inter-graph process, so that its high demand does not interfere with other flows.
- Traffic demands of flow 1 and 3 are completely served due to their relatively low traffic demand levels. Flow 1's assignment in G_2 is slightly above its fair share (33 slots) since assignment for flow 2 and 3 in G_1 has relieved the contention in G_2. Again, the advantage of the inter-graph process is evident.
- Time slots of G_2 are not all utilized, since a flow only accepts the minimum of all assignments it receives from all maximal cliques within a graph.

The time slot assignment in each MCSS-FCG under traffic pattern 2 is shown in Table IV. The following observations can be drawn from traffic pattern 2's results:

- Traffic demand of flow 1 cannot be fully met due to its high level of traffic demand in the only MCSS-FCG, G_2, it resides in. Note that its assignment is above its fair share.
- Flow 2 and 3 are in a topologically similar position, as the degrees of their maximal cliques are equal in G_1 and G_2 and they contend with similar set of flows. When traffic demand levels of flow 2 and 3 become more even compared to theirs in traffic pattern 1, their traffic demand can be fully met.

TABLE IV
SLOT ASSIGNMENT DETAILS UNDER TRAFFIC PATTERN 2

Flow ID	FCG #1	FCG #2	FCG #3
#1		36	
#2	18	32	
#3	18	32	
#4			100

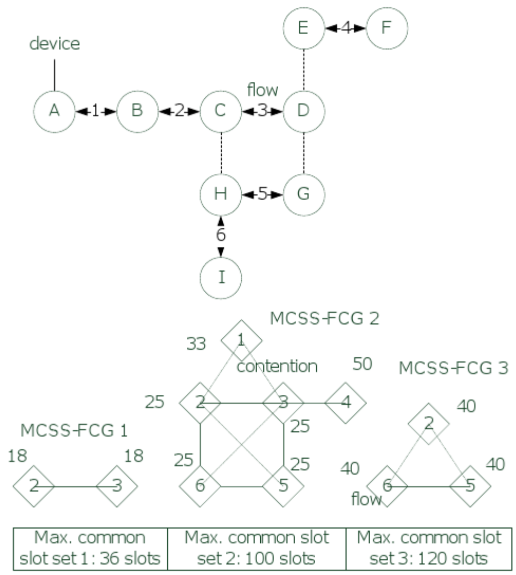

Fig. 8. Network Scenario 2

C. Network Scenario 2

A network scenario with a higher flow contention level is studied to highlight the benefit of the intra-graph process. In this setup, more fairness problems would need to be addressed by the intra-graph process and algorithm. The scenario is illustrated in Fig. 8.

There are 6 new flow arrivals. Three maximal common slot sets and their MCSS-FCGs are identified. Flow 2, 3, 5 and 6 are in a high contention level among each other in various MCSS-FCGS $\{G_1, G_2, G3\}$. Two traffic demand patterns are simulated. They are specified by their corresponding traffic demand vectors in Table V and Table VI. Traffic pattern 4 has slightly higher traffic demand than traffic pattern 3. The results under traffic pattern 3 and traffic pattern 4 are shown in Table V and Table VI, respectively.

DAF is shown to meet the joint QoS and fairness requirement much better compared to DRAND and E-RAND, as it achieves the lowest objective function value. This result highlights the significant benefit of using our intra-graph process.

The time slot assignment in each MCSS-FCG under traffic pattern 3 is shown in Table VII. The following observations can be drawn from traffic pattern 3's results:

- All traffic demand requirements are satisfied. This is due

TABLE V
SLOT ASSIGNMENT COMPARISON UNDER TRAFFIC PATTERN 3

Flow ID	Demand (slots)	DAF	DRAND	E-DRAND
#1	50	50	17	22
#2	50	50	50	50
#3	50	50	35	39
#4	50	50	16	21
#5	50	50	50	50
#6	50	50	50	50
	Objective Value	-39.0000	103.7374	39.7865

TABLE VI
SLOT ASSIGNMENT COMPARISON UNDER TRAFFIC PATTERN 4

Flow ID	Demand (slots)	DAF	DRAND	E-DRAND
#1	60	42	17	19
#2	60	60	60	60
#3	60	60	35	37
#4	60	50	16	20
#5	60	60	60	60
#6	60	60	60	60
	Objective Value	19.1905	158.7374	116.146

TABLE VII
SLOT ASSIGNMENT DETAILS UNDER TRAFFIC PATTERN 3

Flow ID	FCG #1	FCG #2	FCG #3
#1		50	
#2	18		32
#3	18	32	
#4		50	
#5		10	40
#6		10	40

to the fact that G_1 and then G_3 are able to sequentially provide large amount of time slots for flow 2, 5, and 6, which leads to a significantly lowered traffic demand in a highly contended MCSS-FCG G_2. As an example, flow 2 does not need to be assigned any time slots in G_2.

