Algorithmic Aspects of Capacity in Wireless Networks

V.S. Anil Kumar* Madhav V. Marathe† Srinivasan Parthasarathy‡ Aravind Srinivasan§

ABSTRACT

This paper considers two inter-related questions: (i) Given a wireless ad-hoc network and a collection of source-destination pairs $\{(s_i, t_i)\}$, what is the maximum throughput capacity of the network, i.e. the rate at which data from the sources can be transferred to their corresponding destinations in the network? (ii) Can network protocols be designed that jointly route the packets and schedule transmissions at rates close to the maximum throughput capacity? Much of the earlier work focused on random instances and proved analytical lower and upper bounds on the maximum throughput capacity. Here, in contrast, we consider arbitrary wireless networks. Further, we study the algorithmic aspects of the above questions: the goal is to design provably good algorithms for arbitrary instances. We develop analytical performance evaluation models and distributed algorithms for routing and scheduling which incorporate fairness, energy and dilation (path-length) requirements and provide a unified framework for utilizing the network close to its maximum throughput capacity.

Motivated by certain popular wireless protocols used in practice, we also explore "shortest-path like" path selection strategies which maximize the network throughput. The theoretical results naturally suggest an interesting class of congestion aware link-metrics which can be directly *plugged into* several existing routing protocols such as AODV, DSR, etc. We complement the theoretical analysis with extensive simulations. The results indicate that routes obtained using our congestion aware link-metrics consistently yield higher throughput than hop-count based shortest path metrics.

1. INTRODUCTION

Two central questions in communication networks are: what is the throughput capacity of the network, and how can one utilize the network close to the capacity? In other words, given a collection of source-destination pairs $\{(s_i, t_i)\}$, what is the maximum rate (throughput) at which the network can transfer data from the sources to their corresponding destinations? There are many factors effecting this question such as interference, fairness and energy constraints. For a wired network, some of these constraints can be formulated easily as a simple linear program (LP), but this problem is non-trivial to solve in the case of wireless networks due to interference. An influential result on the capacity of wireless networks is that of Gupta and Kumar [9]. They show that, given n identical randomly distributed nodes on a unit square, with each node having an independent randomly chosen destination, the uniform per node throughput capacity in bit-meters/second, is $\Theta(\frac{1}{\sqrt{n \log n}})$: this is sub-linear in the number of nodes, in contrast with the wired setting. Several extensions of the basic result have recently been considered, see Section 9 for additional discussion.

Here, building on the earlier results in [16], [13], and [17], we study the algorithmic aspects of both the inter-related questions posed earlier, namely: (i) What is the maximum throughput capacity of the network? (ii) How to design network protocols that jointly route the packets and schedule transmissions at rates close to the maximum throughput capacity. *In contrast to the results in [9], we focus on (a) arbitrary instances rather than just random node distributions, (b) allow nodes to have varying transmission ranges instead of uniform ranges, (c) consider not only the uniform node-throughput metric but other natural linear functionals of node throughputs and (d) consider linear constraints such as total energy consumed and path length.*

Key technical contributions of our work include a novel definition for congestion which captures the central properties of wireless interference, linear models for a wide class of wireless throughput maximization problems, and the no-

*Basic and Applied Simulation Science (CCS-5) and National Infrastructure and Analysis Center (NISAC), Los Alamos National Laboratory, MS M997, P.O. Box 1663, Los Alamos, NM 87545. The work is supported by the Department of Energy under Contract W-7405-ENG-36. Email: anil@lanl.gov

†Department of Computer Science and Virginia BioInformatics Institute, Virginia Tech, Blacksburg, VA 24061, Email: mmarathe@vbi.vt.edu

‡Department of Computer Science, University of Maryland, College Park, MD 20742. Email: sri@cs.umd.edu.Most of the work was done while visiting Los Alamos National laboratory. Research supported in part by NSF Award CCR-0208005 and NSF ITR Award CNS-0426683.

§Department of Computer Science and Institute for Advanced Computer Studies, University of Maryland, College Park, MD 20742. Email: srin@cs.umd.edu. Research supported in part by NSF Award CCR-0208005 and NSF ITR Award CNS-0426683.

SIGMETRICS'05, June 6–10, 2005, Banff, Alberta, Canada.

tion of rate competitiveness of scheduling algorithms. Our techniques can accommodate a variety of path selection constraints such as low energy, low hop-count, etc. as well as incorporate wireless technologies such as multiple channels and directional antennas. The algorithmic and analytical techniques introduced here are applicable to a variety of interference models including the Protocol Model and could be of independent interest. The main contributions of this paper are as follows:

1. Given an arbitrary wireless network $G = (V, E)$, where each node can have a different transmission range, and a set of k arbitrary source-destination pairs, we describe a polynomial time approximation algorithm that computes the maximum achievable throughput in G to within a constant factor. Thus, this is an algorithmic version of the Gupta and Kumar [9] result, and gives a way of quantifying the capacity for an *arbitrary* wireless network. In contrast the work of [9] focuses on a random wireless network. As noted and emperically shown in [13], the capacity of an arbitrary wireless network could differ substantially from a random network. Our results are based on a linear programming formulation of the problem and crucially uses the properties of wireless interference models. The results hold for various interference models and for variable power levels at individual transceivers. Recently, there has been substantial in the use of directional antennas to improve the overall capacity of ad hoc networks [28]. Our techniques can be easily extended to the case of directional antennas as well. A new stability measure is introduced that provably bounds the optimium capacity to within constant factors.

2. Our approach allows us to incorporate the per flow end-to-end fairness constraints in the throughput maximization problem. The resulting LP formulation can enforce any given (long term) fairness objective; our scheduling algorithm guarantees that the total throughput is within a constant factor of the optimal, for such a fairness constraint. As radio devices become cheaper and smaller, sensor and ad hoc networks are becoming more and more prevalent. This is leading to a new challenge: how to design protocols such that radio devices do not drain battery power very fast. There has been much research on all aspects of routing and scheduling with low energy requirements; see, e.g., [25, 14] and the references therein. Our approach makes it very simple to add energy constraints into the formulation: given such a bound on the energy, we aim to maximize throughput. In fact, we can add any set of requirements that can be modeled by linear constraints. Our LP-formulations differ from the formulations presented in [13, 16] as follows: although [16] presents constant factor approximate LP formulations for a class of scheduling problems, they do not handle wireless interference constraints in their formulations, thus limiting the utility of their approach to most realistic wireless network scenarios. The approach of [13] can model arbitrarily complex interference models; however, they do not discuss how close their computed throughput is to the optimal throughput. Our modeling techniques overcome both these limitations.

3. We also study the empirical performance of a natural class of congestion-aware path selection strategies which arise from our LP formulations. Since these heuristics essentially involve computing shortest paths using congestion-aware link metrics, standard routing protocols such as AODV can be easily modified to incorporate such link metrics. This yields a *unified protocol for MAC, routing and (aspects of) transport layers in a wireless network* which gives good throughput and does not require too many changes from existing routing protocols. The algorithm of [16] involves multiple phases (as does that of [8]), and cannot be easily used this way. As shown in [3, 6, 24, 26], there is a significant interaction between individual layers of a OSI protocol stack and plugging in optimal protocols for each layer does not lead to optimal overall performance [6]. As a result, recent work has focused on designing unified protocols for wireless networks [23]. The unified routing+scheduling protocols developed here overcome this performance loss.

