SUFFICIENT RATE CONSTRAINTS FOR QoS FLOWS IN AD HOC NETWORKS

by

Rajarshi Gupta, John Musacchio and Jean Walrand

Memorandum No. UCB/ERL M04/42

Fall 2004

SUFFICIENT RATE CONSTRAINTS FOR QoS FLOWS IN AD HOC NETWORKS

by

Rajarshi Gupta, John Musacchio and Jean Walrand

Memorandum No. UCB/ERL M04/42

Fall 2004

ELECTRONICS RESEARCH LABORATORY

College of Engineering
University of California, Berkeley
94720

Sufficient Rate Constraints for QoS Flows in Ad-Hoc Networks

Rajarshi Gupta, John Musacchio and Jean Walrand
University of California, Berkeley
{guptar, musacchj, wlr}@eecs.berkeley.edu

Abstract—**The capacity of an arbitrary ad-hoc network is difficult to estimate due to interference between links. We use a conflict graph that models the interference relationships between links to determine if a set of flow rates can be accommodated. Using the cliques (complete subgraphs) of the conflict graph, we derive constraints that are *sufficient* for a set of flow rates to be feasible. We compare these clique constraints to an alternate set of sufficient constraints that can be easily derived from the rows of the matrix representation of the conflict graph. These two sets of constraints are particularly useful because their construction and verification may be distributed across the nodes of a network. We also extend the ad-hoc network model to incorporate variations in the interference range, and obstructions in the network. We use packet level simulations in OPNET to compare the throughput achieved by a distributed MAC protocol like 802.11b with the capacity predicted by our theoretical constraints.**

Keywords: Graph Theory, Ad-Hoc Networks

I. Introduction

Determining the capacity of an arbitrary ad-hoc network is difficult because neighboring links using the same channel interfere, and the interference relationships between all of the links in a network can be quite complex.

Several researchers interested in the capacity of ad hoc networks have modelled the ad-hoc network using randomized models, and evaluated asymptotic bounds on the capacity. Other work has addressed the question of whether a given flow vector is feasible on a particular ad-hoc network, where "feasible" means that a global scheduler with access to all the information in the network could find a link scheduling policy that would achieve the desired rates. In this work, we are also interested in methods for determining whether a flow vector is feasible, but we are particularly interested in methods that are suitable for distributed control in an ad-hoc network.

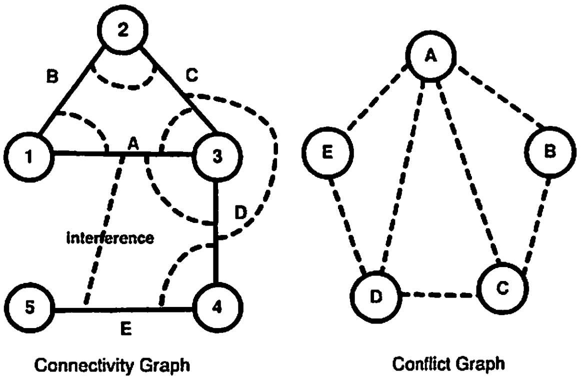
As in [1] and [2], we make use of a "conflict graph" that models the interference relationships between the different links of a network. Every link in the connectivity graph $G = (V, E)$ is represented by a node in the conflict graph $CG = (V^C, E^C)$. Two nodes in CG are connected by an edge if the nodes correspond to links in G that interfere. In Fig. 1, we show an example of a connectivity graph in which the interference between links is marked using dotted lines. The corresponding conflict graph is shown on the right. The authors of [1], [2] show that a set of necessary and sufficient conditions to whether a set of flows is feasible is found by a computationally expensive process of finding all of the independent sets of the conflict graph, and then writing constraints in terms of the independent sets. (We review the details of the "independent set" method, as we call it, in Sec. II-B.)

Because the independent set constraints are computationally expensive and require global information, they are not suitable to be used in a distributed scheme. We therefore look to find a different set of conditions that can be computed in a distributed way and that are at least sufficient, though perhaps not necessary, for a flow vector to be feasible.

One such set of constraints we refer to are the "row" constraints, because they are derived by using the rows of the matrix representation of the conflict graph. While they are sufficient conditions that are relatively easy to compute, they are much more restrictive than is necessary in many cases.

This motivates us to develop a different set of sufficient conditions using the cliques (complete subgraphs) of the conflict graph. While previous work [2] has used cliques to find *necessary* conditions for a set of flow rates to be feasible, our result finds *sufficient* conditions. Our result relies on a recent result in graph theory that bounds the imperfection ratio – the ratio between the chromatic number and the largest clique size – of a class of graphs called unit disk graphs [3]. In this paper, we refer to our sufficient clique constraints as "scaled" clique constraints because they are constructed by modifying the necessary

This work was supported by the Defense Advanced Research Project Agency under Grant N66001-00-C-8062.



Connectivity Graph

Conflict Graph

Fig. 1. Example of a connectivity graph with its conflict graph

clique constraints by a constant scaling factor. Unlike the row constraints, the scaled clique constraints are within a bounded factor of being necessary.

One potential drawback of a scheme based on cliques is that the number of cliques in a graph can grow exponentially with the number of nodes. However, we discuss a technique for identifying approximate cliques that can be implemented distributedly, and that grows only polynomially with the number of nodes. We discuss how the sufficiency of the clique constraints is maintained when using this approximation.

To consider more realistic ad-hoc network scenarios, we augment the network model to incorporate obstructions in the network, and extend the proof of sufficiency to include this model. We also allow for variances in the interference range, and evaluate its effect on the scaling factor.

We validate our theoretical constraint-based approach by simulating random ad-hoc networks, and comparing the achieved rates in the simulation to the theoretical limits predicted by our model. Simulation results show that our proposed constraints correspond well with the performance of the actual network.

The rest of the paper is organized as follows. In Sec. II, we describe the related work in the field. Sections III and IV are used to present the row and clique constraints, and prove sufficiency and bounds on these. In Sec. V, we compare these two sets of constraints. The application of these constraints in distributed algorithms are described in Sec. VI. Simulation results used to validate the constraint-based approach are presented in Sec. VII. Finally we conclude with Conclusion and Future Works sections.

II. Related Work

Many researchers have looked at modelling the capacity of ad-hoc networks. In [4], Gupta and Kumar study how the capacity of an ad-hoc network scales asymptotically with the number of nodes in the network. The authors study the problem under two different models of interference: the *protocol model* and the *physical model.* In the protocol model, a receiver can successfully receive a sender's transmission if the sender is geographically closer, by some margin, than any other node that is actively transmitting. In contrast, the physical model models the transmissions of other nodes as noise, and assumes a receiver successfully receives a sender's transmission if the signal to interference (SIR) power ratio is above a threshold. The authors show that under both the protocol and physical models, the maximum capacity is $\Theta(1/\sqrt{n})$ per node, assuming optimal node placement in a disk of unit area. They also show that under random node placement, and under the protocol interference model, the capacity is $\Theta(1/\sqrt{nlogn})$. Other researchers have extended the work of [4] by considering the effects of node mobility [5], and throughput-delay trade-offs [6].

