Throughput Optimization Routing Under Uncertain Demand for Wireless Mesh Networks

Liang Dai, Yuan Xue, Bin Chang, Yi Cui
Department of Electrical Engineering and Computer Science, Vanderbilt University
{*liang.dai, yuan.xue, bin.chang, yi.cui*}@*vanderbilt.edu.*

Abstract

Wireless mesh networks have attracted increasing attention and deployment as a high-performance and low-cost solution to last-mile broadband Internet access. Network routing plays a critical role in determining the performance of a wireless mesh network. To study the best mesh network routing strategy which can maximize the network throughput while satisfying the fairness constraints, existing research proposes to formulate the mesh network routing problem as an optimization problem. These works usually make ideal assumptions such as known static traffic input. Whether they could be applied for practical use under the highly dynamic and uncertain traffic in wireless mesh network is still an open issue.

The main objective of this paper is to understand the effects of traffic uncertainty in wireless mesh networks and consider these effects in throughput maximization routing. It identifies the appropriate optimization framework that could integrate the statistical user traffic demand model into a tractable throughput maximization problem. The trace-driven simulation study demonstrates that our algorithm can effectively incorporate the traffic demand uncertainty in routing optimization, and outperform the throughput maximization routing which only considers static traffic demand.

1 Introduction

Wireless mesh networks [4, 3] have attracted increasing attention and deployment as a high-performance and low-cost solution to last-mile broadband Internet access. In wireless mesh networks, local access points and stationary wireless mesh routers communicate with each other and form a backbone network which forwards the traffic from mobile clients to the Internet.

Traffic routing in the mesh backbone network plays a critical role in determining the performance of a wireless mesh network. These existing routing solutions usually fall into two ends of the spectrum. On one end of the spectrum are the heuristic algorithms (*e.g.*, [19, 8, 20]). Although many of such approaches are adaptive to the dynamic environments of wireless networks, they lack the theoretical foundation to analyze how well the network performs globally (*e.g.,* whether the network resource is fully utilized, whether the flows share the network in a fair fashion). On the other end of the spectrum, there are theoretical studies that formulate these network planning decisions into optimization problems (*e.g.*, [5, 14]). Yet these results usually make ideal assumptions towards the network such as known static traffic input, which makes them unsuitable for practical use in the highly dynamic wireless networking environments.

To date, how mesh network could optimize its performance under dynamic user demand is still an open question. This question calls for a new framework that could characterize the traffic demand uncertainty and integrate its effect into optimal network routing. To answer this call, this paper investigates the traffic routing problem for throughput optimization with fairness constraint under uncertain demand. In particular, it studies how traffic from(to) local access points could be routed in a mesh network so that the minimum proportion of the demands from all local access points could be maximized.

If the traffic demand from each local access point is known a priori, the throughput optimization problem with fairness constraint could be formulated as a linear programming problem (maximum concurrent flow problem). To incorporate demand uncertainty into this optimization framework, this paper first characterizes the uncertain traffic demand using a random variable. Under this model, the proportions of traffic demands are also random variables for a given routing strategy. Thus this paper extends the maximum concurrent flow problem to a stochastic optimization problem where the expected value of the performance ratio between the achieved traffic demand proportion and its optimum is maximized. This paper further presents two fast $(1-\epsilon)$-approximation algorithms for throughput optimiza-

1-4244-1455-5/07/$25.00

tion under fixed and uncertain demand respectively.

To evaluate and compare the performance of these two algorithms under realistic network traffic environment, we conduct trace-driven simulation study. In particular, we derive the traffic demands from the access points of campus wireless LANs based on the traces collected at Dartmouth College [1]. These traffic demands are used to drive the simulation. Our simulation results demonstrate that our statistical problem formulation could effectively incorporate the traffic demand uncertainty in routing optimization, and our algorithm outperforms the conventional solution which only considers the static traffic demand.

The original contributions of this paper are two-fold. First, to the best of our knowledge, this is the first work that integrates statistical user traffic demand model into a tractable throughput optimization problem for wireless mesh networks. Second, this paper evaluates the practicability of optimization-based routing solutions using the trace data collected in the real wireless network environments.

The remainder of this paper is organized as follows. Sec. 2 presents the network and traffic demand model. Sec. 3 formulates the mesh network routing problem under fixed traffic demand based on maximum concurrent flow and presents a fast approximation algorithm. Sec. 4 extends the routing problem to handle uncertain traffic demand. We show simulation results in Sec. 5, present related work in Sec. 6 and finally conclude the paper in Sec. 7.

2 Model

2.1 Network Model

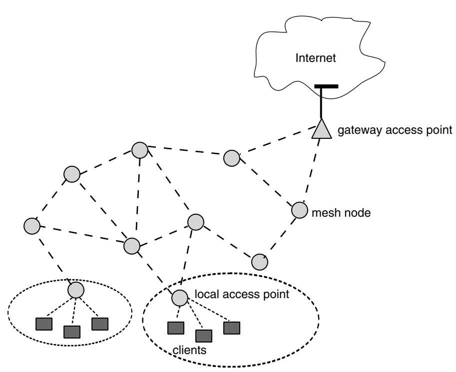


Figure 1. Illustration of Wireless Mesh Network

We consider a multi-hop wireless mesh network as illustrated in Fig. 1. In this network, local access points aggregate the traffic from mobile clients that are associated with them. They communicate with each other, also with stationary wireless routers, forming a multi-hop wireless backbone network which forwards the user traffic to a gateway access point connecting to the Internet. In the following discussion, local access point, gateway access point, and mesh routers are collectively called mesh nodes.

We model the backbone of a wireless mesh network as a directed graph $G = (V, E)$, where each node $u \in V$ represents a mesh node. Among these nodes, $g \in V$ is the gateway access point that connects to the Internet. A directed edge $e = (u, v) \in E$ denotes that u can transmit to v directly. We assume that all mesh nodes have a uniform transmission range denoted by R_T. We denote $r(u, v)$ as the distance between u and v. An edge $e = (u, v) \in E$ if and only if $r(u, v) \leq R_T$. We also use $r(e)$ to represent the length of edge e. Let $b(e)$ bit/sec be the data rate of edge e, which is maximum data that can be carried in a second along e.

2.2 Traffic Demand Model

This paper investigates the throughput optimization routing scheme for wireless mesh backbone network. Thus we consider the aggregated traffic among the mesh nodes. In particular, we regard the gateway access points as the source of all incoming traffic and the destination of all outgoing traffic of a mesh network. Similarly, the local access points, which aggregate the client traffic, serve as the sources of all outgoing traffic and the destinations of incoming traffic. For simplicity, we call the aggregated traffic that shares the same source and destination as a *flow* and denote it as $f \in F$, where F is the set of all aggregated flows. It is worth noting that although we consider only one gateway access point in this paper, our problem formulation and algorithm presented here could be easily extended to handle multiple gateway routers and inter-mesh-router traffic. Finally, we denote the rate of an aggregated flow $f \in F$ as x_f, and use $\boldsymbol{x} = (x_f, f \in F)$ to represent the aggregated flow rate vector.