4. We perform extensive simulations to study the performance of our algorithm and the shortest path heuristics. We obtain explicit tradeoff between fairness and total throughput, which shows the increase in throughput with decreasing fairness requirement; while this behavior is completely expected, our results also *quantify* this tradeoff, i.e., by what fraction does the throughput increase for a given loss in fairness. Similarly, we also study such a relationship between the energy consumed and throughput. Thus, our results provide a formal way of quantifying the tradeoffs between different constraints for wireless networks. Our simulations also indicate that routes obtained using our congestion-aware shortest path heuristics have much better throughput in every instance than the routes obtained using the hop-count based shortest path algorithm.

2. ORGANIZATION

The rest of the paper is organized as follows. In Section 3 we introduce some basic definitions of network flows, network and interference models. In Section 4, we present our centralized and distributed algorithms and related stability results for link-flow scheduling. In Section 5, we present our end-to-end flow scheduling algorithms and analyze their performance using the notion of rate-competitiveness. In Section 6 we discuss our LP formulations for the various flow problems and show how to incorporate wireless interference and fairness, energy and dilation constraints. In Section 7, we deal with our congestion-aware path selection strategies. In Section 8, we present the results of our simulations which measure the impact of various constraints and path selection strategies on the network throughput.

3. PRELIMINARIES

This section contains basic definitions and concepts used in the rest of the paper. We consider multi-hop wireless networks. The network is modeled as a directed graph $G = (V, E)$. The nodes of the graph correspond to individual transceivers and a directed edge (u, v) denotes that u can transmit to v directly. Each edge in $G = (V, E)$ has a capacity $c(e)$ bits/sec and denotes the maximum data that can be carried on e in a second. We assume that the system operates synchronously in a time slotted mode. Each time slot is τ seconds long. Thus, at most $\tau c(e)$ bits of information can be transmitted over link e during any time slot.

A schedule $\mathcal{S}$ describes the specific times at which data is moved over the links of the network. In other words, let $X_{e,t}$ be the indicator variable which is defined as follows:

$$X_{e,t} = \begin{cases} 1 & \text{if } e \text{ transmits successfully at time } t \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

A schedule $\mathcal{S}$ is a $0-1$ assignment to the variables $X_{e,t}$, $e \in E$, $0 \leq t$. We will focus on periodic schedules in this paper. A schedule $\mathcal{S}$ is periodic with period T, if $\forall e, t, i$: $X_{e,iT+t} = X_{e,(i+1)T+t}$. In wireless networks, links can be scheduled to transmit in the same time slot only if they do not interfere. The precise notion of interference will be made clear in the following subsection. *For ease of exposition, we will assume that $c(e)$ is 1 and τ is also 1.* All the results generalize directly, when we relax these constraints.

3.1 Network and Interference Models

We assume that vertices V are embedded in the plane $\mathbf{R}^2$. Each vertex (transceiver) u has an associated range denoted by $range(u)$. A necessary (but *not* sufficient) condition for a transceiver v to hear u is that v be within a distance $range(u)$ of u. Specifically, if transmission is not feasible from u to v either because v is outside the range of u or because of other reasons (such as the presence of an obstruction between u and v), then the edge (u, v) is not present in the graph $G = (V, E)$. This is an important consideration for modeling realistic network scenarios such as indoor wireless networks or even outdoor networks in the presence of obstructions. We assume that all antennas are omnidirectional although generalizations to various path loss models and directional antennas is possible and is omitted in this paper.

Since the medium of transmission is wireless, simultaneous transmissions on proximate edges may *interfere* with each other resulting in collisions. Formally, we say that edges $e_1, e_2 \in E$ interfere with each other if edges e_1 and e_2 cannot both transmit successfully during the same time slot. Let $I(e_1)$ denote the set of edges which interfere with edge e_1, i.e., e_1 cannot transmit successfully whenever an edge $e_2 \in I(e_1)$ is transmitting. An *Interference Model* defines the set $I(e_1)$ for each edge e_1 in the network. Several such models have been studied, because of variations in the underlying technology, protocol, etc. We consider three interference models in this paper; in the first two models below, $\Delta \geq 0$ is a constant that is a parameter of the models.

In the **transmitter model (Tx-model)** a transmission from u is successful (i.e. is received correctly by the intended recipient of the transmission) if and only if any other transmitter w is such that, $d(u, w) \geq (1 + \Delta) \cdot (range(u) + range(w))$. This model was introduced by Yi *et al.* [29] to analyze capacity of random ad hoc networks. Throughout this work, we focus on obtaining good performance guarantees for the Tx-model. We also show how to extend our constant factor approximation guarantees to two popular receiver-based models described next. In the **protocol model [9]**, if a node u transmits to node v, this transmission is successfully received by v iff $v \in range(u)$ and any other transmitter w is such that $d(w, v) \geq (1 + \Delta) \cdot d(u, v)$. Finally, consider the **transmitter-receiver Model (Tx-Rx model) [4, 17]:** let $e_1 = (u, v) \in E$ be an edge along which there is a transmission. Let D denotes the network distance (in terms of hop-count) between the edges and nodes in the network. Specifically, for any two edges e_1 and e_2, $D(e_1, e_2)$ is defined as the least hop-count distance between an incident node of e_1 and an incident node of e_2. The transmission along e_1 is successful if and only if any other transmission along an edge $e_2 \in E$ is such that $D(e_1, e_2) > 2$. In all the three models above, a node can either receive a message or transmit a message (and not both) at the same time. Thus for any edge $e = (u, v)$, all other edges which are incident on u or v are also included in the set $I(e)$.

We note that the restriction that all nodes lie on a 2-dimensional (**2D**) plane is only for ease of analysis. The modeling and algorithmic techniques developed here easily extend to **3D** as well, as we show for instance in the protocol model.

3.2 Network Flows

Given a set of flows, with flow i starting at a source node s_i and ending at a destination node t_i, we will be concerned with the *rates* at which data can be sent along these flows. If the rate for flow i is r_i bits per second, then, on an average, in one time slot, r_i bits sent by s_i are received by t_i. In our LP formulations, rates for each flow i will translate to a per edge rate, $x(e, i)$ for edge e: this is the rate at which flow i is routed through edge e. As in [16], we assume an infinitesimally divisible flow model for data transmission - this leads to flow conservation constraints for the data. Let $\vec{x}$ denote the link flow rate vector; this vector associates a link rate $x(e)$ (the total rate of all flows on link e) with each link e. Recall that $X_{e,t}$ is the indicator variable which denotes if there was a succesful transmission on link e during time t. By definition, as time $t' \to \infty$, we have $x(e) = \sum_{t \leq t'} X_{e,t}/t'$.

For the link scheduling and the end-to-end scheduling problems studied here, the central question we are concerned with is that of *stability*: a schedule is said to be stable if every packet incurs a bounded delay, and consequently, all buffers have bounded sizes. A stable rate vector is one for which there exists a stable schedule. In Section 4, given a link-rate vector $\vec{x}$, we will either show that $\vec{x}$ is not stable, or show how to approximate an optimal schedule for vector $\vec{x}$ by a near optimal schedule with a slightly smaller throughput. This will serve as a useful building block for our end-to-end scheduling and throughput maximization techniques in Sections 5 and 6.

Two of the fundamental end-to-end throughput maximization problems we will consider are the maximum multicommodity flow problem (MFP) and maximum concurrent flow problem (MCFP) [1]. In MFP (as defined in the context of wired networks), given a directed graph $G(V, E)$ and a collection of source-destination pairs $\{(s_i, t_i)\}$, the goal is to find a stable end-to-end rate vector for the (s_i, t_i) pairs such that data can be injected into the network by the sources at these rates without violating individual edge capacities; the objective is to maximize the *total* rate of injection for these pairs; packets injected at such a rate can be scheduled in a wired network, since the only constraints are the edge capacities. Note that this formulation does not consider any notion of fairness among the different flow values; MCFP incorporates fairness by requiring that the total rate of injection be maximized subject to the constraint that all the (s_i, t_i) pairs have the same rate. We note that standard LP formulations exist for optimally solving both MFP and MCFP for wired networks. The problems we consider here for wireless networks are variations of these classical multi-commodity flow problems wherein, flow on the links that interfere with each other cannot be scheduled simultaneously. Thus the task of finding optimal multi-commodity flows in wireless networks becomes considerably more complicated.

