Distributed Scheduling for Multi-Hop Wireless Networks

Nachiket Sahasrabudhe
Centre for Electronics Design and Technology
Indian Institute of Science, Bangalore
Email: snachiket@cedt.iisc.ernet.in

Joy Kuri
Centre for Electronics Design and Technology
Indian Institute of Science, Bangalore
Email:kuri@cedt.iisc.ernet.in

***Abstract*—We study the performance of greedy scheduling in multihop wireless networks where the objective is aggregate utility maximization. Following standard approaches, we consider the dual of the original optimization problem. Optimal scheduling requires selecting independent sets of maximum aggregate price, but this problem is known to be NP-hard. We propose and evaluate a simple greedy heuristic. We suggest how the greedy heuristic can be implemented in a distributed manner. We evaluate an analytical bound in detail, for the special case of a line graph and also provide a loose bound on the greedy heuristic for the case of an arbitrary graph.**

I. Introduction and Related Work

Consider a multihop wireless packet network. We want to "pack" as much traffic into the network as it can handle. Quantitatively: Each source is equipped with a concave increasing utility function, and we want to maximize aggregate utility. We note that the utility-based approach is a standard way of balancing high aggregate throughput and fairness [1]–[4], [11], [17].

In a wireless network, the situation is complicated by the fact that only some subsets of links can be active simultaneously. The notion of "independent sets" is used to capture this inherent constraint [1], [2], [16]. A *maximal* independent set is a subset of links that is maximal in the sense that even one more link cannot be added to the set without violating interference constraints. Maximal independent sets correspond to maximal elements in a partial order.

Research into scheduling, routing and congestion control is several decades old, but has seen a lot of activity recently, following the seminal paper of Tassiulas and Ephremides. The literature can be classified into two broad groups. In the first group, traffic arrives into the network according to specified random processes that cannot be controlled, and the objective is to find the "capacity region", which is the largest set of arrival vectors for which a scheduling and routing policy can be found ensuring stable operation. [5]–[10]

In the second group, each source has an infinite backlog of data to send, is equipped with a utility function and the objective is to maximize aggregate utility. Following the important paper of Kelly, Maulloo and Tan [17], researchers have addressed the issue of obtaining distributed controls to achieve the objective. The basic idea is that the network will provide congestion signals to the sources (in the form of "prices"), and the sources will modify their data rates accordingly. The original problem is shown to decompose into several subproblems, viz., congestion control, routing and scheduling, and the objective is to find distributed solutions to each [1]–[4], [11].

We consider a problem that belongs to the second group above.

II. System Model and Mathematical Formulation

A. Primal optimization problem

Our problem can be stated precisely as follows. We represent a network as a directed graph $\mathcal{G} = (\mathcal{N}, \mathcal{L})$, where $\mathcal{N}$ represents the set of nodes in the network and $\mathcal{L}$ represents the set of wireless links in the network. The total number of nodes in the network is denoted by $N = |\mathcal{N}|$ and the total number of links by $L = |\mathcal{L}|$. A wireless link $(i, j) \in \mathcal{L}$ if nodes i and j are within transmission range of each other, i.e. direct communication between node i and node j is possible. As in [1], we assume that if a link $(i, j) \in \mathcal{L}$, then so does the link (j, i). For simplicity of mathematical expressions, we assume that all links support a bit rate C when active, though the results in the paper can be extended easily to the case when this is not true. Let $\mathcal{F}$ denote the set of all end-to-end multihop flows present in the network. For every flow $f \in \mathcal{F}$, $s(f)$ and $d(f)$ represent the source and destination nodes, respectively. All the source nodes are assumed to have an infinite backlog. The data rate associated with the flow f is represented by x_f and y_{fl} denotes the part of the flow f that is carried by the link l. Let $\mathbf{y}_f = (y_{fl})_{l \in \mathcal{L}}$ be a vector representing the part of flow f carried by each link in the network. $\mathbf{u}_f$ represents a N-sized column vector such that $u_f(s(f)) = x_f$ and $u_f(d(f)) = -x_f$ and all other entries are zero.

Interference Model: In this paper we use the IEEE 802.11 based interference model, for modeling an inter-link interference. Here we reproduce some definitions from [12] in order to define the interference model.

Definition 1: Let $d_S(x, y)$ denote the shortest distance (in terms of number of links) between nodes $x, y \in \mathcal{N}$. Define a function $d : (\mathcal{L}, \mathcal{L}) \to \mathbb{N}$ as follows: For $l_u = (u_1, u_2)$, $l_v = (v_1, v_2) \in \mathcal{L}$, let

$$d(l_u, l_v) = \min_{i,j \in \{1,2\}} d_S(u_i, v_j).$$

In the K-hop link interference model, we assume that any two links l_1 and l_2 for which $d(l_1, l_2) < K$, will interfere

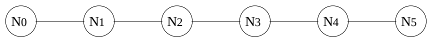

Fig. 1. An example of a line graph. S and D represent the source and destination nodes respectively.