2.3 Interference Model and Schedulability

In a wireless network, packet transmissions are subject to location-dependent contention, which means that simultaneous transmissions in the proximate region may interfere with each other and result in packet collision. Usually the interference range of a node is larger than its transmission range. Thus we denote the interference range of a node as $R_I = (1 + \Delta)R_T$, where $\Delta \geq 0$ is a constant.

In this paper, we consider the *protocol model* for packet transmission [11]. In the protocol model, packet transmission from node u to v is successful if and only if (1) the distance between these two nodes $r(u, v)$ satisfies $r(u, v) < R_T$; (2) any other node $w \in V$ within the interference range of the receiving node v, *i.e.*, $r(w, v) \leq R_I$, is not transmitting. Two edges e, e' interfere with each other, if they can not transmit simultaneously based on the protocol model. We use $I(e)$ denote the set of edges which interfere with edge e.

To study the throughput optimization routing problem, we first need to understand the constraint of the flow rates. Let $\boldsymbol{y} = (y_e, e \in E)$ denote the wireless link rate vector, where y_e is the aggregated flow rate along wireless link e. Link rate vector $\boldsymbol{y}$ is said to be schedulable, if there exists a stable schedule that ensures every packet transmission with a bounded delay. Essentially, the constraint of the flow rates is defined by the schedulable region of the link rate vector $\boldsymbol{y}$.

The link rate schedulability problem has been studied in several existing works, which lead to different models [13, 16, 24]. In this paper, we adopt the model in [16], which presents a sufficient condition under which a link scheduling algorithm is given to achieve stability with bounded and fast approximation of an ideal schedule. Based on this model, we define $S(e)$ as a subset of $I(e)$, where each $e' \in I(e)$ that has a length $r(e')$ greater than or equal to $r(e)$ is in $S(e)$. In the following discussion, we refer $S(e)$ as the *adjusted interference set* of e. Based on the results presented in [16], we have the following claims.

Claim 1. (*Sufficient Condition of Schedulability*) The link rate vector $\boldsymbol{y}$ is schedulable *if* the following condition is satisfied:

$$\forall e \in E, y_e + \sum_{e' \in S(e)} y_{e'} \leq b(e) \quad (1)$$

where $b(e)$ bit/sec is the data rate of edge e. For ease of exposition, we assume that $b(e) = 1$ for all $e \in E$ in the following discussion.

3 Fixed Demand Mesh Network Routing

In this paper, we investigate the throughput optimization routing problem for wireless mesh backbone network. The objective of this problem is to maximize the throughput of aggregated flows among local access points and the gateway, while satisfying the fairness constraints. This problem is usually formulated as a maximum concurrent flow problem. In this section, we first study the formulation of throughput optimization routing in wireless mesh backbone network under fixed traffic demand. We then present a fully polynomial time approximation algorithm (FPTAS) for this problem, which finds an ϵ-approximate solution. The problem formulation and algorithm presented in this section serve as the basis of the uncertain demand routing problem discussed in Sec. 4.

Recall that $f \in F$ is the aggregated traffic flow between local access points and the gateway. We use d_f to denote the demand of flow f and $\boldsymbol{d} = (d_f, f \in F)$ to denote the demand vector consisting of all flow demands. Consider the fairness constraint that, for each flow f, its throughput being routed is in proportion to its demand d_f. Our goal is to maximize λ (so called scaling factor) where at least λd_f amount of throughput can be routed for flow f. We assume an infinitesimally divisible flow model where the aggregated traffic flow could be routed over multiple paths and use $\mathcal{P}_f$ to denote the set of unicast paths that could route flow f.

Notation	Definition
$G = (V, E)$	Network
$u \in V$	Node
$g \in V$	Gateway router
$e = (u, v) \in V$	Edge connecting nodes u and v
$f \in F$	Flow, also known as commodity
$\boldsymbol{x} = (x_f, f \in F)$	Aggregated flow rate vector
$\boldsymbol{y} = (y_e, e \in E)$	Wireless link flow rate vector
$\boldsymbol{d} = (d_f, f \in F)$	Flow demands
$\mathcal{P}_f$	Set of unicast paths that could route f
$x_f(P)$	Rate of flow f over path $P \in \mathbf{P}_f$
λ	Scaling factor
S_e	Adjusted interference set of $e \in E$
$A_{eP} = \lvert S_e \cap P \rvert$	Number of wireless links P passes in S_e
μ_e	Price of S_e
$\lambda(\boldsymbol{d}) = \min_{f \in F}\{\frac{x_f}{d_f}\}$	Scaling factor for $\boldsymbol{d}$
$\lambda^*(\boldsymbol{d})$	Optimal value of $\lambda(\boldsymbol{d})$
$\theta = \lambda(\boldsymbol{d})/\lambda^*(\boldsymbol{d})$	Performance ratio
$p(\boldsymbol{d})$	Probability of $\boldsymbol{d}$

Table 1. Notations

Let $x_f(P)$ be the rate of flow f over path $P \in \mathcal{P}_f$. Obviously the aggregated flow rate y_e along edge $e \in E$ is given by $y_e = \sum_{f: P \in \mathcal{P}_f \& e \in P} x_f(P)$, which is the sum of flow rates that are routed through paths P passing edge e. Based on the sufficient condition of schedulability in Claim 1 (Eq.(1)), we have that

$$\sum_{f: P \in \mathcal{P}_f \& e \in P} x_f(P) + \sum_{e' \in S_e} \sum_{f: P \in \mathcal{P}_f \& e' \in P} x_f(P) \leq 1 \quad (2)$$

The throughput optimization routing with fairness constraint is then formulated as the following linear programming (LP) problem:

$$
\begin{aligned}
\mathbf{P}: \quad & \text{maximize} \quad \lambda && (3)\\
& \text{subject to} \quad \sum_{P \in \mathcal{P}_f} x_f(P) \geq \lambda d_f, \forall f \in F && (4)\\
& \sum_{f: P \in \mathcal{P}_f \& e \in P} x_f(P) && (5)\\
& + \sum_{e' \in S_e} \sum_{f: P \in \mathcal{P}_f \& e' \in P} x_f(P) \leq 1, \forall e \in E \\
& \lambda \geq 0, x_f(P) \geq 0, \forall f \in F, \forall P \in \mathcal{P}_f && (6)
\end{aligned}
$$

In this problem, the optimization objective is to maximize λ, such that at least λd_f units of data can be routed for each aggregated flow f with demand d_f. Inequality (4) enforces fairness by requiring that the comparative ratio of traffic routed for different flows satisfies the comparative ratio of their demands. Thus, the absolute value d_f is meaningless, as we can easily tune the value of λ by scaling up/down all demands, while λd_f stays unchanged. Inequality (5) enforces capacity constraint by requiring the traffic aggregation of all flows passing wireless link $e \in E$ satisfy the sufficient condition of schedulability. This problem formulation follows the classical maximum concurrent flow problem, which has also been used in Internet traffic engineering and load balancing routing.