4. LINK-FLOW SCHEDULING

In this section, we develop a link-flow scheduling algo-

rithm to schedule a set of flows specified on the links of the network. We also develop necessary and sufficient conditions for link flow stability.

4.1 Link-Flow Stability: Necessary Conditions

Recall that for an edge $e = (u, v) \in E$, $I(e)$ denotes the set of edges which interfere with e. Let $I_{\geq(e)}$ be defined as follows.

DEFINITION 1. $I_{\geq}(e) = \{(p, q) : (p, q) \in I(e)$ *and* $d(p, q) \geq d(u, v)\}$.

$I_{\geq}(e)$ is the subset of edges in $I(e)$ which are greater than or equal to e in length. Recall that $X_{e,t}$ is the indicator variable which is 1 iff e transmits successfully during time t. The following claim holds.

CLAIM 2. *In any link schedule,*

$$\forall e \in E, \forall t \qquad X_{e,t} + \sum_{f \in I_{\geq}(e)} X_{f,t} \leq c \tag{2}$$

where c is a fixed constant that depends only on the interference model. In particular, for the Tx-model, the value of c is at most 5.

The intuition behind this claim is as follows. Partition the set of edges in $I_{\geq}(e) \cup \{e\}$ into at most c subsets such that within each subset, each edge interferes with all other edges. Thus, at most one edge can successfully transmit from each subset at any time slot, i.e., only c edges in $I_{\geq}(e) \cup \{e\}$ can simultaneously transmit successfully, as stated in the claim.

Let $\vec{x}$ be a link-flow vector. We define the *congestion* on a link e to be $c(e) = x(e) + \sum_{f \in I_{\geq}(e)} x(f)$. The following lemma imposes a simple necessary condition for link-flow stability.

LEMMA 3. *Let c be the constant in Claim 2. $\vec{x}$ is a stable link-flow only if the following holds:*

$$\forall e \in E, \qquad x(e) + \sum_{f \in I_{\geq}(e)} x(f) \leq c$$

PROOF. Assume that the flow vector $\vec{x}$ is stable, i.e., there exists a stable schedule $\mathcal{S}$ which achieves the link-rates specified by $\vec{x}$. Let $X_{e,t}$ be the transmission indicator variable for this schedule for edge e and time t. As time $t' \to \infty$, we have $\sum_{t \leq t'} X_{e,t}/t' \to x(e)$, since $x(e)$ is the link-rate associated with edge e. The lemma now follows by summing up equation (2) over time slots $[1, \ldots, t']$ and taking the average. □

4.2 Link-Flow Scheduling Algorithm

In this section we present both centralized and distributed algorithms for scheduling a link-flow vector $\vec{x}$. In Section 4.3, we analyze conditions under which this algorithm yields a stable schedule (and hence sufficient conditions for link-flow stability). The algorithm works as follows: time is divided into uniform and contiguous windows or *frames* of length w, where w is a sufficiently large positive integer. (We assume w.l.o.g. that w is such that for all e, $w \cdot x(e)$ is integral.) The algorithm employs a subroutine called *frame-scheduling* which specifies a schedule for each edge e within each frame. This schedule is repeated periodically for every frame to obtain the final schedule. We now present the details of the frame-scheduling algorithm whose pseudo-code is presented in **Algorithm 1**.

Consider a single frame W whose time slots are numbered $\{1, \ldots, w\}$. For each edge e, the subroutine assigns a subset of slots $s(e) \subseteq W$ such that the following hold:

1. $|s(e)| = w \cdot x(e)$, i.e., each edge receives a fraction $x(e)$ of time slots.
2. $\forall f \in I(e)$, $s(f) \cap s(e) = \Phi$, i.e., two edges which interfere with each other are not assigned the same time slot.

For all edges $e \in E$, the set $s(e)$ (set of time slots in W which are currently assigned to e) is initialized to Φ. Edges in E are processed sequentially in the non-increasing order of their lengths. Let the current edge being processed be e. Let $s'(e)$ denote the set of time slots in W which have already been assigned to edges in $I(e)$ (and hence cannot be assigned to e): $s'(e) = \bigcup_{f \in I_{\geq}(e)} s(f)$. In the remaining slots $W \setminus s'(e)$, we choose any subset of $w \cdot x(e)$ time slots and assign them to $s(e)$.

Algorithm 1 SCHEDULE($\vec{x}, w$)

```
1: for all e ∈ E do
2:     s(e) = Φ
3: end for
4: Sort E in non-increasing order of edge-lengths.
5: for i = 1 to |E| do
6:     e = E[i]
7:     s'(e) = ⋃_{f ∈ I(e)} s(f)
8:     s(e) = any subset of (W \ s'(e)) of size w · x(e)
9: end for
```

Distributed Frame Scheduling We now present a synchronized distributed implementation of the centralized scheduling algorithm which terminates in polylogarithmic number of rounds. Our distributed algorithm is based on the ideas sketched in [17, 4]. We assume that each edge in the network has a target for the number of time slots it needs to be assigned within a time window. We view the distributed scheduling problem as a variant of the distributed edge coloring problem: the only difference in our context is that edges are weighted and the number of slots needed by an edge is proportional to its weight. The pseudo-code for the distributed algorithm is presented in **Algorithm 2**. Recall that $range(u)$ denotes the transmission range of u. Let $N(v)$ denote the set of all nodes which can potentially interfere with v in the network. We say that a node v is *finished* if slots have been chosen for all its incident edges. The distributed algorithm proceeds in rounds, and the maximum radius of a node that is yet to be finished decreases sufficiently, with good probability, with each round of the algorithm. We now present the details of our distributed algorithm in Algorithm **2**.

We note that Step 6 of the distributed algorithm involves a candidate node v detecting other candidate nodes $w \in N(v)$. The distributed implementation for this candidate detection step is non-trivial and we defer the details of this step to the full version.

4.3 Link-Flow Stability: Sufficient Conditions

In this section, we analyze conditions under which the link-flow scheduling algorithm achieves stability; hence this also yields sufficient conditions for link-flow stability. Recall

Algorithm 2 DISTRIBUTED-SCHEDULE

1: **for all** $e \in E$ **do**
2: $s(e) = \Phi$
3: **end for**
4: **for** each round i **do**
5: For each node v that is yet to be finished, if $\exists w' \in N(v)$ s.t. $range(w') \geq 2 \cdot range(v)$ and w' is yet to be finished, then v does not participate in this round; else, it elects to become a candidate with probability $1/d$, where d is the number of nodes in $N(v)$ that are unfinished.
6: If v elects to be a candidate, and some other $w \in N(v)$ also elects to be a candidate, v will not participate in this round.
7: If node v participates in this round, it chooses slots greedily for each edge (v, w) for which $s(v, w) = \Phi$, in the same way as in steps 7-8 of Algorithm 1.
8: **end for**

that $s(e)$ is the set of time slots assigned to edge e within a single frame. The following lemma states that our scheduling algorithm produces a conflict-free schedule.