with each other and hence cannot be active simultaneously. Throughout this paper, we use the IEEE 802.11 based interference model which is a 2-hop interference model [12]. A maximal independent set of links is a set, no two links of which interfere with each other under the given interference model and no other link can be added to this set without violating an interference constraint. As an example consider the line graph shown in the Figure 1. Here, links (N_0, N_1) and (N_3, N_4) do not interfere with each other. If we consider any other link in this graph, we notice that it will interfere either with one of them or both of them. Thus, the set formed by links (N_0, N_1) and (N_3, N_4) can be said to be a maximal independent set of links under the 2-hop interference model. A maximal independent set of links I, is represented by a column vector $\mathbf{r}_I$ of size L. If a link $l \in I$, $r_I(l)$ is C; else it is 0. We represent the collection of all the maximal independent sets by the matrix $\mathbf{M_I}$, columns of which are $\mathbf{r}_I$s. As no two maximal independent sets can be activated simultaneously, each of them will get scheduled for some fraction of time. If there are J maximal independent sets present in the network, we represent a schedule associated with them by a J sized column vector $\mathbf{a}$, where the i^{th} entry of vector $\mathbf{a}$ represents the fraction of time an independent set representing the i^{th} column of matrix $\mathbf{M_I}$ is active. We associate a utility function $U(x_f)$ with every end-to-end flow $f \in \mathcal{F}$. $U(x_f)$ is assumed to be a strictly concave, twice differentiable, increasing function with $U(0) = 0$. Let $\mathbf{A}$ be the $N \times L$ node-link incident matrix. Then, the formal representation of our problem is as follows:

$$\max_{x_f \geq 0, \mathbf{a} \geq \mathbf{0}} \sum_{f \in \mathcal{F}} U(x_f) \tag{1}$$

$$\begin{aligned} \text{Subject to: } \mathbf{A}\mathbf{y}_f &= \mathbf{u}_f, \forall f \in \mathcal{F} \\ \sum_{f \in \mathcal{F}} \mathbf{y}_f &\leq \mathbf{M}_I \mathbf{a} \qquad (2) \\ \sum_{k=1}^{K} a_k &= 1 \qquad (3) \end{aligned}$$

Here, the first constraint ensures flow conservation at every node in the network. The next constraint, represented by Equation 2 ensures that the aggregate flow on each link is less than the effective capacity of the link. We elaborate more on this point. Let $I_1, I_2, \ldots, I_J$ be the maximal independent sets of links present in the network and let a_k represent the fraction of time an independent set I_k is active. Then the fraction of time λ_l for which link l is active is given by,

$$\lambda_l = \sum_{k | l \in I_k} a_k \tag{4}$$

Thus, the effective capacity of a link l is given by,

$$c_{l eff} = \lambda_l C$$

The third constraint ensures that there are no idle slots in our schedule. This is a convex optimization problem with affine constraints. Thus by applying Slater's condition [18] we can show that our problem has no duality gap.

B. Dual problem

To obtain a solution in a distributed manner, we follow the approach initiated by Kelly and others [2], [17]: that of considering the dual of this problem. As in these papers, we relax the capacity constraints given by Equation (2); the corresponding Lagrange variables behave as link prices, represented by a vector $\mathbf{p}$. Given a vector of link prices, the problem splits into congestion control, routing and scheduling subproblems that can be solved independently. The dual problem associated with the primal problem stated in Equation 1 is as follows:

$$\min_{\mathbf{p} \geq \mathbf{0}} D(\mathbf{p}) \tag{5}$$

where

$$\begin{aligned} D(\mathbf{p}) &= \Big(\max_{x_f \geq 0, \mathbf{a} \geq \mathbf{0}} \sum_{f \in \mathcal{F}} (U(x_f) - \\ & \qquad \mathbf{p}^t(\mathbf{y}_f - \mathbf{M}_I \mathbf{a})) \Big) \qquad (6) \\ \text{subject to: } \sum_{k=1}^{J} a_k &= 1 \qquad (7) \end{aligned}$$

Although it is a wellknown fact, we state that the dual function $D(\mathbf{p})$, is a convex one. Thus, given a price vector $\mathbf{p}$, $D(\mathbf{p})$ is a solution to the relaxed maximization problem. With some algebraic manipulation, we can split this optimization problem into two independent problems and can solve them separately. Thus, we can split the RHS of Equation 6 as two functions of price vector $\mathbf{p}$:

$$D(\mathbf{p}) = D_1(\mathbf{p}) + D_2(\mathbf{p})$$

$D_1(\mathbf{p})$ corresponds to the congestion control and routing problem while the $D_2(\mathbf{p})$ represents the link scheduling problem.

Congestion control and routing subproblem:

$$D_1(\mathbf{p}) = \max_{x_f \geq 0} \Big(\sum_{f \in \mathcal{F}} (U(x_f) - \sum_{i=1}^{L} p_i y_{fi}) \Big)$$

$$\text{subject to: } \mathbf{A}\mathbf{y}_f = \mathbf{u}_f, \forall f \in \mathcal{F}$$

For a given vector of link prices, each source solves the problem of how much traffic to send into the network so as to maximize its net utility. The amount of traffic to send is

$$x_f = U'^{-1}(p(f)) \tag{8}$$

where $p(f)$ is cost of the least cost path between $s(f)$ and $d(f)$ for a given $\mathbf{p}$.

We note that the route chosen by the source corresponds to the *minimum-priced* path(s) between the source and the destination.