Li, Blake, et. al. analyze the capacity of specific network topologies, and run packet level simulations of both the specific topologies and of random graphs [7]. With a packet level simulator, the authors show that the maximum throughput achieved in the simulation, where the link schedule is determined by a distributed MAC protocol (802.11), is somewhat less than that predicted by the analytical capacity model which assumes an ideal schedule. For example, for a network consisting of a chain of nodes relaying a single flow, analysis suggests that a rate of up to $\frac{1}{4}$ the channel capacity should be feasible, but with nodes using the 802.11 MAC, a rate of only $\frac{1}{7}$ is achieved.

Luo, Lu, and Bhargavan [1] as well as Jain, Padhye, et. al. [2] study the problem of finding rate constraints on a set of flows in an ad-hoc network, modelling what would be possible if there were a global scheduler. Both works model the interference between links in an ad-hoc network using conflict graphs and find capacity constraints by finding the independent sets of the conflict graph. The concepts of conflict graph and independent sets are discussed in more detail in the next section. In [8], Kodialam and Nandagopal model the routing and scheduling of flows in an ad-hoc network as a graph edge coloring problem, and find necessary and sufficient conditions for the achievability of a rate vector. However, this model only considers conflicts between links incident at the same node, and does not take into account interference due to all other neighboring links.

A. Determining the Conflict Graph

We consider a wireless ad-hoc network with N stations. Each station is equipped with a radio with communications range ρ, and a potentially larger interference radius ω. Our model of interference is similar to that of the protocol model introduced in [4]. A transmission from station i to station j is successful if both of the following conditions are satisfied:

$$d_{ij} < \rho \quad \text{and} \quad d_{kj} > \omega \tag{1}$$

for every other station k that is simultaneously transmitting. Here d_{ij} denotes the distance between i and j.

The connectivity graph G is a directed graph whose vertices correspond to wireless stations and the edges correspond to the wireless links. There is a directed link from vertex i to vertex j iff $d_{ij} < \rho$. The nodes of the conflict graph represent links in the connectivity graph. A pair of nodes, l_{ij} and l_{kl} in the conflict graph are connected by an edge if they cannot have simultaneous transmissions according to the protocol interference model.

To avoid confusion in the rest of the paper, we adopt the convention of using the prefix 'CG' (e.g. CG-node, CG-edge) when referring to the conflict graph. Additionally, a wireless device participating in the ad-hoc protocol is sometimes referred to as an ad-hoc station.

We envision the following technique for computing the conflict graph. We assume each station knows its position using GPS and disseminates its position information to other stations in the neighborhood. Each station then geometrically computes which stations are within an interference radius; we call such stations interfering neighbors.

However, the transmission and interference footprints are not perfect circles in reality, due to factors such as obstacles, multi-path fading, etc. A better scheme would use measurements, as well as any available position information from GPS, to determine which stations interfere. We feel that the design of practical interferer discovery algorithms is a very important problem, and discuss this further as part of Future Works in Sec. IX.

In cases where the network's MAC protocol uses RTS/CTS (Request to Send / Clear to Send) or acknowledgements, such as 802.11 [9], one might choose to use stricter rules for identifying conflicting links than the conditions presented in (1). For a successful transmission to occur, these stricter conditions require that $d_{kj} > \omega$ *and* $d_{ki} > \omega$, for every other station k that is simultaneously transmitting *or receiving*. This is because a receiving station will be sending acknowledgements or will have sent a CTS message that silences other stations in the vicinity.

An alternate approach to constructing a conflict graph, is to compare the geometric centers of each link, i.e. the midpoint of a line segment connecting receiver and transmitter. This is in contrast to comparing distances between the stations themselves [10]. A sufficient condition for a pair of links, l_{ij} and l_{kl} not to conflict would be $|c_{ij} - c_{kl}| > (\omega + \rho)$, where c_{ij} is the position of the geometric center of link l_{ij}. Note that this is sufficient whether one uses the conditions (1) or the stricter rules, for a link transmission to be successful. One may then construct a conflict graph by assuming that pairs of links that do not satisfy these link-center conditions conflict. However, this approximation would lead to some pairs of links being modelled as conflicting, even though they do not conflict in reality, and thus would result in a conservative view of network capacity. This approach is often useful (e.g., [11]) in simplifying the identification of conflicting links.

B. Independent Set Method

One can find a set of necessary and sufficient conditions for a proposed set of flow rates to be feasible, by looking at the independent sets in CG [1], [2]. An independent set in the CG is a set of CG-nodes that have no edges between them. The idea is to identify all of the maximal independent sets of CG, calling them $\mathcal{I}_1$, $\mathcal{I}_2$,... $\mathcal{I}_z$. Then, the independent set constraints say that a set of flow rates $\mathcal{F}_1$, $\mathcal{F}_2$,...$\mathcal{F}_n$ is feasible *iff* there exists λ_1, λ_2,...λ_z such that $\sum_{i=1}^{z} \lambda_i \leq 1$ and

$$\mathcal{F}_j \quad \leq \quad C \sum_{i:j\in\mathcal{I}_i} \lambda_i \quad \forall j \in \{1, ..., n\} \tag{2}$$

where C is the capacity of the channel.

Unfortunately, computing all of the independent sets in a conflict graph is exponential in the number of CG-nodes [2], so using this method in a large CG is not practical. Furthermore, the method requires global information about the entire graph. For these reasons, we look for methods that require less computation, and that can be done with local information at each CG-node.

III. Row Constraints

We describe here one set of sufficient conditions for a set of flow rates to be feasible. We represent a set of flow rates as the vector $\mathcal{F}$ of size $n \times 1$, where n is the number of links in the network and $\mathcal{F}_i$ is simply the flow rate assigned to link i. We also make use of a conflict graph incidence matrix $\mathcal{M}$ defined as follows:

$$\mathcal{M}_{jk} = \begin{cases} 1 & \text{if links } j \text{ and } k \text{ are connected by an edge} \\ 0 & \text{otherwise} \end{cases}$$

Theorem 1: A set of flow rate assignments $\mathcal{F}$ has a feasible schedule if

$$\mathcal{M}\mathcal{F} \le \mathcal{C}. \tag{3}$$

where $\mathcal{C}$ is a $n \times 1$ vector, with all entries equal to the channel capacity C. We refer to expression (3) as the "row constraints" because each row of the conflict graph matrix $\mathcal{M}$ is used to construct a linear inequality on the values of $\mathcal{F}_1$, $\mathcal{F}_2$,... , $\mathcal{F}_n$.

Proof: We assume that the entries of the flow rate vector $\mathcal{F}$ are rational multiples of each other. Let T be the smallest integer such that flow rates $\mathcal{F}_i$ are integer multiples of $\frac{C}{T}$ for all $i \in \{1, ..., n\}$, and let $\mathcal{K}$ be the vector of integers such that $\mathcal{F}_i = \mathcal{K}_i \times \frac{C}{T}$. We will construct a periodic schedule with period T.

We begin by transforming the conflict graph CG by replacing a CG-node i by a clique consisting of $\mathcal{K}_i$ nodes. Let this transformed graph be called CG_f. Observe that a coloring of CG_f implies a schedule for CG. This is because each node in the clique replacing i has a unique color. Further, if CG-nodes i and j were adjacent in CG, all the components of their replacement cliques are adjacent in CG_f, and hence have unique colors. Consequently, by letting each color correspond to a slot, we can ensure that CG-nodes i and j are scheduled for appropriate durations, which are disjoint.