This problem could be solved by a LP-solver such as [2]. To reduce the complexity for practical use, we present a fully polynomial time approximation algorithm (FPTAS) for problem **P**, which finds an ϵ-approximate solution. The key to a fast approximation algorithm lies on the dual of this problem, which is formulated as follows. First we define $A_{eP} = |S_e \cap P|$ as the number of wireless links a path P passes in the adjusted interference set S_e. We assign a price μ_e to each set S_e for $e \in E$. The objective is to minimize aggregated price for all adjusted interference sets. As the constraint, Inequality (8) requires that the price $\sum_{e \in E} A_{eP}\mu_e$ of any path $P \in \mathcal{P}_f$ for flow f must be at least μ_f, the price of flow f. Further, Inequality (9) requires that the weighted flow price μ_f over its demand d_f must be at least 1.

$$
\begin{aligned}
\mathbf{D}: \quad & \text{minimize} \quad \sum_{e \in E} \mu_e && (7)\\
& \text{subject to} \quad \sum_{e \in E} A_{eP}\mu_e \geq \mu_f, \forall f \in F, \forall P \in \mathcal{P}_f && (8)\\
& \sum_{f \in F} \mu_f d_f \geq 1 && (9)
\end{aligned}
$$

Based on the above dual problem **D**, our fast approximation algorithm is presented in Table 2. The algorithm design follows the idea of [10]. To start with, we initialize the price

Algorithm I: Mesh Network Routing Under Fixed Demand

1 $\forall e \in E$, $\mu_e \leftarrow \beta$
2 $x_f(P) \leftarrow 0, \forall P \in \mathcal{P}_f, \forall f \in F$
3 **while** $\sum_{e \in E} \mu_e < 1$
4   **for** $\forall f \in F$ **do**
5     $d'_f \leftarrow d_f$
6   **while** $\sum_{e \in E} \mu_e < 1$ **and** $d'_f > 0$ **do**
7       $P \leftarrow$ lowest priced path in $\mathcal{P}_f$ using μ_e
8       $\delta \leftarrow \min\{d'_f, \min_{e \in P} \frac{1}{A_{eP}}\}$
9       $d'_f \leftarrow d'_f - \delta$
10       $x_f(P) \leftarrow x_f(P) + \delta$
11       $\forall e$ s.t. $A_{eP} \neq 0$, $\mu_e \leftarrow \mu_e(1 + \epsilon\delta A_{eP})$
12   **end while**
13   **end for**
14 **end for**

Table 2. Routing Algorithm Under Fixed Demand

on each adjusted interference set S_e as β (Line 1). We also zero the traffic on all paths $P \in \mathcal{P}_f$ (Line 2). Then for each flow f, we route d_f units of data. We do so by finding the lowest priced path in the path set $\mathcal{P}_f$ (Line 7), then filling traffic to this path by its bottleneck capacity (Lines 8 to 10). Then we update the prices for physical edges appeared in this path based on the function defined in Line 11. We keep filling traffic to flow f in the above fashion until all d_f units are routed. This procedure is repeated until the aggregate price of interference sets S_e for all $e \in E$ weighted by the capacity c exceeds 1 (Line 3).

We make following notes to our algorithm. First, it completes in finite time, which is guaranteed by the asymptotic link price update function defined in Line 11. ϵ here is the step size, which controls the growing speed of the link price. Second, as one might see, the algorithm in fact routes more traffic than the network is able to afford. Therefore, a scaling procedure is needed to scale down the routed traffic so it fits the capacity of each physical link in the network. We formally analyze the properties of our algorithm in the following lemmas and theorem. In the analysis, we denote f^* as the result returned by the algorithm and OPT as the optimal value of **D** as well as **P**. The detailed proofs of the following lemmas and theorem are provided in the Appendix.

Lemma 1: If $OPT \geq 1$, scaling the final flow by $\log_{1+\epsilon} 1/\beta$ yields a feasible primal solution of value $f^* = \frac{t-1}{\log_{1+\epsilon} 1/\beta}$, t being the number of phases the algorithm takes to stop.

Lemma 2: If $OPT \geq 1$, then the final flow scaled by $\log_{1+\epsilon} 1/\beta$ has a value at least $(1 - 3\epsilon)$ times OPT, when $\beta = (|E|/(1-\epsilon))^{-1/\epsilon}$.

Lemma 3: If $OPT \geq 1$ and $\beta = (|E|/(1-\epsilon))^{-1/\epsilon}$, **Al-**

gorithm I terminates after at most $t = 1 + \frac{OPT}{\epsilon}\log_{1+\epsilon}\frac{|E|}{1-\epsilon}$ phases.

These lemmas require that $OPT \geq 1$. The running time of the algorithm also depends on OPT. Thus we need to ensure that OPT is at least one and not too large. Let ζ_f be the maximum traffic value of flow f when all other flows have zero traffic. Let $\zeta = \min_f \frac{\zeta_f}{d_f}$. Since at best all single commodity maximum flows can be routed simultaneously, ζ is an upper bound on f^*. On the other hand, routing $1/|F|$ fraction of each flow of value ζ_f is a feasible solution, which implies that $\zeta/|F|$ is a lower bound on OPT. To ensure that $OPT \geq 1$, we can scale the original demands so that $\zeta/|F|$ is at least one. However, by doing so, OPT might be made as large as $|F|$, which is also undesirable.

To reduce the dependence on the number of phases on OPT, we adopt the following technique. If the algorithm does not stop after $T = \frac{2}{\epsilon}\log_{1+\epsilon}\frac{|E|}{1-\epsilon}$ phases, it means that $OPT > 2$. We then double demands of all commodities, so that OPT is halved and still at least 1. We then continue the algorithm, and double demands again if it does not stop after T phases.

Lemma 4: Given ζ_f for each flow f, the running time of **Algorithm I** is $O(\frac{\log|E|}{\epsilon^2}(2|F|\log|F| + |E|)) \cdot T_{mp}$.

Theorem 1: The total running time of **Algorithm I** is $O(\frac{1}{\epsilon^2}[\log|E|(2|F|\log|F| + |E|) + \log U)]) \cdot T_{mp}$.

4 Uncertain Demand Mesh Network Routing

Now we proceed to investigate the throughput optimization routing problem for wireless mesh backbone network when the aggregated traffic demand is uncertain. We model such uncertain traffic demand of an aggregated flow $f \in F$ using a random variable D_f. We assume that D_f follows the following discrete probability distribution

$$Pr(D_f = d_f^i) = q_f^i \tag{10}$$

where $\mathcal{D}_f = \{d_f^1, d_f^2, ..., d_f^m\}$ is the set of of values for D_f with non-zero probabilities. Let $\boldsymbol{d} = (d_f, d_f \in \mathcal{D}_f, f \in F)$ be a sample traffic demand vector of all flows, $\boldsymbol{D}$ be the corresponding random variable, and $\mathcal{D}$ be the sample space. We assume that the demand from different access points are independent from each other. Thus the distribution of $\boldsymbol{D}$ is given by the joint distribution of these random variables as follows.

$$Pr(\boldsymbol{D} = \boldsymbol{d}) = Pr(D_f = d_f^i, f \in F) = \Pi_{f\in F} q_f^i \tag{11}$$