Scheduling subproblem:

$$D_2(\mathbf{p}) = \max_{\mathbf{a}\geq\mathbf{0}} (\mathbf{p^t M}_I \mathbf{a}) \tag{9}$$
$$\text{subject to: } \sum_{k=1}^{J} a_k \;=\; 1$$

It can be shown that, for a given vector of link prices, the solution to this problem is to schedule an independent set of maximum *aggregate* price [19]. Thus, an optimal scheduling vector $\mathbf{a}^*(\mathbf{p})$ for a given price vector $\mathbf{p}$ has all its entries zero except the one corresponding to an independent set with maximum aggregate price.

Price updation algorithm: The price of link l is updated while going from the j^{th} to the $(j+1)^{st}$ iteration using the following equation:

$$p_l[j+1] = (p_l[j] + \delta_j(y_l[j] - c_l[j]))^+. \tag{10}$$

Here, $(x)^+ = \max(x, 0)$. $y_l[j]$ represents the aggregate flow on link l in the j^{th} iteration. If link l is part of an independent set that is a solution to the Equation (9) for the given price vector $\mathbf{p}[j]$, then $c_l[j]$ takes the value of C, else it is zero. This update equation follows from the subgradient algorithm that is used, in particular, for non-differentiable functions [19]. δ_j is the step size associated with the subgradient algorithm in the j^{th} iteration. In this paper we use a constant stepsize δ in all iterations.

C. The overall scheme

Summarizing, the overall scheme is follows:

1) Start with a vector of link prices.
2) Obtain the traffic injected into the network by each source (congestion control and routing problem).
3) Obtain the schedule with maximum aggregate price (scheduling problem).
4) Update prices (price updation) and go to Step 2.

It is well-known that when the step sizes in the price updation algorithm are chosen appropriately, the algorithm given above converges to a neighbourhood of the optimal solution of the primal problem [2].

III. Our work

It has been recognized in [12], [16] that the problem in Step 3 is hard, even when a centralized algorithm is used. The problem has been shown to be NP-hard in [12], [16]. Therefore, we relax the requirement of optimality, as in [12], [13]. We consider the natural heuristic idea of choosing an independent set *containing the maximum-priced link*, instead of the independent set of maximum aggregate price. In particular, it is interesting to explore this heuristic because it lends itself to a distributed implementation. In the literature, this heuristic is commonly referred as "'greedy"' and we stick to the same nomenclature.

A. Distributed Nature of Greedy Heuristic

Our main intention behind selecting the greedy heuristic as a scheduling policy is that- it can be implemented in a distributed manner. Here, we present the Greedy Heuristic as in [12] and also present a way in which it can be implemented in a distributed manner.

Definition 2: A "matching" is a set of links no two of which share a common node.

Definition 3: We call a set of edges W, a K-valid matching if for all $l_1, l_2 \in W$ with $l_1 \neq l_2$, we have $d(l_1, l_2) \geq K$.

Greedy Heuristic:

1) $W := \phi$ and $i := 1$.
2) Arrange links of $\mathcal{L}$ in descending order of price, starting with $l_1, l_2, \ldots$.
3) If $W \cup l_i$ is a valid 2-matching, then, $W := W \cup l_i$. $i = i + 1$.
4) Repeat Step 3 for all links in $\mathcal{L}$.

Distributed version of Greedy Heuristic: **An algorithm to be implemented by every node n, in the k^{th} time slot**

1) *Price dissemination over a 2-hop neighbourhood*: For each attached link, broadcast the link price with a hop-count $= 2$, so that the information reaches all nodes within a 2-hop neighbourhood.
2) *Identification of interfering links*: Receive scoped broadcasts from other nodes, and identify all links that are potential interferers with attached links.
3) *Price sorting*: Sort received link prices in descending order.
4) *Link scheduling*: Check if head-node for max-priced link; if yes, schedule the link, else do not schedule any attached link.
5) *Price updation*: Based on links scheduled in the current slot, update prices according to Equation 10.

At time $t = 0$, we associate a fixed index (some integer) with each node. If two conflicting links have maximum price in a given time slot, then a link that corresponds to a head-node with a larger index will get scheduled.

B. Greedy heuristic: Bound on the maximum link price

The greedy heuristic is characterized by the following. Let $p_{max}[j]$ be the price of the *maximum priced link* in the j^{th} iteration. Let $U(x_f)$, a strictly concave, twice differentiable, increasing utility function with $U(0) = 0$, be associated with the flow $f \in \mathcal{F}$. Then the following proposition holds.

Proposition 1: There exist an $0 < M < \infty$ and $j_0 < \infty$, such that $p_{max}[j] \leq M \;\forall j \geq j_0$, under any scheduling policy that schedules an independent set in each time slot.

Proof: Proof is given in the Appendix. ■

As *Proposition 1* holds for any scheduling policy that schedules an independent set, the result holds for the Greedy Heuristic as well.

C. Greedy heuristic: ϵ-subgradient

Compared to the optimal solution, how much do we lose by using the greedy heuristic? To assess this, we need the notion of an ϵ-subgradient.

Definition 4: [19] Given a convex function $D(\mathbf{p}) : \Re^n \rightarrow \Re$ and $\epsilon \geq 0$, a vector $\mathbf{h}(\bar{\mathbf{p}}) \in \Re^n$ is an ϵ-subgradient of $D(\mathbf{p})$ at point $\bar{\mathbf{p}} \in \Re^n$ if $D(\mathbf{p}) \geq D(\bar{\mathbf{p}}) - \epsilon + (\mathbf{p} - \bar{\mathbf{p}})^T \mathbf{h}(\bar{\mathbf{p}}), \forall \mathbf{p} \in \Re^n$.

