

Interference-Aware Topology Control and QoS Routing in Multi-Channel Wireless Mesh Networks *

Jian Tang, Guoliang Xue and Weiyi Zhang
Department of Computer Science and Engineering
Arizona State University
Tempe, AZ 85287-8809, USA.
{jian.tang, xue, weiyi.zhang}@asu.edu.

ABSTRACT

The throughput of wireless networks can be significantly improved by multi-channel communications compared with single-channel communications since the use of multiple channels can reduce interference influence. In this paper, we study interference-aware topology control and QoS routing in IEEE 802.11-based multi-channel wireless mesh networks with dynamic traffic. Channel assignment and routing are two basic issues in such networks. Different channel assignments can lead to different network topologies. We present a novel definition of co-channel interference. Based on this concept, we formally define and present an effective heuristic for the minimum INterference Survivable Topology Control (INSTC) problem which seeks a channel assignment for the given network such that the induced network topology is interference-minimum among all K -connected topologies. We then formulate the Bandwidth-Aware Routing (BAR) problem for a given network topology, which seeks routes for QoS connection requests with bandwidth requirements. We present a polynomial time optimal algorithm to solve the BAR problem under the assumption that traffic demands are splittable. For the non-splittable case, we present a maximum bottleneck capacity path routing heuristic. Simulation results show that compared with the simple common channel assignment and shortest path routing approach, our scheme improves the system performance by 57% on average in terms of connection blocking ratio.

Categories and Subject Descriptors

C.2.1 [Computer-Communication Networks]: Network Architecture and Design - *Wireless Communication*

*Research supported in part by ARO grant W911NF-04-1-0385 and NSF grant CCF-0431167. The information reported here does not reflect the position or the policy of the federal government.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

MobiHoc'05, May 25–27, 2005, Urbana-Champaign, Illinois, USA.
Copyright 2005 ACM 1-59593-004-3/05/0005 ...\$5.00.

General Terms

Algorithm, Design, Performance

Keywords

Wireless Mesh Network, QoS Routing, Topology Control, Channel Assignment.

1. INTRODUCTION

A wireless mesh network ([1, 21, 22]) is a multihop wireless network consisting of a large number of wireless nodes, some of which are called gateway nodes and connected with a wired network. It has attracted much research attention due to its potential applications, including last-mile broadband Internet access, neighborhood gaming, Video-on-Demand (VoD), distributed file backup, video surveillance and so on ([3]). Due to the limited channel capacity, the influence of interference, the large number of users and the emergence of real-time multimedia applications, improving network throughput and supporting QoS have become two critical requirements in such networks.

Using multiple channels instead of a single channel in multihop wireless networks has been shown to be able to improve the network throughput dramatically ([21, 22]). The IEEE 802.11b standard and IEEE 802.11a standard ([12, 13]) offer 3 and 12 non-overlapping channels respectively. We consider a multi-channel wireless mesh network, in which every node is equipped with multiple Network Interface Cards (NICs) and each of them can be tuned to one of the available channels. A pair of NICs can communicate with each other if they are on the same channel and are within the transmission range of each other. The throughput improvement in such networks is brought mainly by allowing multiple simultaneous transmissions within a neighborhood. In a single channel wireless network, two transmissions in a neighborhood are not allowed to happen at the same time because of contention for the shared wireless channel. However, in a multi-channel network, no collision will be caused by such simultaneous transmissions as long as they work on different channels.

QoS routing in multihop wireless networks is very challenging due to interference among different transmissions. Even in a multi-channel wireless network, interference still exists since two transmissions may interfere with each other if they are using the same channel. A QoS connection request usually comes with a bandwidth requirement and QoS routing seeks a source to destination route with requested

bandwidth. If no such route can be found, the connection request should be blocked. In order to solve the QoS routing problem in multihop wireless networks, we need to consider two types of contentions: *inter-flow contention* and *intra-flow contention* ([26]). Suppose P is a candidate route for the current connection request. Inter-flow contentions are the contentions caused by the interference between a wireless link on P and a wireless link which is not on P and used by existing connection requests. Intra-flow contentions are the contentions caused by the interference between a wireless link on P and another wireless link on P . Intra-flow contentions are very hard to determine during the phase of route discovery since the route is still unknown at that time.

To our best knowledge, this is the first paper addressing QoS provisioning in an IEEE 802.11-based multi-channel wireless mesh network. We study a wireless mesh network with dynamic traffic, i.e., connection demands have random sources, destinations and arrival times, without assuming all traffic demands are given *a priori* by a traffic profile as in [21, 22]. We believe that this dynamic traffic model is more useful in reality because considering the aforementioned applications in the future, we should expect not only some traffic from wireless nodes to Internet via gateway nodes but also substantial random and unpredictable traffic among wireless nodes within the mesh network. It is hard to precisely predict the traffic demands in advance. How to assign channel for each NIC and how to route traffic are two basic issues in such networks. For channel assignment, we adopt a static assignment scheme ([21, 22]), i.e., assigning channels for each NICs before connection requests arrive and keep the computed assignment for a long period of time. Changes or adjustments will only be made if a set of new nodes are added or some nodes fail. The dynamic scheme, i.e., assigning channels for NICs in route discovery and allowing channel switching during packet transmissions, is not very suitable for wireless mesh networks. Because dynamic assignment may cause the deafness problem (transmitter and intended receiver happen to be on different channels) and need fine grained synchronization ([15, 24]). In [15], the authors propose a hybrid channel assignment method to solve the above problems, which assigns one NIC of each node statically to a common control channel, and allows other interfaces dynamically switch among other data channels. However, this method will hold back the throughput improvement, especially for the case in which the number of NICs in each node is very small. Moreover, the channel switching delay of current commercial IEEE 802.11 hardware is in the range of a few milliseconds to a few hundred microseconds ([15]), which is intolerable for most real-time multimedia applications. Once a channel assignment is given, the network topology can be determined. Intuitively speaking, we want the channels assigned to the NICs in a common neighborhood to be as different as possible such that interference can be reduced. In addition, we need to preserve the network connectivity and support survivability.

In this paper, we first present a novel definition of co-channel interference which can capture the impact of interference precisely. Based on this definition and by fully considering both interference and connectivity, we define the *minimum INterference Survivable Topology Control (INSTC)* problem which seeks a channel assignment for the given network such that the induced network topology is interference-minimum among all K -connected topologies. K -

connectivity is required for survivability and load-balancing purposes. We assume the transmission power of each NIC is fixed. So the topology control problem studied here is quite different from all previous topology control problems ([7, 19]) in which the network topology is controlled by carefully adjusting the transmission power at each node. We present an effective heuristic for the INSTC problem which guarantees that the topology is K -connected to enhance survivability. In the second part of the paper, we fully exploit the influence of intra-flow and inter-flow contentions for the multihop QoS routing which has not been well addressed before, especially under the scenario of multi-channel multi-NIC multihop wireless networks. We present an optimal LP-based polynomial time QoS routing algorithm for the given network topology to solve the formulated *Bandwidth-Aware Routing (BAR)* problem by assuming that the traffic demands are splittable. We also present an effective maximum bottleneck capacity path heuristic without assuming the splittability. Like the other higher layer solutions proposed in [15, 21, 22], our scheme can be used without making any change on IEEE 802.11 DCF.

