Complexity and Design of QoS Routing Algorithms in Wireless Mesh Networks

Bahador Bakhshi[a,*], Siavash Khorsandi[a]

[a] *Computer Engineering and Information Technology Department, Amirkabir University of Technology, Hafez Avenue, Tehran, Iran*

Abstract

Quality of Service (QoS) provisioning in Wireless Mesh Networks (WMNs) is an open issue to support emerging multimedia services. In this paper, we study the problem of QoS provisioning in terms of end-to-end bandwidth allocation in WMNs. It is challenging due to interferences in the networks. We consider widely used interference models and show that except a few special cases, the problem of finding a feasible path is NP-Complete under the models. We propose a k-shortest path based algorithmic framework to solve this problem. We also consider the problem of optimizing network performance by *on-line* dynamic routing, and adapt commonly used conventional QoS routing metrics to be used in WMNs. We find the optimal solutions for these problems through formulating them as optimization models. A model is developed to check the existence of a feasible path and another to find the optimal path for a demand; moreover, an on-line optimal QoS routing algorithm is developed. Comparing the algorithms implemented by the proposed framework with the optimization models shows that our solution can find existing feasible paths with high probability, efficiently optimizes path lengths, and has a comparable performance to the optimal QoS routing algorithm. Furthermore, our results show that contrary to wireline networks, minimizing resource consumption should be preferred over load distribution even in lightly loaded WMNs.

Keywords: Wireless Mesh Networks, Bandwidth Constrained Routing, NP-Complete, Routing Metric, Integer Linear Programming

1. Introduction

Wireless mesh networking is a promising technology for future multi-hop wireless access networks. The most distinguishing feature of WMNs is the static multi-hop wireless backbone of the networks compared to other wireless networks. WMNs can act as the last-mile in Internet service provider networks, where multimedia services are an integral ingredient of the networks. Multimedia services need end-to-end QoS support. It is often defined in terms of bandwidth, delay, and delay jitter. However, it is argued that bandwidth

[*] Corresponding author
Email addresses: bbakhshi@aut.ac.ir (Bahador Bakhshi), khorsandi@aut.ac.ir (Siavash Khorsandi)

allocation is the main QoS requirement since it controls delay and jitter as well [1]. The problem of end-to-end bandwidth allocation is, in fact, twofold. First, to accept a given traffic demand with a bandwidth requirement, the *bandwidth constrained routing* algorithm should find a path with sufficient end-to-end bandwidth, which is called *feasible path*. Second, network resources should be utilized efficiently to avoid congestions and effective load distribution throughout the network to maximize network performance.

Bandwidth constrained routing is a long standing problem in the networking literature. It has been extensively studied in both wireline and wireless networks [1–18]. Previous studies in wireline networks mostly focused on the efficient network utilization aspect because feasible paths in these networks are simply found using the network pruning technique [2–6]. These solutions cannot be directly applied for WMNs since they do not consider interference, which is a fundamental issue in multi-hop wireless networks. In wireline networks, a flow routed through a path consumes only the bandwidth of the links in the path. However, in WMNs, each flow consumes bandwidth of all the links in the interference range of the path. The exact bandwidth consumption by a flow is determined by the *interference pattern* that specifies the links interfere with each other.

The studies on the bandwidth constrained routing problem in wireless networks mainly have considered the problem of finding feasible paths. Most the proposed solutions are variations of flooding-based algorithms [1, 7–16]. These solutions are not efficient in WMNs because the significant overhead of the flooding-based algorithms is tolerable only in highly dynamic networks, which is not the case in WMNs. Recently, a few link-state like (and centralized) algorithms have been proposed [17–19]. The major shortcoming of the existing studies is that they do not consider the complexity of bandwidth consumption and its relation to the interference pattern, and usually use (over)simplified interference models.

The algorithmic aspects of the bandwidth constrained routing problem, e.g., complexity of finding a feasible path and the effect of the system models on the complexity, in general WMNs have not yet been studied in the literature. Most of the previous work has not considered these complexities and only proposed adhoc heuristic solutions rather than a systematic approach to solve the problem. Moreover, none of the previous studies have provided evaluations of the ability of their proposed solutions to deal with the complexities.

In this paper, we study the algorithmic aspects of the QoS routing problem, where the QoS requirement is described in terms of end-to-end bandwidth. We consider general multi-rate contention-based WMNs, which can be either single-channel or multi-channel. It is assumed that the pattern of interferences is static, specified by an interference model, and is given. Another assumption is that the network is deployed in a rural area, in which the behavior of the links is stable and predictable as shown in [20]. We consider the WMN as a part of an Internet service provider network. Hence, the network is managed, and routing and resource allocation algorithms are parts of the centralized network management tool. It is supposed that a fairly accurate and complete view of the network is available to the algorithms. Traffic demands arrive in an on-line fashion; so, no prior knowledge of future demands is available. When a new demand arrives, there are some existing flows in the network; the objective is to find a feasible path for the demand. It is a path that bandwidth consumption by the demand through it does not violate the guaranteed bandwidth of the existing flows. Moreover, we need to distribute the load in the network through optimizing the feasible paths to boost network performance. Our contributions to the problem are as follows.

- We analyze the complexity of the bandwidth constrained routing problem under widely used interference models. We identify special situations in which the problem of finding a feasible path is polynomially solvable, and prove its intractability in general cases.
- Based on systematic investigations of the complexity analysis results, we propose the Adjustable Constrained Routing Algorithmic Framework (ACRAF). It is composed of the k-shortest path algorithm, a selector function, and hop-by-hop call admission control.
- We consider different routing metrics traditionally used for QoS routing, and adapt them for the bandwidth constrained routing in WMNs. It yields six routing algorithms, which are implemented through setting the parameters of ACRAF.
- We develop optimization models to check the existence of a feasible path for a given demand and find the minimum length feasible path for it. Furthermore, we develop an on-line optimal QoS routing algorithm. These are used as benchmarks to evaluate the performance of ACRAF.

The rest of the paper is organized as follows. We present an overview of the related work and the differences between our work and the previous studies in Section 2. In Section 3, after describing the needed models, we formulate the problem. In Section 4, we analyze the complexity of the bandwidth constrained routing under various interference models. The proposed solution is discussed in detail in Section 5. We develop the optimization models in Section 6. Simulation results are presented in Section 7; and finally, Section 8 concludes this paper.

2. Related work

The bandwidth constrained routing problem has been the subject of many studies from the early days of network development. This problem has been studied in wireline networks in the context of load balancing and traffic engineering, especially in the MPLS networks [2–6]. These solutions are based on forming a *feasible residual network* by pruning all links that do not have sufficient resources. In the pruned network, every path is feasible. In WMNs, the feasible residual network cannot be constructed by link-level pruning due to the complexity of bandwidth consumption arises from interference in the networks. In fact, we show that the problem of finding a feasible path is NP-Complete, generally.

A number of studies have been carried out on the bandwidth constrained routing problem in multi-hop wireless networks [21, 22]. Some of them, e.g., [1, 7–13], have been specifically dedicated for mobile ad-hoc networks (MANETs). These works focused on the dynamic nature of MANETs, and proposed flooding-based routing algorithms [7–9]; some works attempted to reduce the overhead of the flooding-based algorithm, e.g., [8, 13]. The main objective in these studies is to deal with node mobility. However, due to the static infrastructure of WMNs, this is not the main challenge in WMNs.

In recent years, a few algorithms and protocols have been proposed for QoS routing in single-channel [14–18, 23], and multi-channel multi-radio WMNs [19, 24, 25]. In [14–16], the authors proposed flooding-based algorithms to find a feasible path but they did not consider load distribution throughout the network and the complexity of finding a feasible path. The authors in [23] proposed a flooding-based algorithm to find a path that satisfies multiple QoS constraints. The most closely related studies to this paper are [17–19, 26, 27]. The authors in [17] proposed the IQRouting algorithm, which is a combination of multiple routing algorithms. IQRouting applies the algorithms one-by-one, and if it finds multiple feasible paths, it selects the widest or the least-cost path. Jia et al. dealt with the shortest widest path problem using the k-shortest path algorithm in [18]. The interference model used in [17, 18] is only suitable for single-channel networks; furthermore, these solutions may not find a feasible path, even if it does exist. In [19], a hop-count bounded heuristic algorithm was proposed to find a feasible path with maximum bottleneck capacity, which approximates the widest path. The authors in [26] studied the maximum bandwidth routing problem and proposed a heuristic algorithm and an optimization model. A technique was proposed in [28] to approximate the bandwidth of a given path. The authors in [27] considered the 1-hop interference model [29] and enhanced this technique to approximate the path bandwidth in a distributed manner. Using the distributed approach, the authors proposed a hop-by-hop QoS routing algorithm. However, these studies did not provide analyses of the complexity of the problem.

In terms of complexity analysis, NP-Completeness of the shortest widest path problem was proved in [18]. The authors in [19] conjectured that there is no polynomial time algorithm for the bandwidth constrained routing in multi-channel multi-radio WMNs. In [30, 31], the authors analyzed the complexity of the bandwidth constrained routing in single-channel multi-hop wireless networks and proved NP-Completeness of this problem. In this paper, we analyze the complexity under different interference models; moreover, we evaluate the performance of our proposed pseudo-polynomial algorithm to deal with the intractability of the problem.

Besides these QoS routing algorithms, a number of previous works proposed routing metrics for on-line load balancing in wireless networks [32–36]. They were designed to capture packet loss ratio and interference. These metrics were used for best-effort traffic routing. Such routing metrics are unlikely to be applicable in QoS routing in rural WMNs. They try to distinguish between links that have intermediate loss rates, and since this is not the case in rural WMNs, it will lead to an erratic behavior of the routing layer [20]. In [37], an off-line mechanism was proposed to achieve optimal load balancing whilst satisfying user requirements. Clearly, this mechanism cannot be used for on-line bandwidth constrained routing, which is our concern in this paper, because it needs prior knowledge of the traffic matrix.

3. System model and problem statement

In this section, we first describe the assumptions and models used throughout the paper; then, we formulate the problem we study here. The notations used

Table 1: Notations

Notation	Description
u and v	Node
V	Set of nodes, and $\|V\| = n$
(u, v)	Link
E	Set of links, and $\|E\| = m$
$d(u, v)$	The Euclidean distance between nodes u and v
$I_{(u,v)}$	Interference set of link (u, v)
I	Set of interference sets $I = \{I_{(u,v)} \; \forall(u, v) \in E\}$
$\hat{I}$	Size of of the largest interference set
$\overline{I}_{(u,v)}$	Set of potentially interfering links with (u, v)
$c_{(u,v)}$	Physical channel capacity of link (u, v)
C	Set of physical capacities $C = \{c_{(u,v)} \; \forall(u, v) \in E\}$
$f_{(u,v)}$	Flow on link (u, v)
$ALB(u, v)$	Available link bandwidth of (u, v)
$AAB(u, v)$	Available area bandwidth of (u, v)
T_R	Transmission range
I_R	Interference range
Γ	The number of orthogonal channels in frequency spectrum
r_u	The number of radios of node u
$\deg(u)$	Degree of node u
p	Path $p = <u \to \ldots \to v>$ from u to v
P	Set of paths
$p_1 \oplus p_2$	The concatenation of paths p_1 and p_2
$l(p)$	Path length function
$bw(p)$	Bandwidth of path p
δ	$\delta = (s, d, b, t, e)$: Demand for a path from s to d, required bandwidth $= b$, arrival time $= t$, and exit time $= e$
Δ	Set of demands, $\Delta = \{\delta_i\}$
ϕ	$\phi = (s, d, b, p)$: Flow at rate b from s to d through path p
Φ	The set of existing flows, $\Phi = \{\phi_i\}$
$BC(\phi, (u, v))$	Bandwidth consumption of flow ϕ at link (u, v)
k	The number of hops in k-hop interference model, and The number of paths in k-shortest path algorithm

are summarized in Table 1. For ease of description, we drop the subscripts when they are clear from the context, e.g., c is the physical channel capacity of all links.

3.1. Assumptions

We consider multi-rate contention-based WMNs in which all nodes are static. The network can be single-channel or multi-channel multi-radio; in the latter case, all nodes have multiple radios, and there are Γ orthogonal available channels in the frequency spectrum. We assume that the network is deployed in a rural location where the PHY layer is stable, i.e., links would perform more or less like wired links [20]. Hence, we suppose that the physical channel capacity does not vary over time similar to previous work [15–19, 30, 31]. Moreover, in this paper, we consider a centralized routing algorithm like previous studies [17–19, 31].

3.2. System model

The network is modeled by a digraph $G = (V, E, C)$, where V is the set of n nodes, E is the set of m links, and set C denotes the physical channel capacities. Each $v \in V$ corresponds to a node in the network. Let $d(u, v)$

denote the Euclidean distance between nodes u and v. For a given pair of nodes u and v, there is a link $(u, v) \in E$ if $d(u, v) \leq T_R$, where T_R is the transmission range. Set C is $\{c_{(u,v)}\ \forall(u, v) \in E\}$, where $c_{(u,v)}$ is the physical channel capacity of (u, v).