The following result, from [2], [19], is important for us.

Proposition 2: Suppose at each iteration j an ϵ_j-subgradient is used. If $\epsilon_j \leq \epsilon_0 \ \forall j$ or $\lim_{j \rightarrow \infty} \epsilon_j = \epsilon_0$ and $||h(j)||_2 \leq H \ \forall j$, then the ϵ-subgradient algorithm converges within $\delta H^2/2 + \epsilon_0$ of the optimal value.

Proof: See Theorem 5 from [2] for proof. ■

Now consider $U(x_f) = log(1 + x_f)$. Then under this assumption and K-hop interference model, the following result holds.

Claim 1: Consider a line graph with sufficiently large number of links L, each with capacity C. Then the path-price S satisfies the inequality: $(K+1)/(K+1+C) \leq S \leq (2K+1)/(2K+1+C)$ under any scheduling policy that schedules an independent set in each time slot.

Proof:

Consider a line graph with $L >> K$. We want to know what can be the lower bound on the path price. We notice that when the path-price is minimum then the achieved data-rate is maximum. Also when a path is occupied by a single flow then only that flow can achieve maximum data-rate. Thus in order to find the lower bound on the path-price, we need to assume that a single flow is present on the line graph, between its end nodes.

Let I_{max} be the maximum-sized independent set of links and let I_{min} be the minimum-sized independent set of links. We represent the size of any set I by $|I|$. Then under the K-hop link interference constraints, the following expressions can be obtained:

$$
\begin{aligned}
|I_{max}| &= \left\lceil \frac{L}{K+1} \right\rceil \\
|I_{min}| &= \left\lceil \frac{L}{2K+1} \right\rceil
\end{aligned}
$$

These expressions can be explained as follows. Consider a line graph with L number of links in it, L being sufficiently larger than K. We see that the maximal independent set with minimum size under the K-hop interference model, corresponds to the set of links : $\{K+1, 3K+2, 5K+3, \ldots (2n-1)K+n, \ldots\}$. So we see that out of any $2K+1$ consecutive links, at least one link belongs to I_{min} in order to maintain its maximality. This leads to the conclusion that the size of I_{min} should be $\lceil L/(2K+1) \rceil$. A similar argument works for the size of I_{max}.

Let $I(t)$ be an independent set scheduled at time interval t. Now according to Equation 10, the price updation happens as follows:

If a link l is scheduled in slot k then,

$$
\begin{aligned}
p_{k+1}(l) - p_k(l) &= \delta(U'^{-1}(S_k) - C) \text{ else,} \\
p_{k+1}(l) - p_k(l) &= \delta(U'^{-1}(S_k))
\end{aligned}
$$

δ being a constant step-size and S_k being the path-price for the k^{th} iteration.

Adding the price update equations for all links we get,

$$S_{k+1} = S_k + \delta(L.U'^{-1}(S_k) - |I(t)|C)$$

For $U(x) = log(1+x)$ we have $U'^{-1}(S_k) = (\frac{1}{S_k} - 1)$.

Now we move to the continuous time domain to have a differential equation governing the path-price $S(t)$ as:

$$\frac{dS}{dt} = L\left(\frac{1}{S} - 1\right) - |I(t)|C \tag{11}$$

We see from the differential equation above that if we keep on scheduling the same independent set I at each time instant t, then $\frac{dS}{dt}$ will become zero when S takes value $\frac{L}{L+|I|C}$. Thus $S = \frac{L}{L+|I|C}$ is the *equilibrium point* of the differential equation. As this value depends upon the cardinality of set I, we can see that larger the size of set I, smaller will be the value of S that is associated with the equilibrium point. In the actual network, though the same independent set will not get scheduled at each time instant, in order to find a lower bound on S we consider the case when the set I_{max} is scheduled at all time instants. Thus by substituting I by I_{max} in Equation 11 we get,

$$
\begin{aligned}
S_{min} &= \frac{L}{L + \lceil \frac{L}{K+1} \rceil C} \\
&\approx \frac{K+1}{K+1+C}
\end{aligned} \tag{12}
$$

where S_{min} represents the lower bound on the path-price S. Similarly we can find an upper bound on S by substituting $|I_{min}|$ in place of I in Equation 11 and equating it to zero. Thus we find the upper bound S_{max} on S as:

$$S_{max} \approx \frac{2K+1}{2K+1+C} \tag{13}$$

This completes the proof. ■

As in a line graph, $p_{max} \leq S$, from the above claim we can conclude that $p_{max} \leq (2K+1)/(2K+C)$, in the case of a line graph.

Proposition 3: Under the same set of assumptions for utility function, interference model and number of end-to-end flows, as for *Claim 1*, in an arbitrary network graph $S \leq 1$ under any scheduling policy that schedules an independent set in each time slot

Proof: Among the multiple routes that will be present between a source-destination pair, let us select an arbitrary route r and let S_r be the path-price associated with this route. We focus on the derivative of this path-price in each time instant, irrespective of whether the source node sends data over it or not. Then at any time instant we see that the derivative of S_r can be one of the four possible functions of S_r.

$$\frac{dS_r}{dt} = \begin{cases} -|I_r(t)|C & \text{case } 1 \\ L_r\left(\frac{1}{S_r} - 1\right) & \text{case } 2 \\ L_r\left(\frac{1}{S_r} - 1\right) - |I_r(t)|C & \text{case } 3 \\ 0 & \text{case } 4 \end{cases} \tag{14}$$

Here we enumerate the details of all four cases mentioned above:

1) when $|I_r(t)|$ number of links from the route r get scheduled and source routes no data over r.
2) when data is routed over r and no link from r is scheduled
3) when both routing and scheduling activities are carried out on route r
4) when neither routing nor scheduling is carried out along the route r.

