Proceedings of the
47th IEEE Conference on Decision and Control
Cancun, Mexico, Dec. 9-11, 2008

Admission Control and Routing in Multi-Hop Wireless Networks

Juan José Jaramillo and R. Srikant

Abstract— **To enable services such as streaming multimedia and voice in multi-hop wireless networks it is necessary to develop algorithms that guarantee Quality of Service (QoS). In this paper, we consider the problem of optimal routing and admission control for flows which require a pre-specified bandwidth from the network. We develop an admission control and routing algorithm whose performance is close to that of an omniscient off-line algorithm that has complete *a priori* knowledge of the entire sequence (including the future) of flow arrivals and their bandwidth requests. Our algorithm makes no statistical assumptions on the flow arrival pattern or other parameters of the arriving requests, and can be implemented in a distributed manner.**

I. Introduction

Future multi-hop wireless networks will carry a host of multimedia services such as voice calls and video conferencing. A common feature of such services is that they require Quality of Service (QoS) guarantees; specifically we consider services that require a pre-specified bandwidth between the endpoints of the flow. To support such services, the network must be equipped with a protocol to decide whether or not to accept a new request, and to find a route with sufficient bandwidth for an admitted flow. Optimal admission control and routing for pre-specified bandwidth flows has been extensively studied for wireline networks. In the case of wireless networks, a number of papers have highlighted the difficulties in designing good QoS routing algorithms. In particular, the importance of taking contention count into consideration for available bandwidth estimation has been recognized in [1], [2] and the importance of load balancing to maximize the number of admitted flows has been highlighted in [3]. However, no provably optimal algorithm has been developed to the best of our knowledge.

The idea of achieving provably good performance without any modeling assumptions on the arrival requests was proposed in [4]–[6], based on the concept of *competitive ratio* [7]–[10]. Informally, competitive ratio measures the performance loss of a given algorithm caused by imperfect decisions due to the fact that it is oblivious of the future when compared against an off-line algorithm that has complete *a priori* knowledge of the sequence of requests, including the future, and can therefore make perfect decisions. For reasons that we will describe next, it is difficult to immediately adapt the competitive ratio-based routing algorithms to wireless networks.

Research supported by Motorola through the Motorola Center for Communication.

J. J. Jaramillo and R. Srikant are with the Coordinated Science Laboratory, and the Department of Electrical and Computer Engineering, University of Illinois, Urbana, IL 61801 USA (e-mail: jjjarami@illinois.edu; rsrikant@illinois.edu).

The wireless channel is a shared resource, thus there is resource contention among transmissions from different nodes. As a result, even if a flow requests a pre-specified bandwidth from the network, the load imposed by a flow on a node is a function of the topology of the network (for example, the number of neighbors and hidden terminals) and the choice of the route itself. Hence, unlike a wireline network where the bandwidth consumed by a flow along a link is known at the arrival time of a flow, in a wireless network, the load imposed by a flow on a node can only be determined during route discovery. Therefore, it is not immediately obvious that the techniques for deriving optimal QoS routing algorithms for wireline networks can be applied to wireless networks.

In this paper, our contributions are as follows:

- We first develop a model for QoS routing in multi-hop wireless networks which allows us to derive an admission control and routing algorithm with provable performance guarantees using competitive analysis and the work developed for wireline networks in [4].
- We then show that no other algorithm performs better in an asymptotic sense to be described later. The proof of this result is more involved and relies on the unique characteristics of wireless networks, and uses the techniques developed in [4].
- The optimal algorithm is not in a form that is amenable to distributed implementation. Thus, an important contribution is to convert the algorithm into a form that allows the use of standard shortest-path algorithms.

The rest of the paper is organized as follows. Section II presents an overview of previous related work. Competitive analysis theory is presented in Section III. The network model and definitions are described in section IV, while in section V we introduce our algorithm and use the competitive analysis theory to prove performance guarantees. Section VI proves that our algorithm is asymptotically optimal with respect to the competitive ratio. We show how the algorithm can be implemented in a distributed fashion in section VII. Conclusions are presented in section VIII.

