WIRELESS COMMUNICATIONS AND MOBILE COMPUTING
Wirel. Commun. Mob. Comput. 2005; **5**:933–943
Published online in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/wcm.357

Interference-aware routing and bandwidth allocation for QoS provisioning in multihop wireless networks

Jian Tang, Guoliang Xue*,† and Christopher Chandler
Department of Computer Science and Engineering, Arizona State University, Tempe, AZ 85287-8809, U.S.A.

Summary

In this paper, we study the bandwidth guaranteed routing and timeslot allocation (BANDRA) in TDMA-based multihop wireless networks with dynamic traffic. We formally model BANDRA as an optimization problem and present an integer linear programming (ILP) formulation to provide optimal solutions. This problem turns out to be a hard problem because of the impact of interference. Therefore, we propose a two-step scheme, i.e., seeking a path for routing first and then allocating bandwidth along the found path. We present two routing algorithms to compute interference-optimal cost-bounded paths. In addition, we present an optimal bandwidth allocation algorithm to allocate timeslots along the found paths for connection requests with unit bandwidth requirements. For the general case where the bandwidth requirement is larger than one, we present an effective heuristic algorithm. Our simulation results show that the average difference between solutions given by our efficient scheme and optimal ones in terms of call-blocking ratio is only 7%. Compared with the shortest path routing, our interference-aware routing algorithms combined with our bandwidth allocation algorithm always reduce call-blocking ratios. Copyright © 2005 John Wiley & Sons, Ltd.

KEY WORDS: Multihop wireless networks; interference-aware routing; bandwidth allocation; admission control; QoS; cross-layer design

1. Introduction

In order to provide real-time services, quality of service (QoS) must be well supported. Bandwidth is a basic QoS parameter and the major topic of study in this paper. A QoS connection request will be admitted if there exists a path along which the required bandwidth is available. In this case, the network should select a minimum cost path among all feasible paths and reserve the required bandwidth accordingly. Otherwise, the connection request should be blocked. Finding a minimum cost bandwidth guaranteed path in a wired network can be easily achieved by simply ignoring the links without enough bandwidth and applying a shortest path algorithm in the residual network. However, QoS provisioning in multihop wireless networks is very challenging due to the impact of interference. Since the radio is inherently

*Correspondence to: G. Xue, Department of Computer Science and Engineering, Arizona State University, Tempe, AZ 85287-8809, U.S.A.
†E-mail: xue@asu.edu

Contract/grant sponsor: ARO; contract grant/number: W911NF-04-1-0385.
Contract/grant sponsor: NSF; contract grant/numbers: CCF-0431167; ANI-0312635 (ITR).
Contract/grant sponsor: CEINT (Seed Grant).

a broadcast medium, transmissions among neighboring nodes in a wireless network may interfere with each other because of contention for the shared wireless channel. In order to compute the available bandwidth for a specific wireless link (transmission), we need to consider the interference coming from other transmissions in the neighborhood. Therefore, previous QoS routing schemes for wired networks are not applicable for multihop wireless networks.

The medium access control (MAC) protocols, such as contention-based (802.11 DCF) or time division multiple access (TDMA)-based protocols [9], have been proposed to resolve contention by scheduling transmissions in a neighborhood into different timeslots (TDMA), or randomly delaying some of them (802.11 DCF). In this paper, we study bandwidth guaranteed routing and timeslot allocation in TDMA-based multihop wireless networks with dynamic traffic. To our best knowledge, this is the first paper formally defining this problem and presenting an integer linear programming (ILP) formulation, which can be used to find optimal solutions. We also propose a practical two-step scheme including algorithms for computing a routing path and algorithms for collision-free timeslot allocation along the computed path. If packets are always routed along shortest paths without considering interference, some wireless nodes within a common neighborhood may become heavily loaded, which makes it difficult to allocate required bandwidth for found paths and therefore increases call blocking ratios. A good routing algorithm should be aware of interference as well as transmission cost. We present novel definitions of link and path interference, which are capable of characterizing the properties of wireless interference precisely. Based on our definitions, we present algorithms to solve two minimum interference cost-bounded routing problems, namely, the min-max interference cost-bounded (MICB) and the min-total interference cost-bounded (TICB) single path routing problem. The motivation behind our routing scheme is that we try to minimize the interference influence from and to existing traffic without increasing transmission cost too much. In this way, it is hopeful for the computed path to be admitted in the later bandwidth allocation phase. Furthermore, we propose an optimal algorithm, which can allocate timeslots along found single paths for connection requests with unit bandwidth requirements. Our algorithm guarantees to compute a collision-free transmission schedule. We also present an effective heuristic for the general case, i.e., the bandwidth requirement is large than one unit.

The rest of this paper is organized as follows. We discuss related work in Section 2. We describe the system model and formally define the problems in Section 3. We present our ILP formulation in Section 4. We present our interference-aware routing and bandwidth allocation algorithms in Section 5 and Section 6, respectively. Simulation results are presented in Section 7. We conclude the paper in Section 8.

2. Related Work

QoS provisioning in mobile ad hoc networks (MANETs) has been well studied in the literature. By assuming the network has a CDMA-over-TDMA MAC layer, Lin *et al*. in Reference [6] propose a sequential bandwidth routing protocol to compute routing paths and to allocate bandwidth. Admission control algorithms have also been proposed in References [5,7] under the same assumption. However, integrating CDMA technology in wireless nodes will increase the hardware cost. It may not be realistic for networks with low-cost nodes such as wireless sensor network (WSNs) or MANETs. Therefore, several QoS routing protocols [4,11] have been proposed under the consumption of a simple TDMA MAC layer.