LEMMA 4. *Algorithm 1 and its distributed implementation produce a conflict-free schedule, i.e., for any two interfering edges $e_1, e_2 \in E$, $s(e_1) \bigcap s(e_2) = \Phi$.*

PROOF. Assume w.l.o.g. that e_1 is processed before e_2 by the algorithm. Since e_1 and e_2 interfere, it follows that $e_1 \in I_{\geq}(e_2)$, and hence $s(e_1) \subseteq \bigcup_{f \in I_{\geq}(e_2)} s(f) = s'(e_2)$. Since, $s(e_2) \subseteq W \setminus s'(e_2)$, the lemma follows. □

The following lemma proposes a sufficient condition for which the link-flow scheduling algorithm yields a *valid* schedule, i.e., sufficient number of slots are chosen for each edge within a frame.

LEMMA 5. *The link-flow scheduling algorithm and its distributed implementation produce a valid schedule for $\vec{x}$ if the following holds:*

$$\forall e \in E, \qquad x(e) + \sum_{f \in I_{\geq}(e)} x(f) \leq 1.$$

Suppose we have a set of end-to-end flows $\mathbf{f}_i$ between each $\{s_i, t_i\}$ pair. For each $e \in E$, let $x(e) = \sum_i \mathbf{f}_i(e)$ denote the total flow on link e where $\mathbf{f}_i(e)$ is the amount of flow i on edge e and let $\vec{x}$ denote the link-rate vector. If $\vec{x}$ satisfies the conditions of lemma 5, the following result shows that we get a stable schedule, i.e., each packet is delivered in a bounded amount of time.

OBSERVATION 6. *If the vector $\vec{x}$ above satisfies the conditions of lemma 5, each packet is delivered in at most Wn steps.*

PROOF. Assume that W is such that $W\mathbf{f}_i(e)$ is integral for each i and e. Consider any flow i. The number of packets injected for this flow during the window of size W is exactly $r_i W$. For each edge e, partition the $Wx(e)$ slots into $\mathbf{f}_i(e)W$ slots for each i. Then, it is clear that for each flow i, each packet will move along one edge in W steps. □

5. SCHEDULING END-TO-END FLOWS

In this section, we discuss efficient algorithms for scheduling end-to-end flows. Specifically, given a collection of paths and an associated rate vector R^*, our goal is to find a stable schedule whose rate is αR^*, where α is the scaling factor whose value we seek to maximize. The basic idea behind the end-to-end scheduling algorithm is as follows. Let the vector R^* induce a set of link flows x^*. Let κ denote the maximum congestion on any edge: i.e., $\kappa = \max_e(x^*(e) + \sum_{e' \in N_{\geq}(e')} x^*(e'))$. If $\kappa \leq 1$, then Lemma 5 implies that the induced link rate x^* (and hence the end-to-end rate R^*) can be stably scheduled by the link scheduling algorithm. Hence, we can achieve a stable end-to-end scheduling by simply repeating the link schedule periodically, with the period being the length of the frame. However, if $\kappa > 1$, then we scale the end-to-end rate vector (and hence the link rates too) by a factor κ^{-1}. Crucially, the new rates allow us to stably schedule the scaled end-to-end flows by repeating the link scheduling algorithm periodically. How good is the scaling factor of $\alpha = \kappa^{-1}$? We now turn to the notion of *competitiveness of scheduling algorithms* to answer this question.

5.1 Competitiveness of Scheduling Algorithms

We now introduce the notion of competitiveness for scheduling algorithms; as will be explained next, this metric plays a key role in understanding the *end-to-end* efficiency of such algorithms. Let $\mathcal{P}$ be a collection of paths, and for each $p \in P$, let $r(p)$ denote the rate associated with the path p. These end-to-end flows induce a link-flow vector $\vec{x}$ which specifies the a rate $x(e)$ for each edge $e \in E$: $x(e) = \sum_{p:e \in p} r(p)$. In general, the end-to-end flow vector may not be stable, i.e., there might not exist any scheduling algorithm which achieves the rates specified by the flow vector. Given a end-to-end flow scheduling algorithm $\mathcal{A}$, define its *throughput fraction* to be the maximum scalar value $q = q(\mathcal{A})$ such that a rate of $q \cdot r(p)$ can be scheduled by $\mathcal{A}$ for each $p \in \mathcal{P}$. Let q^* be the optimal throughput fraction, i.e., q^* is the maximum throughput fraction achievable by any scheduling algorithm. The competitiveness of the scheduling algorithm $\mathcal{A}$ is defined as $q(\mathcal{A})/q^*$. The following lemma states that our scheduling algorithm is α-competitive where α is a constant (which depends only on the interference model). To our knowledge, this is the first such guarantee known; such a worst-case guarantee rigorously proves the utility of our algorithms.

LEMMA 7. *The end-to-end flow scheduling algorithm is α-competitive where $\alpha > 0$ is a constant. For the Tx-model, the value of α is at least* 0.2.

PROOF. Given a link rate vector $\vec{x}$, let q^* be the optimal throughput fraction. Thus, the vector $q^*\vec{x}$ can be scheduled by an optimal scheduling algorithm. By Lemma 3, q^* is such that for all edges e, $q^*(x_e + \sum_{f \in I_{\geq}(e)} x(f)) \leq c$, where c is a constant. We now *scale down* this link rate vector by the scalar c to obtain the vector $\vec{y} = \frac{q^*}{c}\vec{x}$. Clearly, we now have for all edges e, $y(e) + \sum_{f \in I_{\geq}(e)} x(f) \leq 1$. Therefore by Lemma 5, the end-to-end scheduling algorithm can schedule vector $\vec{y}$. Hence the throughput fraction of algorithm $\mathcal{A}$ for the link vector $\vec{x}$ is at least $\frac{q^*}{cq^*} = \frac{1}{c}$. Since this is true for all vectors $\vec{x}$, the algorithm is α-competitive where $\alpha = \frac{1}{c} > 0$ is a constant. Since Lemma 3 implies that $c \leq 5$ for the Tx-model, $\alpha \geq 0.2$ for the Tx-model. This concludes the

proof of the lemma. □

6. LINEAR PROGRAMMING FORMULATIONS

We now present the LP formulations for the maximum flow (MFP) and the maximum concurrent flow (MCFP) problems in wireless networks. See Section 3.2 for the relevant definitions. Let $C = \{1, 2, \ldots, k\}$ denote a set of *commodities*. For each commodity i, let s_i and t_i represent the source and destination for the commodity. Let $\mathcal{P}_i$ denote the set of all paths between source s_i and destination t_i of commodity i. For any $p \in \mathcal{P}_i$, let $r(p)$ denote the *data rate* associated with the path p: this is the rate at which data is transferred from s_i to t_i along p. Let r_i denote the total rate at which data is source s_i injects packets for destination d_i: i.e., thus $r_i = \sum_{p:\ p \in \mathcal{P}_i} r(p)$. For any edge $e \in E$, $x(e)$ denotes the total rate at which data is transferred across edge e: i.e., $x(e) = \sum_{p:\ e \in p} r(p)$. As noted in Section 3.2, in MFP, we would like to maximize the sum of all rates r_i subject to the wireless interference constraints. In MCFP, we would like to maximize the sum of the rates r_i subject to the additional constraint that all the r_i's are equal.