II. Related Work

Finding algorithms to support quality of service in multi-hop networks has been an active topic of research in the last several years. Reference [2] studies the problem of bandwidth estimation at a node while [1], [11] study the problem of estimating the impact of contention in the available bandwidth in a multi-hop path. In [12], [13] some heuristics are presented to support QoS but the effect of contention in the admission process is ignored. The work

978-1-4244-3124-3/08/$25.00 ©2008 IEEE

in [14] takes into account contention under the implicit assumption that the interference range of a node is equal to its transmission range.

In [15] the use of packet scheduling to guarantee QoS in multi-hop networks is studied. Some proposals rely on a central algorithm to do admission control [16], [17], while others have proposed strategies assuming a TDMA [18]–[20] or CDMA over TDMA [21], [22] layer. The work in [23] explores how to provide implicit synchronization in CSMA/CA networks to achieve TDM-like performance.

A solution for multichannel multi-hop networks has been proposed in [3] under the assumption that requests can be split; if requests are non-splittable a heuristic is introduced where requests are routed in the least-congested, minimum-hop path. To the best of our knowledge, our algorithm is the first provably optimal for general networks that allows a distributed implementation.

III. Competitive Analysis

The concept of competitive ratio was first introduced by [7] and further developed by [8]–[10]. Here we will present a brief overview of this theory.

In many situations we must develop efficient algorithms which have to deal with a sequence of tasks one at a time, where the efficiency of current decisions is affected by future tasks. One classical example is the well-known *ski rental problem*, where a ski enthusiast plans on skiing for several days while weather permits. Our ski fan is faced with two options every day, either rent skis for the day at a price of r or buy them at cost b. If we knew that the total number of days to ski is d, then the optimal decision algorithm would be to buy skis if $b < rd$. Since we have no knowledge of the future, we have to develop an algorithm that has to make decisions on a day by day basis and still performs well. To do that we introduce the concepts of *on-line* and *off-line* algorithm.

Definition 1: An *on-line algorithm* is an algorithm that has to deal with a sequence of requests one at a time, without having the entire sequence available from the beginning.

Definition 2: An *off-line algorithm* is an algorithm that has complete a priori knowledge of the entire request sequence, including future requests, before it outputs its answer to solve the problem at hand.

One way to measure the performance of an on-line algorithm is by comparing it against the best possible off-line algorithm when both have to deal with the same set of requests. The *competitive ratio* then measures the performance loss of an on-line algorithm caused by imperfect decisions when compared against an off-line algorithm that can make perfect decisions since it has complete knowledge of the request sequence.

Definition 3: The *competitive ratio* of an on-line algorithm is the supremum over all possible request sequences of the performance ratio between the best possible off-line algorithm and the on-line algorithm.

This means that if an on-line algorithm has a competitive ratio of c then its performance is at least $1/c$ the performance of the best possible off-line algorithm for any request sequence, and for a given performance measure.

IV. Network Model

The network is composed of a set $\mathcal{N}$ of N nodes, where node $n \in \mathcal{N}$ has capacity $u(n)$. Without loss of generality, in the rest of the paper we will assume that

$$u(n) = 1 \; for \; all \; n \in \mathcal{N}. \tag{1}$$

The input to the algorithm is a set of flow requests $\mathcal{F} = \{f_1, f_2, \ldots, f_k\}$, where flow j is specified by

$$f_j = \left\{ s_j, d_j, r_j(t), t_j^S, t_j^F, \rho_j \right\}.$$

Nodes s_j and d_j are the source and destination respectively of an unidirectional flow.[1] Flow j requests a bandwidth $r_j(t)$ at time t, where we define $r_j(t) = 0$ for $t \notin [\, t_j^S, t_j^F \,)$. Thus, t_j^S and t_j^F are the start and finish times of flow j. For simplicity, and without loss of generality, we assume that these times are integers. A profit of ρ_j is accrued if the flow is admitted into the network. Given $f_j \in \mathcal{F}$, our algorithm will output a path P_j assigned for the request, with the understanding that $P_j = \emptyset$ if it is rejected.