Recent research has shown that wireless interference can make a significant impact on the performance of a wireless network. As a pioneering work, Gupta and Kumar in Reference [2] show that in a wireless network with n identical nodes, the per-node throughput is $\Theta(1/\sqrt{n \log n})$ and $\Theta(1/\sqrt{n})$ under the consumption of random and optimal node placement and communication pattern respectively. The authors in Reference [8] investigate and propose approximation algorithms to solve the problem of jointly routing flows and scheduling transmissions to achieve a given rate requirement vector. In Reference [3], the authors model the influence of interference using a conflict graph and derive upper and lower bounds on the optimal throughput. The interference-aware topology control is studied by Burkhart *et al*. in Reference [1]. In Reference [10], the authors present a framework for multihop packet scheduling in wireless ad hoc networks to achieve maximum throughput.

Our work is different from all previous works. Specifically, we consider a multihop wireless network with dynamic traffic, i.e., we do not assume that the traffic matrix is given *a priori* as in References [3,8]. Our scheme is an on-demand scheme, which will be more useful in practice. We fully explore the influence of both primary and secondary interference without making assumption of adopting CDMA-over-TDMA channel model [5–7]. Instead of proposing heuristics like References [4,11], we present ILP formulation for optimal solutions and we prove that our polynomial time bandwidth allocation algorithm is guaranteed to find a collision-free timeslot assignment for the path computed by our routing algorithm if it exits and its bandwidth requirement is one unit. Moreover, we propose interference-aware routing algorithms, which reduce call blocking ratios compared with shortest path routing.

3. Problem Definition

In this section, we will describe our system model and formally define the optimization problems we are going to study.

We assume that all nodes in the network use the same fixed transmission power, i.e., there is a fixed transmission range ($R > 0$) associated with every node. We use a directed graph $G(V, E)$ to model the wireless network where V is the set of n vertices and E is the set of m directed edges. Each vertex in V corresponds to a wireless node in the network. There is a directed edge $(u, v) \in E$ connecting vertex u to vertex v if $d(u, v) \leq R$, where $d(u, v)$ is the Euclidean distance between the nodes corresponding to u and v. The edge (u, v) in G corresponds to a wireless link from node u to node v in the wireless network. We will use vertices and nodes as well as edges and links interchangeably throughout this paper.

Due to the broadcast transmission nature and the fixed transmission range, we can imagine that associated with each node u, there is a disk with radius R and center u. We use D_u to denote the set of nodes covered by that disk. As discussed previously, we assume TDMA scheme is used at the MAC layer for multiple access, i.e., time domain is divided into timeslots, each of which has a constant payload rate. Time slots are further grouped into frames of K timeslots each. Every connection request ρ is specified by a source node s, a destination node t, and a bandwidth requirement B indicating the number of timeslots required to be allocated in one frame. The scheduling is performed on per frame basis and a number of timeslots in one frame are assigned to each connection according to its bandwidth requirement. We also assume that all wireless nodes are stationary.

We consider a single channel multihop wireless network in which each node is equipped with a single transceiver. Half-duplex operation is enforced to prevent self-interference, i.e., one node can only transmit or receive at one time. If D_x includes node v or D_u includes node y, then we say link (x, y) interferes with link (u, v), since simultaneous transmissions along (u, v) and (x, y) will lead to collisions on a receiver. The above definition also covers the cases where two links share one or two common nodes.

In each timeslot, a node can either transmit, receive, or stay idle. We define a status, $S(u, k)$, for node u in timeslot k. In timeslot k, $S(u, k)$ is transmitting (T) if node u transmits; receiving (R) if u receives; interfered (I) if u is interfered by another transmission(s); or Free (F) otherwise. These four statuses are disjoint except that a node may be transmitting (T) and interfered (I) at the same time since we only need to make sure that every receiver is free of interference. We say timeslot k is free for link (u, v) if $S(u, k) = I/F$, $S(v, k) = F$ and $S(x, k) = T/I/F, \forall x \in D_u \backslash \{u, v\}$. There is also a link cost, $C(e)$ associated with link e. In this paper, $C(e)$ is equal to 1 for all links since the transmission power is assumed to be fixed, i.e., all one-hop transmissions have the same transmission cost. The path cost is the summation of costs of all links on the path, i.e., the number of hops.

Let P be a given routing path for the current connection request. We need to find a timeslot assignment for the links on path P to avoid any collision. There are two kinds of potential contentions: inter-flow contention and intra-flow contention [10]. Inter-flow contentions are the contentions caused by the interference between a link for the current connection request and a link for an existing connection request. Intra-flow collisions are the contentions caused by the interference between a link for the current connection request and another link for the current connection request. Inter-flow contentions can be easily resolved if we always assign aforementioned free timeslots to the path of the current connection request. Intra-flow contentions are harder to address as the routing path for the current connection request is not known *a priori*. To address this problem, we will need the following definition.

Definition 1: *A collision-free schedule for path P is a timeslot assignment that allocates free timeslots for every link on the path such that it does not create any inter-flow contention or intra-flow contention. The path P is said to be an admissible path if there exists such a collision-free schedule.*

The bandwidth requirement B in this paper is measured as the number of timeslots in one frame. Here we assume that a connection always uses a single

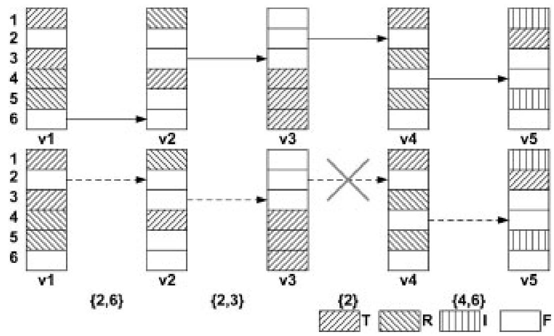


Fig. 1. Timeslot allocation for a path.

path for transmission as in Reference [7], i.e., no traffic splitting is allowed. If the bandwidth requirement B of a connection is larger than 1, we need to allocate each link B timeslots in one frame. We only allocate free timeslots to every link on a path and update node statuses accordingly. In this way, transmissions along the path of a new established connection will not collide with transmissions of existing connections. However, two links on the current routing path may interfere with each other as well, resulting in intra-flow contention [10] and need to be resolved by the timeslot allocation algorithm.