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 interference-aware topology control scheme and QoS routing algorithms in Section 4 and Section 5, respectively. Simulation results are presented in Section 6. We conclude the paper in Section 7.

2. RELATED WORK

Recently, people begin to study multi-channel multihop wireless networks, such as multi-channel wireless mesh networks and multi-channel Mobile Ad hoc NETwork (MANET), since network throughput can be substantially improved by making full use of the non-overlapping channels. In [22], the authors propose and evaluate one of the first IEEE 802.11-based multi-channel multi-hop wireless mesh network architectures. They develop a set of centralized algorithms for channel assignment, bandwidth allocation, and routing. They also present distributed algorithms utilizing only local traffic load information to dynamically assign channels and to route packets in a later paper [21]. Draves *et al.* in [8] present a new metric named Expected Transmission Time/Weighted Cumulative ETT (ETT/WCETT), for multi-radio, multihop wireless networks which can be used for finding a high-throughput path between a source and a destination. They also present a Multi-Radio Link-Quality Source Routing (MR-LQSR) protocol by incorporating this metric. In [15], the authors propose algorithms for channel assignment and routing in multi-channel multi-NIC MANETs. Vaidya *et al.* ([24]) also present a routing protocol for the scenario in which each node only has one NIC. Besides routing protocols, several link layer and MAC layer solutions have also been proposed for multi-channel multihop wireless networks in [2, 5, 23]. We note that previously proposed channel assignment schemes ([21, 22]) are not suitable for the dynamic traffic model here since no traffic demand profile is available as the guideline for channel assignment. Moreover, they do not guarantee K -connectivity. In addition, none of previously known routing schemes can support QoS routing with bandwidth requirements in a 802.11-based multi-channel wireless network.

QoS routing in single channel MANETs has been well

studied in the literature. Perkins *et al.* have extended the basic Ad hoc On-Demand distance Vector (AODV) routing protocol to support QoS [20]. Formats of packets, such as route request (RREQ) packet or route reply (RREP) packet, are modified to specify the service requirements which must be satisfied by the forwarding nodes. Xue and Ganz in [25] introduce a resource reservation-based routing and signaling algorithm named Ad hoc QoS on-demand routing (AQOR), which provides end-to-end QoS support in terms of both bandwidth and end-to-end delay, in IEEE 802.11-based MANETs. Several QoS routing protocols ([16, 17, 18, 27]) have been proposed under the assumption of a TDMA or TDMA-over-CDMA MAC layer. The authors of [10] consider bandwidth as the QoS parameter in TDMA-based wireless ad hoc networks with limited energy resources. A constraint model for the QoS-aware Minimum Energy Multicast (QoS-MEM) problem in terms of Integer Linear Programming (ILP) is presented.

Recent research has shown that interference can make a significant impact on the performance of a wireless network. As a pioneering work, Gupta and Kumar in [11] show that in a wireless network with n identical nodes, the per-node throughput is $\Theta(1/\sqrt{n \log n})$ by assuming random node placement and communication pattern. It becomes $\Theta(1/\sqrt{n})$ under the assumption of optimal node placement and communication pattern. In [14], the authors model the influence of interference using a conflict graph and derive upper and lower bounds on the optimal throughput. Burkhart *et al.* give a concise and intuitive definition of interference in [7]. Based on this definition, they show that most currently proposed topology control algorithms do not effectively constrain interference. They propose algorithms to compute interference-optimal connected subgraphs and spanners. In [19], the authors extend results from [7] and present algorithms for constructing a network topology in wireless ad hoc networks such that the maximum (or average) link (or node) interference of the topology is either minimized or approximately minimized. In [26], the authors present a framework for multihop packet scheduling to achieve maximum throughput by considering the influence of intra-flow and inter-flow contentions.

3. PROBLEM DEFINITION

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

We use a similar network model as described in [21, 22]. The IEEE 802.11 Distributed Coordination Function (DCF) ([12]) is assumed to be used to handle multiple access in the MAC layer. There are totally C non-overlapping frequency channels in the system and each node is equipped with Q NICs where $Q \leq C$. In order to efficiently and fully make use of the network resources, we assume that each NIC is tuned to a channel and that any two NICs at the same node are tuned to different channels. All nodes in the network use the same fixed transmission power, i.e., there is a fixed transmission range ($r > 0$) and a fixed interference range $R > r$ (which is typically 2 to 3 times of r [21]) associated with every node. We also assume that all wireless nodes are stationary.

We use an undirected graph $G(V, E)$ to model the wireless mesh network where V is the set of n vertices and E is the set of m edges. Each vertex $v \in V$ corresponds to a

wireless node in the network with its location known. *By abusing the notation a little bit without confusion, we also use v to denote its corresponding wireless node, or even the location of its corresponding wireless node.* There is an undirected edge $(u, v) \in E$ connecting vertex u and vertex v if $d(u, v) \leq r$, where $d(u, v)$ is the Euclidean distance between u and v . The edge (u, v) in G corresponds to a wireless link between nodes u and v in the wireless network. The use of undirected links reflects the fact that the IEEE 802.11 DCF requires the sender to be able to receive a link layer acknowledgment message from the receiver for every transmitted data packet. Throughout this paper, we will use vertices and nodes interchangeably, as well as edges and links.

A *channel assignment* \mathcal{A} assigns each node $v \in V$ a set $\mathcal{A}(v)$ of Q different channels: $\mathcal{A}(v) \subseteq \{1, 2, \dots, C\}$. The channels in $\mathcal{A}(v)$ correspond to the Q different channels that the Q NICs at node v are tuned to. A channel assignment \mathcal{A} defines a *topology* $G_{\mathcal{A}}(V, E_{\mathcal{A}})$ in the following natural way: There is an *edge* $e = (u, v; k)$ on channel $\lambda(e) = k$ between nodes u and v in $G_{\mathcal{A}}$ if and only if $d(u, v) \leq r$ and $\lambda(e) \in \mathcal{A}(u) \cap \mathcal{A}(v)$. Note that $G_{\mathcal{A}}$ may be a *multi-graph*, i.e., a graph including multiple edges between the same pair of nodes, because a pair of nodes may share two or more channels. An example of channel assignment and its corresponding topology is given in Figure 1, where $C = 3$ and $Q = 2$. The label associated with each node indicates the channels assigned to that node. The label associated with each edge indicates the channel shared by the nodes incident with that edge. Note that the pair of nodes A and B share two channels. Therefore there are two edges connecting vertices A and B in the corresponding topology, one on channel 1 and the other on channel 2.