Next we observe that $\mathcal{M}\mathcal{F} \le \mathcal{C}$ implies $\mathcal{M}\mathcal{K} \le \mathcal{T}$, where $\mathcal{T}$ is a $n \times 1$ vector, with all entries equal to T. This in turn implies that each CG_f-node has strictly less than T neighbors. We may now color the graph with the greedy coloring algorithm that follows. Label the CG_f-nodes with indices $\{1, ..., N\}$ and begin coloring the nodes in increasing order of index, by assigning each CG_f-node a color with index in $\{1, ..., T\}$. For CG_f-node i, we assign it the lowest indexed color not already assigned to a neighbor. We can always find such a color with index in $\{1, ..., T\}$, simply because CG_f-node i has less than T neighbors. Thus we have a valid coloring for CG_f, which implies the existence of a feasible schedule for the flow vector $\mathcal{F}$. ■

The row constraints may be evaluated in a localized and distributed manner by evaluating

$$\mathcal{M}^i\mathcal{F}^i \le \mathcal{C}^i, \forall\ i, \tag{4}$$

where $\mathcal{F}^i$ and $\mathcal{C}^i$ are the flow vector and capacity vector representing only the neighbors of link i. Since all the non-zero elements of. the i^{th} row of $\mathcal{M}$ lies in the interference neighborhood of CG-node i, we only need to consider those relevant entries of $\mathcal{F}$ and $\mathcal{C}$. Each CG-node only needs local information from all its neighbors to check the validity of these constraints.

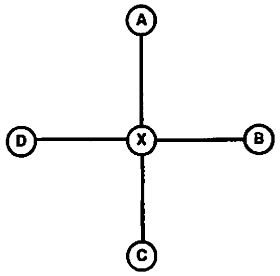


Fig. 2. Row constraints - Star graph

While the row constraints are sufficient, they are not necessary constraints. Indeed, in some examples they can be much more restrictive than is necessary. Fig. 2 shows a conflict graph that illustrates such an example. The row constraints imply that

$$\mathcal{F}_A + \mathcal{F}_B + \mathcal{F}_C + \mathcal{F}_D + \mathcal{F}_X \le C. \tag{5}$$

However, expression (5) is much stronger than what is necessary to be feasible. For example, one could achieve the rates $\mathcal{F}_A = \mathcal{F}_B = \mathcal{F}_C = \mathcal{F}_D = C$ if CG-node X is switched off. Yet, if we used expression (5) as our guide (with $\mathcal{F}_X = 0$), we would have to set all the rates to $C/4$. Thus, in this case, obeying the row constraints would lead us to rates that are only $1/4$ of actually feasible rates. In theory, the row constrained solution could be arbitrarily far from optimal, as in the case if the star had n rays instead of 4.

Because the row constraints could possibly be overly conservative, we are motivated to develop a different set of constraints using cliques. We describe the method in the next section.

IV. Clique Constraints

A. Cliques: Definition and Background

We begin with a few definitions well-known in graph theory. Consider a bi-directional graph with nodes and edges. An *induced subgraph* is a subset of the nodes together with any edges whose endpoints are both in this subset. An induced subgraph that is a complete graph is called a *clique*. A *maximal clique* of a graph is a clique such that it is not contained in any other clique. In the conflict graph in Fig. 1, ABC, ACD and ADE are all maximal cliques.

A clique in a conflict graph is closely related to the capacity of ad-hoc networks. CG-nodes that form a clique are all connected to each other – consequently only one CG-node in a clique may be active at once.

There are two main reasons to utilize cliques in place of independent sets. First, cliques in a CG are inherently local structures and therefore amenable to localized algorithms. Second, as we describe in Section VI-A, and [11], we can approximate the maximal cliques around a link in a computationally simple and distributed fashion.

B. Specifying Clique Constraints

Assume that each CG-node (i.e., link in the connectivity graph) is aware of all the maximal cliques that it belongs to. This information may be described by an incidence matrix $\mathcal{Q}^i$, which is of order $q \times n$. Here, q is the number of maximal cliques that this link i belongs to, and n is the total number of links. In this matrix,

$$\mathcal{Q}^i_{jk} = \begin{cases} 1, & \text{if links } i \text{ and } k \in \text{clique } j \\ 0, & \text{if links } i \text{ or } k \notin \text{clique } j \end{cases}$$

Note that this matrix $\mathcal{Q}^i$ does not include information about the entire network – it covers only the interference neighborhood of link i. The union of the clique matrices across all the links gives the global clique matrix $\mathcal{Q}$.

Since a network must satisfy the capacity constraints for all cliques, we can write the 'clique constraints' in a matrix form. As in Section III, we denote the flow vector $\mathcal{F}$ and the capacity vector $\mathcal{C}$. Hence we have,

$$\mathcal{Q}\,\mathcal{F} \leq \mathcal{C}. \tag{6}$$

Consider the conflict graph as shown in Fig. 1. Let the allocated flow on each CG-node be denoted by $\mathcal{F}_A$, $\mathcal{F}_B$ etc. Then, the clique constraints $\mathcal{Q}\,\mathcal{F} \leq \mathcal{C}$ are:

$$\begin{aligned} \mathcal{F}_A + \mathcal{F}_B + \mathcal{F}_C &\leq C \\ \mathcal{F}_A + \mathcal{F}_C + \mathcal{F}_D &\leq C \\ \mathcal{F}_A + \mathcal{F}_D + \mathcal{F}_E &\leq C. \end{aligned}$$

C. Insufficiency of Clique Constraints

The clique constraints provide a *necessary* condition for a realizable schedule to exist, since there cannot be a feasible schedule over links that form a violated clique constraint. One might hope that these constraints would also be sufficient conditions for a realizable schedule. Unfortunately, that is only true for a special sub-class of graphs called *Perfect Graphs* [12]. Perfect graphs are those whose chromatic number and clique number (size of the largest clique) are equal for all induced subgraphs.

As noted in [2], the simplest example of an imperfect graph where the clique constraints are insufficient is the conflict graph shaped like a pentagon, as seen in Fig. 3. Although the clique constraints suggest a valid flow allocation of $0.5C$ on each link, in reality only $0.4C$ on each link is achievable since at most two out of the five CG-nodes may be active simultaneously.

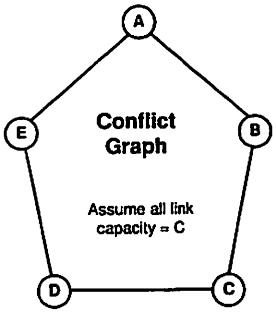


Fig. 3. A pentagon conflict graph

D. Sufficiency using Scaled Clique Constraints

In Sec. III, we proved the sufficiency of row constraints for a schedule to exist. In this section, we demonstrate another *sufficient* condition – based on clique constraints scaled by 0.46.

By modelling a link by its mid-point (Sec. II-A), the resulting CG has an unit disk graph structure. A graph is said to be a *unit disk graph* (UDG) [13] when there is an edge between two nodes if and only if their Euclidean distance is at most 1 (or a constant value ω). We then use properties of UDG to prove the following theorem.