We use Figure 1 to illustrate the concept of timeslot allocation. In this example, we have $B = 1$ and $K = 6$. The routing path P is given as $\langle v_1, v_2, v_3, v_4, v_5 \rangle$. In this figure, different boxes are used to distinguish different node statuses in a specific timeslot. The labels on the bottom show free timeslots available for each link. In this example, there actually exists a collision-free schedule, which is shown in the upper figure. However, if we assign timeslot 2 to link (v_1, v_2) as shown in the lower figure, then we can not assign the only available free timeslot 2 to link (v_3, v_4) since these two links interfere with each other (collision will happen on node v_2 if transmitting simultaneously). Therefore, timeslot allocation for a given path is not trivial. Without a proper method, a connection request might be rejected even though there exists a collision-free schedule.

Now we are ready to define our optimization problems. Let ρ be a connection request with source node s, destination node t, and bandwidth requirement B.

Definition 2 (BANDRA): *The bandwidth guaranteed routing and timeslot allocation (BANDRA) problem seeks a single path from s to t along with a collision-free schedule such that the total path cost is minimum among all admissible s–t single paths.*

The BANDRA problem seems very difficult since even seeking a collision-free schedule for a given single path with bandwidth requirement $B > 1$ is hard to solve [7]. Therefore, we propose a two-step scheme for this problem, i.e., we break the BANDRA problem into two sub-problems, the routing problem and the timeslot allocation problem, and solve them separately. Our scheme works as follows: we first run one of our routing algorithms to find a path, then use our timeslot allocation algorithm to allocate the required bandwidth along the found path. If the allocation algorithm fails to obtain a bandwidth guaranteed collision-free schedule, block the connection request; otherwise, admit it.

The timeslot allocation problem is to find a collision-free schedule for a given path. For the routing part, simply choosing a minimum cost path for every connection may overload some spatially close wireless links, which eventually makes the found path not admissible. In the following, we introduce some concepts to capture the influence of interference and define two interference-aware routing problems. Let e be a link and IE_e be the set of links that interfere with link e.

Definition 3 (Link Interference): *The link load of e, denoted by L(e), is the sum of the bandwidth of the existing connections that use link e. The link interference of e, denoted by I(e), is* $\sum_{e' \in IE_e} L(e')$.

Essentially, L(e) represents the amount of traffic going through link e. We define the interference value of a link e as the weighted (by link load) sum of the links that interfere with e.

Definition 4 (Path Interference): *Let P be a path for the current connection request. The maximum path interference of P, denoted by* $I_{max}(P)$, *is* $max\{I(e) | e \in P\}$. *The total path interference of P, denoted by* $I_{sum}(P)$, *is* $\sum_{e \in P} I(e)$.

Given a connection request, together with a cost threshold, we are interested in finding a cost bounded path which either minimizes the maximum interference or minimizes the total interference. These problems are formally defined below as the MICB problem and the TICB problem. Two auxiliary problems, MIBC and TIBC, are also defined to ease the description of the algorithms for the MICB and TICB problems. In the following four definitions, ρ represents a connection request with source node s, destination node t and bandwidth requirement B. $\mathcal{C}$ represents a cost threshold.

Definition 5 (MICB): *The min-max interference cost-bounded (MICB) routing problem asks for an* *s–t path with minimum–maximum interference among all s–t paths whose cost is bounded by $\mathcal{C}$.*

Definition 6 (TICB): *The min-total interference cost-bounded (TICB) routing problem asks for an s–t path with minimum total interference among all s–t paths whose cost is bounded by $\mathcal{C}$.*

Definition 7 (MIBC): *Let I_{max} be a maximum interference threshold. The maximum interference bounded minimum cost (MIBC) routing problem asks for an s–t path with minimum cost among all s–t paths whose maximum interference is bounded by I_{max}.*

Definition 8 (TIBC): *Let I_{sum} be a total interference threshold. The total interference bounded minimum cost (TIBC) routing problem asks for an s–t path with minimum cost among all s–t paths whose total interference is bounded by I_{sum}.*

4. ILP Formulation

Before presenting the ILP formulation, we need to define some decision variables and introduce several notations.

(i) X_j is a binary variable which is equal to 1 if link j is chosen as a link in the routing path. Otherwise, it is 0.
(ii) Y_j^k is a binary variable which is equal to 1 if link j is chosen as a link in the routing path and assigned to timeslot k; otherwise, it is 0.

We use E_i^{out} (E_i^{in}) to denote the set of outgoing (incoming) links at node i. FS_j is used to denote the set of free timeslots available for link j. Our ILP formulation for the BANDRA problem is presented in the following.

$$\text{minimize} \sum_{j=1}^{m} X_j \tag{1}$$

subject to:

$$\sum_{j \in E_i^{\mathrm{out}}} X_j - \sum_{j \in E_i^{\mathrm{in}}} X_j = 0; \quad \forall i \in V \backslash \{s, t\} \tag{2}$$

$$\sum_{j \in E_s^{\mathrm{out}}} X_j = 1; \tag{3}$$

$$\sum_{k \in FS_j} Y_j^k = X_j * B; \quad \forall j \in E \tag{4}$$

$$\sum_{l \in IE_j} Y_l^k \leq 1; \quad \forall k \in FS_j; \quad \forall j \in E \tag{5}$$

The objective function (1) is set to minimize the cost (hop-count) of the path. Constraints 2 and 3 guarantee that a single path is selected since the unit integer flow is enforced and flow conservation constraints are satisfied. If link j is selected, then B free timeslots should be allocated to link j, which is ensured by the bandwidth constraint (4). Constraint (5) makes sure that a collision-free schedule is assigned, i.e., at any free timeslot, at most one transmission can happen in any set of links with pairwise interference. Clearly, a minimum hop-count admissible path and its corresponding schedule can be found by solving our ILP, as long as such a path exists.

