Optimal Routing for Wireless Mesh Networks With Dynamic Traffic Demand

Liang Dai, Yuan Xue, Bin Chang, Yanchuan Cao, Yi Cui
Department of Electrical Engineering and Computer Science
Vanderbilt University
{*liang.dai, yuan.xue, bin.chang, yanchuan.cao, 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. Traffic routing plays a critical role in determining the performance of a wireless mesh network. To investigate the best routing solution, existing work proposes to formulate the mesh network routing problem as an optimization problem. In this problem formulation, traffic demand is usually implicitly assumed as static and known a priori. Contradictorily, recent studies of wireless network traces show that the traffic demand, even being aggregated at access points, is highly dynamic and hard to estimate. Thus, in order to apply the optimization-based routing solution into practice, one must take into account the dynamic and unpredictable nature of wireless traffic demand.

This paper presents an integrated framework for wireless mesh network routing under dynamic traffic demand. This framework consists of two important components: traffic estimation and routing optimization. By studying the traces collected at wireless access points, we first present a traffic estimation method which predicts future traffic demand based on its historical data using time-series analysis. This method provides not only the mean value of the future traffic demand estimation but also its statistical distribution. We further investigate the optimal routing strategies for wireless mesh network which take these two forms of traffic demand estimations as inputs. The goal is to balance the traffic load so that minimum congestion will be incurred. This routing objective could be transformed into the throughput optimization problem where the throughput of aggregated flows is maximized subject to fairness constraints that are weighted by the traffic demands. Based on linear programming, we present two routing algorithms which consider the mean value and the statistical distribution of the predicted traffic demands, respectively. The trace-driven simulation study demonstrates that our integrated traffic estimation and routing optimization framework can effectively incorporate the traffic dynamics in mesh network routing.

I. Introduction

Wireless mesh networks (*e.g.*, [1], [2]) have attracted increasing attention and deployment as a high-performance and low-cost solution to last-mile broadband Internet access. In a wireless mesh network, local access points and stationary wireless mesh routers communicate with each other and form a backbone structure which forwards the traffic between mobile clients and the Internet.

Traffic routing plays a critical role in determining the performance of a wireless mesh network. Thus it attracts extensive research recently. The proposed approaches usually fall into two ends of the spectrum. On one end of the spectrum are the heuristic routing algorithms (*e.g.*, [3], [4], [5]). Although many of them are adaptive to the dynamic environments of wireless networks, these algorithms lack the theoretical foundation to analyze how well the network performs globally (*e.g.*, whether the traffic shares the network in a fair fashion).

On the other end of the spectrum, there are theoretical studies that formulate mesh network routing as optimization problems (*e.g.*, [6], [7]). The routing algorithms derived from these optimization formulations can usually claim analytical properties such as resource utilization optimality and throughput fairness. In these optimization frameworks, traffic demand is usually implicitly assumed as static and known a priori. Contradictorily, recent studies of wireless network traces [8] show that the traffic demand, even being aggregated at access points, is highly dynamic and hard to estimate. Such observations have significantly challenged the practicability of the existing optimization-based routing solutions in wireless mesh networks.

To address this challenge, this paper investigates the optimal mesh network routing framework which takes into account the dynamic nature of wireless traffic demand. To incorporate the traffic dynamics, the following two components must be seamlessly integrated into this framework.

- *Traffic demand estimation* which derives the traffic model of a wireless mesh network. The model should be dependable at predicting the mean demand at long term, yet agile at containing often uncertain dynamics at short term.
- *Routing optimization* which distributes the traffic along different routes so that minimum congestion will be incurred even under dynamic traffic. The routing strategy should effectively take into account the traffic demand estimation results.

By studying the traces collected at Dartmouth College campus wireless network [9], this paper first presents a traffic prediction method based on time-series analysis. This method derives future traffic demand based on its historical data. The mean value of the predicted demand, together with its prediction error distribution, are used in establishing a statistical model for the traffic demand at a local access point.