From the above equation, we see that when only routing activity is carried out over the route r, then S_r keeps increasing at the maximum possible rate of change. Thus under this scenario (i.e. when only routing activity is carried out over the route r) the maximum possible value that S_r can take is 1 as $S_r = 1$ is the equilibrium point for the differential equation in case 2.

Now consider the situation when S_r keeps decreasing under case 1 or case 3. Let r be the first route among all routes which reaches the value $\frac{K+1}{K+1+C}$. As it is the first route to reach that value, it is the minimum path-priced route when it reaches this particular value. Thus, data will be routed over the route r and hence under either case 2 or case 3, dS_r/dt will start increasing (each route is a line graph and *Claim* 1 provides a lower bound for the path-price of a line graph). Thus S_r cannot reach zero, the case in which path-price would increase unboundedly in the next time instant. Thus S_r remains bounded by 1. ■

As S_r is bounded above by 1 for all routes r, and as $p_l \leq S_r$ for all links l along route r, we can say that p_{max} is bounded above by 1 under any scheduling policy. The last proposition can easily be extended for the case when multiple flows are present in the network and hence the bound on p_{max} continues to hold even in that case.

In the result that follows we will be using the following definitions from [12].

Definition 5: The K-hop interference degree of a link $l \in \mathcal{L}$, denoted by $d_K(l)$, is a size of the maximum sized independent set $I_{max}(l)$, such that $d(l_i, l) < K \; \forall l_i \in I_{max}(l)$.

Definition 6: The K-hop interference degree of a graph $G(\mathcal{N}, \mathcal{L})$, denoted by $d_K(G)$, is defined as

$$d_K(G) = \max_{l \in \mathcal{L}} d_K(l).$$

Now for the greedy heuristic, let $S_{opt}[j]$ and $S_{grd}[j]$ represent the aggregate prices of the independent sets corresponding to the optimal schedule and greedy schedule, respectively, in the j^{th} iteration. In [12], the authors prove that $S_{grd}[j] \geq S_{opt}[j]/d_K(G)$. Based on this fact we have the following proposition.

Proposition 4: The greedy heuristic leads to an $LC(d_K(G) - 1)/d_K(G)$-subgradient under the K-hop interference model.

Proof: Proof is provided in the Appendix . ■

IV. Numerical evaluation

In this section we discuss the results that we obtained after implementing the greedy heuristic on four networks. We do not implement the distributed version of the greedy heuristic. Figure 2 [14], Figure 3, Figure 4 [3], Figure 5 [2] and Figure 6 [15] show the topologies of five networks G_1,G_2,G_3,G_4 and G_5, respectively. All the links in all the networks are assumed to have the capacity of 10 Mbps. We start by assigning unit price to all the links. The plots in the Figures 7, 8 and 11 compare the performance of the greedy algorithm with the optimal scheduling algorithm for the respective networks, when multiple flows are present in the network. From the plots in Figure 7, one sees that the aggregate utility in the case of greedy scheduling varies from that of optimal scheduling by less that 2% and 15% for two networks G_1 and G_2, respectively. In the case of G_3, G_4 and G_5, we observe that the utility achieved under the greedy schedule is almost the same as that obtained under the optimal schedule. The step size δ is taken to be 0.001. Also, Figure 9, Figure 10 and Figure 12 show that p_{max} is bounded in all the five cases.

V. Conclusion and Future Work

The scheduling problem is known to be a bottleneck in the cross-layer optimization approach. We relax the optimality requirement, and propose a natural greedy heuristic. The loss in performance due to the greedy heuristic is shown to be bounded. The greedy heuristic is of interest because it is amenable to distributed implementation. Though the upper bound on p_{max} was shown to be 1, that result was obtained for any scheduling policy which schedules an independent set. The actual bound on p_{max} under greedy scheduling policy might be much less than 1, as suggested by the numerical results. In our ongoing work, we are trying to find this upper bound on p_{max} under greedy scheduling.

References

[1] F.L. Presti, "Joint Congestion Control, Routing and Media Access Control Optimization via Dual Decomposition for Ad Hoc Wireless Networks," In the Proc. of the 8-th International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM 2005) , Montreal, Canada, October 2005.

[2] L. Chen,S. Low,J. Doyle ,"Joint Congestion Control and Media Access Control Design for Ad Hoc Wireless Networks"in Proc. of *IEEE Infocom 2005*,March 2005.

[3] L. Chen, S.H. Low, M. Chiang, J.C. Doyle, "Cross-layer Congestion Control,Routing and Scheduling Design in Ad Hoc Wireless Networks," in *IEEE Infocom 2006*.

[4] X. Lin,N.B. Shroff,R. Srikant "A Tutorial on Cross-Layer Optimization in Wireless Networks"*IEEE Journal on Selected Areas in Communication*,vol.24 no. 8,August 2006.