It is well-known that ILPs may take very long to solve. Therefore, we present our two-step scheme in the following sections.

5. Interference-Aware Routing

Like mentioned before, we will cope with the routing and timeslot allocation separately. We present two optimal algorithms to solve the proposed routing problems in this section.

An efficient algorithm for solving the MIBC problem is described as Algorithm 1. We will use this algorithm as a subroutine for solving the MICB problem.

Algorithm 1: Solving MIBC

Step_1	Construct an auxiliary digraph $GM(V_M, E_M)$ where V_M contains all the wireless nodes and E_M contains all the wireless links whose interference values are at most $I_{\max}$.
Step_2	Apply a shortest path algorithm to compute an *s–t* path P in *GM* with minimum cost.

Theorem 1: *Algorithm 1 correctly computes an s–t path with minimum cost among all s–t paths whose maximum interference value is no more than the given threshold $I_{\max}$, provided that there exists such an s–t path. In addition, the worst-case running time of Algorithm 1 is $O(n^2)$.*

Proof: It follows from the construction of graph *GM* that each link in *GM* has an interference value no more than $I_{\max}$. If there exists an *s–t* path P whose maximum interference is no more than $I_{\max}$, then P must be an *s–t* path in *GM*. Therefore, our algorithm correctly computes an *s–t* path with minimum cost among all *s–t* paths whose maximum interference value is no more than $I_{\max}$.

Step_1 takes $O(n^2)$ time, since the original network G includes $O(n)$ nodes and $O(n^2)$ links. Step_2 takes $O(n^2)$ time if breadth first search (BFS) is used for computing the minimum hop-count path. This completes the proof. □

Algorithm 2: Solving MICB

Step_1	Use bisection on all possible link interference values to find the minimum maximum interference value $I_{\max}$ so that the path P computed by Algorithm 1 with maximum interference threshold $I_{\max}$ has a cost no more than $\mathcal{C}$. Output the path P.
Step_2	Update the load of each link on path P. Update the link interference values in G.

Theorem 2: *Algorithm 2 correctly computes an s–t path with min-max interference among all s–t paths whose cost is no more than $\mathcal{C}$, provided that there exists such an s–t path. In addition, the worst-case running time of the algorithm is $O(n^3)$,*

Proof. The correctness of the algorithm follows from the correctness of Algorithm 1. For the worst-case time complexity, Step_1 will take $O(n^2 \log n)$ since the number of possible link interference values is equal to the number of wireless links, which is $O(n^2)$.

In the following, we show that Step_2 takes $O(n^3)$ time. Once an *s–t* path P is computed, we need to update the load and interference of each of the $O(n^2)$ wireless links in the network. Note that there are at most $(n-1)$ links on path P. The load of a link will change due to the establishment of path P if and only if the link is on path P. We can update the loads of all links on path P in $O(n)$ time. Updating link interferences is more involved because there are changes in interference values of links on path P as well as links not on path P. For each link e. We count the number of links on P interfering with e. Let this value be $N_I(e)$. We increase its interference value by $(N_I(e) \times B)$. This will take $O(n^3)$ time, since there are $O(n^2)$ links in the network and $O(n)$ links on path P. This completes the proof of Theorem 2. □

Please note that the link load and interference values will be updated if and only if the corresponding connection request is admitted. Using the same argument in the time analysis of Theorem 2, we can prove that it takes $O(n^3)$ time to update the link load and interference values when an existing connection leaves the network. In the rest of this section, we turn our attention to solving the TICB problem. Algorithm 3 for solving the TIBC problem can be used as a subroutine for this purpose.

Algorithm 3: Solving TIBC

Step_1	Construct an auxiliary digraph $GT(V_T, E_T)$ as follows. For each wireless node v in G, there are $(I_{\text{sum}}+1)$ nodes $v^0, v^1, \ldots, v^{I_{\text{sum}}}$ in V_T. For each link (u,v) in the network such that the interference value $I(u,v)$ is at most I_{sum}, E_T contains the following directed links: $(u^i, v^{i+I(u,v)})$, $0 \leqslant i \leqslant I_{\max} - I(u,v)$. The costs of all such edges are set to 1. E_T also contains zero cost edges (t^{i-1}, t^i) for $i = 1, 2, \ldots, I_{\text{sum}}$.
Step_2	Apply a shortest algorithm to compute an s^0–$t^{I_{\text{sum}}}$ path π in GT with minimum cost. Obtain the corresponding path P in G.

Theorem 3: *Algorithm 3 correctly computes an s–t path with minimum cost among all s–t paths whose total interference value is no more than I_{sum}, provided that there exists such an s–t path. In addition, the worst-case running time of Algorithm 3 is $O(n^2 I_{\text{sum}})$.*

Proof: It follows from the construction of the graph GT that there is an *s–t* path in the original network G with total interference no more than the given threshold I_{sum} if and only if there is an s^0–$t^{I_{\text{sum}}}$ path in GT. In addition, each s^0–$t^{I_{\text{sum}}}$ path in GT corresponds to an *s–t* path in the original network simply by ignoring the superscript. Therefore, the algorithm correctly computes an *s–t* path with minimum cost among all *s–t* paths whose total interference value is no more than I_{sum}, provided that there exists such an *s–t* path.