The links interfering with (u, v) are denoted by the interference set $I_{(u,v)}$. This set is determined by a particular *interference model*, e.g., k-hop interference model [29] or protocol model [38]. It is supposed that interference pattern is static; the given interference sets do not change over time. We assume that $I_{(u,v)}$ is given for all links, $(u_1, v_1) \in I_{(u_2,v_2)}$ if and only if $(u_2, v_2) \in I_{(u_1,v_1)}$, and $(u, v) \in I_{(u,v)}$. The set of all interference sets is denoted by I.

From the point of view of a flow, there are two kinds of interferences. The *inter-flow* interference is the interference between the flow and other existing flows, and the *intra-flow* interference is the interference between different links in the path of the flow.

3.3. Bandwidth models

3.3.1. Available bandwidth

We use the *row* constraint introduced in [39] to compute link available bandwidths. It is a sufficient condition for feasibility of bandwidth allocation and implies that the aggregate load of the links in the interference set of each link must be less than physical channel capacities. Let $f_{(u,v)}$ denote the flow on link (u, v). This constraint imposes that

$$\sum_{(u',v')\in I_{(u,v)}} \frac{f_{(u',v')}}{c_{(u',v')}} \leq 1 \qquad \forall(u, v) \in E, \tag{1}$$

where $\frac{f_{(u,v)}}{c(u,v)}$ is the fraction of time (u, v) needs to transmit flow $f_{(u,v)}$. We refer (1) as the "capacity constraint" since if it is satisfied, network bandwidth allocation will be feasible. Based on this constraint, we define two bandwidths for each link as follows.

Definition 1. *Available Link Bandwidth of* (u, v)*:* $ALB(u, v) = \max\Big\{0, c_{(u,v)}\Big(1-\sum\limits_{(u',v')\in I_{(u,v)}} \frac{f_{(u',v')}}{c_{(u',v')}}\Big)\Big\}$.

Definition 2. *Available Area Bandwidth of* (u, v)*:* $AAB(u, v) = \min\limits_{(u',v')\in I_{(u,v)}} \Big\{\frac{c_{(u,v)}}{c_{(u',v')}} ALB(u', v')\Big\}$.

$ALB(u, v)$ and $AAB(u, v)$ are, respectively, the maximum bit rates at which link (u, v) can transmit without violating its capacity constraint and the constraint of the other links in its interference set. These definitions are clarified by an illustrative example depicted in Fig. 1. Let $c_{(u_1,v_1)} = 10$, $c_{(u_2,v_2)} = 20$, $c_{(u_3,v_3)} = 20$, $c_{(u_4,v_4)} = 40$, $f_{(u_1,v_1)} = 2$, $f_{(u_2,v_2)} = 0$, $f_{(u_3,v_3)} = 10$, and $f_{(u_4,v_4)} = 15$. Using these definitions, we have $ALB(u_1, v_1) = c_{(u_1,v_1)}\Big(1-\frac{f_{(u_1,v_1)}}{c_{(u_1,v_1)}} - \frac{f_{(u_2,v_2)}}{c_{(u_2,v_2)}}\Big) = 8$, $ALB(u_2, v_2) = 20\Big(1 - \frac{2}{10} - \frac{0}{20} - \frac{10}{20}\Big) = 6$, $ALB(u_3, v_3) = 20\Big(1 - \frac{0}{20} - \frac{10}{20} - \frac{15}{40}\Big) = 2.5$, and $ALB(u_4, v_4) = 40\Big(1 - \frac{10}{20} - \frac{15}{40}\Big) = 5$. Moreover, $AAB(u_1, v_1) = \min\Big\{\frac{10}{10}ALB(u_1, v_1), \frac{10}{20}ALB(u_2, v_2)\Big\} = 3$, $AAB(u_2, v_2)$

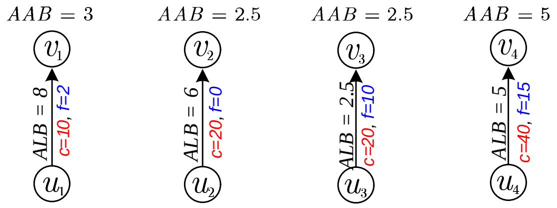


Figure 1: Illustration of ALB (Definition 1) and AAB (Definition 2). The network is a single-channel WMN. Suppose $I_{(u_1,v_1)} = \{(u_1, v_1), (u_2, v_2)\}$, $I_{(u_2,v_2)} = \{(u_1, v_1), (u_2, v_2), (u_3, v_3)\}$, $I_{(u_3,v_3)} = \{(u_2, v_2), (u_3, v_3), (u_4, v_4)\}$, and $I_{(u_4,v_4)} = \{(u_3, v_3), (u_4, v_4)\}$.

$= AAB(u_3, v_3) = \frac{20}{20} ALB(u_3, v_3) = 2.5$, and $AAB(u_4, v_4) = 5$. Note that although $ALB(u_2, v_2) = 6$, link (u_2, v_2) should not transmit in a rate more than $AAB(u_2, v_2) = 2.5$ to maintain the guaranteed bandwidth of flow $f_{(u_3,v_3)}$.

3.3.2. Bandwidth consumption

There are two key observations about the bandwidth consumption of a flow in multi-hop wireless networks. Consider flow $\phi = (s, d, b, p)$ that is from s to d through path p at rate b. First, the flow not only consumes the bandwidth of the links in the path, $\forall(u, v) \in p$, but also it consumes the bandwidth of other links (u', v') in the interference range of the path. We name the links whose bandwidth is consumed by the flow as the "affected links" of the path, which is defined below.

Definition 3. *Affected Links of p:* $AL(p) = \{(u', v') \in I_{(u,v)} \ \forall(u, v) \in p\}$.

Note that by definition $(u, v) \in AL(p)$ if $(u, v) \in p$, and if $(u, v) \notin AL(p)$, its available bandwidth is *not* influenced by creating flow $\phi = (s, d, b, p)$.

The second observation is that a flow may consume the bandwidth of a link multiple times. Consider link $(u, v) \in AL(p)$ and assume $(u', v') \in p$ interferes with (u, v); in other words, $(u', v') \in p \cap I_{(u,v)}$. The amount of the time fraction needed by (u', v') to transmit load b is $\frac{b}{c_{(u',v')}}$. During this time, link (u, v) must be shut down to avoid interference. This happens for each $(u', v') \in p \cap I_{(u,v)}$; therefore, the total time fraction link (u, v) must be shut down due to allocating bandwidth b through path p is $\sum_{(u',v') \in p \cap I_{(u,v)}} \frac{b}{c_{(u',v')}}$. As a result, we have

Definition 4. *The bandwidth consumption by flow $\phi = (s, d, b, p)$ at link $(u, v) \in AL(p)$ is*

$$BC(\phi, (u, v)) = c_{(u,v)} \Bigg(\sum_{(u',v') \in p \cap I_{(u,v)}} \frac{b}{c_{(u',v')}} \Bigg).$$

It is obvious that to maintain the guaranteed bandwidth of the existing flows, we need $BC(\phi, (u, v)) \leq ALB(u, v)$ because otherwise $BC(\phi, (u, v)) > ALB(u, v)$ yields that

$$c_{(u,v)} \Bigg(\sum_{(u',v') \in p \cap I_{(u,v)}} \frac{b}{c_{(u',v')}} \Bigg) > c_{(u,v)} \Bigg(1 - \sum_{(u',v') \in I_{(u,v)}} \frac{f_{(u',v')}}{c_{(u',v')}} \Bigg),$$

$$\sum_{(u',v') \in p \cap I_{(u,v)}} \frac{b}{c_{(u',v')}} + \sum_{(u',v') \in I_{(u,v)}} \frac{f_{(u',v')}}{c_{(u',v')}} > 1,$$

that means the capacity constraint of the link is violated.

3.4. Problem statement

In this paper, we study the QoS routing problem in WMNs. In the problem, there is a set of demands $\Delta = \{\delta_i = (s_i, d_i, b_i, t_i, e_i)\}$; demand δ_i arrives at time t_i, needs a path with bandwidth b_i from node s_i to node d_i. If the QoS routing algorithm can find a *feasible* path p, the demand is admitted that creates flow $\phi = (s, d, b, p)$ in the network. In this case, the demand leaves the network at time e_i. In the context of QoS routing, network performance is usually measured in terms of demand acceptance rate (or the number of accepted demands) [2–6], which needs to be optimized by the QoS routing algorithm. Obviously, this network performance optimization problem is equivalent to the problem of maximizing the probability of finding a feasible path for each demand. Two factors influence this probability. The first one is resource availability in the network that determines the existence of feasible paths. The second factor is the ability of the QoS routing algorithm to find existing feasible paths. Accordingly, the QoS routing problem is composed of two subproblems: the problem of finding a feasible path for a given demand and the problem of efficient utilization of network resources. These subproblems are explicated more formally in the following.

First, we consider the problem of finding a feasible path. Suppose that network $G = (V, E, C)$ is given and a set of flows, Φ, are existing in the network. These flows determine the available bandwidth of each link. At time t, a new demand $\delta = (s, d, b, t, e)$ arrives. The problem is to find a *feasible* path p from s to d. Feasibility of the path implies that transmission at rate b through the path does not violate the capacity constraint (1). In other words, it means that if flow $\phi = (s, d, b, p)$ is created, its bandwidth consumption does not exceed the available bandwidth of any link; otherwise, the capacity constraint is violated as explained in Section 3.3.2. More specifically, the problem is defined as follows.

Problem: Feasible Bandwidth Constrained Path in WMNs (FBCP).
Instance: $G = (V, E, C)$, set I, set Φ of existing flows, and a demand $\delta = (s, d, b, t, e)$.
Question: Is there any path $p = <s \to \ldots \to d>$ such that creating flow $\phi = (s, d, b, p)$ satisfies $BC(\phi, (u, v)) \leq ALB(u, v)\ \forall (u, v) \in AL(p)$?

Since flow ϕ does not affect $ALB(u, v)$ if $(u, v) \notin AL(p)$, satisfaction of this constraint only for the links $(u, v) \in AL(p)$ is the necessary and sufficient condition for the feasibility of the path.

As mentioned, network performance is measured in terms of the number accepted demands. Hence, the second subproblem, efficient utilization of network resources, is defined formally as follows.

Problem: Maximum Acceptance Rate in WMNs (MAR).
Instance: $G = (V, E, C)$, set I, and set Δ of demands.
Question: What is the maximum number of demands that can be accepted?

Here, we assume that there is not any information about a demand before it arrives. Hence, the QoS routing algorithm is *on-line*. When a demand arrives, the algorithm attempts to find a path for it only according to the state of the network at the time.

4. Complexity analysis

In this section, we analyze the complexity of the FBCP and MAR problems.

4.1. Complexity of finding a feasible path

The problem of finding a feasible path with a guaranteed end-to-end bandwidth is polynomially solvable in wireline networks by networking pruning. However, in multi-hop wireless networks, it is substantially difficult. In fact, in general, the FBCP problem is intractable, which is due to interferences in wireless networks. *Interference model*, which specifies the interferences, greatly influences the complexity of the problem. In the following, we first provide an insight into the complexity, give an illustrative example, and prove a theorem on the complexity of FBCP problem under an arbitrary interference model. Then, we consider different interference models widely used in the literature and analyze the complexity of the FBCP problem under each model.

4.1.1. Introduction

The constraint of the FBCP problem, $BC(\phi,(u,v)) \leq ALB(u,v)$, is affected by both intra-flow and inter-flow interferences. However, the problem is that the interferences are not fully determined until the path is completely constructed. For example, in finding a path for demand δ, if link (u,v) is selected because of $ALB(u,v) \geq b$, it cannot be guaranteed that constraint $BC(\phi,(u,v)) \leq ALB(u,v)$ will be satisfied when the path gets completed. This is due to the fact that the links added to the path after (u,v) affect $BC(\phi,(u,v))$, and may violate the constraint. The following theorem and corollary show the complexity of the FBCP problem under an arbitrary interference model.

Theorem 1. *For a given interference model, flow $\phi = (s,d,b,p)$, and link (u,v), if $BC(\phi,(u,v))$ is the same for every path p where $(u,v) \in p$, and it is the same for every path p where $(u,v) \in AL(p) \setminus p$, pruning the network by the following rules*

1. *Link (u,v) is pruned if $ALB(u,v) < BC(\phi,(u,v))$.*
2. *Link (u,v) is pruned if $\exists(u',v')$ s.t. $(u,v) \in I_{(u',v')}$ and $ALB(u',v') < BC(\phi,(u',v'))$.*

yields that

1. *All paths in the pruned network is feasible for the demand $\delta = (s,d,b,t,e)$ corresponding to flow ϕ.*
2. *The pruning does not exclude any feasible path.*

Proof. The proof can be found in Appendix A.1. □

Corollary 2. *If the conditions of Theorem 1 hold, the FBCP problem is polynomially solvable.*

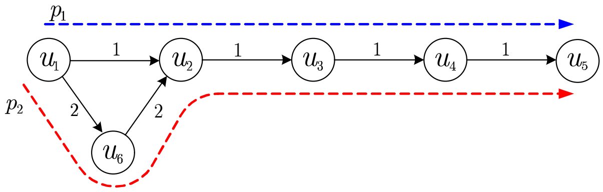


Figure 2: Illustration of the complexity of finding a feasible path. Label of each link is the channel assigned to the link. Two paths for demand $(u_1, u_5, 5, 0, 1)$ are shown by dashed lines. Assume that $c = 15$, and $I_{(u_1,u_2)} = \{(u_1,u_2), (u_2,u_3), (u_3,u_4)\}$, $I_{(u_2,u_3)} = I_{(u_3,u_4)} = \{(u_1,u_2), (u_2,u_3), (u_3,u_4), (u_4,u_5)\}$, $I_{(u_4,u_5)} = \{(u_2,u_3), (u_3,u_4), (u_4,u_5)\}$, and $I_{(u_1,u_6)} = I_{(u_6,u_2)} = \{(u_1,u_6), (u_6,u_2)\}$.