[5] L. Tassiulas and A. Ephremides, "Stability Properties of Constrained Queuing Systems and Scheduling Policies for Maximum Throughput in Multihop Radio Networks".*IEEE Transactions on Automatic Control*, vol.37, no.12, pp. 1936-1948, December 1992

[6] A. Stolyar,"Maximizing queuing network utility subject to stability: Greedy primal-dual algorithm" *Queuing Systems* vol.50, no.4,pp.401-457, 2005.

[7] R. Buche and H.J. Kushner,"Control of mobile communication systems with time-varying channels via stability methods"*IEEE Transactions on Automatic Control*, 2004.

[8] S. Shakkottai and A. Stolyar,"Scheduling of a shared time varying channel:The exponential rule stability". in *INFORMS Applied Probability Conference* New York, July 2001.

[9] S. Shakkottai and A. Stolyar,,"Scheduling for multiple flows sharing a time varying channel: The exponential rule" *Transactions of AMS*, 2001, a volume in memory of F. Karpelevich.

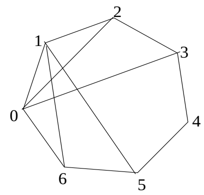

Fig. 2. Network G_1. Flow is present between following pairs of nodes:$(0, 4)$, $(5, 1)$, $(3, 6)$.

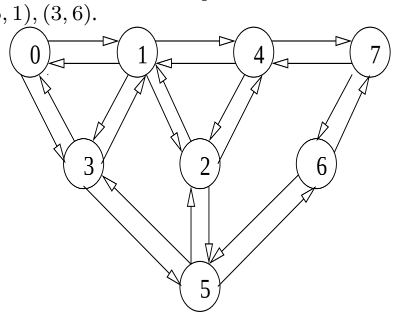

Fig. 3. Network G_2. Flow is present between following pairs of nodes:$(0, 1)$, $(5, 4)$, $(6, 1)$.

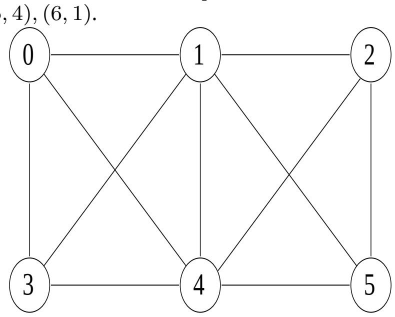

Fig. 4. Network G_3. Flow is present between following pairs of nodes:$(0, 5)$, $(3, 2)$.

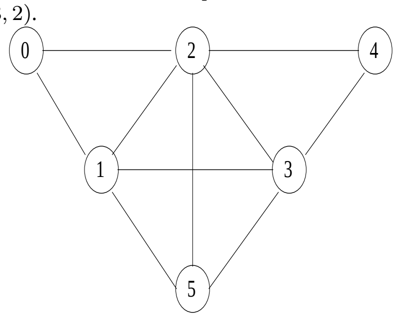

Fig. 5. Network G_4. Flow is present between following pairs of nodes:$(0, 3)$, $(5, 4)$.

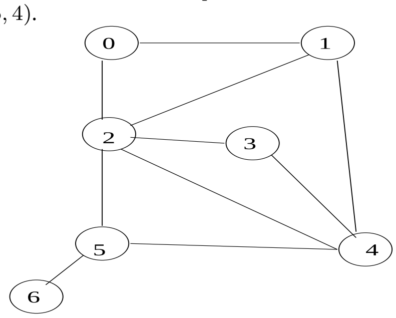

Fig. 6. Network G_5. Flow is present between following pairs of nodes:$(0, 4)$, $(6, 1)$.

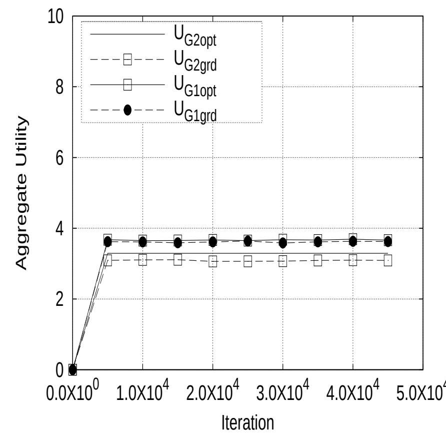

Fig. 7. Utility function under optimal and greedy schedules for networks G_1 and G_2.

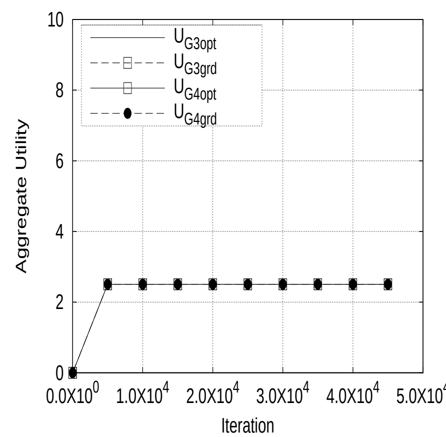

Fig. 8. Utility function under optimal and greedy schedules for networks G_3 and G_4.