For the time complexity analysis, we note that GT has $O(nI_{\text{sum}})$ nodes and $O(n^2 I_{\text{sum}})$ links. Therefore Step_1 and Step_2 require $O(n^2 I_{\text{sum}})$ time in the worst case if BFS is used for computing the minimum hop-count path. This proves the theorem. □

Algorithm 4: Solving TICB

Step_1	Use bisection on the possible values of total path interference to find the minimum total interference value I_{sum} so that the path P computed by Algorithm 3 with total interference threshold I_{sum} has a cost no more than $\mathcal{C}$. Output path P.
Step_2	Update the load of each link on path P. Update the link interference values in G.

Theorem 4: *Algorithm 4 correctly computes an s–t path with min-total interference among all s–t paths whose cost is no more than $\mathcal{C}$, provided that there exists an s–t path with cost no more than $\mathcal{C}$. In addition, the worst-case running time of Algorithm 4 is $O(n^4 HL_{\max} \times (\log n + \log(HL_{\max})))$, where H is the total number of existing connections in the network and $L_{\max}$ is the maximum bandwidth among all existing connections.*

Proof. The correctness of the algorithm follows from the correctness of Algorithm 3. For the worst-case time complexity, we only need to analyze the maximum possible values of the total interference of a path (which consists of $O(n)$ links). Let e be a wireless link, for each of the H existing connections, e interferes with at most $(n-1)$ links used by that connection whose bandwidth is bounded by $L_{\max}$. The interference of link e is bounded by $(n \times H \times L_{\max})$. Therefore, the total interference value of a path I_{sum} is bounded by $(n^2 \times H \times L_{\max})$. This implies that the worst-case time complexity of Algorithm 4 is $O(n^4 HL_{\max} \times (\log n + \log(HL_{\max}))$. This completes the proof. □

6. Bandwidth Allocation

By finding interference-optimal cost-bounded paths, we try to avoid interfering existing traffic or being interfered by existing traffic, with the hope that this will make it easier to allocate required bandwidth for the found paths successfully. The cost threshold is usually set to a bound ratio β multiplied by the path cost of the corresponding minimum cost path. Since there may exist several paths with the same costs (hop-counts) for a specific source–destination pair, we can obtain a minimum cost path with smaller interference value than other corresponding minimum cost paths by using our algorithms and setting $\beta = 1$ (note that minimum hop-count paths are not unique). Hence, setting $\beta = 1$ or slightly larger than 1 will be helpful for timeslot allocation without sacrificing the cost efficiency. After finding a path for a connection request, we need to schedule the transmissions along the found path to make sure the required bandwidth is satisfied and no collision will happen. In the following, we will prove a lemma, which characterizes a nice property of a routing path obtained by our routing algorithms.

Lemma 1: *Let* (v_{s_1}, v_{s_2}), (v_{s_2}, v_{s_3}), *and* (v_{s_3}, v_{s_4}) *be three consecutive links on any path. They must interfere with each other. Let (x, y) and (u, v) be any two links on a path P ((x, y) is the upstream link of (u, v)) found by the shortest path algorithm, Algorithm 2 with* $\beta = 1$ *or Algorithm 4 such that the subpath from y to u contains at least two hops, then (x, y) and (u, v) will not interfere with each other.*

Proof. It is easy to prove the first part. Any two consecutive links on a path will interfere with each other since a node cannot transmit and receive at the same time. Link (v_{s_1}, v_{s_2}) and (v_{s_3}, v_{s_4}) will interfere with each other since collision will happen on node v_{s_2} if two transmissions are allocated to the same timeslot.

The second part of this lemma can be shown by contradiction. Suppose link (v_{s_1}, v_{s_2}) and (v_{s_4}, v_{s_5}) are two links which are two hops apart and interfere with each other. Then $D_{v_{s_1}}$ includes v_{s_5} or $D_{v_{s_5}}$ includes v_{s_2}. In the first case, if we replace links (v_{s_1}, v_{s_2}), (v_{s_2}, v_{s_3}), (v_{s_3}, v_{s_4}), (v_{s_4}, v_{s_5}) with a single link (v_{s_1}, v_{s_5}), we will have a new path P'. The hop-count of P' is less than P. If P is found by the shortest path algorithm, P violates the cost optimality. If it is found by Algorithm 2 with $\beta = 1$, P violates the cost constraint as well. If P is found by Algorithm 4, it violates the interference optimality since the total interference of P' is less than that of P. If $D_{v_{s_5}}$ includes v_{s_2}, then $D_{v_{s_2}}$ includes v_{s_5} because we assume that every node has the same transmission range R. Then if we replace links (v_{s_1}, v_{s_2}), (v_{s_2}, v_{s_3}), (v_{s_3}, v_{s_4}), (v_{s_4}, v_{s_5}) with two links (v_{s_1}, v_{s_2}) and (v_{s_2}, v_{s_5}), we have the same kinds of contradictions. Therefore, link (v_{s_1}, v_{s_2}) and (v_{s_4}, v_{s_5}) do not interfere with each other. If hop distance in between is larger than two, we can show its correctness in the same way. □

For timeslot allocation problem, we first consider the special case where the bandwidth requirement of every connection request is one. We propose a fast and simple algorithm to solve it optimally. We use FS_i to denote the set of free timeslots of the ith link on P and $SC(i)$ to denote the timeslot assigned to the ith link on P in the found schedule, where P is the found path and $(1 \le i \le |P|)$. Every link keeps a list SL_i and every entry of the list is a three-tuple (a, b, c). Field a of the tuple records a free timeslot from FS_{i-2}, field b records a free timeslot from FS_{i-1}, and field c records a free timeslot from FS_i. For easy presentation, we assume that FS_0 includes only one virtual free timeslot, 0 and FS_{-1} includes only one virtual free timeslot, -1. The entry is considered as a valid entry if and only if the values in these three fields are different from each other and it is not identical with another existing entry. The algorithm is formally presented as Algorithm 5.