Let us consider a traffic routing solution $(x_f(P), P \in \mathcal{P}_f, f \in F)$ that satisfies the capacity constraint (Eq. (5)). It is obvious that the λ is a function of $\boldsymbol{d}$:

$$\lambda(\boldsymbol{d}) = \min_{f\in F}\{\frac{x_f}{d_f}\} \tag{12}$$

where $x_f = \sum_{P\in\mathcal{P}_f} x_f(P)$. Further let us consider the optimal routing solution under demand vector $\boldsymbol{d}$. Such a solution could be easily derived based on Algorithm I shown in Table 2. We denote the optimal value of λ as $\lambda^*(\boldsymbol{d})$. We further define the *performance ratio* θ of routing solution $(x_f(P), P \in \mathcal{P}_f, f \in F)$ as follows:

$$\theta = \frac{\lambda(\boldsymbol{d})}{\lambda^*(\boldsymbol{d})}$$

Obviously, the performance ratio is also a random variable under uncertain demand. We denote it as Θ. Θ is a function of random variable $\boldsymbol{D}$. Now we extend the wireless mesh network routing problem to handle such uncertain demand. Our goal is to maximize the expected value of Θ, which is given as follows.

$$E(\Theta) = Pr(\boldsymbol{D} = \boldsymbol{d}) \times \frac{\lambda(\boldsymbol{d})}{\lambda^*(\boldsymbol{d})} \tag{13}$$

We abbreviate $Pr(\boldsymbol{D} = \boldsymbol{d})$ as $p(\boldsymbol{d})$. It is obvious that $\sum_{d\in\mathcal{D}} p(\boldsymbol{d}) = 1$. Formally, we formulate the throughput optimization routing problem with fairness constraints for wireless mesh backbone network under uncertain traffic demand as follows.

$$\mathbf{P_U}: \quad \text{maximize} \quad \sum_{\boldsymbol{d}\in\mathcal{D}} p(\boldsymbol{d})\frac{\lambda(\boldsymbol{d})}{\lambda^*(\boldsymbol{d})} \tag{14}$$

$$\text{subject to} \quad \sum_{P\in\mathcal{P}_f} x_f(P) \geq \lambda(\boldsymbol{d})d_f, \tag{15}$$

$$\forall \boldsymbol{d} \in \mathcal{D}, \forall f \in F$$

$$\sum_{f:P\in\mathcal{P}_f\&e\in P} x_f(P) + \tag{16}$$

$$\sum_{e'\in\hat{I}(e)} \sum_{f:P\in\mathcal{P}_f\&e'\in P} x_f(P) \leq 1,$$

$$\forall e \in E, \forall f \in F, \forall P \in \mathcal{P}_f, x_f(P) \geq 0$$

Similar to problem $\mathbf{P}$, the constraints of $\mathbf{P_U}$ come from the fairness requirement and the wireless mesh network capacity. In particular, Inequality (15) enforces fairness for all demand $\boldsymbol{d} \in \mathcal{D}$, and Inequality (16) enforces capacity constraint as (5) in problem $\mathbf{P}$.

Now we consider the dual problem $\mathbf{D_U}$ of $\mathbf{P_U}$. Similar to $\mathbf{D}$, the objective of $\mathbf{D_U}$ is to minimize the aggregated price for all interference sets. However, in Inequality (19), for each sample demand vector $\boldsymbol{d}$, the aggregated price of all flows weighted by their demand needs to be larger than the ratio of its probability to its optimal value of λ.

$$\mathbf{D_U}: \quad \text{minimize} \quad \sum_{e \in E} \mu_e \tag{17}$$

$$\text{subject to} \quad \sum_{e \in E} A_{eP}\mu_e \geq \mu_f, \quad \forall f \in F, \forall P \in \mathcal{P}_f \tag{18}$$

$$\sum_{f \in F} \mu_f d_f \geq \frac{p(\boldsymbol{d})}{\lambda^*(\boldsymbol{d})}, \forall \boldsymbol{d} \in \mathcal{D} \tag{19}$$

$$\text{where } \boldsymbol{d} = (d_f, f \in F)$$

Algorithm II: Mesh Network Routing Under Uncertain Demand

1 $\forall e \in E, \mu_e \leftarrow \beta$
2 $x_f(P) \leftarrow 0, \forall P \in \mathcal{P}_f, \forall f \in F$
3 **loop**
4 **for** $\forall f \in F$ **do**
5 $\bar{P} \leftarrow$ lowest priced path in $\mathcal{P}_f$ using μ_e
6 $\mu_f \leftarrow \sum_{e \in E} A_{e\bar{P}}\mu_e$
7 **end for**
8 **for** $\forall \boldsymbol{d} \in \mathcal{D}$ **do**
9 $\mu_{\boldsymbol{d}} \leftarrow \sum_{f \in F} \mu_f d_f \frac{\lambda^*(\boldsymbol{d})}{p(\boldsymbol{d})}$
10 **end for**
11 $\mu^{\min} \leftarrow \min_{\boldsymbol{d} \in \mathcal{D}} \mu_{\boldsymbol{d}}$
12 $\boldsymbol{d}^{\min} \leftarrow \arg\min_{\boldsymbol{d} \in \mathcal{D}} \mu^{\min}$
13 **if** $\mu^{\min} \geq 1$
14 **return**
15 **for** $\forall f \in F$ **do**
16 $d'_f \leftarrow d_f^{\min}$
17 **while** $d'_f > 0$ **do**
18 $P \leftarrow$ lowest priced path in $\mathcal{P}_f$ using μ_e
19 $\delta \leftarrow \min\{d'_f, \min_{e \in P} \frac{1}{A_{eP}}\}$
20 $d'_f \leftarrow d'_f - \delta$
21 $x_f(P) \leftarrow x_f(P) + \delta$
22 $\forall e$ s.t. $A_{eP} \neq 0$, $\mu_e \leftarrow \mu_e(1 + \epsilon\delta A_{eP} \times \frac{\lambda^*(\boldsymbol{d}^{\min})}{p(\boldsymbol{d}^{\min})})$
23 **end while**
24 **end for**
25 **end loop**

Table 3. Routing Algorithm Under Uncertain Demand

Now we present an approximation algorithm for $\mathbf{P_U}$ in Table 3. This algorithm (Algorithm II) has the same initialization as the algorithm for problem $\mathbf{P}$ (Algorithm I). Then we march into the iteration, in which we find $\boldsymbol{d}^{\min}$, the demand whose price $\mu^{\min}$ is the minimum among others (Lines 4 to 12). If $\mu^{\min} \geq 1$, then the algorithm stops (Lines 13 and 14), since Inequality (8) and (9) would be satisfied for all demand. Otherwise, we will increase the price of $\boldsymbol{d}^{\min}$ by routing more traffic through its node pairs. This procedure (Lines 16 to 22) is the same as what has been described in the lines 4 to 11 of the previous algorithm. Following the same proving sequence for Algorithm I, we are able to prove the similar properties with Algorithm II, which we illustrate the details in the Appendix.