Table 2: Bandwidth consumption of flows in the topology depicted in Fig. 2.

ϕ	Bandwidth consumption at link					
	(u_1,u_2)	(u_1,u_6)	(u_6,u_2)	(u_2,u_3)	(u_3,u_4)	(u_4,u_5)
ϕ_1	15	0	0	20	20	15
ϕ_2	10	10	10	15	15	15

Proof. This is a direct result of Theorem 1. It is sufficient to prune the network by the rules mentioned in the theorem. It implies that if there is a feasible path for demand δ, it will be present in the pruned network, and since every path is feasible after the pruning, a path can be found by polynomial time graph search algorithms. □

We illustrate the complexity of FBCP by an example when the conditions of Theorem 1 do not hold. Consider Fig. 2, which depicts a general multi-channel multi-radio WMN. Suppose that we use Dijkstra's algorithm. Assume that there is not any flow in the network, $c = 15$ bps, and a demand $(u_1, u_5, 5 \text{ bps}, 0, 1)$ arrives. There are two (not necessarily feasible) paths for the demand: $p_1 = <u_1 \rightarrow u_2 \rightarrow u_3 \rightarrow u_4 \rightarrow u_5>$ and $p_2 = <u_1 \rightarrow u_6 \rightarrow u_2 \rightarrow u_3 \rightarrow u_4 \rightarrow u_5>$. We consider a flow per path: $\phi_1 = (u_1, u_5, 5, p_1)$ and $\phi_2 = (u_1, u_5, 5, p_2)$. Bandwidth consumption by these flows is shown in Table 2. As it seen, in this topology, bandwidth consumption at each link by the demand depends on its path. Note that path p_1 is not feasible, because $BC(\phi, (u_2, u_3)) = BC(\phi, (u_3, u_4)) > ALB(u_2, u_3) = ALB(u_3, u_4) = c$. However, p_2 is feasible.

In this example, if (u_2, u_3) and (u_3, u_4) are pruned due to bandwidth consumption by flow ϕ_1, it excludes the existing feasible path p_2 from the pruned network. If we do not prune the network, Dijkstra's algorithm finds path p_1 as the shortest path which is not feasible. Even if we augment Dijkstra's algorithm to check the feasibility of each *partial path*, it does not solve the problem. Assume that we check the capacity constraint of all the links in the interference set of each link before Dijkstra's algorithm selects the link to be used in a partial path. In this example, the augmented algorithm starts from node u_1 and creates partial paths $p'_1 = <u_1 \rightarrow u_2>$ and $p'_2 = <u_1 \rightarrow u_6>$ by relaxing the node. In the next steps, p'_1 is extended to $p'_1 = <u_1 \rightarrow u_2 \rightarrow u_3 \rightarrow u_4>$ which is a feasible partial path. After this point, no further extension is possible due to the following reasons. First, (u_4, u_5) is not selected since checking the capacity constraint for all $(u, v) \in I_{(u_4,u_5)}$ indicates that the capacity constraint of (u_2, u_3) and (u_3, u_4) is violated if the required bandwidth is allocated through path $<u_1 \rightarrow u_2 \rightarrow u_3 \rightarrow u_4 \rightarrow u_5>$. Second, Dijkstra's algorithm does not extend p'_2 through (u_6, u_2) because u_2 has already been visited. Therefore, even the augmented Dijkstra's algorithm cannot find the existing feasible path p_2.

It should be noted that as we prove in the following, this is not the problem of Dijkstra's algorithm. For every polynomial search algorithm, it is possible to construct a pathological topology in which the algorithm fails to find an existing feasible path. In the following, we analyze the effect of the interference models on the complexity of the problem.

4.1.2. k-hop interference model [29]

In the k-hop interference model, two links within k-hop range interfere with each other. In this section, we assume that the network is single-channel and analyze the effect of the value of k on the complexity of the FBCP problem. Multi-channel networks are discussed in Section 4.1.4.

Case 1. Node-Exclusive model ($k = 1$) [29]: In this model, only links that share an end-node interfere with each other; therefore, $I_{(u,v)} = \{(u', v') \text{ s.t. } u = u' \text{ or } u = v' \text{ or } v = u' \text{ or } v = v'\}$. We show that FBCP is polynomially solvable under this model if routing mechanism (routing algorithm in conjunction with routing metric) meets the *single-hop* requirement which is defined bellow.

Definition 5. *Let $p_1 = <u \rightarrow v>$ and $p_2 = <u \rightarrow u' \rightarrow \ldots \rightarrow v>$ be, respectively, the single-hop and a multi-hop feasible paths from u to v. A routing mechanism meets the single-hop requirement if it always selects p_1 instead of p_2*[1].

To show polynomial solvability of FBCP, it is sufficient to show that the conditions of Theorem 1 hold. Consider $\phi = (s, d, b, p)$, suppose that s and d are not directly connected, and for the sake of simplicity of presentation assume that all links have the same physical channel capacity. We make the following observations.

- $BC(\phi, (u, v)) = BC(\phi, (v, u))$ because $I_{(u,v)} = I_{(v,u)}$.
- $BC(\phi, (u, v)) = 0$ if $u, v \notin p$ because $AL(p) = \{(u, v) \text{ s.t. } u \in p \text{ or } v \in p\}$.
- $BC(\phi, (u, v)) = 3b$ if $(u, v) \in p$ and $u, v \notin \{s, d\}$ because $p \cap I_{(u,v)} = \{(v', u), (u, v), (v, u')\}$.
- $BC(\phi, (u, v)) = 2b$ if $(u, v) \in p$ and $u = s$ or $v = d$ because if $u = s$, we have $p \cap I_{(s,v)} = \{(s, v), (v, u')\}$ and if $v = d$ then $p \cap I_{(u,d)} = \{(v', u), (u, d)\}$.
- $BC(\phi, (u, v)) = 2b$ if $(u, v) \notin p$ and $u \in p \setminus \{s, d\}$ because due to the single-hop requirement $v \notin p$; hence, $p \cap I_{(u,v)} = \{(v', u), (u, v'')\}$.
- $BC(\phi, (u, v)) = b$, if $(u, v) \notin p$ and $u = s$ or $v = d$ because if $u = s$, due to the single-hop requirement $v \notin p$; hence we have $p \cap I_{(s,v)} = \{(s, v')\}$. If $v = d$, in the similar way, we have $p \cap I_{(u,d)} = \{(u', d)\}$.

[1] A simplified version of this requirement was also identified as "triangular inequality" in [10].

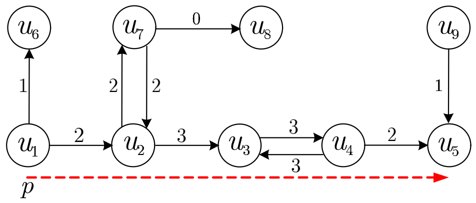


Figure 3: Illustration of bandwidth consumption under the node-exclusive interference model in single-channel WMNs. Label of each is link the bandwidth consumption by flow $(u_1, u_5, 1, p)$ at the link.

An example of the bandwidth consumption by a flow in a single-channel network under the node-exclusive interference model is shown in Fig. 3. Obviously, if the network is single-channel, the bandwidth consumption is the same for all paths from s to d. Consequently, due to Theorem 1 and Corollary 2, the problem is polynomially solvable.

Case 2. General case ($k \geq 2$): The computational complexity of bandwidth allocation in multi-hop wireless network under the *node-oriented* k-hop interference model was studied in [31]. This model implies that two *nodes* within k-hop distance of each other are interfering. The authors proved that the problem is NP-Complete for $k \geq 1$. In this paper, we use the *link-oriented* k-hop model [29]. It is easy to see that the node-oriented k-hop interference model is equivalent to the link-oriented $(k+1)$-hop model. Hence, their analysis shows that the FBCP problem is NP-complete under the link-oriented k-hop interference model for $k \geq 2$.

4.1.3. Interference range model [40] in single-channel networks

The interference range model is a special case of the well-known protocol model [38]. In this model, the interference range I_R is defined besides the transmission range T_R. Under this model, two links (u, v) and (u', v') are interfering if $d(u, u') \leq I_R$ or $d(u, v') \leq I_R$ or $d(v, u') \leq I_R$ or $d(v, v') \leq I_R$. In this section, we assume that network is single-channel and analyze the effect of the value of I_R on the complexity of FBCP.

Case 1. $I_{R,(u,v)} < T_{R,(u,v)}$: This is an *artificial* case where the interference set of (u, v) contains only links that have a common end-node with (u, v). Note that this is the definition of the node-exclusive model; hence, in this case, the FBCP problem is polynomially solvable as discussed in Section 4.1.2.

Case 2. $I_R \geq (1 + \gamma)T_R$: This case is usually used in the literature, where $1 \leq \gamma \leq 2$. It is easy to see that this model is equivalent to a generalization of the link-oriented k-hop interference model with $k \geq 2$, where different values of k are used for different links. Therefore, as proved in [31], the FBCP problem is NP-complete in this case.

4.1.4. Interference range model [40] in multi-channel networks

When the interference range model is used in multi-channel multi-radio WMNs, channel assignment affects the interference sets besides the Euclidean distance between nodes. Two links are *potentially* interfering if they are in the interference range of each other; but to be *actually* interfering they should also be assigned to the same channel. Channel assignment pattern and interference sets are influenced by the number of available channels in the frequency spectrum, Γ, and the number of radios of each node, r_u. We analyze their effects on the complexity of the FBCP problem in this section. The following analyses are based on the interference range model with $I_R \geq (1+\gamma)T_R$; however, they can be easily adapted to the k-hop interference model.

Case 1. No interference: There is not any interference in multi-channel multi-radio WMNs if there are sufficient available channels and radios. In this case, similar to wireline networks, we have $I_{(u,v)} = \big\{(u,v)\big\}$ and $AL(p) = p$. As a result, for a given flow $\phi = (s,d,b,p)$, we have $BC(\phi,(u,v)) = b$ if only if $(u,v) \in p$. Note that it is the same for all paths; hence, the requirements of Theorem 1 are met, and consequently, the FBCP problem can be solved polynomially.

Let $\overline{I}_{(u,v)}$ denote the set of *potentially* interfering links with (u,v). It is easy to see that an interference free channel assignment is achievable if $\Gamma \geq \max|\overline{I}_{(u,v)}|$ and $r_u \geq \deg(u)$. A unique channel should be assigned to each link in every interference set that means $\Gamma \geq \max|\overline{I}_{(u,v)}|$. In each node, a radio should be tuned to the unique channel of each incoming or outgoing link that means $r_u \geq \deg(u)$. It is important to note that these are sufficient conditions[2].

Case 2. Outgoing interfering links: This is a special case, in which only the outgoing links of each node interfere with each other; more formally, $I_{(u,v)} = \big\{(u',v') \text{ s.t. } u = u'\big\}$. This is accomplished by assigning the same channel to all outgoing links of each node; however, the channel must be unique in the interference range of the outgoing links. The number of potentially interfering links with (u,v) is $|\overline{I}_{(u,v)}|$, where $\frac{\deg(u)}{2}$ links are the outgoing links of the node (including the link itself). Therefore, we need $\Gamma \geq \max\big(|\overline{I}_{(u,v)}| - \frac{\deg(u)}{2} + 1\big)$ available channels. In each node, a channel is assigned to all outgoing links and a unique channel is needed for each incoming link; hence, at least $r_u \geq \big(\frac{\deg(u)}{2}+1\big)$ radios are needed at node u. Again, note that these are sufficient conditions to achieve the desired channel assignment[3].

Under this channel assignment, the FBCP problem is polynomially solvable since the conditions of Theorem 1 hold due to the following reasons. First, $BC(\phi,(u,v)) = BC(\phi,(u,v'))\ \forall(u,v),(u,v') \in E$ because both links have the same interference set. Second, $BC(\phi,(u,v)) = b$ if $(u,v) \in p$ since at most one outgoing link of each node belongs to the path and outgoing links of different nodes are not interfering. Hence, $BC(\phi,(u,v)) = BC(\phi,(u,v')) = b$ if and only if $u \in p$, and it is the same for all paths p.