Let $\lambda_n(t)$ be the relative load on node n at time t, which is a function of the flows currently admitted by our algorithm. Flow j's holding time is denoted by $T_j = t_j^F - t_j^S$, where we define the maximum holding time

$$T = \max_j \{T_j\}.$$

As mentioned in Section I, the wireless channel is a shared resource and contention among transmissions from different nodes implies that when admitting a flow we must take into account the impact of a flow in the network. To better understand this, let us use the simple scenario of Fig. 1. Our linear network of 5 nodes is such that any node can only communicate with its nearest neighbors, and where the interference range for each node is illustrated as a concentric circle around each node. There is a flow from node B to node D that requires a rate of r. Due to the exposed terminal problem, node A cannot transmit while B is transmitting, so node A's available capacity is

$$u(A) - r \stackrel{(1)}{=} 1 - r,$$

where $\stackrel{(1)}{=}$ means that the equality follows from equation (1). We use this notation throughout the paper.

Thus, when scheduling this flow, we must reserve a rate of r in node A for it to remain idle. Similarly, node B must transmit at a rate r and must remain silent while C is relaying a packet for this flow, which implies a resource reservation of rate $2r$ at B. Because of the hidden terminal problem, node E must remain idle while D is receiving a packet, so we have to reserve a rate of r at this node. Following the lines of this argument, it can be checked that we must reserve a rate of $2r$ at nodes C and D in order to be able to schedule this flow. These values are shown above each node in Fig. 1.

[1] For the case of bidirectional flows, we simply need to split the request in two unidirectional flows that need to be accepted/rejected simultaneously.

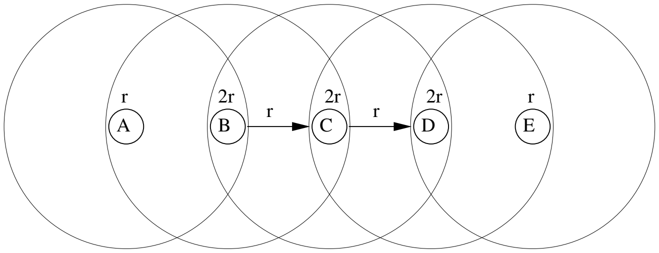

Fig. 1. Example of resource contention among nodes.

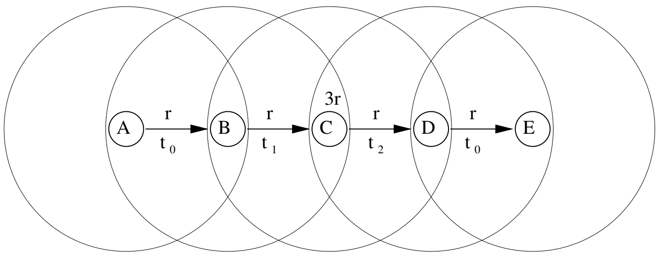

Fig. 2. Example of resource contention among nodes under the assumption of perfect packet transmission scheduling.

This example gives the intuition for the following definition. Let $Q_n(P_j)$ be the impact of flow j on node n if path P_j is used. Formally,

$$Q_n(P_j) = \sum_{n' \in \mathcal{N}_n} I_{\{n' \in P_j\}} \left(I_{\{n' \neq d_j\}} + I_{\{n' \in HT_n(P_j)\}} \right), \quad (2)$$

where $\mathcal{N}_n$ is the set of nodes within interference range of node n (including node n itself), $HT_n(P_j)$ is the set of nodes in $\mathcal{N}_n$ that need to receive a transmission for flow j that node n cannot sense –that is, hidden terminal transmissions–, and $I_{\{\}}$ is the indicator function defined as

$$I_{\{statement\}} = \begin{cases} 1 & if \quad statement = TRUE \\ 0 & if \quad statement = FALSE \end{cases}.$$