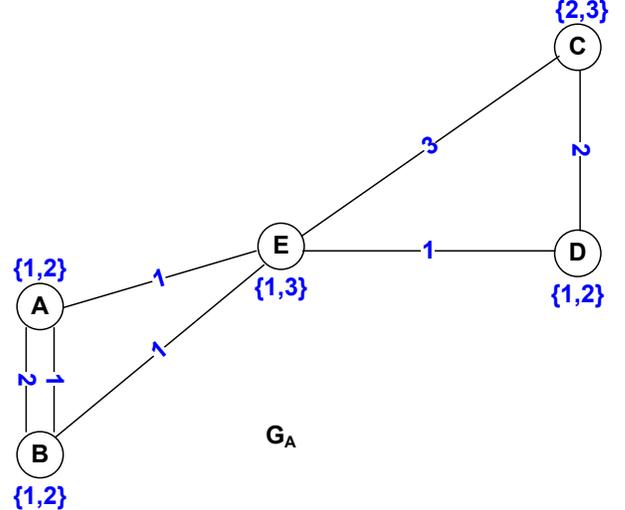


Figure 1: Network topology given by a channel assignment

Half-duplex operation is enforced for each NIC to prevent self-interference, i.e., one NIC can only transmit or receive at one time. However, while one NIC at a node v is transmitting/receiving data on one channel, another NIC at node v can simultaneously transmit/receive data on a *different* channel. Due to the broadcast transmission nature of the radio and the fixed transmission power, we can imagine that

associated with each node u , there is a disk D_u centered at u with radius R (the *interference disk*). Without confusion, we will also use D_u to denote the set of nodes covered by this disk. It follows from the definition that for any pair of wireless nodes u and v , $v \in D_u$ implies $u \in D_v$, and vice versa. Assume that u, v, x, y are wireless nodes such that $d(u, v) \leq r$ and $d(x, y) \leq r$. If node x or node y are covered by one of the interference disks at u and v (correspondingly, node u or node v must be covered by one of the interference disks at x and y) and that there is a channel $k \in \mathcal{A}(u) \cap \mathcal{A}(v) \cap \mathcal{A}(x) \cap \mathcal{A}(y)$, then the link $e = (u, v; k)$ on channel $\lambda(e) = k$ interferes with the link $e' = (x, y; k)$ on channel $\lambda(e') = k$, since simultaneous transmissions along $(u, v; k)$ and $(x, y; k)$ (using channel k) will lead to collision. This definition of *link interference* also includes the cases where the two links share a common node and the case where e and e' are identical (the notion that a link interferes with itself is a technical agreement that simplifies the notations later). In Figure 2, link $(A, B; 2)$ interferes with link $(C, D; 2)$. Let e be a link in G_A . We will use IE_e to denote the set of links in G_A that interferes with link e .

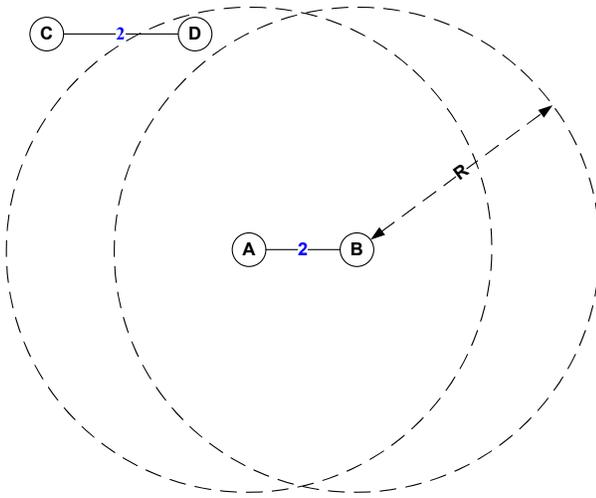


Figure 2: Link $(A, B; 2)$ interferes with link $(C, D; 2)$.

As discussed earlier in this section, a channel assignment defines a corresponding topology. It is desirable to have a channel assignment whose corresponding topology has relatively low interference. We formalize interference-aware topology control and related concepts in the following.

DEFINITION 1 (CO-CHANNEL INTERFERENCE). Let a channel assignment \mathcal{A} and its corresponding network topology G_A be given. Let e be any link in G_A . The **Link Co-Channel Interference** of link e , denoted by $I(e)$, is $|IE_e|$. The **Topology Co-Channel Interference** of G_A , denoted by $I(G_A)$, is $\max_{e \in G_A} I(e)$.

DEFINITION 2 (INSTC PROBLEM). Given the network G and an integer K , the **minimum INterference Survivable Topology Control (INSTC)** problem seeks a channel assignment \mathcal{A} such that the corresponding network topology G_A is K -connected and has the minimum topology co-channel interference among all K -connected topologies induced by channel assignments.

In order to ensure the existence of a K -connected topology G_A , the original network G must be K -connected.

Now we switch our attention to the routing problem. In this paper, we study symmetric communication. The bandwidth requirement in a connection request indicates the total bandwidth required for the communication in both directions. As discussed in Section 1, the intra-flow contention complicates the bandwidth computation. Instead of seeking a single routing path, we compute a splittable flow allocation. Specifically, $f(e, \rho)$ is used to denote the bandwidth to be allocated on link e for connection ρ . The splittable flow approach enables us to compute a route with required bandwidth in polynomial time in a multihop wireless network.

We need to compute the available bandwidth on each wireless link in the given topology when establishing a connection. Let $CAP(e)$ be the (physical) capacity of link e , which is usually a fixed value, e.g. $CAP(e) = 11(\text{Mbps})$, $\forall e \in E_A$ if IEEE 802.11b is used.

DEFINITION 3 (AVAILABLE BANDWIDTH). The **Link Load** of $e \in G_A$, denoted by $L(e)$, is the sum of the bandwidth allocations of all the existing connections that use link e : $L(e) = \sum_{\rho \text{ is an existing connection}} f(e, \rho)$. The **Available Bandwidth** of link $e \in G_A$, denoted by $A(e)$, is $CAP(e) - \sum_{e' \in IE_e} L(e')$.

In our definition of link interference, we have adopted the notion that IE_e contains e itself. This notion is used in the above definition. We note that the above method for computing available bandwidth is a worst-case computation. Suppose that links e_1 and e_2 both interfere with link e , but do not interfere with each other. Then traffic on e_1 and e_2 may be transmitted simultaneously using IEEE 802.11 DCF. In this case, the available bandwidth on link e is actually larger than the value computed based on Definition 3. Therefore $A(e)$ is a lower bound for the actual available bandwidth. This bound is *tight* because it is achievable in the worst case. The defined computation method is necessary to provide required bandwidth for each QoS connection demand. Here, we need not only to allocate enough bandwidth for the incoming connection request, but also to make sure that the bandwidth allocated for existing connections is not affected. Some previous QoS routing papers such as [25] employ similar computation methods.

DEFINITION 4 (BAR PROBLEM). Let there be some existing connections on the topology G_A induced by a given channel assignment \mathcal{A} . Let ρ be a new connection request with source node s , destination node t , and bandwidth requirement B . The **Bandwidth-Aware Routing (BAR)** problem seeks a flow allocation \mathcal{F} such that the total s - t flow is equal to B and that $\sum_{e' \in IE_e} f(e', \rho) \leq A(e)$, $\forall e \in G_A$.

If such a flow allocation can be found, the connection request ρ will be admitted and the available bandwidth will be updated on related links accordingly. Otherwise, the connection request will be blocked.

4. INTERFERENCE-AWARE TOPOLOGY CONTROL

In this section, we will present an effective heuristic for the INSTC problem defined in the previous section.

We say link $(u, v) \in G$ and link $(x, y) \in G$ *potentially interfere* with each other if $x \in D_u \cup D_v$ or $y \in D_u \cup D_v$.