Theorem 2: When the conflict graph is modelled as a UDG, a set of flow rate assignments $\mathcal{F}$ has a feasible schedule if

$$\mathcal{Q}\mathcal{F} \leq \mathcal{C} \times 0.46. \tag{7}$$

Proof: Following in the lines of the proof of Theorem 1, we impose an integer flow rate vector $\mathcal{K}$ by assuming a T-periodic slotted time schedule, where $\mathcal{F}_i = \mathcal{K}_i \times \frac{\mathcal{C}}{T}$. Next, we transform the conflict graph CG by replacing a CG-node i by a clique consisting of $\mathcal{K}_i$ nodes. Let this transformed graph be called CG_f. As we observed in the proof of Theorem 1, a coloring of CG_f implies a schedule for CG.

Denote $\chi(CG_f)$ as the chromatic number of CG_f, and let $\kappa(CG_f)$ be the clique number of the graph. Then it is well known that $\chi(CG_f) \geq \kappa(CG_f)$, since we at least need to use a different color for every member of the largest clique.

We would have a *feasible* schedule for CG if we could color CG_f using at most T colors (i.e., $\chi(CG_f) \leq T$). This would ensure that all CG-nodes in a clique are scheduled for disjoint slots, yet the number of available slots in the periodic schedule is not exceeded.

In [3], the authors define the imperfection ratio $imp(G)$ of a graph as the supremum of the ratio between its weighted chromatic number and its weighted clique number. They further show bounds on $imp(G)$ if the graph belongs to the class of UDG. For a UDG,

$$imp(CG) = \sup_f \frac{\chi(CG_f)}{\kappa(CG_f)} \leq 2.155 \approx \frac{1}{0.46}. \quad (8)$$

We sketch the proof of this result in Sec. IV-F.

To complete our proof, we observe that $\mathcal{QF} \leq \mathcal{C} \times 0.46$ implies that $\mathcal{QK} \leq 0.46\mathcal{T}$. This in turn implies that $\kappa(CG_f) \leq 0.46T$, since the clique number $\kappa(CG_f)$ is simply the largest element of $\mathcal{QK}$. Now applying expression (8), we have $\chi(CG_f) \leq \frac{1}{0.46}\kappa(CG_f) \leq T$. Thus we have a sufficient condition for the existence of a feasible schedule. ■

Consider the scaled clique constraints of expression (7): In addition to being sufficient, they are within a bounded factor of the necessary conditions of expression (6). Consequently, we are assured that flow vectors satisfying the scaled clique constraints are no further than a factor of 0.46 from an optimally feasible flow vector.

It is also worth noting that the clique constraints may be evaluated in a distributed fashion by checking

$$\mathcal{Q}^i\mathcal{F}^i \leq \mathcal{C}^i \times 0.46, \forall\ i, \quad (9)$$

where $\mathcal{F}^i$ and $\mathcal{C}^i$ are the flow vector and capacity vector as known to link i. The only non-zero elements of the clique matrix $\mathcal{Q}^i$ lie in the interference neighborhood of CG-node i, and so the only affected elements of $\mathcal{F}^i$ and $\mathcal{C}^i$ are the corresponding ones.

E. Obstructions in Ad-Hoc Network

Typically, interference regions in a real ad-hoc network are not shaped like perfect disks, due to the presence of obstructions. In such situations, the underlying CG may not be a UDG. Thus, condition (7) may not be sufficient to guarantee the existence of a feasible schedule, as the proof of Theorem 2 depends on the UDG property.

One solution to this problem is to construct what we call a "virtual" conflict graph, where links that lie within an interference range of each other are always modelled as conflicting, even if in reality an obstacle prevents the links from interfering each other. Note that the virtual-CG is a UDG by construction, even if the underlying CG is not. Using the virtual-CG we may state and prove the following lemma.

Lemma 1: A set of flow rate assignments $\mathcal{F}$ has a feasible schedule if

$$\mathcal{Q}_V\mathcal{F} \leq \mathcal{C} \times 0.46. \quad (10)$$

where $\mathcal{Q}_V$ is the clique incidence matrix of the virtual-CG, as defined above. This is valid even if the "true" CG is not UDG.

Proof: Note that the virtual-CG has the same set of CG-nodes (corresponding to links in the connectivity graph) as the true-CG, while the set of CG-edges of the virtual-CG are a superset of the edges of the true-CG. We may define integer flow rates $\mathcal{K}_i$ where $\mathcal{K}_i = \mathcal{F}_i \times \frac{\mathcal{C}}{\mathcal{T}}$, a virtual-$CG_f$, and a true-$CG_f$ in the same way that we did in the proof of Theorem 1. Again the virtual-CG_f and the true-CG_f have the same CG-nodes, but the CG-edges of the former are a superset of the latter's. By Theorem 2, condition (10) implies that there exists a coloring of the virtual-CG_f with at most T colors. However, a valid coloring of the virtual-CG_f is a valid coloring of the true-CG_f as well, because each node in the true-CG_f has a subset of the neighbors it has in the virtual-CG_f. Thus, there exists a valid coloring of the true-CG_f, which in turn implies the existence of a feasible schedule to achieve the flow rate vector $\mathcal{F}$. ■

F. Variance in Interference Range

Often, the interference region may not be shaped like a perfect disk, but be uneven near the edges, due to fading effects. Such an interference region may be modelled as being bounded between two disks. Two CG-nodes cannot interfere if their distance > 1, and two CG-nodes will always interfere if their distance $\leq x \leq 1$. We would like to take this variance into account while employing our constraint-based approach to QoS.

In order to present the proof of our extension, it is useful to sketch the proof of the original imperfection theorem, as given in [3].

Theorem 3: [3] For a UDG,

$$imp(G) \leq \frac{\frac{\sqrt{3}}{2}+1}{\frac{\sqrt{3}}{2}} \approx 2.155. \quad (11)$$

Proof: The proof uses the 'Stripe Lemma' from [13] which implies that in a UDG, stripes of width $\sqrt{3}/2$ are perfect. The technique used in the proof is to cover the UDG with a large number of randomly positioned stripe-graphs. A stripe-graph consists of stripes of width $\sqrt{3}/2$ separated by a distance of 1. Since each stripe is perfect, and their separation is greater than 1, the entire stripe-graph is perfect. Then, the probability p that any node is covered by a particular stripe-graph is given by $p = \frac{\frac{\sqrt{3}}{2}}{\frac{\sqrt{3}}{2}+1}$. Extending a probability argument, the authors show that the imperfection of the graph is bounded above, by the reciprocal, i.e., $imp(G) \leq \frac{1}{p}$. ■

We define the 'connectivity band' of a node in a stripe in the CG. In Fig. 4, it follows from geometry that a node A will necessarily be connected to all other nodes in the stripe which lie within a *connectivity band* of height $1/2$ in either direction (shaded region in the figure). So

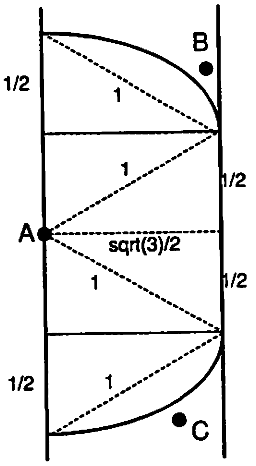


Fig. 4. Connectivity band of a node in a stripe

if two nodes are not connected to each other, their Y co-ordinates must be separated by at least $1/2$.