We now present a generalized *LP*-formulation, called the MAXFLOWLP which captures both these problems by incorporating end-to-end fairness constraints. The central notion in this formulation is that of the *fairness index* λ. The fairness index $\lambda \in [0, 1]$ denotes the ratio between the minimum and maximum rates: $\lambda = \frac{\min_i r_i}{\max_i r_i}$. Note that λ equal to 0 and 1 correspond to the special cases of MFP and MCFP respectively. Our formulation is as follows:

$$\begin{aligned}
\max \quad & \sum_{i \in C} r_i \qquad \text{subject to} \\
\forall i \in C, \quad & r_i = \sum_{p \in \mathcal{P}_i} r(p) \\
\forall i \in C, \forall j \in C \setminus \{i\}, \quad & r_i \geq \lambda r_j \\
\forall e \in E, \quad & x(e) = \sum_{p:\ e \in p} r(p) \\
\forall e \in E, \quad & x(e) + \sum_{f \in I_{\geq}(e)} x(f) \leq 1 \\
\forall i \in C, \forall p \in \mathcal{P}_i, \quad & r(p) \geq 0
\end{aligned}$$

We make the following observations about this LP formulation. First, we note that the stability conditions derived in Lemmas 3 and 5 are crucial for modeling the effect of interference in the LP and still guarantee a constant-factor performance ratio. This is a significant distinction between our techniques and those of [13] and [16]: the former does not guarantee good performance bounds and the latter does not model wireless interference; Next, we observe that the size of this program may not be polynomial in the size of the network G as there could be exponentially many paths $\mathcal{P}_i$. However, using standard techniques, the same program could be equivalently stated as a polynomial-size network flow formulation [1]; we choose to present this standard formulation here for ease of exposition.

The first set of constraints define the total rate r_i for each commodity. The second set of constraints are the fairness constraint which ensure that the ratio between the minimum and maximum end-to-end rates is at least λ. The third set of constraints define the link rates $x(e)$ for each link e. The fourth set of constraints capture wireless interference. These constraints along with the end-to-end scheduling algorithm discussed in Section 5 ensure that the flows computed by the LP can be feasibly scheduled. Finally, the objective value of this LP is at most a constant factor away from an optimal solution: the optimal schedule induces a rate $x^*(e)$ on each link e; further the rates x^* also satisfy the conditions of Lemma 5. Hence, scaling down the optimal end-to-end rates and hence the link rates by a factor c (the constant which appears in Lemma 3) results in a feasible solution. The following lemma formalizes the last two observations.

THEOREM 8. *The MAXFLOWLP formulation always results in a solution which can be stably scheduled. Further, the value of the objective function computed by the MAXFLOWLP is within a constant factor from the optimal solution to the corresponding flow problem. This factor is has a value of at most* 5 *for the Tx-model.*

Additional Constraints Any set of linear constraints can be added to the LP formulation. Recall that $x(e, i) = \sum_{p \in \mathcal{P}_i : e \in p} r(p)$ denotes the rate for flow i on edge e. Let $d(e)$ denote the length of edge e. To bound the total amount of energy used for the ith flow by some quantity q, we can add a constraint of the form for each i:

$$\sum_e d(e)^\beta x(e, i) \leq q r_i \tag{3}$$

The constant β is the exponent that relates the energy needed to transmit over a given distance. Similarly, for bounding the total hops used in a flow by some number h, the following constraint can be added for each i:

$$\sum_e x(e, i) \leq h r_i \tag{4}$$

7. HEURISTICS FOR PATH SELECTION

Several ad hoc routing protocols such as AODV, DSR, and DODV use hop-count as the path metric for selecting routes: whenever a route needs to be established between a source and destination, amongst all the available routes, the protocol selects the one with the shortest number of hops. In general, hop-count based shortest paths do not optimize network throughput since several shortest paths could potentially pass through a small region in the network, resulting in "hot-spots" or regions of heavy congestion in the network. Several recent approaches have been proposed for devising path metrics to avoid hot-spots (see [20] for instance).

Motivated by the theoretical techniques developed in this paper, we propose some congestion aware path selection strategies to alleviate hot-spots. The basic idea behind our path selection strategies is as follows: each link e in the network is associated with a length function $l(e)$ which is an increasing function of the link-congestion $c(e)$. Recall that the congestion $c(e)$ as defined in Section 4 takes into account, the load on e as well as the load on the edges which interfere with e. In practical settings, this can be estimated at the MAC layer by passively hearing the transmissions by neighboring nodes. A simple alternative would be to use the number of end-to-end flows through a link as a substitute for the load on the link. The length of the path is the sum of all the length of its edges. Whenever we need to choose a

route between two nodes s and t, we choose the route with the least length according to this metric.

Two functions suggest themselves naturally for the length metric: the linear length metric and the exponential length metric. In the linear length metric, the length of an edge $l(e) = \alpha c(e) + \beta$, where $\alpha > 0$ and $\beta \geq 0$ are protocol parameters. In the exponential length metric, the length of an edge $l(e) = e^{\epsilon c(e)}$, where $\epsilon > 0$ is a protocol parameter. In Section 8, we experiment with these link metrics as well as the hop-count based link metric. Our simulations indicate that both the linear and exponential link metric which takes into account congestion from interfering links, significantly outperform the hop-count based link metric in terms of network throughput.

8. SIMULATIONS

This section deals with the experimental performance evaluation of our algorithms and LP formulations through simulations. There are two main goals of our simulations: (i) understand the unconstrained network throughput of a random geometric network, as determined by the LP solution, and the impact of various constraints such as fairness, energy, and dilation on it, and (ii) a comparison of different path selection heuristics, which convert the LP based multi-path routing solution into a single path solution- this also helps in quantifying the single path vs multi-path tradeoff. Our simulation setup is described in Figure 1.

8.1 Impact of Network Constraints

We now present our simulations which deal with the impact of the network conditions on the throughput. All the experiments here were performed on the uniformly random distribution in the unit square. We study the unconstrained network throughput as well as throughput subject to fairness, energy and dilation constraints. The objective of each experiment, the results and an analysis of these results are presented.

Experiment #1: Study the variation of the maximum throughput (sum over all rates) as a function of the number of end-to-end flows subject to wireless interference constraints, without any other additional constraints.

Results and Explanation: Figure 2 plots the results of our experiments averaged over ten runs and for a single run of the experiment. The maximum aggregate throughput, on an average increases steadily with the number of end-to-end pairs. However, observe that the for a single run of the experiment, the throughput exhibits a step-like behavior w.r.t. the number of end-to-end flows. This is due to the fact that the maximum network throughput is achieved at the cost of assigning a rate of zero to certain end-to-end flows. In other words, certain flows could be completely starved so that the maximum possible aggregate throughput can be achieved by the remaining flows. Also, the total throughput flattens out, as the number of flows is increased-signifying that the absolute bound on the total capacity for the instances is reached at that point.

Experiment #2: Study the variation of aggregate throughput as a function of end-to-end fairness. Recall the definition of the fairness index λ from Section 6: λ is the ratio between the minimum rate and the maximum rate across all flows; $\lambda = 0$ implies complete starvation of flows would be allowed while $\lambda = 1$ implies that all flows have identical throughput.

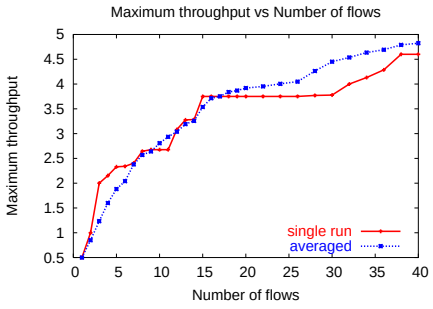


Figure 2: Experiment # 1: Variation of throughput w.r.t the number of flows. There is no fairness constraint.

Results and Explanation: Figure 3 plots the maximum aggregate throughput as a function of the fairness index λ. As expected, the aggregate throughput decreases monotonically as a function of λ; the "fair" throughput is almost half of the maximum total throughput for the current choice of parameters. The results of this experiment should be contrasted with those of Experiment # 1; they provide a tradeoff between system and user optimum.

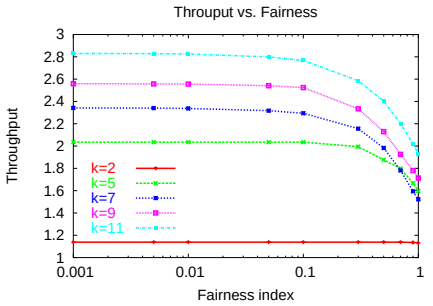


Figure 3: Experiment #2: Variation of throughput w.r.t fairness.