Case 3. General case: In general multi-channel multi-radio WMNs with an arbitrary channel assignment, the FBCP problem is intractable as formally stated in the following theorem.

Theorem 3. *The FBCP problem in multi-channel multi-radio WMNs with an arbitrary channel assignment under the interference range model with $I_R \geq (1+\gamma)T_R$ is NP-Complete.*

[2] The necessary and sufficient conditions are $r_u = \deg(u)$ and $\Gamma = \chi(\overline{IG})$, where $\chi(\overline{IG})$ is the chromatic number of the potentially interference graph. In potentially interference graph $\overline{IG}$, each vertex represents a link in G, and there is an edge between two vertices if their corresponding links are potentially interfering with each other.

[3] Deriving the necessary conditions is not easy in this case.

Table 3: Summary of the complexity of the FBCP problem

Interference Model	Network/Model Configuration	The Complexity of FBCP
k-hop model	$k = 1$ & single hop requirement & single-channel network	Polynomially solvable
	$k \geq 2$	NP-Complete
Interference range model in single-channel WMNs	$I_{R,(u,v)} < T_{R,(u,v)}$	Polynomially solvable
	$I_R \geq (1+\gamma)T_R$	NP-Complete
Interference range model in multi-channel WMNs with $I_R \geq (1+\gamma)T_R$	$\Gamma \geq \max \lvert\overline{I}_{(u,v)}\rvert$ & $r_u \geq \deg(u)$	Polynomially solvable
	$\Gamma \geq \max \left(\lvert\overline{I}_{(u,v)}\rvert - \frac{\deg(u)}{2} + 1\right)$ & $r_u \geq \left(\frac{\deg(u)}{2} + 1\right)$	Polynomially solvable
	General case	NP-Complete
Physical model	General case	NP-Complete

Proof. The proof can be found in Appendix A.2. □

4.1.5. Physical model [38]

The physical interference model is another commonly used model. It is based on the Signal to Interference Noise Ratio (SINR) concept. A transmission on link (u, v) is successful if SINR of the signal received at v is greater than a predefined threshold. In [40], the authors showed that the interference range model with $I_R \geq (1+\gamma)T_R$ is a special case of the physical model. Consequently, according to the case 2 in Section 4.1.3 and case 3 in Section 4.1.4, the FBCP is NP-complete under the physical interference model in both single and multi-channel networks.

Table 3 summarizes the results of the complexity analyses presented in this section.

4.2. Complexity of efficient network utilization

Efficient network resource utilization, which is formally stated by the MAR problem, is a network-wide optimization problem. Finding its optimal solution is very difficult. In fact, its off-line version in the wireline network, where there is not any interference and the information about all demands is given at the beginning, is NP-Hard [5]. Clearly, the on-line version in WMNs that have very complicated interference patterns and there is not any information about future demands is much more difficult.

Dynamic routing is a well-developed approach to obtain a good approximate solution for this problem [6]. In this approach, the *minimum length feasible* path is selected for each demand, where the length of the path is a monotonically increasing function of link loads. Using this approach in WMNs has its own complexities. If the FBCP problem is polynomially solvable, the *minimum length feasible* path problem is also solved in polynomial time using the shortest path algorithms. However, if FBCP is NP-Complete, this problem is extremely difficult. From the complexity theory point of view, the problem is NP-Hard since it is the optimization version of an NP-Complete decision problem, the FBCP problem.

An example, where Dijkstra's algorithm fails to find the minimum length feasible path is illustrated in Fig. 4. Assume that $c = 10$ bps, demand $(u_1, u_8, 6\text{ bps}, 0, 1)$ has been arrived, and the weight of each link is one. Similar to Fig. 2, bandwidth consumption depends on path; so, the pruning rules in Theorem 1 do

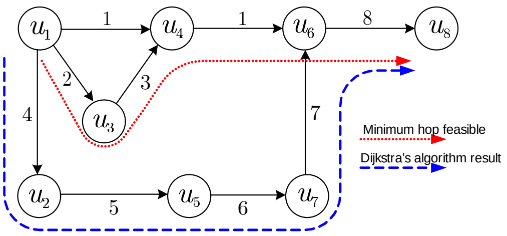


Figure 4: A pathological topology where Dijkstra's algorithm cannot find the minimum hop feasible path from u_1 to u_8 with bandwidth 6. The label of each link is the channel assigned to the link and $c = 10$ bps. Under a given interference model, $I_{(u_1,u_4)} = I_{(u_4,u_6)} = \{(u_1, u_4), (u_4, u_6)\}$ and the remaining links do not interfere with each other due to the channel assignment.

not solve the problem. We use the augmented Dijkstra's algorithm, which is explained in Section 4.1.1. It starts from u_1; in relaxing this node, it visits u_2, u_3, and u_4. In relaxing node u_4, it does not select (u_4, u_6) because partial path $<u_1 \to u_4 \to u_6>$ is not feasible due to the interference between (u_1, u_4) and (u_4, u_6). In relaxing u_3, it does not select (u_3, u_4) because u_4 has already been visited. The algorithm continues relaxing u_2, u_5, u_7, and u_6 sequentially. Finally, it selects path $p_1 = <u_1 \to u_2 \to u_5 \to u_7 \to u_6 \to u_8>$ while the minimum hop feasible path is $p_2 = <u_1 \to u_3 \to u_4 \to u_6 \to u_8>$.

5. Proposed solution

In this section, we first explain how to deal with the problems of finding a feasible path and efficient network utilization. Then, we integrate our solutions into an algorithmic framework. Finally, we analyze the computational complexity of the framework.

5.1. Finding a feasible path

Finding a feasible path is composed of three functionalities: network pruning, searching, and satisfying feasibility. Our proposed mechanisms for them are as follows.

5.1.1. Network pruning

For a given instance (G, I, Φ, δ) of the FBCP problem, we prune the network according to $AAB(u, v)$; (u, v) is pruned if $AAB(u, v) < b$. The reason is that if $AAB(u, v) < b$, routing the demand through (u, v) violates the capacity constraint of at least one link in the interference set of (u, v). It must be noted that this pruning does not affect the *solution space* but shrinks the *search space* more than pruning according to $ALB(u, v)$.

5.1.2. Satisfying feasibility

Since network pruning does not guarantee feasibility of paths in the pruned network, we use a call admission control (CAC) mechanism to maintain path feasibility. The CAC is performed in a *hop-by-hop* manner as follows. Assume $p = <s \to \ldots \to u>$ is a feasible partial path from s to u. This path can be extended one hop through (u, v) only if allocating the required bandwidth b

through path $p' = p \oplus <u \to v>$ does not violate the capacity constraint of any link. More precisely, p can be extended to p' only if for flow $\phi = (s, v, b, p')$, we have $BC(\phi, (u, v)) \leq ALB(u, v) \ \ \forall (u, v) \in AL(p')$. Obviously, if $v = d$, path p' will be a feasible path for the demand and it is admitted.

5.1.3. Search strategy

To search for a feasible path, we use the k-shortest path algorithm that allows revisiting each node up to k times. We use the k-shortest path strategy due to the shortcoming of Dijkstra's algorithm in finding feasible paths as demonstrated by the example in Fig. 2. It arises from the fundamental property of Dijkstra's algorithm that *each node is visited only one time*. Reconsider Fig. 2; in this figure, Dijkstra's algorithm does not find feasible path $p_2 = <u_1 \to u_6 \to u_2 \to u_3 \to u_4 \to u_5>$, since it does not allow *revisiting* node u_2 through partial path $<u_1 \to u_6 \to u_2>$. In this example, if we use the 2-shortest path algorithm, it extends partial path $<u_1 \to u_6>$ to $<u_1 \to u_6 \to u_2>$, and finally finds the feasible path p_2.

5.2. Efficient network utilization

We use the dynamic routing technique to find an approximate solution for the MAR problem. Dynamic routing is composed of two subproblems: selecting an appropriate routing metric, which will be discussed later, and finding the minimum length paths. We use a selector function besides the k-shortest path algorithm to approximate the minimum length feasible paths. We run the k-shortest path algorithm using path length function $l(p)$, but do *not* terminate the algorithm as soon as it reaches the destination. The algorithm always attempts to find k paths, which are stored in set P. We select the best path among the k paths by a selector function $s(P)$. The selector function can be either $\text{argmin}_{p \in P} \ l(p)$ or a combination of $l(p)$ and other functions. This approach improves optimality of the result. For example, in Fig. 4, if we use the 2-shortest path algorithm with $l(p) = \sum_{(u,v) \in p} 1$ and selector function $s(P) = \text{argmin}_{p \in P} \ l(p)$, the 2-shortest path algorithm finds both paths $p_1 = <u_1 \to u_2 \to u_5 \to u_7 \to u_6 \to u_8>$ and $p_2 = <u_1 \to u_3 \to u_4 \to u_6 \to u_8>$, $P = \{p_1, p_2\}$, and the selector function selects p_2, which is the minimum hop feasible path.

The efficiency of dynamic routing technique depends on the path length function in use. Conventional routing metrics proposed for this purpose in wireline networks are based on the *bandwidth of path*, $bw(p)$. It is the maximum bit rate at which data can be transmitted through the path without violating the capacity constraint. Assume that a flow at rate f is transmitting through path p, its bandwidth consumption at link (u, v) is

$$c_{(u,v)} \Big(\sum_{(u',v') \in p \cap I_{(u,v)}} \frac{f}{c_{(u',v')}} \Big) = f \Big(\sum_{(u',v') \in p \cap I_{(u,v)}} \frac{c_{(u,v)}}{c_{(u',v')}} \Big).$$

This bandwidth consumption can be at most $ALB(u, v)$ in order to satisfy the capacity constrain (1); hence,

$$f \Big(\sum_{(u',v') \in p \cap I_{(u,v)}} \frac{c_{(u,v)}}{c_{(u',v')}} \Big) = ALB(u, v),$$

and consequently,

$$f = \frac{ALB(u,v)}{\sum_{(u',v')\in p\cap I_{(u,v)}} \frac{c_{(u,v)}}{c_{(u',v')}}}.$$

Therefore, the bandwidth of path is defined as follows.

Definition 6. *The bandwidth of path p:* $bw(p) = \min_{(u,v)\in AL(p)} \frac{ALB(u,v)}{\sum\limits_{(u',v')\in p\cap I_{(u,v)}} \frac{c_{(u,v)}}{c_{(u',v')}}}$.

In the following, we consider a few commonly used QoS routing metrics in wireline networks, and explain how they can be adapted for WMNs.

5.2.1. Hop count and least usage metrics

In spite of the fact that the hop count path length function, $l_{HC}(p) = \sum_{(u,v)\in p} 1$, is *not* a dynamic function, it is still being used in bandwidth constrained routing in wireline networks, since it minimizes network resource consumption [41]. In WMNs, resource consumption by flows at a link depends on the size of the interference set of the link. Consequently, a path with minimum resource usage can be found by minimizing the following path length function.

Definition 7. *Least Usage path length function [42]:* $l_{LU}(p) = \sum\limits_{(u,v)\in p} |I_{(u,v)}|$.

5.2.2. Widest Shortest Path (WSP)

We find an *approximate* solution for the WSP problem using the k-shortest path algorithm and an appropriate selector function. We find k minimum hop paths using the k-shortest path algorithm, and if there are multiple same length paths, the selector function selects the widest one. The bandwidth of the path is defined by Definition 6.

5.2.3. Shortest Widest Path (SWP)

The authors in [18] approximated the shortest widest path in single-channel networks under the 2-hop interference model. Finding SWP is harder than WSP because according to Definition 6, the bandwidth of path cannot be determined before it is completed. Here, we *approximate* a solution for SWP using path length function $l_{WP}(p)$ defined as follows.

Definition 8. *Widest Path length function:* $l_{WP}(p) = \max_{(u,v)\in p} \frac{1}{AAB(u,v)}$.

We use the k-shortest path algorithm and find the k widest paths according to $l_{WP}(p)$. If there are multiple paths with the same width, we select the one that has the minimum number of hops.

5.2.4. Minimum criticality metric

Comparisons between the path length functions proposed for dynamic bandwidth constrained routing in wireline networks were carried out in [6, 41], and showed that the following path length function has superior performance.

Definition 9. *Reversed Link Bandwidth path length function:* $l_{RLB}(p) = \sum\limits_{(u,v)\in p} \frac{1}{ALB(u,v)}$.

Some previous works, e.g., [31, 43], used this routing metric in WMNs. We augment it by two observations. First, in wireless networks, congestion occurs in an *interference region* not at a single link. Therefore, we use $AAB(u,v)$ instead of $ALB(u,v)$. Second, in addition to the available bandwidth, criticality of a link also depends on the size of its interference set. Larger interference sets imply that links have to share bandwidth with more other interfering links; in other words, it means more criticality. Hence, we take the interferences into account, and define the following path length function.

Definition 10. *Minimum Criticality path length function:* $l_{MC}(p) = \sum\limits_{(u,v)\in p} \frac{|I_{(u,v)}|}{AAB(u,v)}$.