5 Simulation Study

5.1 Simulation Setup

We evaluate the performance of our algorithms via simulation study. In the simulated wireless mesh network, 30 mesh nodes are randomly deployed over a $800 \times 800m^2$ region, among which 10 nodes are local access points that forward traffic for clients. Node 10 which resides in the center of the deploy region is selected as the gateway router. Each mesh nodes has a transmission range of $250m$. The simulated network topology is shown in Fig. 2

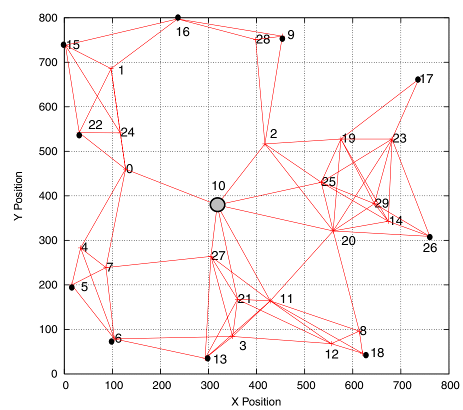
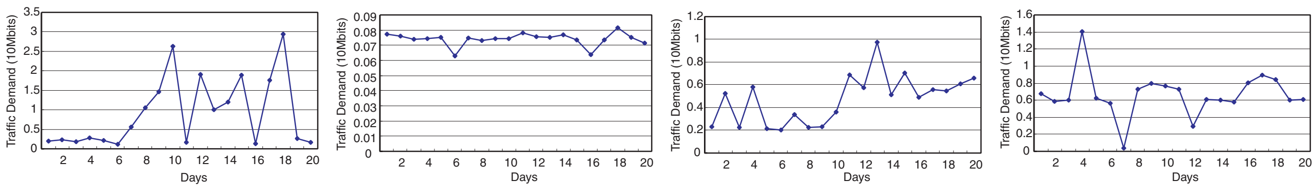


Figure 2. Mesh Network Topology.

5.2 Traffic Demand Generation

To realistically simulate the traffic demand at each local access point, we analyze the traces collected in the campus wireless LAN network. The network traces used in this work are collected in Spring 2002 at Dartmouth College and provided by CRAWDAD [1].

By analyzing the *snmp* log trace at each access point, we are able to derive its incoming and outgoing traffic volume in a 5-minute period. We argue that the local access points of a wireless mesh network serve a similar role as the access points of wireless LAN networks at aggregating and forwarding client traffic. Thus, we select 10 access points from the Dartmouth campus wireless LAN and assign their traffic traces to the local access points in our simulation.



(a) Node 15 (b) Node 16 (c) Node 18 (d) Node 22

Figure 3. Traffic Time Series.

Fig. 3 plots the time series of the traffic volume during one-hour period at the same time of a day (12pm-1pm) from 4 access points for 20 consecutive work days. We remove the weekend days from the traces due to their extreme low traffic volume. From the figure, we observe that the traffic at each access point is highly dynamic and unpredictable due to the insufficient level of aggregation. This observation motivates the need of mesh routing schemes that are aware of the traffic uncertainty.

Based on the one-hour traffic volume data from the traces, we further derive the traffic demand distribution for each access point. Essentially, We round the obtained raw traffic volume into finite collection of traffic demand ranges. We treat each piece of traffic volume data as a sample and derive its probability density function. Fig. 4 plots the probability density function for the 4 corresponding access points in Fig. 3.

5.3 Performance Comparison

We compare the performance of three traffic routing solutions described as follows. In particular, the first two employ the throughput maximization algorithm under fixed demand (Algorithm I), while the last one employs the algorithm under uncertainty demand (Algorithm II).

- *Online Solution*: This solution keeps track of the dynamically changing demand and maximizes the throughput based on the current demand of each access point, meanwhile maintaining the fairness among them. Since the access point demand keeps changing, it has to continuously rerun Algorithm I (Table 2) to adapt to the new demand. This solution yields the optimal routing result at the cost of frequent routing computation and update.

- *Average-Demand-based Solution*: This solution estimates the dynamic traffic demand using the mean value from its probability distribution for each access point. It computes a fixed route based on this average demand vector using Algorithm I. Using only the average demand, this solution does not consider the uncertainty of the traffic demand.

- *Uncertainty-aware Solution*: This solution accommodates the uncertainty in traffic demand by maximizing the throughput for all demand vectors normalized by their occurring probabilities. In particular, it employs Algorithm II presented in Table 3 with the traffic distribution derived from the traces.

We evaluate the above three routing solutions under a series of experiments. For each experiment, the traffic demand of each access point is generated based on their probability distribution. We repeat the experiment for 100 times with 100 randomly generated traffic demand vectors. For each experiment, we derive its scaling factor λ, which is the minimum ratio of throughput and demand among all access points. Fig. 5 plots the sorted values of λ in these 100 experiments. Evidently, the online solution keeps delivering the optimal scaling factor, at the cost of rerouting for each experiment. Comparatively, average-demand solution provides a single set of routes for all demand vectors, but still achieves a scaling factor no worse than 50% of the optimum, except at the last 5 experiments. Uncertainty-aware solution demonstrate the same trend, but continuously outperforms the average-demand solution by 20%. Although there are exceptions in the first 20 experiments when the optimal value of λ is low, the difference is minor.

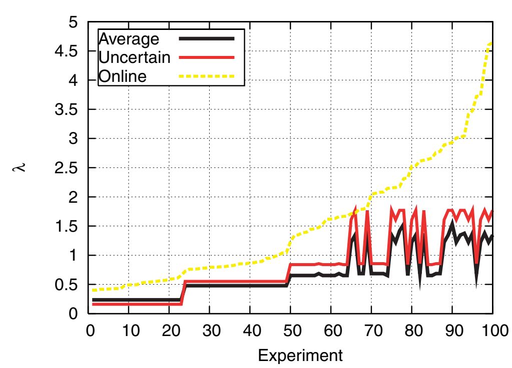
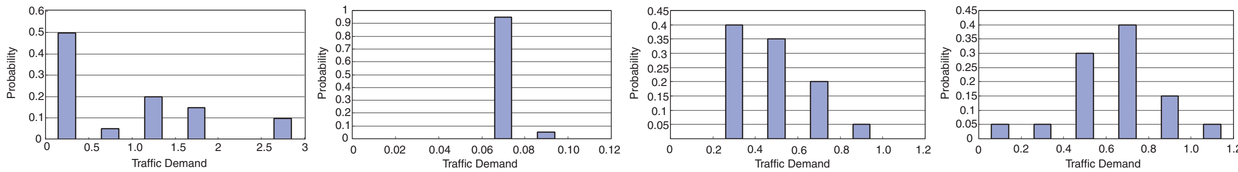


Figure 5. Comparison of Three Algorithms

After evaluating the overall performance of these solutions, we then study them in the granularity of a single experiment. In particular, we are interested to learn the ability



(a) Node 15 (b) Node 16 (c) Node 18 (d) Node 22

Figure 4. Traffic Demand Distribution.

of each algorithm at maintaining fairness among different access points. In Fig. 6, we randomly choose one experiment listed in Fig. 5, and plot the scaling factor of aggregated flows achieved on four local access points (node 15, 16, 18, and 22). Here, the average-demand solution achieves the worst fairness among four nodes, as it gives the highest scaling factor for node 18, and the lowest for 22[1]. On the other hand, online algorithm maintains the best fairness among four nodes, since its result is tuned on-the-fly to the specific demand. Finally, the uncertainty-aware solution trades off well between the former two, by returning the scaling factor which is much higher than average-demand solution (observing node 22), yet only slightly lower than the online solution (observing node 15 and 18).