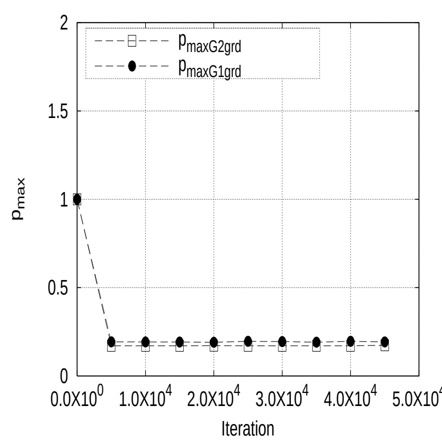

Fig. 9. p_{max} remains bounded in networks G_1 and G_2

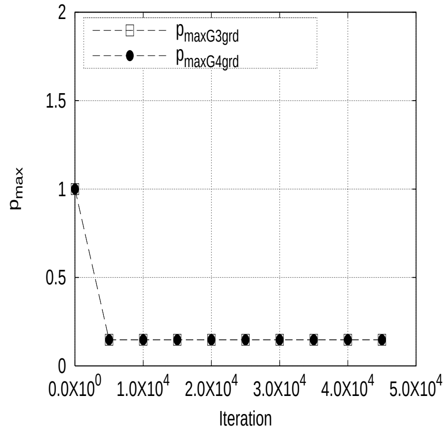

Fig. 10. p_{max} remains bounded in networks G_3 and G_4

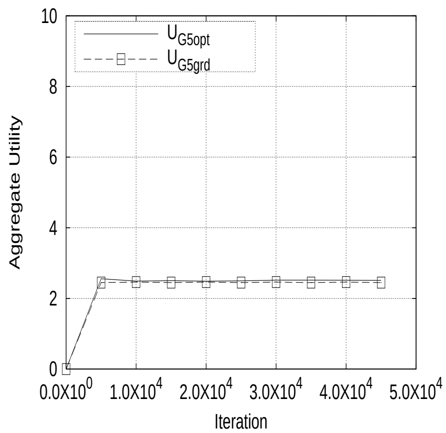

Fig. 11. Utility function under optimal and greedy schedules for network G_5

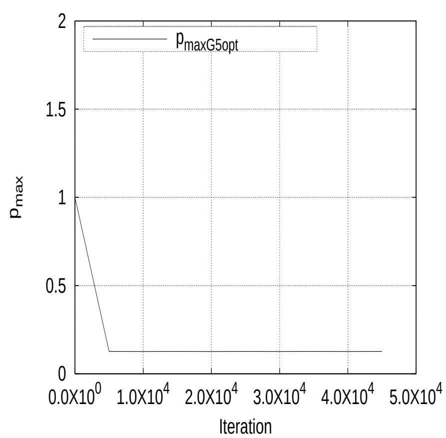

Fig. 12. p_{max} remains bounded in network G_5

[10] A. Andrews, K. Kumaran, K. Ramanan, A. Stolyar, P. Whiting and K. Vijaykumar "Providing quality of service over a shared wireless link",*IEEE Communications Magazine* vol.39,pp.150-153, February 2001.

[11] X. Lin,N.B. Shroff "The Impact of Imperfect Scheduling on Cross-Layer Congestion control in Wireless Networks" in *in IEEE/ACM Transactions on Networking*, vol. 14, no. 2, pp302-315, April 2006

[12] G. Sharma,R.R. Mazumdar,N.B. Shroff "On the Complexity of Scheduling in Wireless Networks"in Proc. of *ACM MobiCom 2006*, September 2006.

[13] A. Gupta, X. Lin and R. Srikant, Low Complexity Distributed Scheduling Algorithms for Wireless Networks, in Proc.*IEEE INFOCOM 2007*.

[14] A. Brzezinski, G. Zussman, E. Modiano "Enabling Distributed Throughput Maximization in Wireless Mesh Networks-A Partitioning Approach" in Proc.*ACM MobiCom 2006*, September 2006.

[15] P. Chaporkar, K. Kar, and S. Sarkar, "Throughput Guarantees Through Maximal Scheduling in Wireless Networks. 43^{rd} Annual Allerton Conference on Communications , Control and Computing, Sept. 2005.

[16] K. Jain, J. Padhye, V.N. Padmanabhan, L. Qiu "Impact of Interference On Multi-hop Wireless Network Performance" in Proc. of *IEEE MobiCom 2003*, September 2003.

[17] F.P. Kelly,A. Maullo, and D. Tan, "Rate control in communication networks:shadow prices,proportional fairness and stability,"*J.Oper.Res.Soc.*,vol. 49,pp. 237-252,1998.

[18] S. Boyd and L. Vandenberghe, "Convex Optimization",Cambridge University Press

[19] D. Bertsekas "Nonlinear Programming", Athena Scientific, 2003.

[20] N. Sahasrabudhe, J. Kuri, "Greedy Scheduling for Wireless Networks", *Technical Report* ,Indian Institute of Science, Bangalore -560012, August 2008, http://www.cedt.iisc.ernet.in/people/kuri

VI. Appendix

A. *Proof of Proposition* 1

Proof:

1) Suppose the price of the max-priced link is not bounded above. Then in a slot if the price of the max-priced link is increased, it must be the case that in the immediately preceding slot, the routing algorithm resulted in some flows (at least one) choosing paths through that link, and for such a flows, the path passing through the max-priced link of the current iteration, was the minimum-priced path.
2) The above can be restated as follows: In the immediately preceding slot, there is a nonempty subset of flows for which the minimum-priced path passed through the max-priced link of this slot.
3) As the iterations proceed, such subsets have to keep appearing repeatedly, so that the price increase in p_{max} can be sustained.
4) For the flows in the subset, the price of the minimum-priced path keeps increasing (because p_{max} keeps increasing).
5) There are a finite number of subsets. Eventually, for any subset of flows, there is no incentive to send traffic, because the price of the minimum-path price is too high.(When S_{min} crosses $U^{'}(0)$ for a particular p_{max}.)
6) So, eventually, all flows will stop sending traffic. In this situation, the increase of price of the max-priced link cannot be sustained. And we have a contradiction. Thus we conclude that, there exists an $0 < M < \infty$, such that $p_{max} \leq M$.