As an example, in Fig. 1 we have that for path $P = \{B, C, D\}$, $Q_A(P) = 1$, $Q_C(P) = 2$, and $Q_E(P) = 1$. It must be noted that this definition is only an upper bound on the actual impact of a flow in a given node, and can only be improved by assuming a specific, possibly ideal, transmission scheduling algorithm. To illustrate this, in Fig. 2 we assume that there is a flow from node A to E and that we schedule node transmissions such that nodes A and D simultaneously transmit at time t_0, node B transmits at t_1, and C is scheduled to transmit at time t_2. In this case, the impact of the flow on node C is 3 instead of 4, as estimated using $Q_n(P_j)$. [2]

Finding methods for estimating $Q_n(P_j)$ has been an active topic of research. For related work, the reader is referred to [1], [11], [14].

[2]It must be highlighted that for the ease of explanation we have assumed that due to the exposed terminal problem a node cannot transmit while another in its interference neighborhood is transmitting. If a certain physical layer technology does not preclude such transmissions, the definition of $Q_n(P_j)$ should be modified accordingly and the results in the rest of the paper still apply with minor modifications.

Define $Q_T(P_j)$ to be the total impact of flow j (routed on path P_j) on the network. Thus,

$$Q_T(P_j) = \sum_{n \in \mathcal{N}} Q_n(P_j). \quad (3)$$

Additionally, define Q_T and Q as follows:

$$Q_T = \max_j \{Q_T(P_j)\}$$
$$Q = \max_{j,n} \{Q_n(P_j)\}. \quad (4)$$

We normalize the cost such that for any flow $f_j \in \mathcal{F}$ and any time such that $r_j(t) > 0$

$$1 \leq \frac{\rho_j}{Q_T r_j(t) T_j} \leq F \quad (5)$$

for F large enough. For example, if we have that $r_j(t) = r_j$ for $t \in [\, t_j^S, t_j^F \,)$ and the profit is defined to be proportional to the bandwidth-holding time product, i.e. throughput, then we can make $\rho_j = Q_T r_j T_j$ and let $F = 1$.

Finally, we assume that rate requirements are small enough compared to node capacity. Specifically,

$$r_j(t) \leq \frac{\min_n \{u(n)\}}{Q \log \mu} \stackrel{(1)}{=} \frac{1}{Q \log \mu} \quad (6)$$

where

$$\mu = 2\left(1 + Q_T T F\right), \quad (7)$$

and log means $\log_2$.

Later we will prove in Section VI that (6) is a necessary condition for any algorithm to achieve logarithmic competitive ratio.

V. ALGORITHM

The main goal is to develop an admission control and routing algorithm that enforces capacity constraints, that is

$$\lambda_n(t) \leq 1 \; for \; all \; t \; and \; n \in \mathcal{N}. \quad (8)$$

and maximizes profit:

$$\sum_{j:P_j \neq \emptyset} \rho_j.$$

Furthermore, the algorithm cannot rely on knowledge about future flows to make admission decisions and once a flow has been admitted no rerouting is allowed and no flow is to be interrupted.

To do that, it will sequentially analyze flows from $\mathcal{F}$ and decide whether to admit them or not.

A. Definition

The decision rule for admitting flow $f_j \in \mathcal{F}$ and assigning a path is:

1) For all $t \in [\, t_j^S, t_j^F \,)$, $n \in \mathcal{N}$ let the cost of node n at time t be

$$c_n(t) = u(n)\left[\mu^{\lambda_n(t)} - 1\right] \stackrel{(1)}{=} \mu^{\lambda_n(t)} - 1. \quad (9)$$

2) If there exists a path P_j from node s_j to d_j such that

$$\sum_{n\in\mathcal{N}} \sum_{t_j^S \le t < t_j^F} Q_n(P_j)\frac{r_j(t)}{u(n)}c_n(t) \le \rho_j \qquad (10)$$

then route f_j using P_j and set

$$\lambda_n(t) \leftarrow \lambda_n(t) + Q_n(P_j)\frac{r_j(t)}{u(n)} \qquad (11)$$

for all $n \in \mathcal{N}$, $t_j^S \le t < t_j^F$.