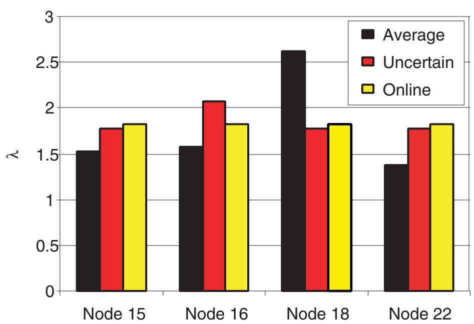


Figure 6. Comparison of Aggregated Flows from Different Access Points

In Fig. 7, we compare the aggregated throughput of all local access points enabled by these three solutions. Obviously, throughout all experiments, the aggregated throughput remains the same for average-demand and uncertainty-aware solutions since they stick to only one set of routes. The latter solution outperforms the former one, as confirmed by our observation in Fig. 5 in terms of scaling factor. The online solution, however, results in unstable aggregate throughput over different experiments. While its maximum value is higher than the other two solutions, the minimum throughput is also well below both of them.

[1]Recall in Fig. 5, the scaling factor achieved at each experiment is the lowest value among all access nodes.

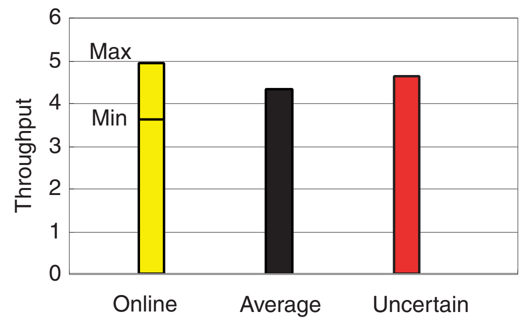


Figure 7. Comparison of Aggregated Throughput

In summary, our simulation results demonstrate that our statistical problem formulation could effectively incorporate the traffic demand uncertainty in routing optimization, and its algorithm outperforms the algorithm which only considers the static traffic demand.

6 Related Work

We evaluate and highlight our original contributions in light of previous related work.

The problem of wireless mesh network routing has been extensively studied in the existing literature. For example, routing algorithms are proposed to improve the throughput for wireless mesh networks via integrating MAC layer information [7], such as expected packet transmission time [8]. Joint solutions for channel allocation and routing are explored in [20] using a centralized algorithm and in [19] in a distributed fashion. These heuristic solutions lack the theoretical foundation to analyze how well the network performs globally (*e.g.,* whether the network resource is fully utilized, whether the flows share the network in a fair fashion) under their routing schemes.

There are also theoretical studies that formulate these network planning decisions into optimization problems. For example, the works of [5], [14] study the optimal solution of joint channel assignment and routing for maximum throughput under a multi-commodity flow problem formulation and solves it via linear programming. The work of

[21] presents bandwidth allocation schemes to achieve maximum throughput and lexicographical max-min fairness respectively. [9] presents a rate limiting scheme to enforce the fairness among different local access points. These results provide valuable analytical insights to the mesh network design under ideal assumptions such as known static traffic input. It is not clear whether they will be unsuitable for practical use under highly dynamic traffic situation. Different from these existing works, our work explicitly incorporates traffic uncertainty into the routing optimization, thus better fits the routing need in the dynamic wireless mesh networks. Distributed algorithms have presented for joint scheduling and routing in [17], and for joint channel assignment, scheduling and routing in [17]. These distributed algorithms only use local information for traffic routing, thus have the potential to accommodate dynamic traffic. However, their crucial properties, such as convergence speed and messaging overhead, are yet to be evaluated under realistic traffic conditions.

Trace analysis has been used to study the behavior of wireless networks in many recent works. To name a few, the works of [15, 12] analyze a campus-wide wireless network traffic and its changes in two years. [18] statistically characterizes both static flows and roaming flows in a large campus wireless network. In [22], the flow level traffic pattern is used to design a scheduling algorithm for end-host multi-homing. Different from these existing works, which focus on either user behavior, network flow or link performances, we provide a framework that integrates traffic uncertainty model with its performance optimization.

Our work is also related to dynamic traffic engineering [23] in Internet and oblivious routing [6], which also consider the impact of demand uncertainty in make routing decisions. The major difference between our work and these existing works lies in the different network models of wireless mesh network and Internet and the different problem formulations. In particular, traffic engineering tries to minimize the congestion (utilization) of the wired links of a network. In multihop wireless network, wireless link utilization can not be used to characterize the network performance due to the location dependent contention in the vicinity area. The objective of our research is to maximize the ratio between flow throughput and its demand, subject to the schedulability and fairness constraints.

7 Conclusion

This paper studies the throughput optimization routing problem for wireless mesh networks. Different from existing works which assume fixed traffic demand known a priori, this work considers the traffic demand uncertainty. It models such uncertain demand with random variables, extends the classical maximum concurrent flow problem with statistical demand input, and derives approximation algorithms for uncertainty-aware traffic routing. Simulation study is conducted based on the traffic demands from the real wireless network traces. The results show that our problem formulation and algorithm could effectively incorporate the traffic demand uncertainty.

References

[1] A community resource for archiving wireless data at dartmouth. http://crawdad.cs.dartmouth.edu/.

[2] Ilog cplex mathematical programming optimizers. http://www.ilog.com/products/cplex.

[3] Mit roofnet. http://www.pdos.lcs.mit.edu/roofnet/.

[4] Seattle wireless. http://www.seattlewireless.net.

[5] M. Alicherry, R. Bhatia, and L. Li. Joint channel assignment and routing for throughput optimization in multi-radio wireless mesh networks. In *Proc. of ACM MobiCom*, 2005.

[6] Y. Azar, E. Cohen, A. Fiat, H. Kaplan, and H. Racke. Optimal oblivious routing in polynomial time. *J. Comput. Syst. Sci.*, 69(3), 2004.

[7] S. Biswas and R. Morris. Exor: opportunistic multi-hop routing for wireless networks. In *Proc. of ACM SIGCOMM*, pages 133–144, 2005.

[8] R. Draves, J. Padhye, and B. Zill. Routing in multi-radio, multi-hop wireless mesh networks. In *Proc. of ACM Mobicom*, pages 114–128, 2004.

[9] V. Gambiroza, B. Sadeghi, and E. W. Knightly. End-to-end performance and fairness in multihop wireless backhaul networks. In *Proc. of ACM MobiCom*, 2004.

[10] N. Garg and J. Konemann. Faster and simpler algorithms for multicommodity flow and other fractional packing problems. In *Proc. of IEEE FOCS*, 1998.