The following theorem now follows from Theorem 3.

Theorem 4: When the interference range ω in a CG varies between $\frac{1}{2} \leq x \leq \omega \leq 1$, the imperfection ratio is bounded by,

$$imp(CG) \leq \frac{1+z}{z}, \text{ where } z = \sqrt{x^2 - (\frac{1}{2})^2}. \quad (12)$$

Proof: In [13], the authors show that the Stripe Lemma holds if for all nodes located within the stripe, two nodes on either side of a third node, but not adjacent to the third node (e.g. nodes B and C in Fig. 4) are never connected themselves. This is a corollary of having a connectivity band of width $1/2$ on either side of a node – B and C on either side of A are necessarily separated by a distance greater than 1.

When the interference range varies between x and 1, a modification of the Stripe Lemma using triangle geometry ensures that the stripe is always perfect provided $z^2 + (\frac{1}{2})^2 \leq x^2$, where z is the width of the stripe. Consider Figure 4 again: We need to ensure that the band of height $1/2$ on either direction of A is within its range of connectivity, i.e. the diagonal of the connectivity band $\leq x$. At the limit, we get $z = \sqrt{x^2 - (1/2)^2}$. Note that since $x \leq 1$, we have $z \leq \sqrt{3}/2$.

Following the proof technique of Theorem 3 [3], the imperfection ratio in this case is bounded by $imp(CG) \leq \frac{1+z}{z}$. ■

Hence, existence of a feasible schedule is guaranteed provided $Q\mathcal{F} \leq C \times \frac{z}{1+z}$.

Table I tabulates the values of the bound on imperfection ratio as a function of the unevenness in the interference region. As seen, the imperfection can grow as the interference range varies more. When the minimum and the maximum interference radii differs by a factor of 2, the imperfection ratio could become arbitrarily large.

x	z	imp	l/imp
1	0.866	2.155	0.46
0.9	0.75	2.33	0.43
0.8	0.7	2.43	0.41
0.7	0.49	3.04	0.33
0.6	0.33	4	0.25
0.5	0	∞	0

TABLE I
IMPERFECTION RATIO BOUND AS A FUNCTION OF INTERFERENCE UNEVENNESS

This extension can also account for the approximation error introduced by representing a link by its mid-point (Section II-A). As noted in [10], the ideal way to model the interference between two links is to consider the distance between the stations themselves. To account for this, we can model the interference range as lying between two discs of radius ω and $\omega + \rho$.

Once again it is important to realize that the clique constraints applied to a virtual conflict graph akin to the one described in Sec. IV-E will also imply a feasible schedule. In this case, the virtual CG corresponds to the *maximum* interference range (i.e., modelling the variance in the interference range by its upper bound). A feasible schedule in the virtual CG ensures a feasible schedule in the true CG as well. Depending on the value of the imperfection bound as given in Table I, we may choose to use the clique constraints on the virtual CG ($Q_V\mathcal{F} \leq C \times 0.46$) instead of the constraints on the true CG ($Q\mathcal{F} \leq C \times \frac{z}{1+z}$).

V. Row Constraints vs Clique Constraints

We have presented two sets of constraints, *both* of which may be evaluated in a distributed fashion. In this section, we discuss the efficacy of using one set of constraints versus the other.

A. Relation between Row and Clique Constraints

We have seen that the row constraints are sufficient, but may be arbitrarily far from being necessary, as we observed in Sec. III. On the other hand, the scaled clique constraints are not only sufficient, but also within a bounded factor of the necessary constraints (Sec. IV-D).

We present a schematic of the relationship between the various sets of constraints discussed earlier. Each of the constraint sets define a polytope in $\mathbb{R}^n_+$ describing the feasible range of values for the flow vector $\mathcal{F}$. We

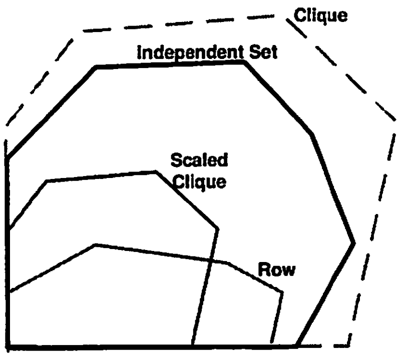


Fig. 5. Polytopes describing feasible regions for the flow rate vector, as defined by the row, clique and independent set constraints

show a two dimensional representation of these regions in Fig. 5.

The independent set (I.S.) polytope corresponds to the necessary and sufficient condition for a feasible schedule to exist, and is therefore the benchmark. The outer clique polytope is derived from necessary conditions and so contains the I.S. polytope. Scaling this by a factor of 0.46 gives us the scaled clique polytope. This corresponds to sufficient conditions for feasibility, and is hence entirely contained within the I.S. polytope. The row constraints also describe sufficient conditions, and so the row polytope lies within the I.S. polytope as well. The row and the clique constrained polytopes certainly overlap (e.g. at the point when the flow vector is 0), but do not contain one another.

Indeed, there are notable cases (typically simple networks, and/or few flows) where the row constraints are less pessimistic than scaled clique constraints. An example is the pentagon CG (Fig. 3), where row constraints say that a flow rate of $C/3$ is possible on each link, while scaled clique constraints say that only a flow rate of $0.46 \times C/2 = 0.23C$ is possible on each link.

B. Using Both Row and Clique Constraints

Because the row constraints may be less restrictive in some circumstances than the clique constraints, we would like to give each station in the network the flexibility to choose to use either the row constraints or the scaled clique constraints, depending on which are less restrictive for the situation. However, we need to be sure that in a network where some stations use clique constraints, and others use row constraints, the union of the constraints being checked across the network constitute sufficient conditions for a flow rate vector to be feasible. We show that this is indeed the case in the lemma that follows.

Lemma 2: Suppose the CG-nodes are partitioned into two disjoint sets $\mathbf{A}$ and $\mathbf{B}$. Let $\mathcal{Q}^{\mathbf{A}}$ denote a reduced clique matrix, containing only the rows of $\mathcal{Q}$ that correspond to cliques containing one or more CG-nodes in $\mathbf{A}$. Similarly let $\mathcal{M}^{\mathbf{B}}$ denote a reduced conflict graph incidence matrix, containing only the rows of $\mathcal{M}$ that describe the neighbors of CG-nodes in $\mathbf{B}$.

The flow rate vector $\mathcal{F}$ is feasible if

$$\begin{aligned} \mathcal{Q}^{\mathbf{A}}\mathcal{F} &\leq 0.46 \times C \\ \mathcal{M}^{\mathbf{B}}\mathcal{F} &\leq C \end{aligned}$$

Proof: As in the proof of Lemma 1 we define integer flow rates $\mathcal{K}_i$ where $\mathcal{K}_i = \mathcal{F}_i \times \frac{C}{T}$, and we define the graph CG_f by replacing each node i of CG with a clique of $\mathcal{K}_i$ "children" nodes. Recall that a coloring of graph CG_f using at most T colors implies the existence of a feasible schedule. Let $CG_f^{\mathbf{A}}$ denote the subgraph of CG_f restricted to CG_f-nodes whose parents are in $\mathbf{A}$. Theorem 2 implies the existence of a coloring for $CG_f^{\mathbf{A}}$. We color the nodes of $CG_f^{\mathbf{A}}$ using such a coloring, and now we seek to color the remaining nodes of CG_f.