Algorithm 5: Bandwidth Allocation ($B = 1$)

```
Step_1 Compute FS_i for each link on P;
       SL_0 := {(-2, -1, 0)};
       for i = 1 to |P|
           SL_i := Ø;
       endfor
```

Step_2 for $i = 1$ to $|P|$
  forall $sl2 \in FS_{i-2}$
    forall $sl1 \in FS_{i-1}$
      forall $sl \in FS_i$
        new_entry := ($sl2$, $sl1$, sl);
        if (new_entry is valid)
          add new_entry into SL_i;
        endif
      endforall
    endforall
  endforall
endfor
Step_3 if ($SL_{|P|} \neq \emptyset$)
  select entry $\in SL_{|P|}$;
  $SC(|P|)$:= entry.c;
  for $i = |P| - 1$ to 1
    select new_entry $\in SL_i$ s.t.
    new_entry.c = entry.b and
    new_entry.b = entry.a;
    $SC(i)$:= new_entry.c;
    entry := new_entry;
  endfor
else
  Output no collision-free schedule is found.
endif

We are going to use the same example in Figure 1 to illustrate the proposed algorithm. Refer to Figure 2. Here, the labels on the top indicate the free timeslots in each link. By running Step_2, we have $SL_1 = \{(-1, 0, 2), (-1, 0, 6)\}$. Then we obtain $SL_2 = \{(0, 2, 3), (0, 6, 2), (0, 6, 3)\}$, $SL_3 = \{(6, 3, 2)\}$, and $SL_4 = \{(3, 2, 4), (3, 2, 6)\}$. After running Step_3, we have a collision-free schedule for each link, i.e., $SC(4) = 4$, $SC(3) = 2$, $SC(2) = 3$, and $SC(1) = 6$.

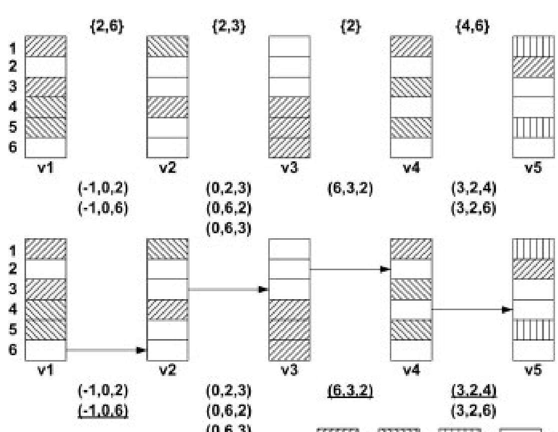


Fig. 2. Timeslot allocation ($B = 1$).

After finding a schedule, we need to update node status information in the following way: for each link (u, v) on the path, which is assigned timeslot k, $S(u, k) = T$, $S(v, k) = R$. $S(x, k) = I$ if it was F for each $x \in D_u \setminus \{u, v\}$; otherwise, no change for the status. After tearing down a connection, we change statuses of affected nodes similarly.

Theorem 5: *Algorithm 5 correctly computes a collision-free schedule for path P, provided that there exists such a schedule and its worst-case running time is* $O(K^3|P|)$.

Proof. In each iteration of Step_2, the algorithm computes all possible valid allocation combinations for ith, $(i-1)$th, and $(i-2)$th links and keep them in SL_i. When considering timeslot allocation for ith link, we only consider valid assignment combinations with $(i-1)$th and $(i-2)$th links. So every entry in the list only has three fields. This is because the timeslot assignment of $(i-3)$th link does not affect that of ith link according to Lemma 1. Assigning timeslot using Step_3 guarantees that any three consecutive links on P are assigned different timeslots because of the definition of the valid entry. Computing SL_i for every link will take $O(K^3)$ time since every link has at most K free timeslots. We need to compute this list for all links on P. Therefore, both Step_2 and Step_3 take $O(K^3|P|)$ time. Step_1 takes linear time in terms of path length. Hence, the worst-case time complexity of Algorithm 5 is $O(K^3|P|)$. □

If there are connections with bandwidth requirement $B > 1$, we apply the same idea for timeslot allocation. However, each field of the aforementioned SL list become a B-tuple here, each of which keeps a valid timeslot assignment combination for a specific link. We note that there exists at most $Q = \binom{K}{B}$ possibilities for timeslot allocation of a single link. Therefore, this list has at most Q^3 entries. When K is relatively large, computing such a list may take very long time. So we propose to keep only Z valid entries in SL list for each link, where Z is a given parameter. By tuning Z, we can find a tradeoff between running time and performance. If the total number of valid entries is less than Z, then we simply keep all combinations. In order to increase the chance of finding a collision-free schedule successfully, the entries kept by our algorithm should be as different as possible.

7. Performance Evaluation

In this section, we evaluate the performance of our algorithms via simulations. We consider static wireless networks with nodes randomly located in a $900 \times 900\,\mathrm{m}^2$ region. Every node has a fixed transmission range of 250 m. In all scenarios, each connection request is generated with a randomly chosen source-destination pair. Totally, 1000 connection requests will be injected. The lifetime (T) of each connection specifies how many time units it will last. It is also a random number uniformly distributed between 1 and a maximal value ($T_{\max}$). In the simulations, we compare the performance of our two-step scheme with the optimal solution given by our ILP formulation in terms of blocking ratio (the ratio between the number of admitted connections and total number of connection requests). We will apply both our interference-aware algorithms and the shortest (minimum hop-count) path algorithm for routing. The bound ratio β is always set to 1 for our routing algorithms, which guarantees that one of minimum hop-count paths for a specific source-destination pair will be selected every time.