[11] P. Gupta and P. Kumar. The capacity of wireless networks. *IEEE Trans. on Information Theory*, pages 388–404, 2000.

[12] T. Henderson, D. Kotz, and I. Abyzov. The changing usage of a mature campus-wide wireless network. In *Proc. of ACM MobiCom*, pages 187–201, 2004.

[13] K. Jain, J. Padhye, V. Padmanabhan, and L. Qiu. Impact on interference on multi-hop wireless network performance. In *Proc. of Mobicom*, September 2003.

[14] M. Kodialam and T. Nandagopal. Characterizing the capacity region in multi-radio multi-channel wireless mesh networks. In *Proc. of IEEE WiMesh*, 2005.

[15] D. Kotz and K. Essien. Analysis of a campus-wide wireless network. In *ACM MobiCom*, pages 107–118, 2002.

[16] V. S. A. Kumar, M. V. Marathe, S. Parthasarathy, and A. Srinivasan. Algorithmic aspects of capacity in wireless networks. In *Proc. of ACM SIGMETRICS*, pages 133–144, 2005.

[17] X. Lin and S. Rasool. A distributed joint channel-assignment, scheduling and routing algorithm for multi-channel ad hoc wireless networks. In *INFOCOM*, 2007.

[18] X. G. Meng, S. H. Y. Wong, Y. Yuan, and S. Lu. Characterizing flows in large wireless data networks. In *Proc. of ACM MobiCom*, 2004.

[19] A. Raniwala and T. Chiueh. Architecture and algorithms for an ieee 802.11-based multi-channel wireless mesh network. In *Proc. of IEEE INFOCOM*, 2005.

[20] A. Raniwala, K. Gopalan, and T. Chiueh. Centralized channel assignment and routing algorithms for multi-channel wireless mesh networks. *Mobile Computing and Communications Review*, 8(2):50–65, 2004.

[21] J. Tang, G. Xue, and W. Zhang. Maximum throughput and fair bandwidth allocation in multi-channel wireless mesh networks. In *Proc. of IEEE INFOCOM*, 2006.

[22] N. Thompson, G. He, and H. Luo. Flow scheduling for end-host multihoming. In *Proceedings of IEEE INFOCOM*, 2006.

[23] H. Wang, H. Xie, L. Qiu, Y. R. Yang, Y. Zhang, and A. Greenberg. Cope: traffic engineering in dynamic networks. In *Proc. of ACM SIGCOMM*, 2006.

[24] Y. Xue, B. Li, and K. Nahrstedt. Optimal resource allocation in wireless ad hoc networks: A price-based approach. *IEEE Transactions on Mobile Computing*, 5(4):347–364, April 2006.

8 Appendix

We show the proofs to the lemmas and theorem in the paper. Here, f^* is the result returned by the algorithm. OPT is the optimal value of $\mathbf{D}$ as well as $\mathbf{P}$.

8.1 Proof for Lemma 1

Proof:

We first make the following denotations. Regarding a set of price assignments μ_e for S_e ($e \in E$), the objective function of $\mathbf{D}$ is $L^{\mu_e} \triangleq \sum_{e\in E} \mu_e$. Let $P^{\mu_e}(f)$ be the minimum path of the flow $f \in F$ using μ_e. $\mu(P^{\mu_e}(f)) \triangleq \sum_{e\in E} A_{eP^{\mu_e}(f)}\mu_e$ is the aggregated price of $P^{\mu_e}(f)$. Each phase contains $|F|$ iterations, where traffic for each flow in F is routed according to its demand. In each iteration, the price of an interference set is updated. We use $\mu_e^{(i)(j)}$ to denote the price of S_e for $e \in E$ after the jth iteration of the ith phase. Regarding $\mu_e^{(i)(j)}$, we simplify the notation $L^{\mu_e^{(i)(j)}}$ into $L^{(i)(j)}$, $P^{\mu_e^{(i)(j)}}$ into $P^{(i)(j)}$, and $\mu(P^{\mu_e^{(i)(j)}})$ into $\mu(P^{(i)(j)})$. Based on the price update function (Line 11 in Tab. 2), we have

$$\begin{aligned} & L^{(i)(j)} \\ = \; & \sum_{e\in E} \mu_e^{(i)(j-1)} + \epsilon \sum_{e\in P^{(i)(j-1)}} A_{eP^{(i)(j-1)}} \mu_e^{(i)(j-1)} d(f_j) \\ = \; & L^{(i)(j-1)} + d(f_j)\mu(P^{(i)(j-1)}) \end{aligned}$$

The price assignment at the start of the $(i+1)$th phase are the same as that at the end of the ith phase, i.e., $\mu_e^{(i+1)(0)} = \mu_e^{(i)(|F|)}$. The price of any interference set S_e is initialized as $\mu_e^{(1)(0)} = \mu_e^{(0)(|F|)} = \beta$. Hence,

$$L^{(i)(|F|)} \le L^{(i)(0)} + \epsilon \sum_{j=1}^{|F|} d(f_j)\mu(P^{(i)(|F|)})$$

since the edge lengths are monotonically increasing.

Let us define $\mu^{(i)(|F|)} = \sum_{j=1}^{|F|} d(f_j)\mu(P^{(i)(|F|)})$. Then the objective of $\mathbf{D}$ is to minimize $L^{(i)(|F|)}$, subject to the constraint that $\mu^{(i)(|F|)} \ge 1$. This constraint can be easily satisfied if we scale the length of all inference sets by $1/\mu^{(i)(|F|)}$. So $\mathbf{D}$ is equivalent to finding a set of inference set lengths, such that $\frac{L^{(i)(|F|)}}{\mu^{(i)(|F|)}}$ is minimized. Thus the optimal value of $\mathbf{D}$ is $OPT \triangleq \min_{\mu^{(i)(|F|)}} \frac{L^{(i)(|F|)}}{\mu^{(i)(|F|)}}$.

Since $\frac{L^{(i)(|F|)}}{\mu^{(i)(|F|)}} \ge OPT$, we have

$$L^{(i)(|F|)} \le \frac{\beta|E|}{1-\epsilon} e^{\frac{\epsilon(i-1)}{OPT(1-\epsilon)}}$$

Since $L^{(0)(|F|)} = \beta|E|$, we have

$$\begin{aligned} L^{(i)(|F|)} &\le \frac{\beta|E|}{(1-\epsilon/OPT)^i} \\ &= \frac{\beta|E|}{(1-\epsilon/OPT)}(1+\frac{\epsilon}{OPT-\epsilon})^{i-1} \\ &\le \frac{\beta|E|}{(1-\epsilon/OPT)} e^{\frac{\epsilon(i-1)}{OPT-\epsilon}} \\ &\le \frac{\beta|E|}{1-\epsilon} e^{\frac{\epsilon(i-1)}{OPT(1-\epsilon)}} \end{aligned}$$

where the last inequality assumes that $OPT \ge 1$. The algorithm stops at the first phase t for which $L^{(t)(|F|)} \ge 1$. Therefore,

$$1 \le L^{(t)(|F|)} \le \frac{\beta|E|}{1-\epsilon} e^{\frac{\epsilon(t-1)}{OPT(1-\epsilon)}}$$

which implies

$$\frac{OPT}{t-1} \le \frac{\epsilon}{(1-\epsilon)\ln\frac{1-\epsilon}{\beta|E|}} \tag{20}$$