Note again that this condition implies $u \in D_x \cup D_y$ or $v \in D_x \cup D_y$. So the concept of link potential interference is symmetric. We need this concept because our topology control algorithm tries to find a channel assignment (before a channel assignment is known, the actual interference of links are unknown). Therefore the previous definition for co-channel interference cannot be employed here. Similarly, we use PE_e to denote the set of links in G that potentially interfere with link $e \in G$. We need the following definition.

DEFINITION 5 (LINK POTENTIAL INTERFERENCE). *Let e be any link in G . The **Link Potential Interference (LPI)** of e , denoted by $I_P(e)$, is $|PE_e|$.*

Our topology control algorithm is formally presented as Algorithm 1. In the algorithm, we try to compute the channel assignment for the nodes by going through the edges in a K -connected subgraph of the network in a non-increasing order of link potential interference values.

Algorithm 1 Interference-aware topology control

- Step_1 Use binary search on all LPIs to find the smallest possible LPI, I_P^{min} and a subgraph of G , $G'(V, E')$ such that G' is K -connected and includes all links whose LPI is at most I_P^{min} .
- Step_2 Initialize $\mathcal{A}(u)$ to \emptyset for all $u \in V$. Select the link in G' one by one in a descending order of LPIs. Update the channel assignment for nodes u and v corresponding to link $e = (u, v) \in G'$ based on the following rules:
1. If $\mathcal{A}(u) \cap \mathcal{A}(v) \neq \emptyset$, then do nothing.
 2. If $\mathcal{A}(u) \cap \mathcal{A}(v) = \emptyset$ but $|\mathcal{A}(u)| < Q$ and $|\mathcal{A}(v)| < Q$, then $\mathcal{A}(u) := \mathcal{A}(u) \cup \{k\}$, $\mathcal{A}(v) := \mathcal{A}(v) \cup \{k\}$, where k is the least used channel in PE_e .
 3. If $\mathcal{A}(u) \cap \mathcal{A}(v) = \emptyset$, but $|\mathcal{A}(u)| < Q$ and $|\mathcal{A}(v)| = Q$, then $\mathcal{A}(u) := \mathcal{A}(u) \cup \{k\}$, where k is the least used channel in PE_e among channels in $\mathcal{A}(v)$.
 4. If $\mathcal{A}(u) \cap \mathcal{A}(v) = \emptyset$, $|\mathcal{A}(u)| = |\mathcal{A}(v)| = Q$, let k be the least used channel in PE_e among channels in $\mathcal{A}(u) \cup \mathcal{A}(v)$. Without loss of generality, assume that $k \in \mathcal{A}(u)$. Let $k' \neq k$ be a channel in $\mathcal{A}(v)$ that is most used in PE_e . Replace k' in $\mathcal{A}(v)$ by k . For all the edges (v, w) already considered for which the change of $\mathcal{A}(v)$ makes $\mathcal{A}(v) \cap \mathcal{A}(w) = \emptyset$ (this implies $k' \in \mathcal{A}(w)$), replace k' in $\mathcal{A}(w)$ by k . This replacement may be performed recursively.
- Step_3 Assign nodes having unassigned NICs with the least used channels among assigned channels from their neighboring nodes.
-

THEOREM 1. *Algorithm 1 correctly computes a node channel assignment whose corresponding network topology is K -connected in $O(Kn^3 \log m + m^2)$ time.*

PROOF. The induced topology $G_{\mathcal{A}}$ based on the node channel assignment \mathcal{A} computed by Algorithm 1 must be K -connected, because we will obtain a K -connected topology $G'(V, E')$ after running **Step_1**. In **Step_2**, the edges in G' are checked one by one and the channel assignment for the end nodes of the selected edge are updated. In this way, every edge in G' will have at least a corresponding edge in $G_{\mathcal{A}}$. By running **Step_3**, new edges may be added into $G_{\mathcal{A}}$ and no existing edge will be removed from $G_{\mathcal{A}}$. Therefore, K -connectivity can be guaranteed by Algorithm 1.

Step_1 takes $O(Kn^3 \log m)$ time since K -connectivity testing on G can be done in $O(Kn^3)$ if the algorithm in [9] is used.

It is obvious that case 4 is the most time-consuming case in **Step_2**. In this case, computing $\mathcal{A}(u) \cap \mathcal{A}(v)$ will take $O(Q)$ time. Finding the least used link in PE_e takes $O(m)$ time since there will be at most m links in PE_e . The channel replacement process will stop after $O(n)$ steps, as there are $O(n)$ nodes involved. Q can be considered as a constant because the total number of available channels, C , is normally a small constant (3 or 12). Q is usually less than C , otherwise, the channel assignment will become trivial. So **Step_2** takes $O(m^2)$ time.

Step_3 can be done in $O(n^2)$ because we need to check the channel assignment for all nodes in **Step_3**. Each node may have $O(n)$ neighbors in G . Channel assignment can be decided after checking the channel assignment of all those neighbors.

Therefore, the total worst-case time complexity of Algorithm 1 is $O(Kn^3 \log m + m^2)$. This completes the proof. \square

Essentially, a K -connected topology with minimum maximum link potential interference will be found by **Step_1** with the hope that the finally computed channel assignment will lead to a topology with low co-channel interference. In addition, a link will be kept in G' as long as it has relatively low potential interference, i.e., the G' will be relatively dense enough, which hopefully will benefit the performance of routing because a large number of candidate routes are available. We always assign the least used channels to links in **Step_2-Step_3**. In this way, the channels assigned to nodes will be as different as possible in a potential interference neighborhood.

In [15], the authors classify the static channel assignment schemes for multi-channel wireless networks into two groups, the common channel approach ([4]) and the varying channel approach ([21, 22]). Like discussed before, the channel assignment schemes proposed in [21, 22] cannot be applied here since we do not assume that the traffic demands are given or can be precisely predicted, and moreover K -connectivity cannot be guaranteed. In the common channel approach, NICs of all nodes are assigned to a common set of channels. Obviously, the topology given by such method is a feasible solution for our INSTC problem. So we will compare our scheme with the common channel assignment approach in the simulations.

5. INTERFERENCE-AWARE QOS ROUTING

In this section, we will present a polynomial time optimal algorithm for solving the bandwidth-aware routing problem. Our optimal algorithm is based on solving a Linear Programming (LP). As a result, the computed route is flow based, which is not guaranteed to be a single path. Computing a

single path with required bandwidth subject to both inter-flow contention and intra-flow contention does not seem to be possible in polynomial time since intra-flow contention can be determined only if the path is given but there may exist exponential number of single paths between a source and a destination node. Therefore, we will present a maximum bottleneck capacity path heuristic for this purpose.

5.1 Optimal Algorithm

We first present an LP formulation for the routing problem and then formally present the complete algorithm. We assume that we have already obtained a good channel assignment \mathcal{A} and its corresponding topology $G_{\mathcal{A}}(V, E_{\mathcal{A}})$. Our QoS routing will be operated on the topology $G_{\mathcal{A}}(V, E_{\mathcal{A}})$.

Given a new connection request ρ with source node s , destination node t , and bandwidth requirement B . We want to find a route with required bandwidth on the given topology.