The time slot assignment in each MCSS-FCG under traffic pattern 4 is shown in Table VIII. The following observations can be drawn from traffic pattern 4's results:

- Flow 1's traffic demand cannot be fully accommodated in G_2, since its contending flow, flow 3 has an increased its demand. Note that flow 1's assignment in G_2 is surely still above its fair share.
- The tradeoff between meeting traffic demand and maintaining fairness thereby reducing potential congestion is addressed by DAF. The time slots of G_2 are not all utilized, although flow 1 still has traffic demand to meet. This is because many flows have been allocated time slots beyond their fair shares.

V. CONCLUSIONS

In this paper, a demand-aware fair resource allocation algorithm is proposed to allocate time slots in TDMA-based multi-hop wireless networks with the objective of meeting both traffic demand as much as possible while enforcing a predefined fairness level. The algorithm projects the new network flow arrivals onto multiple Maximal Common Slot Set based Flow Contention Graphs and then executes an intra-graph resource allocation algorithm over contention graphs in a carefully selected order. The execution order strives to reduce the flow contention as much as possible before the intra-graph algorithm starts allocating time slots over each particular graph. The proposed intra-graph algorithm is proven to minimize, over a maximal clique, a generic cost function value incurred when inadequately serving traffic demand and serving beyond fair shares. Numerical experiments are conducted to demonstrate the effectiveness of the proposed algorithm. The proposed algorithm is shown to well meet the traffic demand and achieve the predefined fairness.

TABLE VIII
SLOT ASSIGNMENT DETAILS UNDER TRAFFIC PATTERN 4

Flow ID	FCG #1	FCG #2	FCG #3
#1		42	
#2	18	2	40
#3	18	42	
#4		50	
#5		20	40
#6		20	40

REFERENCES

[1] L. B. Jiang, and S. C. Liew, *Proportional fairness in wireless LANs and ad hoc networks*, Proceedings of IEEE WCNC 3, volume 3, pp. 1551–1556, March 2005.

[2] X. L. H., and B. Bensaou, *On Max-min Fairness Bandwidth Allocation and Scheduling in Wireless Ad Hoc Networks: Analytical Framework and Implementation*, Proceedings of ACM MobiHoc, pp. 221–231, 2001.

[3] L. Chen, S. H. Low, and J. C. Doyle, *Joint congestion control and media access control design for ad hoc wireless networks*, Proceedings of IEEE INFOCOM 3, volume 3, pp. 2212–2222, March 2005.

[4] I. Rhee, A. Warrier, and J. Min: DRAND, *Distributed Randomized TDMA Scheduling for Wireless Ad-hoc Networks*, Proceedings of ACM MobiHoc, pp. 190–201, 2006.

[5] H. Zhai, *QoS Support over UWB Mesh Networks*, Proceedings of IEEE WCNC, pp. 2283–2288, March 2008.

[6] *Standard ECMA-368 High Rate Ultra Wideband PHY and MAC Standard*, http://www.ecma-international.org/publications/standards/Ecma-368.htm, December 2008.

[7] *Standard ECMA-387 High Rate 60 GHz PHY, MAC and HDMI PAL*, http://www.ecma-international.org/publications/standards/Ecma-387.htm, December 2008.

[8] A. M. Chou, and V. O. K. Li, *Slot Allocation Strategies for TDMA Protocols in Multihop Packet Radio Networks*, Proceedings of IEEE INFOCOM 2, volume 2, pp. 710–716, May 1992.

[9] J. Grnkvist, *Traffic Controlled Spatial Reuse TDMA in Multi-hop Radio Networks*, Proceedings of IEEE PIMRC 3, volume 3, pp. 1203–1207, September 1998.

[10] P. Bjrklund, Vrbrand, P., and Yuan, D., *Resource Optimization of Spatial TDMA in Ad Hoc Radio Networks: A Column Generation Approach*, Proceedings of IEEE INFOCOM 2, volume 2, pp. 818–824, April 2003.

[11] H. L. Chao, and W. Liao, *Credit-Based Slot Allocation for Multimedia Mobile Ad Hoc Networks*, IEEE Journal on Selected Areas in Communications 21(10), volume 21, pp. 1642–1651, December 2003.

[12] T. Nandagopal, T. E. Kim, X. Gao, and V. Bharghavan, *Achieving MAC layer fairness in wireless packet networks*, Proc. ACM MobiCom, pp. 87–98, September 2000.