Now consider an interference set S_e. For every 1 units of flow routed through S_e, we increase its price by at least a factor $(1+\epsilon)$. Initially, its length is β and after $t-1$ phases, since $L^{(t)(|F|)} < 1$, the price of S_e satisfies $\mu_e^{(t-1)(|F|)} < 1$. Therefore the total amount of flow through S_e in the first $t-1$ phases is strictly less than $\log_{1+\epsilon}\frac{1}{\beta} = \log_{1+\epsilon} 1/\beta$ times its capacity. Thus, scaling the flow by $\log_{1+\epsilon} 1/\beta$ will yield a feasible solution. Since in each phase, d_f units of data are routed for each flow, we have $f^* = \frac{t-1}{\log_{1+\epsilon} 1/\beta}$.
□

8.2 Proof for Lemma 2

Proof:

By **Lemma 1**, scaling the final flow by $\log_{1+\epsilon} 1/\beta$ yields a feasible solution. Therefore,

$$\frac{OPT}{f^*} < \log_{1+\epsilon} 1/\beta \tag{21}$$

Substituting the bound on $OPT/(t-1)$ from In Equality (20), we get

$$\frac{OPT}{f^*} < \frac{\epsilon \log_{1+\epsilon} 1/\beta}{(1-\epsilon)\ln\frac{1-\epsilon}{\beta|E|}} = \frac{\epsilon}{(1-\epsilon)\ln(1+\epsilon)} \frac{\ln 1/\beta}{\ln\frac{1-\epsilon}{\beta|E|}}$$

When $\beta = (|E|/(1-\epsilon))^{-1/\epsilon}$, the above in Equality becomes

$$\begin{aligned} \frac{OPT}{f^*} &< \frac{\epsilon}{(1-\epsilon)^2\ln(1+\epsilon)} \\ &\leq \frac{\epsilon}{(1-\epsilon)^2(\epsilon-\epsilon^2/2)} \frac{1}{\leq (1-\epsilon)^3} \\ &\leq (1-3\epsilon) \end{aligned}$$

□

8.3 Proof for **Lemma 3**

Proof: From In Equality (21) and weak-duality, we have

$$1 \leq \frac{OPT}{f^*} < \log_{1+\epsilon} 1/\beta$$

Hence, the number of phases t is strictly less than $1 + OPT\log_{1+\epsilon} 1/\beta$. If $\beta = (|E|/(1-\epsilon))^{-1/\epsilon}$, then $t \leq 1 + \frac{OPT}{\epsilon}\log_{1+\epsilon}\frac{|E|}{1-\epsilon}$ □

8.4 Proof for **Lemma 4**

Proof: The above demand-doubling procedure is repeated for at most $\log|F|$ times. Thus, the total number of phases is at most $T\log k$. Since each phase contains k iterations, the algorithm runs for at most $kT\log k$ iterations.

Now we observe how many steps are within each iteration. For each step except for the last step in an iteration, the algorithm increases the length of some edge (the bottleneck edge on t) by $1+\epsilon$. μ_e has initial value β and value at most 1 before the final step of the algorithm. Otherwise, the stop criterion of the algorithm, $\sum_{e\in E}\mu_e \geq 1$, would have been reached. This means that the length of an edge can be updated in at most $\log_{1+\epsilon}\frac{1}{\beta} = \frac{1}{\epsilon}\log_{1+\epsilon}\frac{|E|}{1-\epsilon}$ steps. Therefore, the algorithm contains at most $\frac{|E|}{\epsilon}\log_{1+\epsilon}\frac{|E|}{1-\epsilon} \leq \frac{|E|}{\epsilon^2}\log\frac{|E|}{1-\epsilon}$ such "normal" steps, and $kT\log k \leq \frac{2k\log k}{\epsilon^2}\log\frac{|E|}{1-\epsilon}$ "final" steps. Each step contains a minimum overlay spanning tree operation. □

8.5 Proof for **Theorem 1**

Proof: Computing ζ_i corresponds to the maximum flow problem, where f_i is the only commodity. The running time of getting ζ_i is $O(\frac{|E|}{\epsilon^2}(\log U)) \cdot T_{mp}$, where U is the length of the longest unicast route, and T_{mp} denotes the running time to find the minimum path. Such an operation has to be repeated for each flow. Also from the result of **Lemma 4**, we can obtain the total running time as described by the theorem. □

8.6 Proof for Algorithm under Uncertain Demand

The proof for the algorithm under uncertain demand follows the same sequence as the proof for the algorithm under fixed demand, with minor modification. We start with **Lemma 1**. Each phase of the algorithm contains $|F|$ iterations, where traffic for each flow in F is routed according to its demand. We reuse the same denotations defined in the original proof to **Lemma 1**. We further introduce $\boldsymbol{d}^{(i)}$ as the demand vector chosen at the ith phase.

Based on the price update function (Line 11 in Tab. 2), we have

$$\begin{aligned} & L^{(i)(j)} \\ = \; & L^{(i)(j-1)} + d(f_j)\mu(P^{(i)(j-1)})\frac{\lambda^*(\boldsymbol{d}^{(i)})}{p(\boldsymbol{d}^{(i)})} \end{aligned}$$

The price assignment at the start of the $(i+1)$th phase are the same as that at the end of the ith phase, i.e., $\mu_e^{(i+1)(0)} = \mu_e^{(i)(|F|)}$. The price of any interference set S_e is initialized as $\mu_e^{(1)(0)} = \mu_e^{(0)(|F|)} = \beta/c$. Hence,

$$\begin{aligned} L^{(i)(|F|)} &= L^{(i)(0)} + \epsilon\sum_{j=1}^{|F|} d(f_j)\mu(P^{(i)(j-1)})\frac{\lambda^*(\boldsymbol{d}^{(i)})}{p(\boldsymbol{d}^{(i)})} \\ &\leq L^{(i)(0)} + \epsilon\sum_{j=1}^{|F|} d(f_j)\mu(P^{(i)(|F|)})\frac{\lambda^*(\boldsymbol{d}^{(i)})}{p(\boldsymbol{d}^{(i)})} \end{aligned}$$

since the edge lengths are monotonically increasing.

Let us define $\mu^{(i)(|F|)} = \sum_{j=1}^{|F|} d(f_j)\mu(P^{(i)(|F|)})\frac{\lambda^*(\boldsymbol{d}^{(i)})}{p(\boldsymbol{d}^{(i)})}$. Then the objective of $\mathbf{D}$ is to minimize $L^{(i)(|F|)}$, subject to the constraint that $\mu^{(i)(|F|)} \geq 1$, i.e., $\frac{L^{(i)(|F|)}}{\mu^{(i)(|F|)}} \geq OPT$.

The rest of the proof follows the same as the original proof to **Lemma 1**. The proofs to **Lemma 2**, **3**, **4**, and **Theorem 1** remain the same.