■

B. Proof of Proposition 4

Proof: Let $\mathbf{p_1}, \mathbf{p_2} \in \Re_+^L$. Let $\mathbf{r_{opt}}(\mathbf{p})$ and $\mathbf{r_{grd}}(\mathbf{p})$ represent the independent sets corresponding to an optimal and greedy schedule, respectively for a given price vector $\mathbf{p}$. Let $S_{opt}(\mathbf{p})$ and $S_{grd}(\mathbf{p})$ be the aggregate prices of $\mathbf{r_{opt}}(\mathbf{p})$ and $\mathbf{r_{grd}}(\mathbf{p})$ respectively. $p_{max}(\mathbf{p})$ represents the price of the *max priced link(s)* for the price vector $\mathbf{p}$. We represent the optimal flow rate and optimal routing vector of a flow $f \in \mathcal{F}$, for a given link price vector $\mathbf{p}$, by $x_f(\mathbf{p})$ and $\mathbf{y}_f(\mathbf{p})$, respectively. $\mathbf{y}(\mathbf{p})$ represents a L sized column vector, l^{th} entry of which indicates the aggregate traffic of all the flows carried by the link l for the price vector $\mathbf{p}$, i.e.$\sum_{f\in\mathcal{F}} \mathbf{y}_f(\mathbf{p})$. Then at $\mathbf{p_2}$ we have the following:

$$\begin{aligned}
D(\mathbf{p}_2) &= \sum_{f\in\mathcal{F}} U(x_f(\mathbf{p_2})) - \mathbf{p}_2^T(\mathbf{y}(\mathbf{p_2}) - \mathbf{r_{opt}}(\mathbf{p}_2)) \\
&\geq \sum_{f\in\mathcal{F}} U(x_f(\mathbf{p}_1)) - \mathbf{p}_2^T(\mathbf{y}(\mathbf{p_1}) - \mathbf{r_{grd}}(\mathbf{p}_1)) \\
&= \sum_{f\in\mathcal{F}} U(x_f(\mathbf{p}_1)) - \mathbf{p_1}^T(\mathbf{y}(\mathbf{p_1}) - \mathbf{r_{opt}}(\mathbf{p_1})) + \\
&\quad ((\mathbf{p_2} - \mathbf{p_1})^T(\mathbf{r_{grd}}(\mathbf{p}_1) - \mathbf{y}(\mathbf{p_1})) - \\
&\quad \mathbf{p_1}^T(\mathbf{r_{opt}}((\mathbf{p_1}) - \mathbf{r_{grd}}(\mathbf{p_1})) \\
&= D(\mathbf{p}_1) + (\mathbf{p}_2 - \mathbf{p}_1)^T(\mathbf{r_{grd}}(\mathbf{p_1}) - \mathbf{y}(\mathbf{p_1})) \\
&\quad -\mathbf{p_1}^T(\mathbf{r_{opt}}((\mathbf{p_1}) - \mathbf{r_{grd}}(\mathbf{p_1}))
\end{aligned}$$

We notice that $\mathbf{p_1}^T(\mathbf{r_{opt}}((\mathbf{p_1}) - \mathbf{r_{grd}}(\mathbf{p_1}))$ is always a non-negative quantity and hence for any $\mathbf{p} \in \Re_+^L$, $\mathbf{r}_{grd}(\mathbf{p}) - \mathbf{y}_f(\mathbf{p})$ is an $\epsilon(\mathbf{p})$-subgradient of $D(\mathbf{p})$ at $\mathbf{p}$. From the above set of equations, it is clear that,

$$\begin{aligned}
\epsilon(\mathbf{p}) &= \mathbf{p}^T(\mathbf{r_{opt}}((\mathbf{p}) - \mathbf{r_{grd}}(\mathbf{p})) \\
&= (S_{opt}(\mathbf{p}) - S_{grd}(\mathbf{p})).C \\
\text{But, } S_{grd}(\mathbf{p}) &\geq S_{opt}(\mathbf{p})/d_K(G). \\
&\quad \text{(From Theorem 5 in [12])} \\
\Rightarrow S_{opt}(\mathbf{p}) - S_{grd}(\mathbf{p}) &\leq S_{opt}(\mathbf{p})(d_K(G)-1)/d_K(G). \\
&\leq L.p_{max}(\mathbf{p})(d_K(G)-1)/d_K(G). \\
&\leq LM(d_K(G)-1)/d_K(G). \\
\Rightarrow \epsilon(\mathbf{p}) &\leq LMC(d_K(G)-1)/d_K(G). \ \forall \mathbf{p}
\end{aligned}$$

Thus $\epsilon(\mathbf{p})$ is bounded above for all $\mathbf{p}$ by $LMC(d_K(G) - 1)/d_K(G)$ under greedy scheduling. ∎