We observe that $\mathcal{M}^{\mathbf{B}}\mathcal{F} \leq C$ implies $\mathcal{M}^{\mathbf{B}}\mathcal{K} \leq T$ which in turn implies that each node in $CG_f^{\mathbf{B}}$ has less than T neighbors. It is worth emphasizing that each node in $CG_f^{\mathbf{B}}$ has less than T neighbors in total, including neighbors in $CG_f^{\mathbf{B}}$ and neighbors in $CG_f^{\mathbf{A}}$. We may now color the remaining nodes of CG_f by using the same greedy coloring algorithm as in Theorem 1. Because the remaining nodes each have less than T neighbors, we can always find a color with index in $\{1, ..., T\}$. Thus we have found a coloring for all of CG_f, and therefore there exists a feasible schedule to accommodate flow rate vector $\mathcal{F}$. ■

VI. Distributed Algorithms

We can use the theoretical frameworks of row and clique constraints (Sections III and IV) to propose simple and distributed algorithms for capacity estimation and admission control in ad-hoc networks. We also need to describe the approximation algorithm used to compute maximal cliques.

A. Computing Cliques

For our purposes, we would like to compute maximal cliques in a computationally simple, distributed and localized manner. General algorithms to generate cliques in a graph (e.g. [14], [15], [16]) are centralized in nature. Also, these are exponential algorithms since the number of maximal cliques in a graph (even in a UDG) is exponential. So we aim for a polynomial approximation algorithm.

The essence of the approximation is to use slightly 'super-maximal' cliques. When the number of cliques grows large, several nearby cliques will have a significant intersection, i.e., their membership will differ only at a few nodes. In this case, the approximation algorithm generates the *union* of these as the super-maximal clique. The exact set of maximal cliques generated depends upon the location of the nodes.

By using approximate cliques, the constraints are only strengthened further as multiple individual constraints are replaced by their union. Thus, the sufficiency of the approximated clique constraints implies the sufficiency of the actual set of clique constraints.

The heuristic algorithm described by Gupta and Walrand in [11] distributedly approximates all maximal cliques that a CG-node belongs to. It recognizes two key features about the geographic nature of ad-hoc networks. First, two CG-nodes that are part of a clique must be within an interference radius ω of each other. Second, if a group of CG-nodes form a clique, then the maximum distance between any pair of them must be ω.

The heuristic approximation uses a small disk of diameter ω (i.e. radius = $\omega/2$) to scan a larger disk of radius ω around a CG-node. Each position of the scanning disk generates a clique. The generated set of cliques is reduced to result in the set of maximal cliques around the link.

B. Capacity Estimation

We can use the clique constraints to estimate the capacity of an ad-hoc network, in a localized and distributed way. Assume that each CG-node is aware of all its interference neighbors, and their allocated flows, by means of a link state protocol. Using this localized information, each CG-node can estimate its available capacity Γ^i as

$$\Gamma^i = \min\left\{(\mathcal{C}^i \times 0.46) - \mathcal{Q}^i\mathcal{F}^i\right\}. \qquad (13)$$

Γ^i is the available bandwidth on link i, taking into account flows allocated on i, as well as interference from neighboring links. We consider all maximal cliques that i belongs to, and take the worst case available capacity over all the cliques.

C. Admission Control

A distributed admission control scheme may now be overlaid on the capacity estimation framework. We assume as above that the CG-nodes in the conflict graph are aware of all their interference neighbors, and their allocated flows.

The admission control algorithm is effected when a new flow request $\{src, dest, path, bw\}$ is received by the network. In this paper, we assume that the route for the flow is known ahead of time (we discuss more about the routing issue in Sec. IX). The new flow description is sent out along the path of the flow. Every station that receives the flow request updates its flow vector $\mathcal{F}^i$ with the new flow parameters. And then it recomputes $\Gamma^i = \min\left\{(\mathcal{C}^i \times 0.46) - \mathcal{Q}^i\mathcal{F}^i\right\}$ for all its links. If $\Gamma^i < 0$ for any link i, an 'admission denied' message is generated.

As we showed in Section V, a station may instead choose to use row constraints and check if $\mathcal{C} - \mathcal{M}^i\mathcal{F}^i < 0$ to decide whether 'admission denied' is generated. However, in the mixed constraint case, the constraints need to be evaluated not only along the path of the flow, but also at the neighboring nodes, since a row constraint may be violated at a neighboring node even if the clique constraints are not. This suggests a strong reason for utilizing the clique constraints alone while employing a distributed admission control mechanism.

D. Feasible vs Actual Schedule

In Sec. IV-D, we have shown that a feasible schedule exists when the scaled clique constraints are satisfied. However, it is important to note that the mere existence of a feasible schedule does not imply our being able to find it, let alone impose it on all the ad-hoc stations. Consequently, we do not strive to determine such a schedule, one that we could not hope to implement.

Instead, we compare our theoretical models with a practical MAC protocol – the default 802.11b. We make no changes to the existing 802.11b, and simply use it to check against the capacity limits predicted by our model. As presented in Sec. VII-C, simulations results show that capacity estimation using the scaled clique constraints mirrors the behavior of a real ad-hoc network reasonably well.

VII. Simulation Results

In this section, we present simulation results to test the various ideas discussed earlier in the paper. We perform the simulations using OPNET [17], which implements detailed packet level simulation models of channels, interference, as well as the 802.11 MAC and ad-hoc routing protocols.

A. Row Constraints

First, we evaluate the row constraints presented in Sec. III. The conflict graph evaluated here is shaped like a star, as shown earlier in Fig. 2. All the links A, B, C, and D interfere with link X, but none of these interfere

Comment	Generated	Received
One outer link	5	5
Two outer link	2×5	2×4.5
Three outer link	3×5	3×4.5
Four outer link	4×5	4×4.5
Two outer + center	$2 \times 5 + 5$	$2 \times 4.5 + 0.25$
Four outer + center	$4 \times 5 + 5$	$4 \times 4.5 + 0.01$
Limit outer links	$4 \times 2.5 + 2$	$4 \times 2.5 + 0.85$
Limit outer links	$4 \times 0.5 + 4$	$4 \times 0.5 + 3.5$

TABLE II

SIMULATION RESULTS - STAR CONFLICT GRAPH

with each other. Assuming the effective capacity of the channel to be $5Mbps$, row constraints (at CG-node X) imply that

$$\mathcal{F}_A + \mathcal{F}_B + \mathcal{F}_C + \mathcal{F}_D + \mathcal{F}_X \leq 5.$$

In fact, *each* of the links A, B, C, and D should be able to achieve close to the $5Mbps$ capacity.

The results of this simulation are presented in Table II. All rates are in $Mbps$, and the notation $3 \times p + q$ in the table implies that three of the outer links are all generating/receiving p $Mbps$, while link X at the center gets q $Mbps$.