Note that

$$Q_n(P_j)\frac{r_j(t)}{u(n)}$$

is the fraction of node n's capacity that would be used by flow j. Thus, the algorithm compares the link cost weighted by this quantity to the profit and admits the call if the cost is less than or equal to the profit.

B. Performance Guarantees

Now that the wireless model and the algorithm are defined, we can use the techniques developed for wireline networks in [4]. We will first proceed to prove that our algorithm enforces capacity constraints, which from now on we will call the *Admission Control and Routing (ACR) algorithm*. In other words, if the ACR algorithm decides to admit a flow request, then there is sufficient available capacity.

Lemma 1: Capacity constraints are enforced by the ACR algorithm. That is,

$$\lambda_n(t) \le 1 \; for \; all \; t \; and \; n \in \mathcal{N}.$$

(Due to lack of space, the proof of the Lemma will be presented in a longer version of the paper.)

The proof of the competitiveness of the ACR algorithm is done in two steps. In Lemma 2 we prove that the total network cost is at most within a factor of the accrued gain, and in Lemma 3 we prove that the profit due to requests rejected by the ACR algorithm and accepted by the optimal off-line algorithm is bounded by the total network cost. These two results then imply that the profit of the ACR algorithm is within a factor of the profit accrued by the off-line algorithm, and hence the competitive ratio is bounded.

For the following two lemmas, define $\lambda_n(t,j)$ to be the relative load on node n at time t when only the first $j-1$ flow requests have been either admitted or rejected. That is,

$$\lambda_n(t,j) \overset{(11)}{=} \sum_{i<j} Q_n(P_i)\frac{r_i(t)}{u(n)} \overset{(1)}{=} \sum_{i<j} Q_n(P_i) r_i(t), \qquad (12)$$

with the understanding that $P_i = \emptyset$ if $f_i \in \{f_1, f_2, \ldots, f_{j-1}\}$ is rejected.

Similarly, and from (9), let $c_n(t,j)$ be defined as

$$c_n(t,j) = \mu^{\lambda_n(t,j)} - 1. \qquad (13)$$

Lemma 2: Let $\mathcal{A}_{ACR}$ be the set of indices of accepted flows by the ACR algorithm and k be the index of the last flow request in $\mathcal{F}$. Then,

$$2\log\mu \sum_{j\in\mathcal{A}_{ACR}} \rho_j \ge \sum_t \sum_n c_n(t, k+1).$$

Lemma 3: Let $\mathcal{A}_{O\setminus A}$ be the set of indices of accepted requests by the optimal off-line algorithm but rejected by the ACR algorithm. Let $m = \max\{\mathcal{A}_{O\setminus A}\}$. Then

$$\sum_{j\in\mathcal{A}_{O\setminus A}} \rho_j \le \sum_n \sum_t c_n(t,m).$$

(Due to lack of space, the proofs of the Lemmas will be presented in a longer version of the paper.)

Now, we are ready to prove the following:

Theorem 1: The ACR algorithm enforces capacity constraints and achieves a competitive ratio of $O(\log(Q_T TF))$. (Due to lack of space, the proof of the Theorem will be presented in a longer version of the paper.)

Remark: It is important to highlight that for the lemmas and theorem of this section we only assume that $Q_n(P_j) \ge 1$, but the precise definition given in (2), which depends on the assumptions about the physical layer, is only used in the following sections.

VI. Optimality

Now we will prove that no other algorithm can achieve a better competitive ratio than the ACR algorithm in an asymptotic sense and that assumption (6) is a necessary condition to achieve a good competitive ratio. To do that, we will first show that even if flow rates are small enough, the profit of the optimal off-line algorithm will exceed the profit of any on-line algorithm that is oblivious to the future by a factor of $\Omega(\log(Q_T TF))$. Finally, we will present stronger bounds for the case when flow rates are allowed to be relatively large, showing that the competitive ratio degrades when we allow large rates.