We need to construct an auxiliary digraph $G'_{\mathcal{A}}(V'_{\mathcal{A}}, E'_{\mathcal{A}})$ from the topology $G_{\mathcal{A}}(V, E_{\mathcal{A}})$ for our LP formulation. We will describe the construction of $G'_{\mathcal{A}}(V'_{\mathcal{A}}, E'_{\mathcal{A}})$ in a way that is easy to understand. For each node $v \in V$, $V'_{\mathcal{A}}$ contains Q nodes $v^{\lambda_1(v)}, v^{\lambda_2(v)}, \dots, v^{\lambda_Q(v)}$, where $\lambda_1(v) < \lambda_2(v) < \dots < \lambda_Q(v)$ are the Q channels constituting the set $\mathcal{A}(v)$. For each $v \in V$ and $1 \leq i < Q$, $E'_{\mathcal{A}}$ contains a directed edge from $v^{\lambda_i(v)}$ to $v^{\lambda_{i+1}(v)}$ and a directed edge from $v^{\lambda_{i+1}(v)}$ to $v^{\lambda_i(v)}$, both with capacity set to ∞ . Such edges are called *intra-node edges*. We will use E'_{AI} to denote the set of intra-node edges. For each undirected edge $(u, v; k) \in E_{\mathcal{A}}$, $E'_{\mathcal{A}}$ contains two directed edges (u^k, v^k) and (v^k, u^k) . Such edges are called *inter-node edges*. The capacity of such edges are set to $A(u, v; k)$ (available bandwidth). We will use E'_{AO} to denote the set of inter-node edges. Clearly, $E'_{\mathcal{A}} = E'_{AI} \cup E'_{AO}$. To reduce the network complexity, we also eliminate some of the intra-node edges in the following way. Let v^i be any node in $V'_{\mathcal{A}}$. If there does not exist a node $w^i \in V'_{\mathcal{A}}$ and $j \neq i$ such that $(v^i, w^i) \in E'_{\mathcal{A}}$ and $(w^j, v^j) \in E'_{\mathcal{A}}$, then we can *shrink* the intra-node edges incident with v^i . After this shrinking step, we will have a graph where intra-node edges only exist at nodes which are incident with multi-edges in the topology. By abusing the notation a little bit, we will use $G'_{\mathcal{A}}(V'_{\mathcal{A}}, E'_{\mathcal{A}})$ to denote this *reduced* graph. Figure 3 shows the directed graph $G'_{\mathcal{A}}$ constructed from the network topology in Figure 1. Note that we only have intra-node edges at nodes A^1, A^2, B^1, B^2 . All other intra-node edges are shrunk. We therefore drop the superscript at the nodes E^i, C^i and D^i .

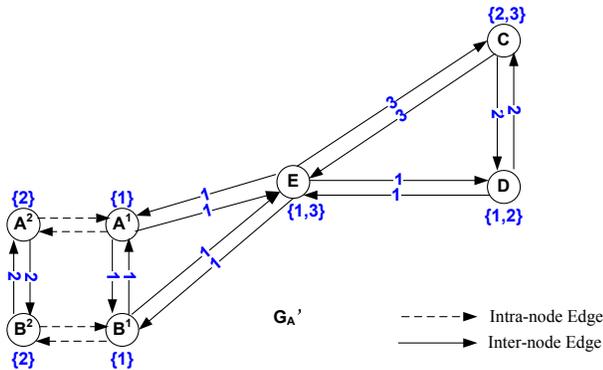


Figure 3: Construction of $G'_{\mathcal{A}}$

We fix s' to any of the vertices $s^i \in V'_{\mathcal{A}}$ and fix t' to any of the vertices $t^j \in V'_{\mathcal{A}}$. We define the flow allocation variables, $f_e : e \in E'_{\mathcal{A}}$, which indicates the bandwidth to be allocated on link e . We present our LP formulation as follows.

$$\text{minimize } \sum_{e \in E'_{AO}} (|IE_e| * f_e) \quad (1)$$

Subject to :

Flow conservation constraint:

$$\sum_{e \in E_v^{out}} f_e - \sum_{e \in E_v^{in}} f_e = 0 \quad \forall v \in V'_{\mathcal{A}} \setminus \{s', t'\} \quad (2)$$

Bandwidth requirement constraint:

$$\sum_{e \in E_s^{out}} f_e - \sum_{e \in E_t^{in}} f_e = B \quad (3)$$

Interference constraint:

$$\sum_{e' \in IE_e} f_{e'} \leq A(e) \quad \forall e \in E'_{AO} \quad (4)$$

$$f_e \geq 0 \quad \forall e \in E'_{\mathcal{A}} \quad (5)$$

In the above formulation, IE_e denotes the set of inter-node edges interfering with link $e \in E'_{AO}$. The intra-node edge will be ignored when interference is considered. E_v^{out} and E_v^{in} are the outgoing and incoming edge sets of node v in $G'_{\mathcal{A}}$ respectively. $A(e)$ denotes the available bandwidth on link e , whose value is the same as the available bandwidth of its corresponding link in $G_{\mathcal{A}}$ (refer to Definition 3).

Constraint (2) is a general flow conservation constraint which makes sure an $s'-t'$ flow allocation is computed. Constraint (3) guarantees that the total bandwidth requirement B from the connection request is satisfied. With the intra-flow and inter-flow contentions fully considered, Constraint (4) ensures that the available bandwidth on each link is large enough for the flow (bandwidth) allocation. The objective function (1) is set to minimize the interference influence from the current connection, which hopefully will increase the chance for the upcoming connection requests to be admitted. The optimal algorithm for solving BAR problem is presented as Algorithm 2 in the following.

THEOREM 2. *Algorithm 2 correctly solves the BAR problem in polynomial time.*

PROOF. As we discussed earlier, the correctness of the algorithm lies in the fact that Equations 2-5 ensure that the constraints in Definition 4 are all satisfied.

The auxiliary graph $G'_{\mathcal{A}}$ includes at most nQ nodes and at most $2mQ + 2nQ$ edges. Since Q is a small constant, we will ignore Q in our analysis. It is obvious that **Step_1** can be done in polynomial time. The LP formulation has $O(n + m)$ constraints and $O(n + m)$ variables. Therefore, **Step_2** takes polynomial time. Clearly, updating the link loads and available bandwidth information can be done in polynomial time as well. \square

Algorithm 2 Optimal BAR Algorithm

- Step_1 Construct $G'_A(V'_A, E'_A)$ from the given topology $G_A(V, E_A)$.
- Step_2 Solve LP formulation specified by Constraints (1-5) on G'_A .
- Step_3 **if** a feasible solution can be found
 Output the corresponding flow allocation in G_A . Update the load and available bandwidth of each affected link in G_A .
else
 block the connection request.
endif
-

5.2 Maximum Bottleneck Capacity Path Heuristic

In order to route packets based on the flow allocation computed by Algorithm 2, traffic needs to be splitted at some intermediate nodes. Sometimes traffic splitting is not preferred because of difficulties in packet fragmentation and reassembly. So we also propose a heuristic for the BAR problem, which computes a single path for packet routing. Note here no splitting means for a specific connection, traffic can only be received by a single NIC and/or be transmitted by a single NIC (these two NICs can be different) at any involved node.