As seen from the first set of rows of the table, all four of the outer links can indeed achieve $4.5Mbps$ each – so these rates exceed the row constrained rates by a large margin. This corroborates the fact that the row constraints may be overly pessimistic, as discussed in Sec. III.

We note the unfair nature of the sharing of links. The second set of rows show that link X is starved when multiple of its neighbors are transmitting simultaneously. Only by rate limiting the neighboring traffic (third set of rows) can link X hope to achieve its share.

B. Clique Constraints and Scaling

The conflict graph shown in Fig. 2 is perfect, and so the unscaled clique constraints of expression (6) should be sufficient [2]. The clique constraints for this graph look like $\mathcal{F}_A + \mathcal{F}_X \leq 5$, $\mathcal{F}_B + \mathcal{F}_X \leq 5$ etc. Indeed, as seen from the received rates in Table II, the clique constraints are always satisfied.

In order to observe the insufficiency of the unscaled clique constraints, we need to simulate an imperfect CG – the pentagon in Fig. 3. The unscaled clique constraints on this conflict graph suggests that a rate of $2.5Mbps$ should be achievable on each link (assuming a $5Mbps$ channel capacity); although analysis confirms that each link is limited to $2Mbps$ at most (Sec. IV-C).

The simulation results support the analysis quite well. When the traffic on each link is shaped to the predicted limit of $2Mbps$, all links are able to achieve $2Mbps$ each. Increasing the offered rates further, to $2.5Mbps$ on each link, only makes matters worse – the links get saturated and the throughput is reduced to only $1.8Mbps$ per link.

C. Randomly Generated Network

1) Description of OPNET model: Using MATLAB [18] we generate a random ad-hoc network consisting of 50 ad-hoc stations in a $2.5km \times 2.5km$ area. The locations of the stations are then fed into OPNET. Transmission range is set to $500m$, and the interference range is $1km$. These numbers roughly correspond to a battalion of tanks in a battlefield, with powerful radios.

We pick five pairs of stations at random and set up video flows between them (running for 5 minutes). We alter these rates in order to change the load on the network. For these simulations, we use a packet size of 1000 *bytes*, which yields a maximum unhindered MAC throughput of $5.5Mbps$.

We use the DSR routing protocol [19] to determine the routes, and measure the amount of traffic received at each of the receiving stations. In parallel, we feed the routes generated into the theoretical model implemented in MATLAB. Using equation (13), we calculate the minimum spare capacity Γ^i on each link i, to compare against the actual rates received by the flows.

2) Plotting spare capacity against traffic received: The minimum spare capacity Υ in the network is calculated by taking the minimum of Γ^i over all the links. This spare capacity, expressed as a % of the theoretical clique capacity ($0.46 \times 5.5Mbps = 2.3Mbps$ in this case), is used in the x-axis of Fig. 6. In the y-axis we plot the percentage of traffic received on the various flows. The blue (darker) symbols show the average rate received over all the flows, while the green (lighter) symbols show the rate received on the worst of all the flows.

We run simulations involving three, four and five flows. In many of the simulations, the calculated spare capacity on the network is negative, which predicts that some of the flows are likely to be losing traffic. In all cases, the rates and the routes of the flows together determine the Γ^i values for each link, and thereby Υ for the network.

If the theoretical model were entirely accurate, every point to the right of the $\Upsilon = 0$ line would be at 100% and points to the left of the line would be below 100% (i.e. experience drops). While the model is not exact, it seems to correspond surprisingly well. Indeed, for all simulation runs where $\Upsilon > 5\%$, all the flows receive well over 90% of their traffic. And even the worst affected flow always receives over 80% of its traffic.

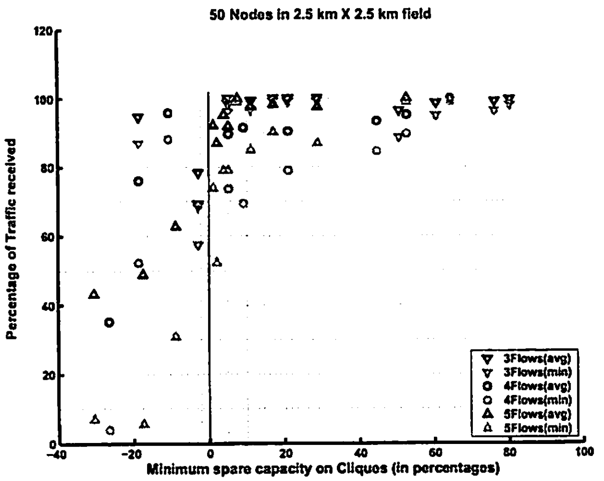


Fig. 6. Fitting clique model to real simulation

3) Plotting predicted limits and traffic received: A different representation of the same simulations are shown in Fig. 7. For this plot, we consider only a subset of the simulations – those that have the same rates on all the flows. By looking at the routes, and interpolating between the various video rates, we can determine the transmitted rate at which $\Upsilon = 0$ for this set of flows. We determine this limit for each of the three, four and five flow cases, and plot these using the dotted vertical lines.

As seen from the figure, the flows receive almost all their traffic until the predicted limit. In each case, the flows experience a sharp loss of quality soon after the theoretical limit is crossed.

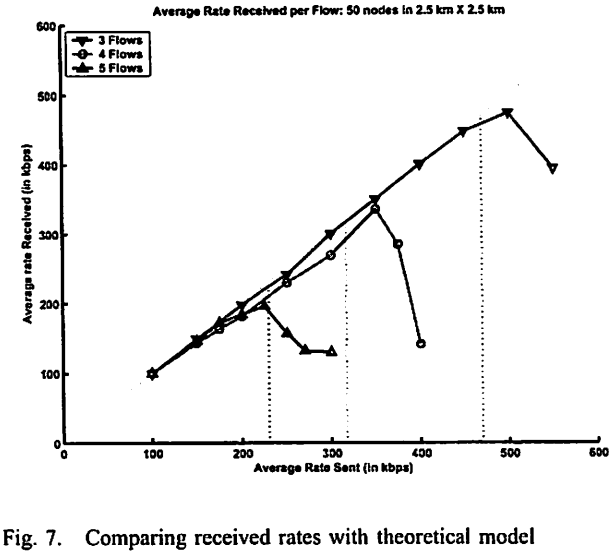


Fig. 7. Comparing received rates with theoretical model

4) MAC inefficiencies and graph imperfection: Our simulation results show that achieved throughput falls off when the send rates exceed the clique constraints scaled by $0.46 = \frac{1}{2.155}$. However, this bound of 2.155 on the imperfection ratio may be significantly larger than the typical case imperfection ratio. In fact, [3] states a conjecture that the imperfection ratio is in fact bounded above by 3/2.

Recall that the imperfection ratio is the ratio between the chromatic number $\chi(CG_f)$ and the largest clique size $\kappa(CG_f)$, maximized across *all* possible flow vectors. Consequently, for the random networks generated for our simulations, it is not possible to analytically determine that imperfection ratio. At best, we can find $\chi(CG_f)/\kappa(CG_f)$ for *particular* flow vectors.