The following three system parameters will influence the performance: network size (n), the frame length (K), the traffic load. In each scenario, we adjust the traffic load by fixing the mean request arrival interval to be ten time units and change the maximal connection lifetime ($T_{\max}$) from 250 to 500 time units. We run simulations on networks with different sizes and frame lengths. We divide our simulation scenarios into two parts. In the first part, the connection requests always come with unit bandwidth requirement ($B = 1$). The results are shown in Figures 3–5. In the second part, four types connection requests with bandwidth requirement 1,2,3, and 4, respectively, are randomly generated. If $B = 1$, Algorithm 5 will be called and if $B > 1$, the aforementioned heuristic with $Z = 1000$ will be used. We present the corresponding simulation results in Figures 6–8. In each figure, OPT, MICB, TICB, and SP stand for optimal solutions,

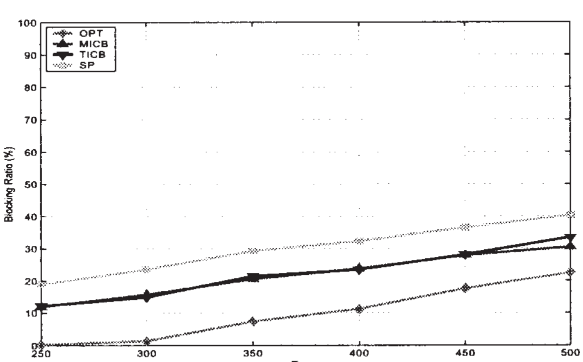


Fig. 3. $n = 20, K = 20, B = 1$.

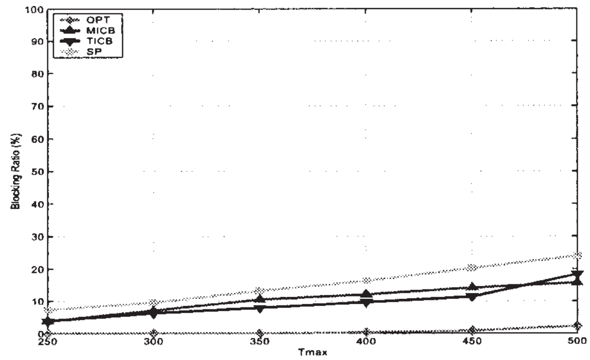


Fig. 4. $n = 20, K = 30, B = 1$.

solutions given by MICB, TICB, and shortest path routing algorithm along with our bandwidth allocation algorithms, respectively. We briefly call them MICB, TICB, and SP scheme in the following.

We make the following observations from the simulation results.

- Our MICB scheme performs best in most of cases. For the special case ($B = 1$), the blocking ratios given by this scheme are approximately 10% larger than those of the optimal solution on average

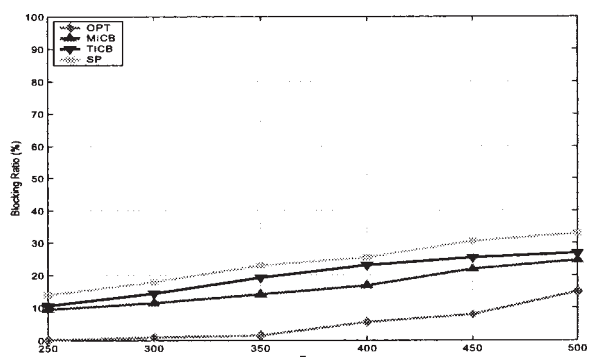


Fig. 5. $n = 30, K = 20, B = 1$.

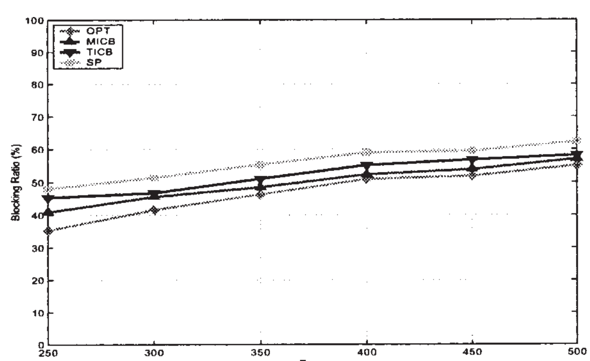


Fig. 6. $n = 20, K = 20, B = [1, 4]$.

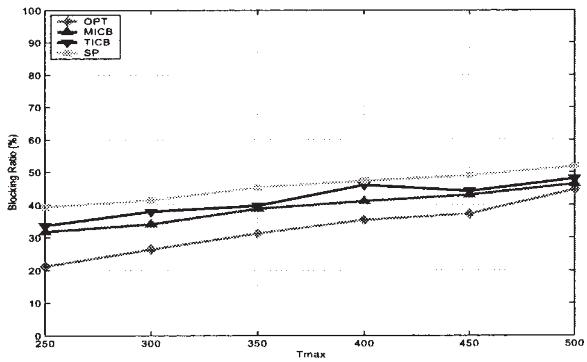


Fig. 7. $n = 20, K = 30, B = [1, 4]$.

(Figures 3–5). For the general case ($B = [1, 4]$), the blocking ratio increases dramatically no matter which scheme is used because of heavier traffic load. However, the average difference of blocking ratio between solutions by our MICB scheme and optimal solutions is only about 4% (Figures 6–8). Our MICB and TICB schemes always outperform the SP scheme with regards to blocking ratio in all simulation runs.