Experiment #3: Study the variation of throughput w.r.t the energy consumption and dilation (number of hops traversed) per unit flow respectively.

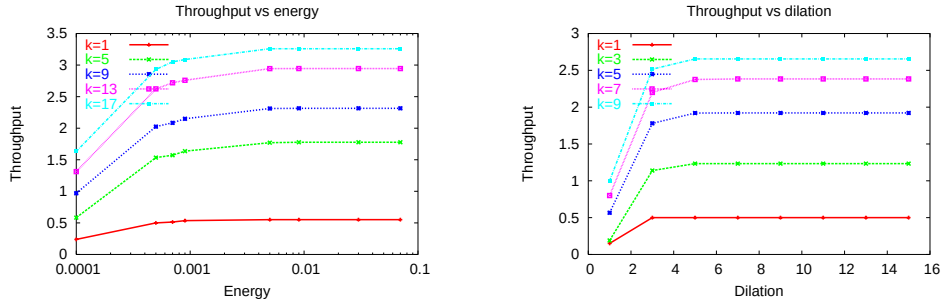
Results and Explanation: Figures 4(a) and (b) summarize the results. In both these plots, note how the throughput increases as a concave function of energy and dilation and reaches an upper limit. The similarity in the trends observed in these two plots can be explained by the fact that the constraints which model energy consumption as well as dilation are both packing constraints and are similar in structure (see Section 6). Also, in both of these plots, the throughput flattens out at some point after which allowing longer paths or paths with more energy does not make any difference to the throughput.

8.2 Impact of Path Selection Strategies

We now discuss the impact of various path selection strategies on the throughput. In all the measurements in this subsection, the fairness index $\lambda = 1$, i.e., we maximize aggregate throughput subject to the constraint that all end-to-end flows have equal rates.

1. **Network type:** We consider the networks resulting from two types of point distributions. The first is a random distribution of 245 points in a 7×7 square. The second corresponds to a distribution of cars in a region of downtown Portland, OR, obtained by running the TRANSIMS simulation [27]. Figure 6(a) presents a map of the node distribution in this network. This consists of 500 points in a $3km$ by $3km$ region . We will denote it by `real-network`.
2. **Number of flows:** We experiment with varying number of end-to-end flows, each of which has a randomly chosen source and destination node. Also denoted by k.
3. **Edge Capacities:** All edges have a transmission rate of one unit of data per time unit.
4. All nodes have a unit transmission radius.
5. All data points are averaged over ten runs of the experiment.

Figure 1: Summary of Simulation Setup.



(a) (b)

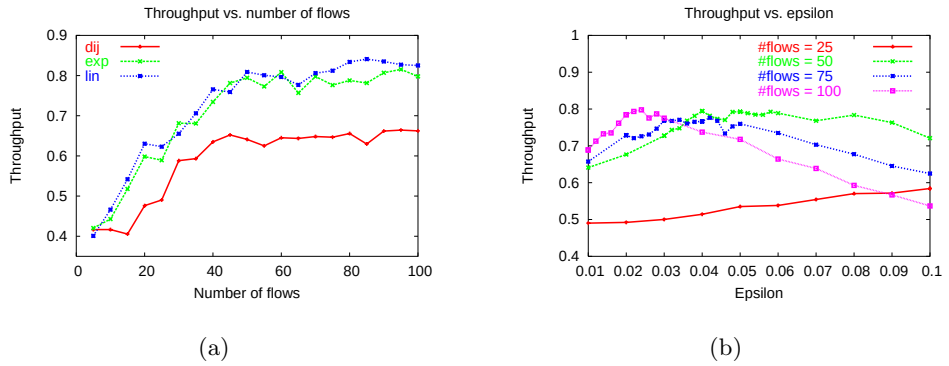
Figure 4: Results for Experiment #3. (a) Variation of throughput w.r.t energy. Each curve represents the throughput for a given number of flows (denoted by k). (b) Variation of throughput w.r.t dilation.

Experiment #4: Study the impact of three different path selection strategies on the aggregate throughput. The three strategies considered here are the simple hop-count based shortest path strategy (marked dij; stands for Dijkstra), the shortest path strategy based on the linear link-metric (marked lin) and shortest path based on the exponential link-metric (marked exp).[1]

Results and Explanation: Figures 5(a) and (b) summarize the results. Figure 5(a) plots the aggregate throughput w.r.t. the number of flows for the three path selection strategies. Clearly, both the linear and exponential congestion based strategies outperform the hop-count based shortest path algorithm significantly. Further, between the linear and the exponential link metrics, the linear metric seems to be slightly better than the exponential link metric as the number of flows increase. It is interesting to compare this plot with the LP solution in Figure 3. Both these plots are for the same scenario, but Figure 3 allows multi-path routing while Figure 5(a) only considers single path routing. This comparison shows that the capacity drops by more than a factor of five by restricting the routes to be single paths; however, multi-path routing protocols clearly incur more overhead in terms of route maintenance, size of the routing table, etc. and these factors need to be taken into consideration for any serious comparison between single-path and multi-path routing.

Figure 5(b) measures the sensitivity of the throughput to the value of the exponent (denoted epsilon) in the exponential link metric. In general, note how the throughput initially increases as a function of epsilon, reaches a peak, and starts decreases as a function of epsilon. Further, when the number of flows is higher, observe that the peak value is attained for a smaller value of epsilon. Intuitively, this implies that a lower value of epsilon is more effective during times of heavy traffic than a higher value of epsilon.

[1]See Section 7 for details of the last two strategies; for the exponential link-metric, we choose the exponent for each set of flows which maximizes the throughput



(a) (b)

Figure 5: Results for Experiment #4. (a) Variation of throughput vs. number of flows for three path selection strategies (b) Variation of throughput w.r.t the exponent epsilon.

8.3 Impact of Network Structure

In this subsection, we describe our simulations on realistic ad-hoc network topologies. The network we consider is described earlier and is henceforth referred to as `real-network`.

Experiment #5: Study the impact of path selection strategy on the total throughput in `real-network`. Figure 6(a) presents a map of the node distribution in this network. Contrast the node densities with those obtained by random distribution of nodes.

Results and Explanation: Figures 6(b) and (c) summarize our results. Once again, both the linear and exponential link-metric based strategies significantly outperform the shortest-path algorithm. In general, the results for this realistic network seem to be qualitatively the same as those for the random network.

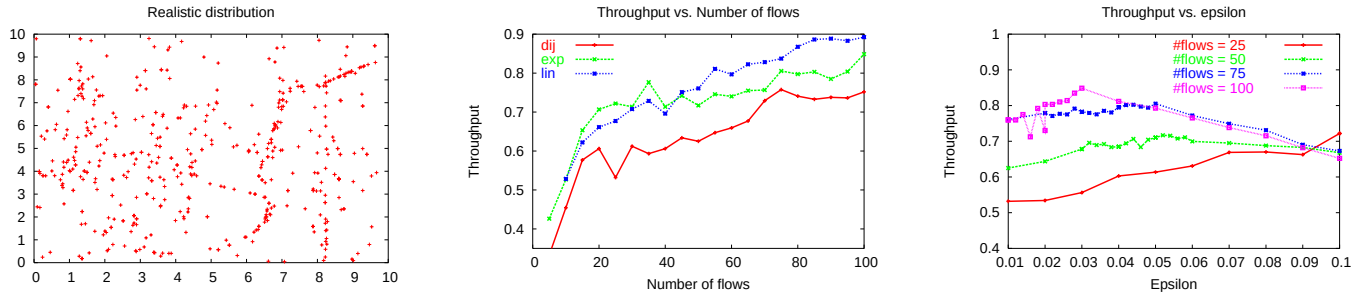
9. RELATED WORK

There has been much research on determining the optimal rates to maximize throughput via LP formulations. The first attempts can be traced back to Hajek and Sasaki [11], and to Baker *et al.* [2]. Jain *et al.* [13] propose LP-formulations for max-flow and related problems in a wireless network; in fact, they formulate their constraints in terms of arbitrary conflict graphs which can incorporate any interference model. Their formulations do not fully exploit the properties of wireless interference constraints; further, they do not discuss how close their LP-formulations are with respect to the optimal solution, or how actual scheduling protocols can be derived from the LP solution.