On the other hand, we are aware of several inefficiencies in the distributed nature of the MAC protocol (e.g. [20]). When graph imperfection is significantly less than 2.155, our observation that the throughput falls off soon after the offered load exceeds the scaled clique constraints, must be because MAC inefficiency in addition to any graph imperfection is limiting the achievable rates. In fact, for the simulations illustrated by Fig. 7, the unscaled clique constraints ought to be necessary and sufficient, since we find that $\chi(CG_f)/\kappa(CG_f) = 1$ in this case. Thus, the fact that throughput falls off soon after the scaled clique constraints are exceeded must be entirely due to the inefficiency of the MAC.

The scaling factor of 0.46 is *required*, to account for the worst case imperfection in a *CG*, since we cannot predict the flow vectors ahead of time. In simple networks and/or few flows (e.g. Sec. VII-B), the effect of the graph imperfection is easily visible. In general situations however, the effects of MAC inefficiency, rather than graph imperfection, might be the dominating cause for the gap between a flow vector satisfying the unscaled clique constraints and one that is actually achievable with a 802.11 MAC.

These inefficiencies of the distributed scheduling in 802.11b, we believe, are also responsible for the losses seen by the flows in Fig. 6 even when the model calculates available spare capacity. While the spare capacity indicates the existence of a feasible schedule to achieve the desired flows, the MAC protocol and the routing algorithm are unable to converge to this schedule, resulting in some losses for these flows.

VIII. Conclusions

This paper presents a theoretical model to predict the capacity of an *arbitrary* ad-hoc network, with a given set of desired flows. We model the ad-hoc network, and the underlying interference between links, as a conflict graph. We then propose two sets of constraints – the row constraints and the clique constraints – to determine if a flow vector is feasible on this network. Our first main contribution is to prove that each of the above constraints are sufficient for the existence a feasible schedule. We also discuss the tightness of these constraints under ad-hoc network conditions.

Our second contribution is to expand the model of the network to incorporate variations in the interference range, and consider obstructions (like buildings or hills) in the network. We extend the above proofs of sufficiency to incorporate these changes.

An important motivation to utilize the row and clique constraints is that they are localized in nature and amenable to distributed approximation algorithms. Our third contribution is to propose distributed algorithms for capacity estimation and admission control, that utilize these constraints.

The theoretical framework only indicates the existence of a maximal schedule. We do not attempt to determine this global schedule, since it would be impossible to impose anyway. Consequently we leave the 802.11b protocol untouched, and use the theoretical constraints only to compare with the behavior of a default 802.11b MAC protocol.

IX. Extensions and Future Work

The algorithms assume that every link is aware of all its interference neighbors. However, it is unclear how that information may be reliably obtained, since stations may not be able to understand the transmission of a potentially interfering neighbor. An "interference discovery" phase in the physical layer protocol is one option to address this issue. During this phase, an ad-hoc station transmits at a higher power during a pre-designated time slot, allowing even its interference neighbors to note its presence, and location. Local link-state exchange protocols may also work for well-connected networks, but may not be acceptable for less dense situations. To account for the unreliability of this information, our algorithms should be robust to some degree of incomplete knowledge in this regard.

Also, we do not deal with any routing issues in this paper, and use the DSR algorithm to determine our routes. The effect of interference in an ad-hoc network is quite overwhelming, and this ought to be the primary metric for choosing QoS routes. We are currently working on a distributed QoS routing algorithm for ad-hoc networks that gives due consideration to interference between links.

Acknowledgements

Special thanks to Dr. Stephanie Gerke for her advice and comments. We are also grateful to Bill Hodge for his help in running many of the laborious OPNET simulations.

References

[1] H. Luo, S, Lu, and V. Bhargavan, "A New Model for Packet Scheduling in Multihop Wireless Networks," *Proceedings ACM Mobicom*, 2000, pp. 76-86.

[2] K. Jain, J. Padhye, V. N. Padmanabhan, and L. Qiu, "Impact of Interference on Multi-hop Wireless Network Performance," *Proceedings ACM Mobicom 2003*, San Diego, CA, September 2003.

[3] S. Gerke and C. McDiarmid, "Graph Imperfection I," *Journal of Combinatorial Theory, Series B*, vol. 83, pp. 58-78, 2001.

[4] P. Gupta, and P. R. Kumar, "The Capacity of Wireless Networks," *IEEE Transactions on Information Theory*, vol. 34, no. 5, pp. 910-917, 2000.

[5] M. Grossglauser and D. Tse, "Mobility Increases the Capacity of Ad-hoc Wireless Networks," *IEEE/ACM Transactions on Networking* vol. 10, no. 4, pp. 477-486, August 2002.

[6] A. El Gamal, J. Mammen, B. Prabhakar, and D. Shah, "Throughput-Delay Trade-off in Wireless Networks," *Proceedings IEEE INFOCOM*, Hong Kong, March 2004.

[7] J. Li, C. Blake, D. S. J. De Couto, H. I. Lee, and R. Morris, "Capacity of Ad Hoc Wireless Networks," *Proceedings ACM Mobicom 2001*, Rome, Italy, July 2001.

[8] M. Kodialam, and T. Nandagopal, "Characterizing the Achievable Rates in Multihop Wireless Networks," *Proceedings ACM Mobicom 2003*, San Diego, CA, September 2003.

[9] M. Gast, "802.11 Wireless Networks: The Definitive Guide," O'reilly and Associates, 2002.

[10] R. Negi, and A. Rajeswaran, "Physical Layer Effect on MAC Performance in Ad-Hoc Wireless Networks," *Proceedings Communications, Internet and Information Technology* CIIT, 2003.

[11] R. Gupta, and J. Walrand, "Approximating Maximal Cliques in Ad-Hoc Networks," To appear in *Proceedings PIMRC 2004*, Barcelona, Spain, September 2004. Available at: *www.eecs.berkeley.edu/~guptar/RGpublications.html*

[12] L. Lovasz, "A Characterization of Perfect Graphs," *Journal of Combinatorial Theory, Series B*, vol. 13, pp. 95-98, 1972.

[13] A. Graf, M. Stumpf, and G. Weisenfels, "On Coloring Unit Disk Graphs," *Algorithmica*, vol. 20 (1998), pp. 277-293.

[14] F. Harary, and I. C. Ross, "A Procedure for Clique Detection Using the Group Matrix," *Sociometry*, vol. 20, pp. 205-215, 1957.

[15] J. G. Augustson, and J. Minker, "An Analysis of Some Graph Theoretical Cluster Techniques," *Journal of the ACM (JACM)*, vol. 17, no. 4, pp. 571-588, October 1970.

[16] C. Bron and J. Kerbosch, "Finding All Cliques in an Undirected Graph," *Communications of the ACM*, vol. 16, pp. 575–577, 1973.

[17] OPNET Modeller, OPNET Technologies Inc. *http://www.opnet.com.*

[18] Matlab Simulation Environment, The Mathworks Inc. *http://www.mathworks.com*

[19] D. B. Johnson, and D. A. Maltz, "Dynamic Source Routing in Ad-Hoc Wireless Networks," *Mobile Computing*, T. Imielinski and H. Korth, Eds., pp. 153-181, Kluwer, 1996.

[20] G. Bianchi, "Performance Analysis of the IEEE 802.11 Distributed Coordination Function," *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 3, March 2000.