The techniques used here are again similar to the ones developed for wireline networks in [4], but rely on the unique characteristics of wireless networks, specially to find worst case scenarios in Lemmas 4 and 7.

Lemma 4: Any on-line algorithm has a competitive ratio of $\Omega(\log Q_T)$.

Lemma 5: Any on-line algorithm has a competitive ratio of $\Omega(\log T)$.

Lemma 6: Any on-line algorithm has a competitive ratio of $\Omega(\log F)$.

(Due to lack of space, the proofs of the Lemmas will be presented in a longer version of the paper.)

Hence, we have just proved the following.

Theorem 2: Any on-line algorithm has a competitive ratio of $\Omega(\log(Q_T TF))$.

Proof: It is a direct consequence of Lemmas 4, 5 and 6. ■

For the proof of Theorem 2 we assume that (6) holds. We will now proceed to prove that if this bound does not hold then no on-line algorithm can achieve logarithmic competitive ratio.

Lemma 7: If we allow requests of rate up to $\frac{1}{4k}$ then the competitive ratio is $\Omega(Q_T^{\frac{1}{4k}})$ for any positive integer k.

Lemma 8: If we allow requests of rate up to $\frac{1}{k}$ then the competitive ratio is $\Omega(T^{\frac{1}{k}})$ for any positive integer k.

Lemma 9: If we allow requests of rate up to $\frac{1}{k}$ then the competitive ratio is $\Omega(F^{\frac{1}{k}})$ for any positive integer k.
(Due to lack of space, the proofs of the Lemmas will be presented in a longer version of the paper.)

Therefore, we have the following theorem.

Theorem 3: If we allow requests of rate up to $\frac{1}{4k}$ then the competitive ratio is $\Omega(Q_T^{\frac{1}{4k}} + T^{\frac{1}{4k}} + F^{\frac{1}{4k}})$ for any positive integer k.

Proof: This follows from Lemmas 7, 8 and 9. ∎

Thus, in order to achieve logarithmic competitive ratio we need to let k be greater than

$$\frac{\log(Q_T TF)}{4\log(\log(Q_T TF))}.$$

VII. Distributed Implementation

In its present form, checking

$$\sum_{n\in\mathcal{N}}\sum_{t_j^S\le t<t_j^F} Q_n(P_j)\frac{r_j(t)}{u(n)}c_n(t)\le \rho_j$$

in the ACR algorithm requires first to specify a path P_j from source to destination and then its associated cost can be calculated. Since we ideally would like to use the minimum cost path, this means that it would be required to first identify all possible paths and then find the cost for each one of them.

The contribution of this section is to show how this can be implemented using a distributed algorithm that can find the minimum cost path without the need to first identify all possible solutions, and where every node only needs to get access to information from a local neighborhood.

Define the directed graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$, where $\mathcal{N}$ is the set of nodes in the wireless network and $\mathcal{E}$ is the set of all directed transmission edges. Formally, $e \in \mathcal{E}$ if and only if $e = (s(e), d(e))$, where $s(e), d(e) \in \mathcal{N}$, $s(e) \neq d(e)$, and node $s(e)$ can transmit to node $d(e)$.

Furthermore, define $c_n^a(j)$ to be the aggregate cost that node $n \in \mathcal{N}$ has during the holding time of flow request $f_j \in \mathcal{F}$. That is,

$$c_n^a(j) = \sum_{t_j^S\le t<t_j^F}\frac{r_j(t)}{u(n)}c_n(t).$$

With these definitions we have the following:

$$\begin{aligned}
&\sum_{n\in\mathcal{N}}\sum_{t_j^S\le t<t_j^F} Q_n(P_j)\frac{r_j(t)}{u(n)}c_n(t) = \sum_{n\in\mathcal{N}} Q_n(P_j)c_n^a(j)\\
&\overset{(2)}{=} \sum_{n\in\mathcal{N}}\sum_{n'\in\mathcal{N}_n} I_{\{n'\in P_j\}}c_n^a(j)\left[I_{\{n'\neq d_j\}} + I_{\{n'\in HT_n(P_j)\}}\right]\\
&= \sum_{e\in P_j}\sum_{n\in\mathcal{N}_{s(e)}\bigcup\mathcal{N}_{d(e)}} c_n^a(j) \qquad (14)\\
&= \sum_{e\in P_j} C_e(j),
\end{aligned}$$

where (14) is simply summation reordering and

$$C_e(j) = \sum_{n\in\mathcal{N}_{s(e)}\bigcup\mathcal{N}_{d(e)}} c_n^a(j) \qquad (15)$$

is the cost of using edge e for routing flow f_j. It must be noted that (15) decouples the cost of an edge from the path cost, allowing a distributed implementation of a shortest path algorithm to find the optimal route. Furthermore, for every edge $e \in \mathcal{E}$ we only need to gather information from the local set $\mathcal{N}_{s(e)}\bigcup\mathcal{N}_{d(e)}$ to find $C_e(j)$. It must be noted that for *any* admission algorithm this is the minimal information that must be gathered and updated in order to check resource availability, since the transmission in this link will affect the load of all nodes in the set $\mathcal{N}_{s(e)}\bigcup\mathcal{N}_{d(e)}$.

VIII. Conclusions

We have developed a model for QoS routing in multi-hop wireless networks which allows us to derive an algorithm with provable performance guarantees using competitive analysis. We proved that our algorithm has a performance close to that of an omniscient off-line algorithm that has complete *a priori* knowledge of the entire sequence of flow arrivals (including the future) and their bandwidth requests. Our algorithm makes no statistical assumptions on the flow arrival pattern or other parameters of the arriving requests. We also proved that no other algorithm performs better in an asymptotic sense of the competitive ratio. Finally, we showed that our algorithm is amenable to a distributed implementation.

References

[1] K. Sanzgiri, I. D. Chakeres, and E. M. Belding-Royer, "Pre-reply probe and route request tail: Approaches for calculation of intra-flow contention in multihop wireless networks," *Mobile Networks and Applications*, vol. 11, no. 1, pp. 21–35, Feb. 2006.

[2] I. D. Chakeres, E. M. Belding-Royer, and J. P. Macker, "Perceptive admission control for wireless network quality of service," *Ad Hoc Networks*, vol. 5, no. 7, pp. 1129–1148, Sep. 2007.

[3] J. Tang, G. Xue, and W. Zhang, "Interference-aware topology control and QoS routing in multi-channel wireless mesh networks," in *Proc. 6th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, Urbana-Champaign, IL, USA, May 25–27, 2005, pp. 68–77.

[4] B. Awerbuch, Y. Azar, and S. Plotkin, "Throughput-competitive on-line routing," in *Proc. 34th Annual Symposium on Foundations of Computer Science*, Palo Alto, CA, USA, Nov. 3–5, 1993, pp. 32–40.

[5] B. Awerbuch, Y. Azar, S. Plotkin, and O. Waarts, "Competitive routing of virtual circuits with unknown duration," in *Proc. 5h Annual ACM-SIAM Symposium on Discrete Algorithms*, Arlington, VA, USA, Jan. 23–25, 1994, pp. 321–327.

[6] S. Plotkin, "Competitive routing of virtual circuits in ATM networks," *IEEE J. Sel. Areas Commun.*, vol. 13, no. 6, pp. 1128 – 1136, Aug. 1995.

[7] D. D. Sleator and R. E. Tarjan, "Amortized efficiency of list update and paging rules," *Communications of the ACM*, vol. 28, no. 2, pp. 202–208, Feb. 1985.

[8] A. R. Karlin, M. S. Manasse, L. Rudolph, and D. D. Sleator, "Competitive snoopy caching," *Algorithmica*, vol. 3, no. 1, pp. 79–119, Mar. 1988.