This paper further identifies an optimization framework which integrates the demand prediction into traffic routing so that minimum congestion will be incurred. This routing objective could be transformed into the throughput optimization problem where the throughput of aggregated flows is maximized subject to fairness constraints that are weighted by the traffic demands. In particular, two forms of traffic demands are considered as the inputs for routing optimization, namely the *mean value* of the demand prediction and its *statistical distribution*. We present two routing algorithms for each form of the traffic demand estimation respectively. For the first case, based on the classical maximum concurrent flow problem, we formulate optimal mesh network routing as a linear programming problem to maximize, among all flows, the minimum scaling factor of throughput to fixed-value demand (λ) and present a fast $(1-\epsilon)$-approximation algorithm (*i.e.* fixed-demand mesh network routing (**FMR**) algorithm) which could accept the mean value of the demand prediction as the input. For the second case, in order to incorporate the statistical distribution of the demand estimation into the problem formulation, we characterize the traffic demand using a random variable. Now the scaling factor λ under a given routing solution is also a random variable. The throughput optimization problem is then extended to a stochastic optimization problem where the expected value of the scaling factor λ is maximized. Finally, based on the design of **FMR** algorithm, a $(1-\epsilon)$-approximation algorithm (uncertain-demand mesh network routing (**UMR**)) is presented for optimal mesh network routing under uncertain demand.

To evaluate the performance of our algorithms under realistic wireless networking environment, we conduct trace-driven simulation study. In particular, we derive the traffic demand for the local access points of our simulated wireless mesh network based on the traffic traces collected at Dartmouth College campus wireless networks. Our simulation results demonstrate that our integrated traffic estimation and optimal routing framework could effectively incorporate the traffic dynamics into the routing optimization of wireless mesh networks.

The original contributions of this paper are two-fold. Practically, the integration of traffic estimation and routing optimization effectively improves the routing performance of wireless mesh networks under dynamic and uncertain traffic. The full-fledged simulation study based on real wireless network traffic traces provides convincing validation of the practicability of our solution. Theoretically, upon the classical linear optimization algorithm which only accepts the fixed-value demands as inputs, we extend it into a stochastic optimization solution capable of serving uncertain demands that are modelled by their statistical distributions.

The remainder of this paper is organized as follows. Sec. II presents the network and traffic demand model. Sec. III describes the traffic prediction method. Sec. IV formulates the mesh network routing problem under fixed-value traffic demand and presents a fast approximation algorithm (**FMR**). Sec. V extends the routing problem to handle uncertain traffic demand and presents the **UMR** routing algorithm. We show simulation results in Sec. VI, present related work in Sec. VII and finally conclude the paper in Sec. VIII.

II. Model

A. Network Model

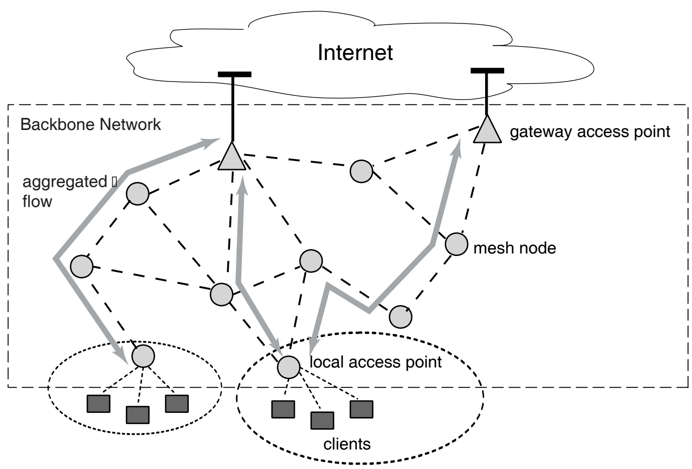


Fig. 1. Illustration of Wireless Mesh Network

We consider a multi-hop wireless mesh network as illustrated in Fig. 1. In this network, local access points aggregate and forward the traffic from the mobile clients that are associated with them. They communicate with each other, also with the stationary wireless routers to form a multi-hop wireless backbone network. This wireless mesh backbone network forwards the user traffic to the gateway access points which are connected to the Internet. In the following discussion, local access point, gateway access point and mesh router are collectively called mesh nodes. We formally 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, $W \subset V$ are the gateway access points that connect to the Internet.

In a wireless network, packet transmissions and interferences are location-dependent. In this paper, we consider the *protocol model* [10]. 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. 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 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 $r(u, v) \leq R_T$; (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. Formally, a edge $e = (u, v) \in E$ is formed if and only if $r(u, v) \leq R_T$. We also use $r(e)$ to represent the length of edge e. Two edges e, e' interfere with each other, if they can not transmit simultaneously based on the protocol model. We use I_e to denote the set of edges which interfere with edge e.