5.3. Adjustable algorithmic framework

In this section, we present the Adjustable Constrained Routing Algorithmic Framework (ACRAF), where the aforementioned algorithms and routing metrics are integrated into a single framework. The value of k in the k-shortest path algorithm, the path length function $l(p)$, and the selector function $s(P)$ are the adjustable parameters of this framework, which are discussed in more detail in Section 5.3.1.

Algorithm 1 shows how an ACRAF-based QoS routing is implemented. Algorithms 2 and 3, respectively, depict the pseudo-code of ACRAF and the k-SP algorithm. ACRAF prunes the network in step 1. In step 2, the k-SP algorithm attempts to find a set of k minimum length feasible paths. Finally, the selector function selects the best path in step 3.

Algorithm k-SP is the integration of the k-shortest path algorithm and the CAC mechanism described in Section 5.1.2. In k-SP, each node v, except the source node, stores up to k shortest paths from the source to itself together with the corresponding length in the array $v[1 \dots k]$. The predecessor of v in the ith shortest path and the corresponding length are stored in $v[i].\pi$ and $v[i].l$, respectively. There are two parameters in k-SP: $w_{l(p)}(u,v)$ and $\otimes_{l(p)}$, which are depended on the path length function $l(p)$. Parameter $w_{l(p)}(u,v)$ is the length of link (u,v) and $\otimes_{l(p)}$ is the operator that computes path length from the lengths of the links. The values of the parameters for the aforementioned path length functions are shown in Table 4.

Algorithm 1 : An ACRAF-based QoS routing

```
Inputs: $G = (V, E, C)$, $I$, and $\Delta$
Outputs: Set of accepted demands
Require: $\Delta$ is sorted in ascending order of $t_i$
Set parameters $k$, $l(p)$, and $s(P)$
for $i = 1$ to $|\Delta|$ do
    $\delta \leftarrow \Delta[i]$
    ACRAF$(G, I, \delta)$
    if there is feasible path then
        Add $\delta$ to the set of accepted demands
return The set of accepted demands
```

The k-SP algorithm initializes the required data structures in lines 1–4. ExtractMin finds the unvisited instance $u[j]$ that has the minimum $u[j].l$ and

Algorithm 2 : ACRAF

```
Inputs: $G = (V, E, C)$, $I$, and $\delta = (s, d, b, t, e)$
Outputs: $p = <s \rightarrow \ldots \rightarrow d>$
Parameters: $k$, $l(p)$, and $s(P)$
Prune $(u, v) \in E$ if $AAB(u, v) < b$
$P \leftarrow k$-SP$(G, I, \delta, k, l(p))$
$p \leftarrow s(P)$
```

Algorithm 3 : k-SP

```
Inputs: $G$, $I$, $\delta$, $k$, $l(p)$
Outputs: $P = \{p_1, \ldots, p_k\}$
 1: for $\forall v \in V \setminus s$ do
 2:     for $i = 1$ to $k$ do
 3:         $v[i].\pi \leftarrow$ NIL, $v[i].l \leftarrow \infty$, and add $v[i]$ to $Q$
 4: $s[1].\pi \leftarrow$ NIL, $s[1].l \leftarrow 0$, and add $s[1]$ to $Q$
 5: while $Q$ is not empty do
 6:     $u[j] \leftarrow$ ExtractMin$(Q)$
 7:     $p' \leftarrow$ GetPartialPath$(G, u[j])$
 8:     for $\forall (u, v) \in E$ and $v \notin p'$ do
 9:         if IsFeasible$(p' \oplus <u \rightarrow v>)$ then
10:             update $\leftarrow$ false
11:             for $i = 1$ to $k$ do
12:                 if update $=$ false and $v[i].l > \big(w_{l(p)}(u, v) \otimes_{l(p)} u[j].l\big)$ then
13:                     $v[i].l \leftarrow \big(w_{l(p)}(u, v) \otimes_{l(p)} u[j].l\big)$
14:                     $v[i].\pi \leftarrow u[j]$
15:                     update $\leftarrow$ true
16:                     DecreaseKey$(Q, v[i])$
17: for $i = 1$ to $k$ do
18:     Add GetPartailPath$(G, d[i])$ to $P$
19: return $P$
```

Table 4: Length of link and path length computing operator for the path length functions

$l(p)$	$\otimes_{l(p)}$	$w_{l(p)}(u, v)$
$l_{HP}(p)$	$+$	1
$l_{LU}(p)$	$+$	$\lvert I_{(u,v)} \rvert$
$l_{RLB}(p)$	$+$	$\frac{1}{ALB(u,v)}$
$l_{MC}(p)$	$+$	$\frac{\lvert I_{(u,v)} \rvert}{AAB(u,v)}$
$l_{WP}(p)$	max	$\frac{1}{AAB(u,v)}$

removes it from Q. GetPartialPath$(G, u[j])$ returns the partial path p', from the source node to node u. In lines 8–16, k-SP relaxes $u[j]$, in which u updates the length of $v[i]$ if the following conditions hold: i) Node v is not in the partial path p', which is checked in line 8; and ii) Path $p' \oplus <u \rightarrow v>$ is a feasible path, which is checked by IsFeasible(p) in line 9; and iii) As examined in line 12, the current length of the partial path $<s \rightarrow \ldots \rightarrow v[i]>$ is greater than the length of the new partial path $p' \oplus <u \rightarrow v>$, that is $u[j].l \otimes_{l(p)} w_{l(p)}(u, v)$. In the relaxation of $u[j]$, k-SP must update *only one* of $v[i]$s through link (u, v),

Table 5: QoS routing algorithms based on ACRAF

Name	$l(p)$	$s(P)$
Wk-MHC	$l_{HC}(p)$	argmin $l_{HC}(p)$
Wk-WSP	$l_{HC}(p)$	argmax bw(argmin $l_{HC}(p)$)
Wk-SWP	$l_{WP}(p)$	argmin l_{HC}(argmin $l_{WP}(p)$)
Wk-RLB	$l_{RLB}(p)$	argmin $l_{RLB}(p)$
Wk-WLU	$l_{LU}(p)$	argmax bw(argmin $l_{LU}(p)$)
Wk-MC	$l_{MC}(p)$	argmin $l_{MC}(p)$

which is enforced by variable *update*.

5.3.1. Parameters of ACRAF

There are three parameters in ACRAF: k, $l(p)$, and $s(P)$. The value of k determines the efficiency of ACRAF to deal with the NP-Completeness of the FBCP problem and to find the minimum length paths. If the value of k is not limited, ACRAF will be exact. It will find the optimal feasible path if any one exists at the cost of an exponential running-time. To achieve a pseudo-polynomial running-time, the value of k should be restricted. Obtaining the minimum value of k that guarantees the exactness of the algorithm is a very difficult problem.

Different combinations of path length function $l(p)$ and selector function $s(P)$ lead to different routing algorithms. The selector function may select the best path according to multiple metrics. We use the notation "argmin l_2(argmin $l_1(p)$)," if the best path is selected according to $l_1(p)$ and in the case of existing multiple minimum length paths, the selection is performed based on $l_2(p)$. Table 5 shows six different algorithms that can be implemented by adjusting the parameters of ACRAF.

5.3.2. Computational complexity analysis

The worst-case complexity of ACRAF depends on the complexity of the k-SP algorithm. We assume that it uses the Fibonacci heap. It creates heap Q in lines 1–4, which takes $O(kn\log(kn))$ times. The complexity of ExtractMin is $O(\log(kn))$. The complexity of GetPartialPath is $O(n)$. They execute kn times, therefore the total complexity is $O(kn(n + \log(kn))$. Lines 8–16 execute per $(u, v) \in E$ in finding a shortest path to all nodes; since k-SP finds k paths to each node, these lines execute km times. In these lines, checking $v \notin p'$ takes $O(n)$ times, IsFeasible takes $O(n\hat{I})$ times, where $\hat{I}$ is the size of the largest interference set, and the remaining operations are $O(1)$. Lines 17–18 take at most $O(kn)$ times. Combining all these running times yields to $O(k\text{-SP}) = O(kn\log(kn) + kn(n+\log(kn)) + km(n\hat{I}+n) + kn) = O(kn\log(kn) + kn^2 + knm\hat{I})$. The worst-case complexity of $s(P)$ is $O(kn\hat{I})$. Therefore $O(\text{ACRAF}) = O(k\text{-SP}) = O(kn\log(kn) + kn^2 + knm\hat{I})$.

6. Optimal solutions

In this section, we find optimal solutions for the FBCP and MAR problems through formulating them as optimization models. We first consider the FBCP problem and develop an optimization model to check the existence of at least one feasible path for a given demand. Then, we extend the model to find the minimum length feasible paths. Finally, for the MAR problem, we develop an optimal QoS routing algorithm.

6.1. Finding any feasible path

To check the existence of a feasible path for demand δ in network G, we do not need to optimize any objective function. The feasible path must satisfy two constraints. The first one is the conventional flow conservation constraint which is

$$\sum_{(u,v)\in E} x_{(u,v)} - \sum_{(v,u)\in E} x_{(v,u)} = \begin{cases} 1, & \text{if } u = s \\ -1, & \text{if } u = d \\ 0, & \text{otherwise} \end{cases} \qquad \forall u \in V, \tag{2}$$

and

$$x_{(u,v)} \in \{0,1\} \qquad \forall (u,v) \in E. \tag{3}$$

Where the *binary* variable x specifies the route of the demand; $x_{(u,v)} = 1$ if the demand is routed through (u,v) and $x_{(u,v)} = 0$ otherwise. The flow on each link is specified by

$$f_{(u,v)} = bx_{(u,v)} \qquad \forall (u,v) \in E. \tag{4}$$

Note that if there is a set of existing flows, $\Phi \neq \{\}$ in the FBCP problem, (4) should be $f_{(u,v)} = f'_{(u,v)} + bx_{(u,v)}$ where $f'_{(u,v)}$ is the existing flow on (u,v).

The second constraint is the capacity constraint. However, the capacity constraint for a link needs to be satisfied only if the link carries a load. We partition E into two subsets $E_1 = \big\{(u,v) \in E \text{ s.t. } f'_{(u,v)} > 0\big\}$ and $E_2 = E \setminus E_1$. For the links belong to E_1, the capacity constraint is

$$\sum_{(u',v')\in I_{(u,v)}} \frac{f_{(u',v')}}{c_{(u',v')}} \leq 1 \qquad \forall (u,v) \in E_1. \tag{5}$$

For the links in set E_2, the capacity constraint should be satisfied if the link is used to route the new demand. To model this *conditional* constraint, we use the *big M* technique and reformulate the capacity constraint (1) as

$$\sum_{(u',v')\in I_{(u,v)}} \frac{f_{(u',v')}}{c_{(u',v')}} \leq 1 + M\big(1 - x_{(u,v)}\big) \qquad \forall (u,v) \in E_2, \tag{6}$$

where the parameter M is a very big value[4]. When (u,v) is not used in routing, $x_{(u,v)} = 0$, the right hand side of (6) becomes a very big value, hence (6) does not impose any restriction. When $x_{(u,v)} = 1$, (6) turns into (5).

Putting these constraints altogether yields the following model. Its solution indicates whether there is a feasible path for demand δ in network G or not.

Model:	FEASIBLEPATH(G, I, δ)
Objective:	No objective function
Subject to:	(2)–(6).

It is important to note that this model is an Integer Linear Programming (ILP) model due to the binary variables $x_{(u,v)}$. ILP models are NP-Complete in general, and since we proved that FBCP is NP-Complete, the FEASIBLEPATH is also intractable, and may not be solved in a reasonable time.

[4] It is easy to show that M must be greater than the size of the largest interference set.

6.2. Optimizing path lengths

We mentioned before that the dynamic routing technique needs minimum length paths. The optimization model to find the minimum length feasible path for demand δ in network G using path length function $l(p)$ is obtained by extending FEASIBLEPATH. Let us assume link weights $w_{l(p)}(u,v)$ are additive; therefore, we need to optimize the following objective function.

$$\text{minimize} \sum_{(u,v)\in E} x_{(u,v)} w_{l(p)}(u,v). \tag{7}$$

In this model, similar to FEASIBLEPATH, the path should be feasible; thus, its constraints are (2)–(6). Consequently, the desired optimization model is

Model: OPTIMALPATH$(G, I, \delta, l(p))$
Objective: (7)
Subject to: (2)–(6).

6.3. Optimal on-line QoS routing

We develop an optimal QoS routing algorithm to solve the *on-line* version of the MAR problem according to the following observation. Routing metrics and route selection are issues in the network routing problem since it is assumed that existing flows in the network *cannot be rerouted*. If we relax this constraint, and assume that we can reoptimize flow routes at any given time, it is not important which path is selected for each flow. In other words, the routing metric is not a matter. Obviously, this is an optimal on-line strategy since we reoptimize the network every time it is needed.