Over the last few years, the capacity of random wireless ad-hoc networks has been a subject of active research; see [5, 15, 10, 9, 18, 19, 21, 17, 22] and the references therein. Researchers have considered random ad-hoc networks, hybrid networks wherein one has infrastructure support, energy constraints, maximum power range constraints and mobility effects.

Our paper builds upon two different results: [16] and [17], and can be viewed as a synthesis of these two results. The approach of formulating the interaction between the MAC, routing and transport layers using linear programming was first considered by Kodialam and Nandagopal [16], who propose similar LP-formulations and a scheduling algorithm for determining the maximum transmission rates for a given network, and for three specific interference models (PCA, RCA, TRCA). They also show how to use the approach in [8] for solving the LPs using a sequence of shortest-path computations. They do not show how to use the LP solution to get an actual schedule with provable performance guarantees. Solving LPs is very time consuming, and an LP based method is generally impractical, especially for mobile computing applications. To remedy this, Kodialam and Nandagopal show how to use the framework of Garg and Konemann [8] to device a combinatorial method of solving such an LP, within any desired level of accuracy. The work in [17] formulates the problem of minimizing latency for the MAC and routing layers together and develops efficient distributed algorithms for this problem in geometric graphs, which are a commonly used abstraction of radio transmission. However, this work only deals with a static setting, i.e., when packets are not injected continuously into the network, and obtaining provable results for the unified protocol design problem with continuous packet injections has been a very interesting open problem. De Couto *et al.* [7] present a link-metric for shortest-path based routing algorithms which optimizes the total expected number of transmissions of a packet. Our link-metrics differ from the one presented in [7] by accounting for interference between links which may belong to different routes and by adapting to load changes on a link and the set of links which interfere with it.

In practice, the problem of transmitting packets between each source-destination pair in a OSI protocol stack based model is broken down into sub-problems, the most important of which are: (i) choosing routes for each such pair - a protocol like AODV chooses some sort of (single) shortest path for each pair, (ii) MAC scheduling of the packets along these paths - this resolves contention, and determines which nodes transmit at a given time slot, (iii) actual transmission of the packets on the physical channel, and (iv) choosing rates of transmission for each source-destination pair - this is achieved dynamically by a TCP like protocol, which uses feedback from the network to regulate the flow. While this modularity is useful in designing the network, it is almost impossible to determine the quality of the performance of such protocols, and how to improve the performance. In



(a) (b) (c)

Figure 6: Results for Experiment #5. (a) Node Distribution in a realistic network `real-network`. (b) Throughput variation for various the three path selection strategies for `real-network`. (c) Throughput variation for the exponential metric w.r.t epsilon for `real-network`.

fact, there is a significant interaction between protocols at different layers [6], and plugging in optimal protocols for each layer does not lead to optimal overall performance [3, 6, 24, 26]. This has motivated the study of unified protocols, and unified measures that capture the overall performance. The work of Anil Kumar et al. [17] presents unified algorithms for routing+scheduling with provable guarantees for wireless networks under static packet injections.

10. REFERENCES

[1] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin. *Network flows: theory, algorithms, and applications*. Prentice Hall, Englewood Cliffs, New Jersey, 1993.

[2] D. J. Baker, J. E. Wieselthier, and A. Ephremides. A distributed algorithm for scheduling the activation of links in self-organizing mobile radio networks. In *IEEE Int. Conference Communications*, pages 2F6.1–2F6.5, 1982.

[3] H. Balakrishnan. Challenges in Reliable Data Transport Over Heterogeneous Wireless Networks. Ph.D. Thesis, Department of Computer Science, University of California at Berkeley, 1998.

[4] H. Balakrishnan, C. Barrett, V. S. Anil Kumar, M. Marathe, and S. Thite. The distance 2-matching problem and its relationship to the mac layer capacity of ad-hoc wireless networks. Technical Report LA-UR-03-5980, Los Alamos National Laboratory. To appear in *IEEE J. Selected Areas in Communications*, 22(6), August, 2004.

[5] N. Bansal and Z. Liu, Capacity, delay and mobility in wireless ad-hoc networks, *IEEE INFOCOM*, April 2003.

[6] C. L. Barrett, M. Drozda, A. Marathe, and M. V. Marathe. Characterizing the interaction between routing and MAC protocols in ad-hoc networks. In *Proceedings of the ACM Symposium on Mobile Ad Hoc Networking and Computing*, pages 92–103, 2002.

[7] D. De Couto, D. Aguayo, J. Bicket, and R. Morris. A high-throughput path metric for multi-hop wireless routing. In *Proceedings of the 9th annual international conference on Mobile computing and networking*, pp. 134–146, 2003.

[8] N. Garg and J. Koenemann. Faster and simpler algorithms for multicommodity flow and other fractional packing problems. In *Proceedings of the 39th Annual Symposium on Foundations of Computer Science*, page 300. IEEE Computer Society, 1998.

[9] P. Gupta and P. R. Kumar. The capacity of wireless networks. *IEEE Transactions on Information Theory*, 46(2):388–404, March 2000.

[10] M. Grossglauser and D. Tse. Mobility increases the capacity of ad-hoc wireless networks *Proc. IEEE INFOCOM* 2001.

[11] B. Hajek and G. Sasaki. Link scheduling in polynomial time. *IEEE Transactions on Information Theory*, 34:910–917, 1988.

[12] F. Meyer auf der Heide, C. Schindelhauer, K. Volbert and M. Grünewald Energy, Congestion and Dilation in Radio Networks *Proc. ACM Symposium on Parallel Algorithms and Architectures (SPAA)*, Mannitoba, 2002.

[13] K. Jain, J. Padhye, V. Padmanabhan, and L. Qiu. Impact of interference on multi-hop wireless network performance. In *Proceedings of the 9th annual international conference on Mobile computing and networking*, pages 66–80. ACM Press, 2003.

[14] K. Kar, M. Kodialam, T. V. Lakshman, and L. Tassiulas. Routing for network capacity maximization in energy-constrained ad-hoc networks. In *IEEE INFOCOM*, 2003.

[15] U. Kozat and L. Tassiulas. Throughput Capacity in Random Ad-hoc Networks with Infrastructure Support, *Proc. 9th Annual ACM International Conference on Mobile computing and networking*, Sept. 2003.

[16] M. Kodialam and T. Nandagopal. Characterizing achievable rates in multi-hop wireless networks: the joint routing and scheduling problem. In *Proceedings of the 9th annual international conference on Mobile computing and networking*, pages 42–54. ACM Press, 2003.

[17] V. S. Anil Kumar, M. Marathe, S. Parthasarathy, and A. Srinivasan. End-to-end packet scheduling in ad hoc networks. In *ACM-SIAM Symposium on*

Discrete Algorithms, SODA, pages 1014–1023, 2004.

[18] B. Liu, Z. Liu, D. Towsley, On the capacity of hybrid wireless networks *Proc. IEEE INFOCOM*, April 2003.