- The blocking ratio increases with the increase of the maximal connection lifetime ($T_{\max}$) because longer lifetimes usually lead to more connections in the network at one time, which will essentially reduce the chance for a connection to be admitted. As expected, we can see that the blocking ratio drops substantially when the frame length is prolonged from 20 timeslots to 30 timeslots. We can see from results that the network size will also influence the blocking ratio. Suppose the region size is fixed, the more wireless nodes we have in a network, the less blocking ratio we can obtain (Figures 3 and 5, 6 and 8). This is because the traffic will be more spread out in a larger network, which gives a connection more chance to be admitted.

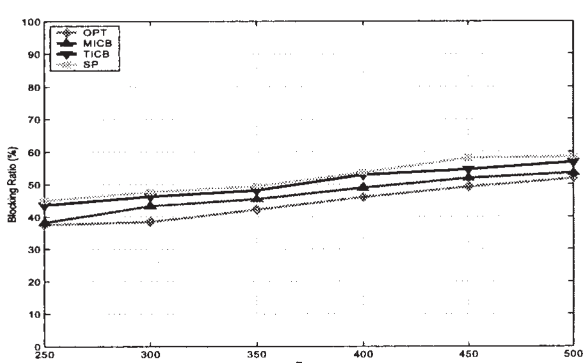


Fig. 8. $n = 30, K = 20, B = [1, 4]$.

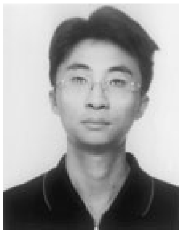
8. Conclusions

In this paper, we have formally defined BANDRA problem in TDMA-based multihop wireless networks with dynamic traffic. We have presented an ILP formulation to provide optimal solutions. Considering its computational complexity, we have proposed a two-step scheme, i.e., computing a routing path first and then allocating bandwidth along the found path. We have presented two routing algorithms to find interference-optimal cost-bounded paths. We have also presented an optimal bandwidth allocation algorithm to allocate timeslots along the found paths for connection requests with unit bandwidth requirements and an effective heuristic for the general case. Simulation results show that the average blocking ratio given by our MICB scheme is only 7% larger than that of the optimal solution. Moreover, our interference-aware routing algorithms always outperform the shortest path routing.

References

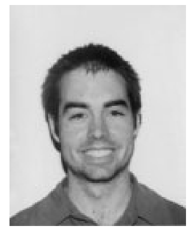
1. Burkhart M, Rickenbach PV, Wattenhofer R, Zollinger A. Does topology control reduce interference. In *Proceedings of ACM MobiHoc* 2004, pp. 9–19.
2. Gupta P, Kumar PR. The capacity of wireless networks. *IEEE Transactions on Information Theory* 2000; **46**(2): 388–404.
3. Jain K, Padhye J, Padmanabhan V, Qiu L. Impact on interference on multihop wireless network performance. In *Proceedings of ACM MobiCom* 2003, pp. 66–80.
4. Liao WH, Tseng YC, Shih KP. A TDMA-based bandwidth reservation protocol for QoS routing in a wireless mobile ad hoc network. In *Proceedings of IEEE ICC* 2002, pp. 3186–3190.
5. Lin CR. Admission control in time-slotted multihop mobile networks. *IEEE Journal on Selected Areas in Communications* 2001; **19**(10): 1974–1983.
6. Lin CR, Liu JS. QoS routing in ad hoc wireless networks. *IEEE Journal on Selected Areas in Communications* 1999; **17**(8): 1426–1438.
7. Liu H, Jia X, Li D, Lee CH. Bandwidth guaranteed call admission in TDMA/CDMA ad hoc wireless networks. *Elsevier Journal of Ad Hoc Networks* 2000; **3**(6): 689–701.
8. Kodialam M, Nandagopal T. Characterizing achievable rates in multihop wireless networks: the joint routing and scheduling problem. In *Proceedings of ACM MobiCom* 2003, pp. 42–54.
9. Yeh CH, Zhou H, Mouftah HT. A time space division multiple access (TSDMA) protocol for multihop wireless networks with access points. In *Proceedings of IEEE VTC* 2002, pp. 1185–1189.
10. Zhai H, Wang J, Fang Y. Distributed packet scheduling for multihop flows in ad hoc networks. In *Proceedings of IEEE WCNC* 2004, pp. 1081–1086.
11. Zhu C, Corson S. QoS routing for mobile ad hoc networks. In *Proceedings of IEEE INFOCOM* 2002, pp. 958–967.

Authors' Biographies



Jian Tang received the B.E. and M.E. degrees from Beijing University of Posts and Telecommunications, China, in 1998 and 2001 respectively. Currently, he is a Ph.D. candidate in the Computer Science and Engineering Department of Arizona State University. His research interest is in the area of wireless networking with a special emphasis on cross-layer design, QoS provisioning, scheduling and routing algorithm design. He is a student member of the IEEE.

Guoliang Xue is a professor in the Department of Computer Science and Engineering at Arizona State University. He received his Ph.D. in Computer Science from the University of Minnesota in 1991 and has held previous positions at the Army High Performance Computing Research Center and the University of Vermont. His research interests include efficient algorithms for optimization problems in networking, with applications to fault tolerance issues in WDM optical networks and robustness issues in mobile wireless ad hoc and sensor networks. He has published over 100 papers in this area. He is an associate editor of the *IEEE Transactions on Circuits and Systems-I* and an Editorial Board Member of the *Computer Networks Journal* and the *IEEE Network Magazine*.



Christopher Chandler is currently pursuing a bachelors degree in Computer Science at Arizona State University. He has been involved with research in wireless networks for about a year as an undergraduate. His other research interests are primarily in information assurance.