In the optimal QoS routing problem, we only need to reoptimize routs at demand arrival times. When a new demand arrives, the optimal algorithm checks the existence of a feasible solution for the *set active demands* through solving an optimization problem. It accepts the new demand if a feasible solution exists for the set; otherwise, the new demand is rejected. The set of active demands denoted by $\overline{\Delta}$ contains the new demand and all the existing flows (demands that were accepted and have not finished yet). The optimization model we need in this algorithm is very similar to FEASIBLEPATH with the difference that we should find feasible paths for a set of flows. We relax FEASIBLEPATH to get a Linear Programming (LP) model, which is much easier than the ILP model. In this relaxation, we remove the binary variable x and assume that flows are splittable; in other words, we use multi-path routing. Furthermore, we use (1) for the capacity constraint instead of (5) and (6). This shrinks the solution space of the problem because we enforce the capacity constraint for all links even if they do not carry any load. However, its effect is negligible specially when there are several flows in the network. Because in this case, the flows are split over almost all links, and consequently the combination of (5) and (6) is equivalent to (1). The optimization model is obtained as follows. Constraints (3) and (4) are removed, and constraint (2) now becomes

$$\sum_{(u,v)\in E} f_{(u,v),\delta_i} - \sum_{(v,u)\in E} f_{(v,u),\delta_i} = \begin{cases} b_i, & \text{if } u = s_i \\ -b_i, & \text{if } u = d_i \\ 0, & \text{otherwise} \end{cases} \quad \forall \delta_i \in \overline{\Delta}, \forall u \in V, \tag{8}$$

where $f_{(u,v),\delta_i}$ is the flow of demand δ_i on link (u,v). The capacity constraint is

$$\sum_{(u',v')\in I_{(u,v)}} \frac{\sum_{\delta_i\in\overline{\Delta}} f_{(u',v'),\delta_i}}{c_{(u',v')}} \leq 1 \qquad \forall (u,v) \in E. \tag{9}$$

Finally, the model is

Model: FeasibleSet$(G, I, \overline{\Delta})$
Objective: No objective function
Subject to: (8) and (9).

Note that solving this model may *reroute* all the exiting flows. The optimal QoS routing algorithm is developed using this optimization model as shown in Algorithm 4. In this algorithm, if δ_i is rejected, it is removed from the set of active demands in line 9; moreover, it is removed in line 12 if it is finished.

Algorithm 4 : OptimalQR

```
Inputs: $G$, $I$, and $\Delta$
Outputs: The set of accepted demands
Require: $\Delta$ is sorted in ascending order of $t_i$
 1: Create empty set $\overline{\Delta}$
 2: for $i = 1$ to $|\Delta|$ do
 3:     $\delta_i \leftarrow \Delta[i]$
 4:     Add $\delta_i$ to $\overline{\Delta}$
 5:     Solve FeasibleSet$(G, I, \overline{\Delta})$
 6:     if FeasibleSet has a feasible solution then
 7:         Add $\delta_i$ to the set of accepted demands
 8:     else
 9:         Remove $\delta_i$ from $\overline{\Delta}$
10:     for each $\delta_j \in \overline{\Delta}$ do
11:         if $e_j < t_{i+1}$ then
12:             Remove $\delta_j$ from $\overline{\Delta}$
13: return The set of accepted demands
```

7. Simulation results

In this section, we present simulation results to evaluate the performance of ACRAF and the routing algorithms based on it. First, we consider the performance of ACRAF to deal with the NP-Completeness of the FBCP problem. Next, we investigate the problem of optimizing path lengths. Then, the performance of the routing algorithms in Table 5 is compared. Finally, we present the results on the overhead of ACRAF.

7.1. Simulation setup

The performance of bandwidth constrained routing algorithms depends on network topology (average node degree) and network load [44]. We evaluate the algorithms in different topologies, which are shown in Table 6. These topologies are general multi-channel multi-radio WMNs, in which $c = 100$ Mbps, $\Gamma = 10$, and the number of radios of each node is a uniform random variable in the

Table 6: Simulated topologies

Name	Area (m^2)	n	m	Node placement	Average node degree
Dense-10	675×675	100	1004	Grid	10.04
Dense-8	525×525	64	612	Grid	9.56
Random	1000×1000	100	656	Random	6.56
Sparse	1350×1350	100	360	Grid	3.6

range from 2 to 5. A static channel assignment is performed by the Greedy minimum interference algorithm [45]. The interference sets are computed using the interference range model, where $T_R = 150$ m and $I_R = 350$ m. In dense and sparse grid topologies, the distance between nodes in each row (column) is $\frac{T_R}{2}$ and T_R, respectively.

We use a flow-level event-driven simulator developed in Java. In each experiment, there is a set of demands $\Delta = \{\delta_i = (s_i, d_i, b_i, t_i, e_i)\}$. The source and destination of each demand are randomly chosen. Its bandwidth requirement is a uniform random variable in the interval [1,10] Mbps. The holding time, $e_i - t_i$, is an exponential random variable with mean 5 minutes. Demand arrival rate is a Poisson random variable.

In the FBCP problem, it is assumed that there is a number of exiting flows, which are denoted by set Φ. To simulate it, which is needed in Sections 7.2 and 7.3, we perform the following steps. At the beginning of the simulation, we create a demand, attempt to accept it using Wk-SWP, and allocate the required bandwidth if accepted; we repeat this procedure until the desired number of flows is generated. These accepted demands are not removed until the end of the simulation. This leads to an almost random distribution of $ALB(u, v)$ in the network. The results presented in the following are the average of 10 experiments with different sets Φ and Δ.

7.2. Dealing with NP-Completeness

To evaluate the performance of ACRAF to deal with the NP-Completeness of the FBCP problem, we compare it to the FeasiblePath model and use the following metric.

Definition 11. *The success rate of algorithm $\mathscr{A}$, $Sr(\mathscr{A})$, is the number of demands that algorithm $\mathscr{A}$ accepts divided by the number of demands that* FeasiblePath *accepts.*

This metric is computed as follows. At the beginning, as explained in Section 7.1, we create a number of existing flows. Then, for each demand $\delta \in \Delta$, we attempt to find a feasible path by the ACRAF-based routing algorithm and by the FeasiblePath model. We count the number of demands that each approach can accept. Since we only evaluate the ability to find a feasible path in this section, we do not create a flow for the accepted demands.

Fig. 5 shows the success rate of the Wk-MHC algorithm, which is selected as a sample of the ACRAF-based routing algorithms in this simulation, versus the number of existing flows, $|\Phi|$. This figure shows that, first, the FBCP problem is not very difficult especially in lightly loaded sparse networks as ACRAF achieves a high success rate. Second, ACRAF with $k = 5$ improves Sr(Wk-MHC) up to 9% in comparison to Dijkstra's algorithm (ACRAF with $k = 1$) in Fig. 5(a).

Third, there is a critical number of existing flows where ACRAF has the worst success rate, e.g., 35 in Fig. 5(a), 42 in Fig. 5(b), and 60 in Fig. 5(c). When the number of existing flows is much less than the critical value, there are multiple feasible paths for each demand; in this case, ACRAF can find one of them easily. When the number of existing flows is higher than the critical value, there is not any feasible path for most of the demands; in this case, the network pruning and CAC procedures shrink search space significantly which leads to the high success rates.

7.3. Achieving path length optimality

We evaluate the ability of ACRAF to find the minimum length feasible paths by comparing it to the optimal solution obtained by the OptimalPath model. We use the optimality ratio metric, which is defined as follows, to quantify the ability.

Definition 12. *Optimality ratio of algorithm $\mathscr{A}$, $Or(\mathscr{A})$, is the length of the feasible path found by the algorithm for a given demand, divided by the length of the path found by the* OptimalPath *model for the demand.*

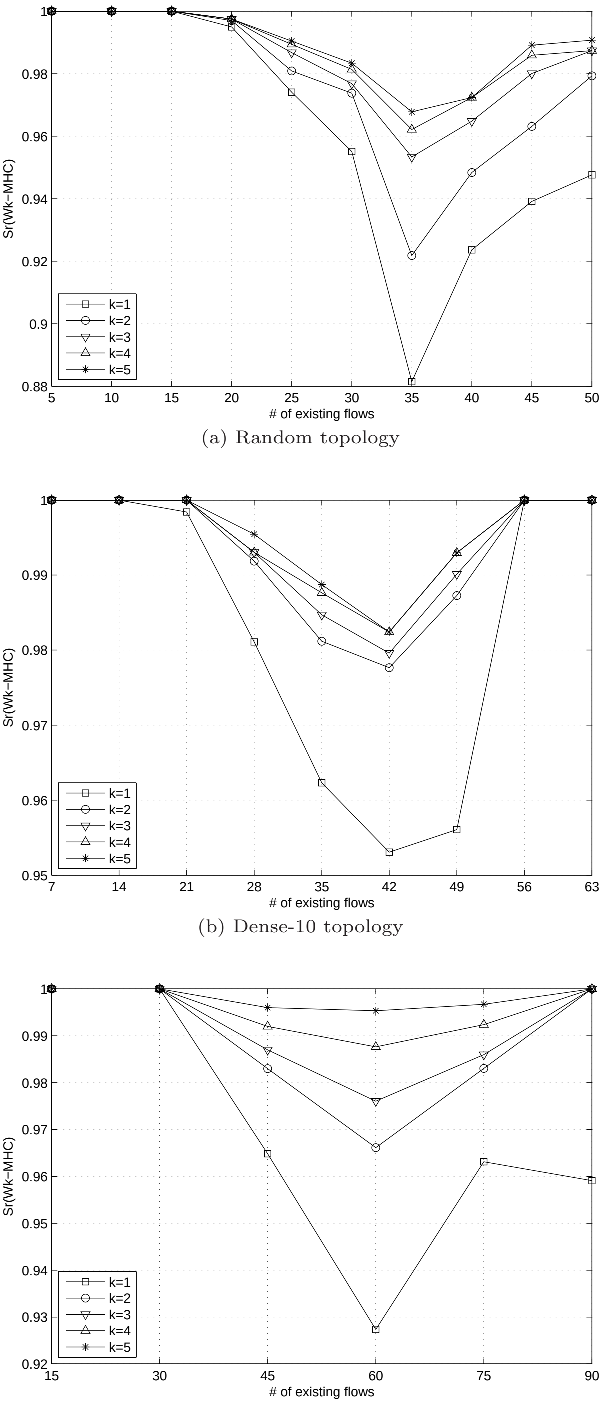
This metric is computed in a similar way that $Sr(\mathscr{A})$ is computed. We first create a number of existing flows; then, we compare the length of the feasible paths found by $\mathscr{A}$ and OptimalPath for each demand. Here, we again selected Wk-MHC as a sample of the ACRAF-based routing algorithms. The optimality ratios of Wk-MHC versus the number of existing flows in the Random, Dense-10, and Sparse topologies are shown in Fig. 6. These figures show that ACRAF, even with $k = 2$, performs notably better than Dijkstra's algorithm. In addition, it is seen that the paths found by ACRAF are typically not longer than the optimal path more than 0.2–0.6%. Optimality ratio has a similar behavior to the success rate versus the number of existing flows. When there are very few flows in the network, one of the k paths found by ACRAF, is the minimum length path; therefore, $Or \approx 1$. In the case of a high number of existing flows, a very limited number of demands are accepted; and for most of them, there is only one feasible path, which is the minimum length feasible path. Consequently, the paths found by ACRAF are optimal, that implies $Or \approx 1$.

7.4. Effect of routing metrics

In this section, we consider the MAR problem and study the effect of the routing metric used in ACRAF on the approximate solution obtained from it. In addition to the six ACRAF-based routing algorithms shown in Table 5, we simulated the OptimalQR algorithm. To evaluate the performance of the algorithms, we consider the following widely used metric.