[9] M. Manasse, L. McGeoch, and D. Sleator, "Competitive algorithms for on-line problems," in *Proc. 20th Annual ACM Symposium on Theory of Computing*, Chicago, IL, USA, 1988, pp. 322–333.

[10] A. Borodin, N. Linial, and M. E. Saks, "An optimal on-line algorithm for metrical task system," *Journal of the ACM*, vol. 39, no. 4, pp. 745–763, Oct. 1992.

[11] Y. Yang and R. Kravets, "Contention-aware admission control for ad hoc networks," *IEEE Trans. Mobile Comput.*, vol. 4, no. 4, pp. 363–377, Apr./May 2005.

[12] S.-B. Lee, G.-S. Ahn, X. Zhang, and A. T. Campbell, "INSIGNIA: An IP-based quality of service framework for mobile ad hoc networks," *Journal of Parallel and Distributed Computing*, vol. 60, no. 4, pp. 374–406, Apr. 2000.

[13] G.-S. Ahn, A. T. Campbell, A. Veres, and L.-H. Sun, "SWAN: Service differentiation in stateless wireless ad hoc networks," in *Proc. IEEE INFOCOM*, vol. 2, New York, NY, USA, Jun. 23–27, 2002, pp. 457–466.

[14] Q. Xue and A. Ganz, "Ad hoc QoS on-demand routing (AQOR) in mobile ad hoc networks," *Journal of Parallel and Distributed Computing*, vol. 63, no. 2, pp. 154–165, Feb. 2003.

[15] H. Luo, S. Lu, V. Bharghavan, J. Cheng, and G. Zhong, "A packet scheduling approach to QoS support in multihop wireless networks," *Mobile Networks and Applications (MONET)*, vol. 9, no. 3, pp. 193–206, Jun. 2004.

[16] R. Ramanathan and M. Steenstrup, "Hierarchically-organized, multi-hop mobile wireless networks for quality-of-service support," *Mobile Networks and Applications (MONET)*, vol. 3, no. 1, pp. 101–119, Jun. 1998.

[17] S. H. Shah and K. Nahrstedt, "Guaranteeing throughput for real-time traffic in multi-hop IEEE 802.11 wireless networks," in *Proc. Military Communication Conference (Milcom)*, Atlantic City, NJ, USA, Oct. 17–20, 2005.

[18] W.-H. Liao, Y.-C. Tseng, and K.-P. Shih, "A TDMA-based bandwidth reservation protocol for QoS routing in a wireless mobile ad hoc network," in *Proc. IEEE International Conference on Communications (ICC)*, vol. 5, New York, NY, USA, Apr. 28/May 2, 2002, pp. 3186–3190.

[19] C. Zhu and M. S. Corson, "QoS routing for mobile ad hoc networks," in *Proc. IEEE INFOCOM*, vol. 2, New York, NY, USA, Jun. 23–27, 2002, pp. 958–967.

[20] S. Guo and O. Yang, "QoS-aware minimum energy multicast tree construction in wireless ad hoc networks," *Ad Hoc Networks*, vol. 2, no. 3, pp. 217–229, Jul. 2004.

[21] C. R. Lin and J.-S. Liu, "QoS routing in ad hoc wireless networks," *IEEE J. Sel. Areas Commun.*, vol. 17, no. 8, pp. 1426–1438, Aug. 1999.

[22] H. Liu, X. Jia, D. Li, and C. Lee, "Bandwidth guaranteed call admission in TDMA/CDMA ad hoc wireless networks," *Ad Hoc Networks*, vol. 3, no. 6, pp. 689–701, Nov. 2005.

[23] S. Singh, P. A. K. Acharya, U. Madhow, and E. M. Belding-Royer, "Sticky CSMA/CA: Implicit synchronization and real-time QoS in mesh networks," *Ad Hoc Networks*, vol. 5, no. 6, pp. 744–768, Aug. 2007.