The basic idea of our heuristic is to carefully select a candidate $s-t$ single path on the given network topology G_A for an incoming connection request, which most likely will satisfy all bandwidth constraints. Obviously, a single path also corresponds to a flow allocation, where the allocated bandwidth in each link on the path is equal to B . Then we check its feasibility according to Definition 4. If no constraint is violated, the connection request is accepted and the found candidate path is used for routing. Otherwise, block the connection request. Before describing the algorithm, we need to introduce a new metric for the candidate path computation.

DEFINITION 6 (BOTTLENECK CAPACITY). *The **Link Bottleneck Capacity** of link e , denoted by $BC(e)$ is $BC(e) = \min_{e' \in I E_e} [A(e')/B]$. The **Path Bottleneck Capacity** of a single path P , denoted by $BC(P)$, is $BC(P) = \min_{e \in P} BC(e)$.*

When a path P is established for a connection, two or more links on this path may interfere with some common link e . The available bandwidth on link e must be large enough to tolerate such interference. Otherwise, path P cannot be used for routing because bandwidth requirements cannot be satisfied in the system. Intuitively, we prefer to select those links with relatively higher bottleneck capacity values because by doing so, we can reduce the probability for the bandwidth constraint to be violated. The total required bandwidth on each link cannot exceed its available bandwidth. On the other hand, we prefer a path with relatively low hop-count because the more hops a candidate path has, the longer delay packets will suffer, and more importantly, the higher bandwidth invalidation probability it will cause. Therefore, we try to find a *hop-count bounded maximum bottleneck capacity path* as the candidate path,

which has the maximum bottleneck capacity value among all $s-t$ single paths and its hop-count is bounded by a given threshold H . For ease of presentation, we first present an algorithm to find a bottleneck capacity bounded minimum hop-count path on a given network topology, which is a minimum hop-count $s-t$ single path among all $s-t$ single paths with bottleneck capacity bounded by a given threshold T .

Algorithm 3 Bottleneck capacity bounded minimum hop-count path algorithm

- Step_1 Construct an auxiliary digraph $G_B(V_B, E_B)$ where V_B contains all the wireless nodes and E_B contains all the links in G_A whose bottleneck capacity are at least T .
- Step_2 Apply Breadth First Search (BFS) algorithm to compute an $s-t$ path P in G_B with minimum hop-count.
-

Our maximum bottleneck capacity path heuristic is listed as Algorithm 4.

Algorithm 4 Maximum bottleneck capacity path heuristic

- Step_1 Use binary search on all link bottleneck capacity values to find the maximum link bottleneck capacity T so that the corresponding path P on G_A computed by Algorithm 3 with bottleneck capacity threshold T has a total hop-count no more than the given threshold H .
- Step_2 Check the feasibility of path P according to Definition 4.
- Step_3 **if** Path P is feasible
 Output path P . Update the load and available bandwidth of each affected link in G_A .
else
 Block the connection request.
endif
-

Algorithm 3 takes $O(m)$ time since G_A contains at most mQ edges. Therefore, **Step_1** of Algorithm 4 can be done in $O(m \log m)$ time. Similarly, **Step_2** takes $O(mn)$ time since a path in G_A has at most n hops. **Step_3** needs $O(m^2)$ time. Thus, the total time complexity for Algorithm 4 is $O(m^2)$.

The hop-count threshold H is usually set to a *Bound Ratio* β multiplied by the number of hops of the corresponding minimum hop-count path. Since the minimum hop-count path is not unique for a specific source-destination pair, we can obtain a minimum hop-count path with larger bottleneck capacity value than other corresponding minimum hop-count paths by using our heuristic and setting $\beta = 1$, which will make the found path more likely to be a feasible solution. Setting β larger than 1 will allow the algorithm to select a path with a larger bottleneck capacity. However, having more hops may counteract such gain. We will discuss this in more detail in Section 6.

6. PERFORMANCE EVALUATION

In this section, we evaluate the performance of our algorithms via simulations. We consider static wireless mesh networks with n nodes randomly located in a $900 \times 900 m^2$ region. In all simulation scenarios, we require 2-connectivity ($K = 2$) be preserved by the topology control algorithm. Each node has a fixed transmission range of $250m$ and interference range of $500m$ ([21]). In all scenarios, each connection request is generated with a randomly chosen source-destination pair and a random bandwidth requirement B which is no more than a given maximal bandwidth requirement (B_{max}). In total, 1000 connection requests are injected in each simulation run. The lifetime (LT) of each connection specifies how many time units it will last. This is also a random number uniformly distributed between 1 and a maximal value (LT_{max}) which is set to 200 in all simulations.

The connection *Blocking Ratio*, i.e., the ratio between the number of blocked connections and the total number of connection requests, is the metric used for our performance evaluation. The following five system parameters can influence the performance: network size (n), the number of available non-overlapping channels (C), the number of NICs (Q) in one node, the channel capacity (CAP) and the traffic load. According to IEEE 802.11 specifications, we set the channel number (C) to be 3 (802.11b) in some scenarios and 12 (802.11a) in others, and the corresponding channel capacity (CAP) are $11(Mbps)$ and $54(Mbps)$ respectively. We adjust the traffic load by fixing the mean connection request arrival interval to be 15 time units and vary the maximal bandwidth requirement (B_{max}). The number of NICs (Q) at each node is set to 2 in some scenarios and 3 in others.

We first show the performance of combined topology control and routing scheme with regards to connection blocking ratio. The corresponding results are presented in Figures 4-8. We compare our solutions, i.e., our interference-aware topology control heuristic (Algorithm 1) along with optimal BAR algorithm (Algorithm 2) or the Maximum Bottleneck Capacity Path heuristic (Algorithm 4), with a simple approach including the Common channel assignment approach ([4]) and minimum hop-count (Shortest) Path routing. These schemes are labelled as BAR, MBCP and CSP, respectively in the figures and will also be called the BAR, the MBCP and the CSP scheme for brevity. Notice that we set the bound ratio β to 1.0 and 1.5 for the maximum bottleneck capacity path heuristic respectively in each scenario.

We make the following observations from our simulation results. Our BAR scheme performs best in all cases. Compared with the CSP scheme, it reduces the blocking ratios from 32.5% to 13.9% on average, which is a 57% improvement. This improvement becomes more substantial when the network resources (Q, C, CAP) become relatively large (Figures 6-8). This is because our scheme is fully aware of the influence of the interference and makes use of available resources more wisely. For example, the more channels we have, the more likely the nodes in the neighbor are assigned to different channels so that the influence of interference can be reduced. This will eventually increase the chance for a connection request to be admitted. However, more channels will not lead to any improvement if common channel assignment approach is applied.