[19] R. Negi and A. Rajeswaran, Capacity of Power Constrained Ad-hoc Networks, *IEEE INFOCOM*, 2004.

[20] Jitendra Padhye, Richard Draves, Brian Zill Routing in multi-radio, multi-hop wireless mesh networks *Proc. MOBICOM*, 2004

[21] C. Peraki and S. Servetto. On the maximum stable throughput problem in random networks, *Proc. ACM MobiHoc*, 2003.

[22] Bozidar Radunovic and Jean-Yves Le Boudec. Rate Performance Objectives of Multihop Wireless Networks, *IEEE Trans. on Mobile Computing* 3(4): 334–349, 2004.

[23] Siuli Roy, Dola Saha, S. Bandyopadhyay, Tetsuro Ueda and Shinsuke Tanaka A network-aware MAC and routing protocol for effective load balancing in ad hoc wireless networks with directional antenna *Proceedings of the 4th ACM international symposium on Mobile ad hoc networking & computing*, 2003.

[24] E. Royer, S. Lee and C. Perkins. The Effects of MAC Protocols on Ad hoc Network Communications. *Proc. IEEE Wireless Communications and Networking Conference,* Chicago, IL, September 2000.

[25] A. Srinivas and E. Modiano. Minimum energy disjoint path routing in wireless ad-hoc networks. In *Proceedings of the 9th annual international conference on Mobile computing and networking*, pages 122–133, 2003.

[26] K. Tang, M. Correa and M. Gerla, Effects of Ad Hoc MAC Layer Medium Access Mechanisms under TCP. *MONET* 6(4): 317-329 2001.

[27] TRANSIMS. *http://transims.tsasa.lanl.gov/.*

[28] Romit Roy Choudhury, Xue Yang, Ram Ramanathan, and Nitin Vaidya. Using Directional Antennas for Medium Access Control in Ad Hoc Networks In *Proceedings of the Conference on Mobile Computing and Networking (MobiCom), ACM MobiCom*, September, 2002.

[29] S. Yi, Y. Pei and S. Kalyanaraman, On the capacity improvement of ad hoc wireless networks using directional antennas. In *Proceedings of the 4th ACM international symposium on Mobile ad hoc networking & computing*, pp. 108–116, 2003.

APPENDIX

PROOF. **(For Claim 2)** We now prove this claim for the Tx-model and the protocol model. The proof for the transmitter-receiver model can be found in [17].

Tx-Model: We first note that unlike the other two models, in the transmitter model we may treat interference as occuring between nodes (senders) rather than edges, since the interference condition depends solely on the transmission ranges of the senders and the distance between them. For any node u, define $I(u)$ and $I_{\geq}(u)$ analogous to the definition for edges as follows: $I(u) = \{w : d(u,w) < (1+\Delta)\cdot(range(u)+range(w))\}$. $I_{\geq}(u) = \{w : range(w) \geq range(u)$ and $w \in I(u)\}$. For any edge $e = (u,u')$, $I(e)$ is now defined as follows: $I(e) = \{e' = (w,v) : w \in I(u)\}$. Similarly, $I_{\geq}(e) = \{e' = (w,v) : w \in I_{\geq}(u)\}$. We now show that for any node u, at most five nodes in $I_{\geq}(u)$ can simultaneously transmit in any time slot without interfering with each other.

Consider any node u and a large disc C centered at u which contains all the nodes in the network. Consider any sector which subtends an angle of $\frac{\pi}{3}$ at u. Let $w, w' \in I_{\geq}(u)$ be two nodes in this sector. W.l.o.g., assume that $d(u,w') \geq d(u,w)$. It is easy to see that $d(w,w') \leq d(u,w')$. Further, we have $range(w') \geq range(u)$. Thus, w' has a bigger range than u and is closer to w than u. Since u and w interfere with each other, clearly, w and w' also each interfere with each other and hence can not transmit simultaneously. Thus the angle subtended at u by any two simultaneous transmitters in the set $I_{\geq}(u)$ is strictly greater than $\frac{\pi}{3}$. Hence, there can be at most five successful transmitters from this set which proves the claim.

Protocol Model: Consider an edge $e = (u,v)$ of length r. Let $\mathcal{C}$ be a disk of radius $r(1+\Delta)$ centered at v. We now *pack* the disk $\mathcal{C}$ with a set of disks S of radius $\frac{r\Delta}{2}$ such that the following hold:

1. The centers of all disks in S are within $\mathcal{C}$.
2. No disk in S is such that its center is within another disk in S.
3. The union of all the disks in S cover the region covered by $\mathcal{C}$.

It is easy to construct such a covering: start with the empty set S; as long as the current set of disks in S do not completely cover disk $\mathcal{C}$, choose an uncovered point in $\mathcal{C}$, draw a disk of radius $\frac{r\Delta}{2}$ around this point and add this disk to S. Let $e_1 = (p,q)$ and $e_2 = (r,s)$ belong to $I_{\geq}(e)$. We claim that if p and r belong to the same disk in S, then e_1 and e_2 cannot both transmit successfully in the same time slot. To see this, observe that if p and r belong to the same disk in S, then $d(p,r) \leq r\Delta$. Hence we have

$$\begin{aligned} d(q,r) &\leq d(p,q) + d(p,r) \quad \text{(triangle inequality)} \\ &\leq d(p,q) + r\Delta \\ &\leq d(p,q)(1+\Delta) \quad \text{(since } (p,q) \in I_{\geq}(e)\text{)} \end{aligned}$$

Hence, by the definition of the protocol model, edges e_1 and e_2 interfere with each other.

We now bound the number of disks within S. Imagine shrinking each disk in S to a smaller disk of radius $\frac{r\Delta}{4}$. It is easy to see that the areas of the shrunken disks do not overlap with each other. Also, the total area covered by the shrunken disks is $O(r^2(1+\Delta)^2)$. Since each of the shrunken disk is of area $\pi(\frac{r\Delta}{4})^2$, the total number of disks in S is at most $O((1+\frac{1}{\Delta})^2)$, which is the required constant c. Hence, the claim holds.

We note that the above arguments can be easily extended to **3D** in the following way. The disks of radii r, $\frac{r}{2}$ and $\frac{r}{4}$ in the **2D** arguments above are replaced by spheres of the respective radii in **3D**. The total number of spheres which can be packed within S is now bound by $O((1+\frac{1}{\Delta})^3)$, which is a constant. Hence, the claim extends to **3D** wireless networks as well. Finally, for sake of completeness, we note that the above arguments can also be extended to higher dimensions with the constant on the R.H.S. depending exponentially on

the number of dimensions. □

PROOF. **(for Lemma 5)** The schedule produced by the link-flow scheduling algorithm is stable if step 8 in Algorithm 4.2 is well defined, i.e., there are always $w \cdot x(e)$ slots available in the set $W \setminus s'(e)$. We now show that this is the case for all edges. Assume otherwise, i.e., there exists an edge e such that $|W \setminus s'(e)| < w \cdot x(e)$. Hence,

$$\begin{aligned} |W| &< |s'(e)| + w \cdot x(e) \\ &\leq |\bigcup_{f \in I_{\geq}(e)} s(f)| + w \cdot x(e) \\ &\leq \sum_{f \in I_{\geq}(e)} |s(f)| + w \cdot x(e) \\ &\leq \sum_{f \in I_{\geq}(e)} w \cdot x(f) + w \cdot x(e) \end{aligned}$$

Dividing both sides above by w and rearranging the terms, we have

$$x(e) + \sum_{f \in I_{\geq}(e)} x(f) > 1$$

which contradicts our assumption. This completes the proof of the lemma. □