Definition 13. *Acceptance rate of algorithm $\mathscr{A}$, $Ar(\mathscr{A})$, is the number of accepted demands by the algorithm divided by the total number of demands.*

To measure this metric in each experiment, we create a set of demands and apply the algorithms on them. Contrary to Sections 7.2 and 7.3, there is *not* any existing flow at the beginning, and we allocate the required bandwidth for each accepted demand to measure the efficiency of network resource utilization. Fig. 7 depicts the performance of the algorithms in the Random, Dense-8, and Sparse topologies. In these simulations, we used the Dense-8 topology instead

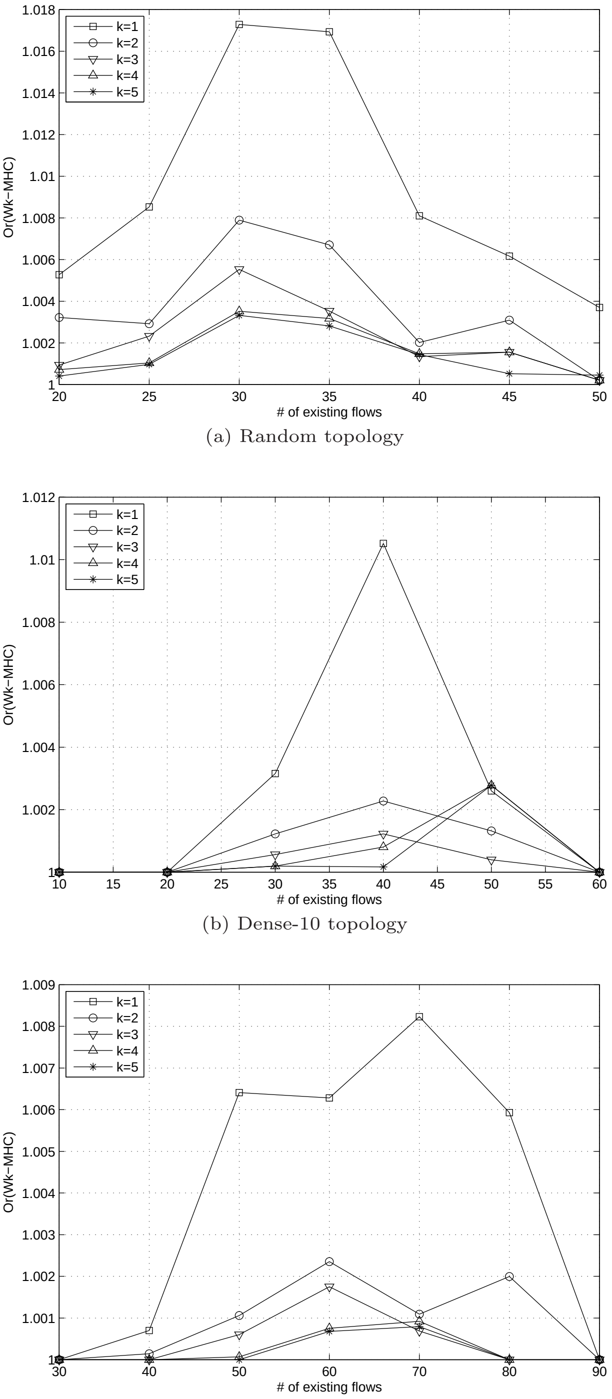


(a) Random topology

(b) Dense-10 topology

(c) Sparse topology

Figure 5: Success rate (Definition 11) of the Wk-MHC algorithm versus the number of existing flows, $|\Delta| = 200$.



(a) Random topology

(b) Dense-10 topology

(c) Sparse topology

Figure 6: Optimality ratio (Definition 12) of the Wk-MHC algorithm versus the number of existing flows, $|\Delta| = 200$.

of Dense-10 because solving the FeasibleSet model, which is needed in each iteration of OptimalQR, is time consuming in the Dense-10 topology.

These figures show that, first, the adapted versions of the traditional routing metrics outperform their corresponding wireline versions; for instance, Wk-MC outperforms Wk-RLB, and Wk-WLU outperforms Wk-WSP. Second, the average node degree has a considerable influence on the performance of the algorithms. Comparing Fig. 7(c) and 7(b) shows that the average acceptance rate of each algorithm in Dense-8 is much less than in the Sparse topology. Third, Wk-SWP and Wk-MC are, respectively, the worst and best routing algorithms, independent of the network topology and load. This is contrary to the results in wireline networks [41], which showed to optimize network performance in lightly loaded networks, routing metrics should give preference to load distribution, i.e., we should use SWP like algorithms. However, our results show that in WMNs, resource consumption should be preferred over load balancing regardless of the network load and topology.

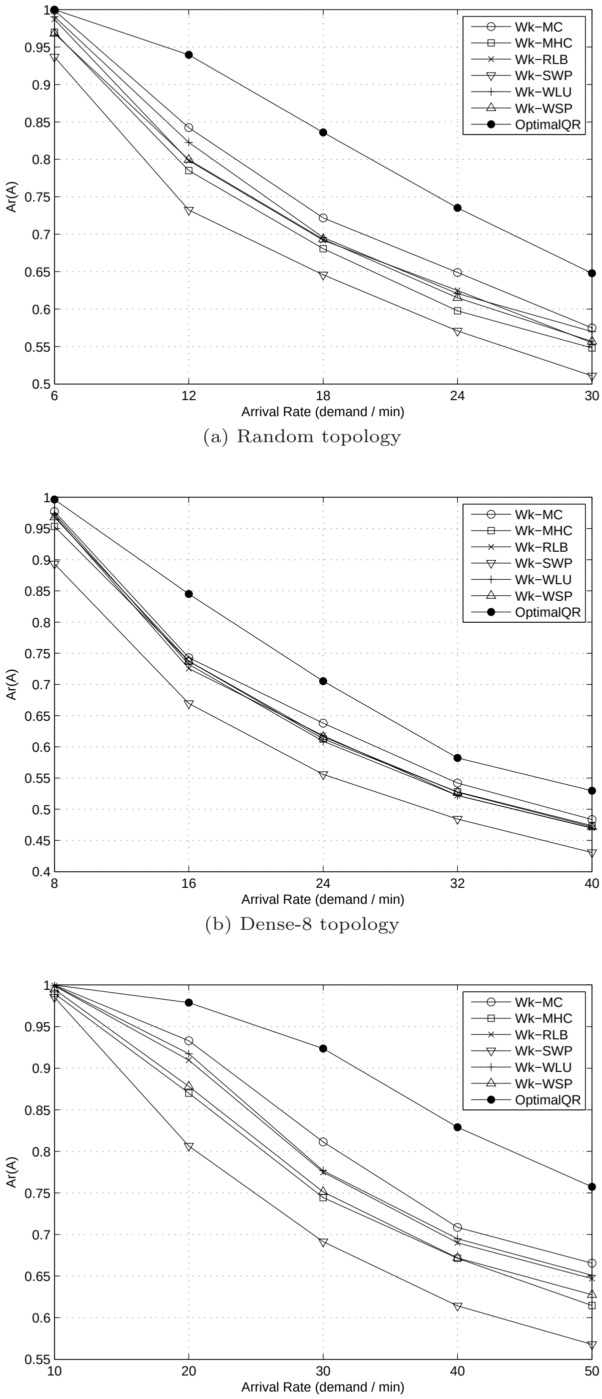
In these figures, as it is expected, the acceptance rate of the OptimalQR algorithm is better than the ACRAF-based algorithms due to two reasons. First, the algorithm is allowed to reroute all existing flows. Therefore, at each demand arrival time, if it is needed, OptimalQR reroutes existing flows to find a feasible path for the new demand. However, ACRAF-based algorithm cannot reroute existing flows; they have to provide sufficient resources for upcoming demands through finding an appropriate feasible path for each demand. Second, OptimalQR uses multi-path routing while the ACRAF-based algorithms are single-path. Even when there is not any single feasible path for a bandwidth intensive demand, OptimalQR can accept it by splitting the demand into multiple low-bandwidth flows and routing them.

7.5. Overhead

In Section 5.3, we mentioned that the running-time of ACRAF is pseudo-polynomial since the value of k is limited. This limitation causes that the ACRAF-based routing algorithms cannot find arbitrary feasible paths. In this section, we present simulation results on the trade-off between the overhead of ACRAF and its ability to find a feasible path with respect to the value of k.

We analyzed the computational complexity of ACRAF in Section 5.3.2, and showed it is mainly determined by the computational complexity of the k-SP algorithm. The computational overhead of k-SP is mostly due to relaxing links in lines 8–16. When the conditions in lines 8, 9, and 12 hold, link (u, v) is selected that updates the weight and predecessor of $v[j]$ in lines 13–16. According to these observations, we measure the overhead of ACRAF in terms of the number of updates in lines 13–16 per accepted demand, which is the overhead we have to pay to find each feasible path. In this section, since we intend to measure the trade-off between the overhead and the path-finding ability of ACRAF, we slightly modified it, ACRAF finishes as soon as it finds a feasible path.

We simulate ACRAF with three different values of k, small ($k = 3$), medium ($k = 20$), and large ($k = 200$). With the large value of k, ACRAF becomes an exhaustive search algorithm in our simulations, it finds a feasible path if any exists. Table 7 shows the simulation results, where the success rate of Wk-MHC and its overhead are presented. For each topology, we used three different numbers of existing flows, $|\Phi|$. These numbers were selected according to the critical value of each topology shown in Fig. 5. The first number is less



(a) Random topology

(b) Dense-8 topology

(c) Sparse topology

Figure 7: Acceptance rate (Definition 13) of the algorithms in Table 5 versus demand arrival rate. $k = 4$, and $|\Delta| = 500$.

Table 7: Overhead of ACRAF in terms of the number of updates per accepted demand

Configuration		k					
Topology	$\|\Phi\|$	3		20		200	
		Overhead	Sr	Overhead	Sr	Overhead	Sr
Sparse	50	146.53	0.996	778.48	1.0	3415.77	1.0
	60	133.01	0.982	730.93	0.996	3092.74	1.0
	70	87.41	1.0	412.05	1.0	1163.59	1.0
Dense-10	30	193.25	1.0	1001.21	1.0	6562.58	1.0
	40	147.78	0.98	731.75	0.995	4265.03	1.0
	50	137.41	0.991	664.56	0.991	3599.43	1.0
Random	25	180.69	1.0	944.91	1.0	6364.93	1.0
	35	153.46	0.962	788.41	0.995	5174.21	1.0
	45	85.06	1 .0	392.27	1.0	2171.53	1.0

than the critical value, the second one is near the value, and the last number is greater than the value. This table shows the following results. First, ACRAF achieves high success rates with an acceptable overhead using small values of k. However, to be an exact algorithm, the value of k should be very large that yields a significant overhead, e.g., up to 30 times in comparison to the small values. Second, the overhead diminishes considerably by increasing the number of existing flows. It is due to the network pruning. When many flows exist in the network, most of the links have not sufficient available bandwidth and are pruned, which shrinks the search space significantly. Third, the worst success rate of ACRAF in each topology is the case where $|\Phi|$ is about the critical value of the topology, this confirms the previous results depicted in Fig. 5.

8. Conclusions and future work

We have studied the problem of bandwidth constrained routing in WMNs. We analyzed the effect of interference models on the complexity of the problem, and showed that except a few special cases, the problem of finding a feasible path is NP-Complete. We proposed ACRAF to solve the problem. We also investigated the problem of optimum utilization of network resources. To achieve this, we used the dynamic routing technique and developed routing metrics to consider both interferences and bandwidth. We developed six routing algorithms by adjusting the parameters of ACRAF. Moreover, we developed three optimization models: a model to check the existence of a feasible path, a model to optimize path lengths, and another to find the maximum number of acceptable demands.

Comparisons between ACRAF and the optimization models showed that it can find most of existing feasible paths, optimizes path length efficiently, and has comparable performance to the optimal QoS routing. We simulated the ACRAF-based routing algorithms in three networks with different average node degrees under various network loads. It showed that the performance of the algorithms depends on network topology and offered load; however, in all cases, the Wk-MC algorithm outperforms the others.

In this paper, we assumed that the static interference sets are given, and solved the QoS routing problem. In the future, we plan to study the problem of joint QoS routing and interference management, which can be accomplished by e.g., channel assignment and power control.

9. Acknowledgment

We would like to thank the reviewers for their helpful comments and Fernando A. Kuipers for commenting on the proof of the NP-Completeness of FBCP.

Appendix A. Proofs

Appendix A.1. Proof of Theorem 1

The proof of Theorem 1 is as follows.

Proof. We prove both parts by contradiction.

Assume that there is an infeasible path p in the pruned network. Since flow ϕ through path p does not consume $ALB(u,v)$ if $(u,v) \notin AL(p)$, infeasiblity of the path can only be due to the following cases: there is $(u,v) \in p$ s.t. $ALB(u,v) < BC(\phi,(u,v))$ *or* there is $(u',v') \in AL(p) \setminus p$ s.t. $ALB(u',v') < BC(\phi,(u',v'))$. The first case is not possible because of the pruning rule 1. The second case implies that $\exists(u,v) \in p$ s.t. $(u',v') \in I_{(u,v)}$ and therefore $(u,v) \in I_{(u',v')}$, which is due to the assumptions about interference sets mentioned in Section 3.2. However, this is also impossible because if $ALB(u',v') < BC(\phi,(u',v'))$ then (u,v) must be pruned by the second rule.

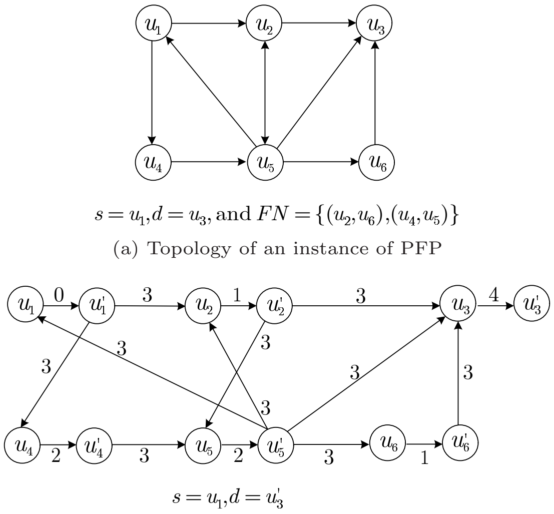
To prove the second part, let p be a feasible path that is excluded from the pruned network. Without loss of generality, assume that the path is excluded due to pruning link (u,v). Pruning (u,v) implies that there is a path p' and its corresponding flow $\phi' = (s,d,b,p')$ such that $ALB(u,v) < BC(\phi',(u,v))$ or $\exists(u',v')$ s.t. $(u,v) \in I_{(u',v')}$ and $ALB(u',v') < BC(\phi',(u',v'))$. Since the bandwidth consumption is the same for all paths, we have $ALB(u,v) < BC(\phi,(u,v))$ or $ALB(u',v') < BC(\phi,(u',v'))$ where ϕ is the flow corresponding to p. However, these mean that p is not feasible because either (u,v) or $(u',v') \in AL(p)$ has not sufficient available bandwidth. □

Appendix A.2. Proof of Theorem 3

We prove Theorem 3 by reducing the Path with Forbidden Pairs (PFP) problem to the FBCP problem. The PFP problem is defined as follows [46].