Our MBCP scheme always outperforms the CSP scheme. Setting the bound ratio to different values does not change

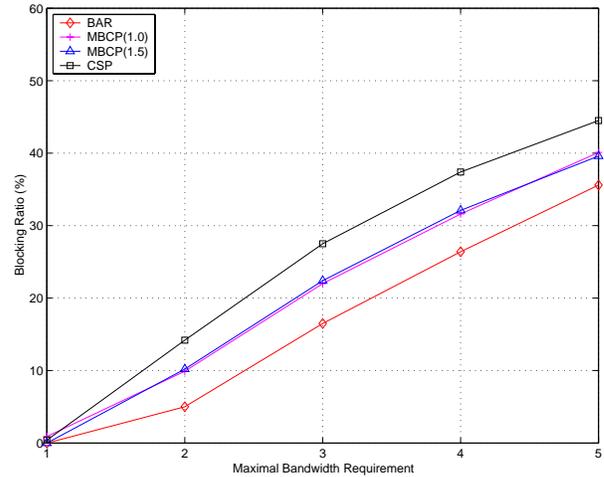


Figure 4: $n = 25, C = 3, Q = 2, CAP = 11$

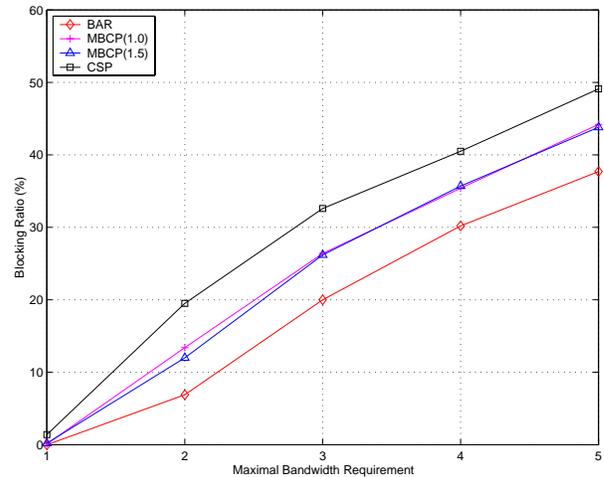


Figure 5: $n = 40, C = 3, Q = 2, CAP = 11$

the blocking ratios too much. This is due to the fact that by setting the bound ratio to a relatively high value, links with high bottleneck capacity may be selected for routing which will reduce the probability for the path to be blocked. On the other hand, the selected path may have more hops, which on the contrary, will increase the bandwidth violation probability and eventually cause the connection to be blocked. These two factors will counteract with each other.

Furthermore, the blocking ratio increases with the increase of the maximal bandwidth requirement since more bandwidth needs to be allocated for each connection. This reduces the chance for an upcoming connection request to be accepted. Substantial blocking ratio reduction is achieved by having one more NICs in each node (Figure 7-8). NICs on a node can be simultaneously used for transmission as long as they are assigned to different channels. Therefore, more bandwidth is brought to system when one more NIC is provided for each node.

In the second part, we try to verify the effectiveness of our topology control algorithm on different networks, i.e.,

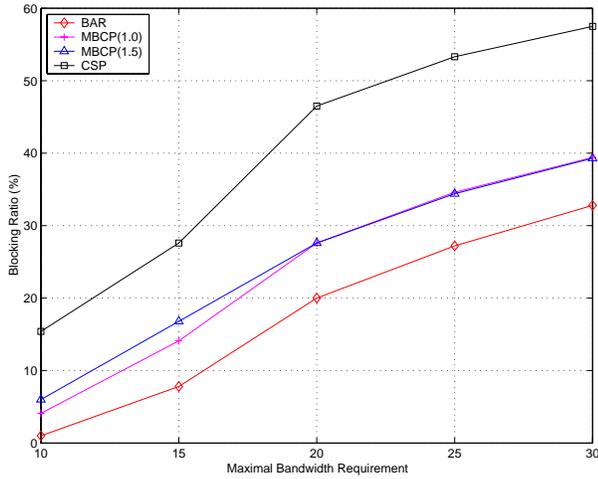


Figure 6: $n = 25, C = 12, Q = 2, CAP = 54$

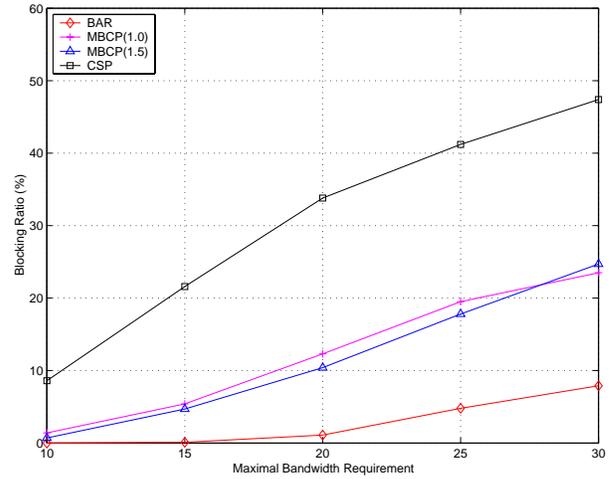


Figure 8: $n = 40, C = 12, Q = 3, CAP = 54$

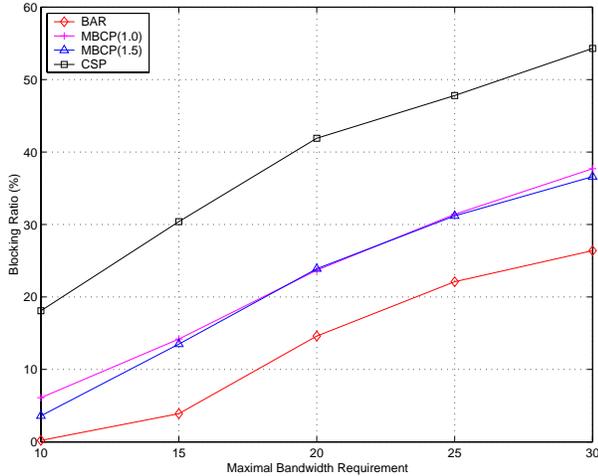


Figure 7: $n = 40, C = 12, Q = 2, CAP = 54$

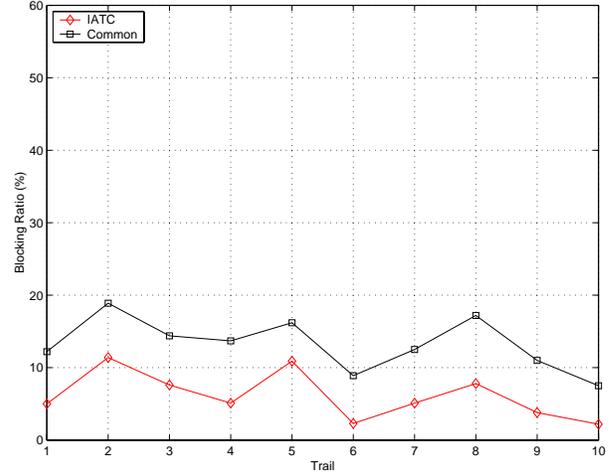


Figure 9: $n = 25, C = 3, Q = 2, CAP = 11, B_{max} = 2$

different node deployments. Our optimal BAR algorithm are always used for QoS routing. In each scenario, we randomly generate 10 networks, each with 25 nodes.

In each trial, we employ our Interference-Aware Topology Control (IATC) algorithm and the common channel assignment approach (Common) respectively to assign channels to nodes and observe the obtained connection blocking ratios. The results are shown in Figures 9–10.

As expected, our interference-aware topology control algorithm always achieves lower blocking ratios than the common channel assignment approach no matter how the nodes are deployed in the plane. Quite consistent with the previous results, the performance improvement becomes more remarkable when more network resources are available.

7. CONCLUSIONS

In this paper, we have studied interference-aware topology control and QoS routing in IEEE 802.11-based multi-channel wireless mesh networks with dynamic traffic. We

have presented a novel definition of co-channel interference to precisely capture the influence of the interference. According to this definition, we formally defined the minimum INterference Survivable Topology Control (INSTC) problem and presented an effective heuristic for INSTC. We also formulated and presented a polynomial time optimal algorithm to solve the Bandwidth-Aware Routing (BAR) problem on a given network topology induced by topology control. Simulation results show that our scheme outperforms the existing scheme substantially.

Studying distributed algorithms for the proposed problems will be of our interest in the future.

8. REFERENCES

- [1] I. F. Akyildiz, X. Wang, W. Wang, Wireless mesh networks: a survey, *Elsevier Journal of Computer Networks*(2005), In press.
- [2] A. Adya, P. Bahl, J. Padhye, A. Wolman, and L. Zhou, A multi-radio unification protocol for IEEE

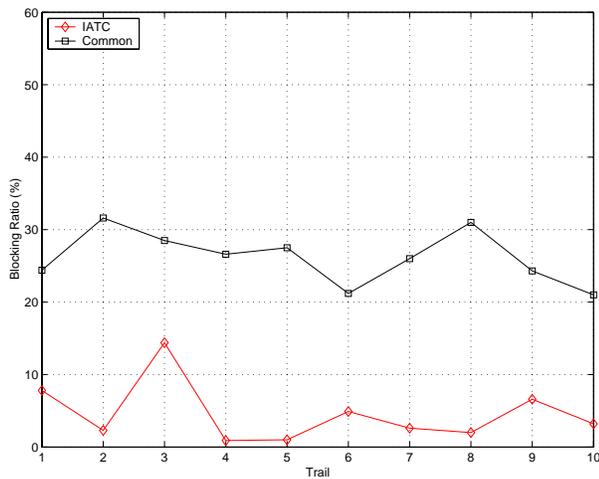


Figure 10: $n = 25, C = 12, Q = 2, CAP = 54, B_{max} = 15$

802.11 Wireless Networks, *Proceedings of IEEE BroadNets'2004*.

- [3] P. Bahl, Opportunities and challenges of community mesh networking, *Keynote Presentation @ MICS Workshop in ETH Zurich*, 2004.
- [4] V. Bahl, A. Adya, J. Padhye, A. Wolman. Reconsidering the wireless LAN platform with multiple radios, *Workshop on Future Directions in Network Architecture*, 2003.
- [5] P. Bahl, R. Chandra, and J. Dunagan, SSCH: Slotted seeded channel hopping for capacity improvement in IEEE 802.11 ad-hoc wireless networks, *Proceedings of ACM MobiCom'2004*, pp. 216–230.
- [6] J. A. Bondy and U. S. R. Murthy, *Graph Theory with Applications*, North Holland, 1976.
- [7] M. Burkhart, P. von Rickenbach, R. Wattenhofer, A. Zollinger, Does topology control reduce interference, *Proceedings of ACM MobiHoc'2004*, pp. 9–19.
- [8] R. Draves, J. Padhye, and B. Zill, Routing in multi-radio, multi-hop wireless mesh networks, *Proceedings of ACM MobiCom'2004*, pp. 114–128.
- [9] S. Even, An algorithm for determining whether the connectivity of a graph is at least k , *SIAM Journal of Computing* Vol. 4(1975), pp. 393-396.
- [10] S. Guo, O. Yang, QoS-aware minimum energy multicast tree construction in wireless ad hoc networks, *Elsevier Journal of Ad Hoc Networks*, Vol. 2(2004), pp. 217–229.
- [11] P. Gupta, P. R. Kumar, The capacity of wireless networks, *IEEE Transactions on Information Theory*, Vol. 46(2000), pp. 388–404.
- [12] IEEE 802.11 Working Group, Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, 1997.
- [13] IEEE 802.11a Working Group, Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications - Amendment 1: High-speed Physical Layer in the 5 GHz band, 1999.
- [14] K. Jain, J. Padhye, V. Padmanabhan and L. Qiu, Impact on interference on multihop wireless network performance, *Proceedings of ACM MobiCom'2003*, pp. 66–80.
- [15] P. Kyasanur, N. H. Vaidya, Routing and interface assignment in multi-channel multi-interface wireless networks, *Proceedings of WCNC'2005*.
- [16] W. H. Liao, Y. C. Tseng, K. P. Shih, A TDMA-based bandwidth reservation protocol for QoS routing in a wireless mobile ad hoc network, *Proceedings of IEEE ICC'2002*, pp. 3186–3190.
- [17] C. R. Lin, J. S. Liu, QoS routing in ad hoc wireless networks, *IEEE Journal on Selected Areas in Communications* Vol. 17(1999), pp. 1426–1438.
- [18] H. Liu, X. Jia, D. Li, C. H. Lee, Bandwidth guaranteed call admission in TDMA/CDMA ad hoc wireless networks, *Elsevier Journal of Ad Hoc Networks*(2005), In press.
- [19] K. Moaveninejad, X. Y. Li, Low-interference topology control for wireless ad hoc networks, *Journal of Ad Hoc and Sensor Wireless Networks*(2005), Accepted for publication.
- [20] C. E. Perkins, E. M. Royer, S. R. Das, Quality of service for ad hoc on-demand distance vector routing (work in progress), *IETF Internet Draft*, 2000.
- [21] A. Raniwala, T. Chiueh, Architecture and algorithms for an IEEE 802.11-based multi-channel wireless mesh network, *Proceedings of IEEE INFOCOM'2005*.
- [22] A. Raniwala, K. Gopalan, T. Chiueh, Centralized channel assignment and routing algorithms for multi-channel wireless mesh networks, *ACM Mobile Computing and Communications Review (MC2R)*, Vol. 8(2004), pp. 50–65.
- [23] J. So, N. H. Vaidya, Multi-channel MAC for ad hoc networks: Handling multi-channel hidden terminals using a single transceiver, *Proceedings of ACM MobiHoc'2004*, pp. 222–233.
- [24] J. So, N. H. Vaidya, A routing protocol for utilizing multiple channels in multi-hop wireless networks with a single transceiver, *UIUC Technical Report*, 2004. Available at: <http://www.crhc.uiuc.edu/wireless/groupPubs.html>
- [25] Q. Xue, A. Ganz, Ad hoc QoS on-demand routing (AQOR) in mobile ad hoc networks, *Journal of Parallel Distributed Computing* Vol. 63(2003), pp. 154–165.
- [26] H. Zhai, J. Wang, Y. Fang, Distributed packet scheduling for multihop flows in ad hoc networks, *Proceedings of IEEE WCNC'2004*, pp. 1081–1086.
- [27] C. Zhu, S. Corson, QoS routing for mobile ad hoc networks, *Proceedings of IEEE INFOCOM'2002*, pp. 958–967.