Problem: Path with Forbidden Pairs (PFP).
Instance: Graph $G = (V,E)$, a set of forbidden pairs $FN = \{\{u_1,v_1\},\ldots,\{u_q,v_q\}\}$, and nodes s and d.
Question: Is there any path $p = <s \to \ldots \to d>$ such that contains at most one vertex from each pair in FN?

Proof. For the sake of simplicity of presentation, we assume that in PFP each node belongs to only one forbidden pair. This restriction will be removed later. For a given instance (G,FN,s,d) of PFP, we construct an instance (G',I,Φ,δ) of FBCP. We assume that there is no restriction on the number of radios and channels. The key observation is that nodes are the conflicting entities in PFP but in FBCP with the interference range model, links are interfering with each other. Hence, to construct G', we replace nodes in G by links in G' as follows.



(a) Topology of an instance of PFP

(b) Topology of the instance of FBCP corresponding to the instance of PFP in (a).

Figure A.8: An instance of PFP and its corresponding instance of FBCP.

- Each node $u \in G$ is replaced by two nodes u and u' and a link (u, u') in G' where $c_{(u,u')} = 1$.
- The outgoing links of node u in G, $(u, v) \in E$, are represented by (u', v) in G' where $c_{(u',v)} = \infty$.

To form the desired interference sets,

- Set $I_R = \infty$, all links are in the interference range of each other.
- Assign a common channel to all links $\{(u', v)\ \forall u' \in G'\}$.
- For each pair $\{u, v\} \in FN$, assign the same channel to links (u, u') and (v, v'). However, the channel should be unique in the network.
- Assign a unique channel to every remaining link (u, u') where u does not belong to any forbidden pair in FN.

An instance of PFP and its corresponding transformed topology are depicted in Fig. A.8. To complete the description of the instance of FBCP, we set $\Phi = \{\}$ and $\delta = (s, d', b = 1, 0, 1)$. We use the following facts to complete the proof.

F1. $I_{(u,u')} = \{(u, u'), (v, v')\}$ if and only if $\exists\{u, v\} \in FN$, and since $c_{(u,u')} = c_{(v,v')} = 1$, only (u, u') or only (v, v') can be present in the solution of FBCP.

F2. $I_{(u,u')} = \{(u, u')\}$ if and only if $\nexists\{u, v\} \in FN$, and since $c_{(u,u')} \geq b$, it can be in the solution of FBCP independent of other links.

F3. Whereas all links $\{(u', v)\ \forall u' \in G'\}$ are interfering with each other, they can be in the solution of FBCP independent of each other because $c_{(u',v)} = \infty$.

We claim that path $p' = <s \rightarrow s' \rightarrow u_1 \rightarrow u'_1 \rightarrow u_2 \rightarrow u'_2 \rightarrow \ldots \rightarrow d \rightarrow d'>$ is a feasible solution for FBCP, if and only if path $p = <s \rightarrow u_1 \rightarrow u_2 \rightarrow \ldots \rightarrow d>$ is a feasible solution for PFP. Let p' be feasible. If $\{u, v\} \in FN$, due to fact **F1**, only (u, u') or (v, v') is in p' which implies only u or v is in p and hence, p is feasible. Assume p is a feasible path for PFP, p' is a feasible solution for FBCP because the capacity constraint of (u', v) $\forall u' \in G'$ is satisfied due to fact **F3**, this constraint is satisfied for (u, u') due to **F2** if $\nexists\{u, v\} \in FN$ and due to feasibility of p and **F1** if $\exists\{u, v\} \in FN$. These complete the proof.

If node u belongs to multiple forbidden pairs, the proof is the same; however, the node should be replaced by multiple nodes and links. For example, assume $\{u, v_1\}$, $\{u, v_2\} \in FN$, node u is replaced by three nodes u, u', and u'' and two links (u, u') and (u', u''). The outgoing link (u, v) is represented by (u'', v), the same channel is assigned to both links (u, u') and (v_1, v'_1), and the same channel is assigned to both links (u', u'') and (v_2, v'_2). □

References

[1] Y. Yang, R. Kravets, Contention-aware admission control for ad hoc networks, IEEE Transactions on Mobile Computing 4 (2005) 363–377.

[2] R. Guerin, A. Orda, D. Williams, QoS routing mechanisms and OSPF extensions, in: IEEE GLOBECOM, 1997.

[3] Z. Wang, J. Crowcroft, Quality-of-service routing for supporting multimedia applications, IEEE Journal on Selected Areas in Communications 14 (1996) 1228–1234.

[4] M. Kodialam, T. Lakshman, Minimum interference routing with applications to MPLS traffic engineering, in: IEEE INFOCOM, 2000.

[5] S. Suri, M. Waldvogel, D. Bauer, P. R. Warkhede, Profile-based routing and traffic engineering, Computer Communications 26 (2003) 351–365.

[6] F. Ricciato, U. Monaco, Routing demands with time-varying bandwidth profiles on a MPLS network, Computer Networks 47 (2005) 47–61.

[7] Q. Xue, A. Ganz, Ad hoc QoS on-demand routing (AQOR) in mobile ad hoc networks, Journal of Parallel and Distributed Computing 63 (2003) 154–165.

[8] S. Chen, K. Nahrstedt, Distributed quality-of-service routing in ad hoc networks, IEEE Journal on Selected Areas in Communications 17 (1999) 1488–1505.

[9] Y. Ge, T. Kunz, L. Lamont, Quality of service routing in ad-hoc networks using OLSR, in: Hawaii International Conference on System Sciences, 2003.

[10] H. Zhu, I. Chlamtac, Admission control and bandwidth reservation in multi-hop ad hoc networks, Computer Networks 50 (2006) 1653–1674.

[11] A. Derhab, Low-Cost and Accurate Intra-flow Contention-Based Admission Control for IEEE 802.11 Ad Hoc Networks, in: ADHOC-NOW, 2008.

[12] L. Chen, W. Heinzelman, QoS-aware routing based on bandwidth estimation for mobile ad hoc networks, IEEE Journal on Selected Areas in Communications 23 (2005) 561–572.

[13] P. Sinha, R. Sivakumar, V. Bharghavan, CEDAR: a core-extraction distributed ad hoc routing algorithm, in: IEEE INFOCOM, 1999.

[14] V. Kone, S. Das, B. Y. Zhao, H. Zheng, QUORUM: quality of service routing in wireless mesh networks, in: QSHINE, 2007.

[15] Q. Xue, A. Ganz, QoS Routing for Mesh-Based Wireless LANs, International Journal of Wireless Information Networks 9 (2002) 179–190. 10.1023/A:1016085627790.

[16] X. Cheng, P. Mohapatra, S.-J. Lee, S. Banerjee, MARIA: Interference-Aware Admission Control and QoS Routing in Wireless Mesh Networks, in: IEEE ICC, 2008.

[17] R. Gupta, Z. Jia, T. Tung, J. Walrand, Interference-aware QoS routing (IQRouting) for ad-hoc networks, in: IEEE GLOBECOM, 2005.

[18] Z. Jia, R. Gupta, J. Walrand, P. Varaiya, Bandwidth guaranteed routing for ad-hoc networks with interference consideration, in: IEEE ISCC, 2005.

[19] J. Tang, G. Xue, W. Zhang, Interference-aware topology control and QoS routing in multi-channel wireless mesh networks, in: ACM MobiHoc, 2005.

[20] B. Raman, K. Chebrolu, D. Gokhale, S. Sen, On the Feasibility of the Link Abstraction in Wireless Mesh Networks, IEEE/ACM Transactions on Networking 17 (2009) 528–541.

[21] L. Chen, W. Heinzelman, A Survey of Routing Protocols that Support QoS in Mobile Ad Hoc Networks, IEEE Network 21 (2007) 30–38.

[22] T. B. Reddy, I. Karthigeyan, B. Manoj, C. S. R. Murthy, Quality of service provisioning in ad hoc wireless networks: a survey of issues and solutions, Ad Hoc Networks 4 (2006) 83–124.

[23] C. Liu, K. Leung, A. Gkelias, A Novel Cross-Layer QoS Routing Algorithm for Wireless Mesh Networks, in: IEEE ICOIN, 2008.

[24] H. Wang, K. C. Yow, QoS routing in multi-channel multihop wireless networks with infrastructure support, in: InterSense, 2006.

[25] T. Liu, W. Liao, Interference-aware QoS routing for multi-rate multi-radio multi-channel IEEE 802.11 wireless mesh networks, IEEE Transactions on Wireless Communications 8 (2009) 166 –175.

[26] C.-C. Hu, Y.-L. Kuo, C.-Y. Chiu, Y.-M. Huang, Maximum bandwidth routing and maximum flow routing inwireless mesh networks, Telecommunication Systems 44 (2010) 125–134. 10.1007/s11235-009-9217-2.

[27] R. Hou, K.-S. Lui, H.-S. Chiu, K. L. Yeung, F. Baker, Routing in multi-hop wireless mesh networks with bandwidth guarantees, in: ACM MobiHoc, 2009.

[28] T. Salonidis, M. Garetto, A. Saha, E. Knightly, Identifying High Throughput Paths in 802.11 Mesh Networks: a Model-based Approach, in: IEEE ICNP, 2007.

[29] G. Sharma, R. R. Mazumdar, N. B. Shroff, On the Complexity of Scheduling in Wireless Networks, in: ACM MobiCom, 2006.

[30] C.-Y. Chiu, Y.-L. Kuo, E.-K. Wu, G.-H. Chen, Bandwidth-Constrained Routing Problem in Wireless Ad Hoc Networks, IEEE Transactions on Parallel and Distributed Systems 19 (2008) 4–14.

[31] L. Georgiadis, P. Jacquet, B. Mans, Bandwidth reservation in multihop wireless networks: complexity and mechanisms, in: International Conference on Distributed Computing Systems, 2004.

[32] D. M. Shila, T. Anjali, Load aware traffic engineering for mesh networks, Computer Communications 31 (2008) 1460–1469. Special Issue: Resource Management and routing in Wireless Mesh Networks.

[33] R. Draves, J. Padhye, B. Zill, Routing in multi-radio, multi-hop wireless mesh networks, in: ACM MobiCom, 2004.

[34] Y. Yang, J. Wang, R. Kravets, Designing routing metrics for mesh networks, in: IEEE Workshop on Wireless Mesh Networks (WiMesh), 2005.

[35] Y. Yang, J. Wang, R. Kravets, Interference-aware Load Balancing for Multihop Wireless Networks, Technical Report UIUCDCS-R-2005-2526, Department of Computer Science, University of Illinois at Urbana-Champaign, 2005.

[36] A.P. Subramanian, M.M. Buddhikot, S.C. Miller, Interference aware routing in multi-radio wireless mesh network, in: WiMesh, 2006.

[37] R. Pal, A Lexicographically Optimal Load Balanced Routing Scheme for Wireless Mesh Networks, in: IEEE ICC, 2008.

[38] P. Gupta, P. Kumar, The capacity of wireless networks, IEEE Transactions on Information Theory 46 (2000) 388–404.

[39] R. Gupta, J. Musacchio, J. Walrand, Sufficient rate constraints for QoS flows in ad-hoc networks, Ad Hoc Networks 5 (2007) 429–443.

[40] A. Iyer, C. Rosenberg, A. Karnik, What is the Right Model for Wireless Channel Interference?, IEEE Transactions on Wireless Communications 8 (2009) 2662–2671.

[41] Q. Ma, P. Steenkiste, On path selection for traffic with bandwidth guarantees, in: International Conference on Network Protocols, 1997.

[42] Y. Yang, J. Wang, R. Kravets, Load-balanced routing for mesh networks, ACM Mobile Computing and Communications Review 10 (2006) 3–5.

[43] H. Aiache, V. Conan, L. Lebrun, S. Rousseau, A load dependent metric for balancing Internet traffic in Wireless Mesh Networks, in: IEEE MASS, 2008.

[44] K. Kowalik, M. Collier, Should QoS routing algorithms prefer shortest paths?, in: IEEE ICC, 2003.

[45] A. Subramanian, H. Gupta, S. Das, J. Cao, Minimum Interference Channel Assignment in Multiradio Wireless Mesh Networks, IEEE Transactions on Mobile Computing 7 (2008) 1459–1473.

[46] M. Garey, D. Johnson, Computers and intractability: A guide to the theory of NP-completeness, WH Freeman and Company, San Francisco, Calif, 1979.