B. Traffic Model

This paper investigates the optimal routing strategy for wireless mesh *backbone* network. Thus it only considers the aggregated traffic among the mesh nodes. In particular, we regard the gateway access points as the sources of all incoming traffic and the destinations 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. Since the traffic of a local access point can be routed to the Internet via any gateway, we introduce a virtual gateway w^* into the network, *i.e.*, $V' = V \cup \{w^*\}$. This virtual gateway access point is connected to each gateway access point with a virtual edge $e' = (w^*, w), w \in W$. The virtual edge could be regarded as a wireline link with unlimited capacity which does not interfere with any of the wireless transmission. The new network graph is represented as $G' = (V', E')$, where $E' = E \cup \{e' = (w^*, w), w \in W\}$.

For simplicity, we call the aggregated traffic from a local access point as a *flow* and denote it as $f \in F$, where F is the set of all aggregated flows. All the flows f will take w^* as either its source or destination. It is worth noting that although we consider only the aggregated traffic between gateway access points and local access points in this paper, our problem formulations and algorithms could be easily extended to handle 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.

C. Schedulability

To study the optimal 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 $\boldsymbol{y}$. For ease of exposition, we assume that the wireless link data rate, which is maximum data that can be carried in unit time, is the same for all $e \in E$ and is denoted as c. c is also referred to as *channel capacity* in the following discussions.

The link rate schedulability problem has been studied in several existing works, which lead to different models [11], [12], [13]. In this paper, we adopt the model in [12], 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 I'_e as a subset of I_e where each $e' \in I'_e$ has a length $r(e')$ greater than or equal to $r(e)$. We further define $S_e = \{e\} \cup I'_e$ as the *adjusted interference set* of e. Based on the scheduling algorithm and its properties presented in [12], we have the following claim.

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

$$\forall e \in E, \sum_{e' \in S_e} y_{e'} \leq c \tag{1}$$

The notations used in this paper are summarized in Table I.

Notation	Definition
$G = (V, E)$	Network
$G' = (V', E')$	Network with virtual gateway/links
$u \in V$	Node
$e = (u, v) \in E$	Edge connecting nodes u and v
$f \in F$	Aggregated flow
$\boldsymbol{x} = (x_f, f \in F)$	Aggregated flow rate vector
$\boldsymbol{y} = (y_e, e \in E)$	Wireless link rate vector
$\boldsymbol{d} = (d_f, f \in F)$	Flow traffic demands
$p(\boldsymbol{d})$	Probability of $\boldsymbol{d}$
$\mathcal{P}_f$	Set of paths that can route f
$x_f(P)$	Rate of flow f over path $P \in \mathcal{P}_f$
S_e	Adjusted interference set of $e \in E$
$A_{eP} = \|S_e \cap P\|$	Number of wireless links P passes in S_e
$x(t)$	Raw traffic series
$z(t)$	Adjusted traffic series
$\bar{x}(t)$	Average traffic series
$\hat{x}(t)$	Predicted traffic series
$\hat{z}(t)$	Predicted adjusted traffic series
$\epsilon_x(t), \epsilon_z(t)$	Prediction error
$\lambda = \min_{f \in F}\{\frac{x_f}{d_f}\}$	Scaling factor
$\theta = \max_{e \in E}\{\frac{\sum_{e' \in S_e} y'_e}{c}\}$	Congestion, maximum adjusted independent set utilization
μ_e	Price of S_e

TABLE I
NOTATIONS

III. Traffic Estimation

In this section, we study the dynamic behavior of aggregated traffic at local access points. Our goal is to (1) develop a reliable estimation method that is able to predict the aggregated traffic demand of an access point based on its historical data, and (2) develop a statistical model to characterize the prediction results. The estimated traffic demand will serve as the input of mesh network routing algorithms which will be presented in Sec. IV and Sec. V.

In order to develop such a traffic demand model, we study the traces collected at the campus wireless LAN network of Dartmouth College in Spring 2002 [9]. By analyzing the *snmp* log from each access point, we derive the dynamic behavior of the aggregated traffic demand. We argue that the access points of a wireless LAN serve a similar role and thus exhibit similar behavior as the local access points of a wireless mesh network. Thus this trace set is the best one that resembles the traffic condition of a wireless mesh network, given the present availability of wireless network traces. Though the traffic conditions may vary from a wireless LAN to a wireless mesh network, and among mesh networks of different usages and scales, we believe that the traffic modelling and prediction approach based on time-series analysis could be general enough to handle traffic conditions from mesh networks of different scales (*e.g.*, metropolitan, campus, etc).

To illustrate our analysis procedure, we choose one of the access points (ResBldg97AP3) as an example. The time series of its incoming traffic is plotted in Fig. 2. From the figure, we can easily observe that (1) the traffic demand is non-stationary over large time scales due to the diurnal and weekly working cycles; (2) compared with the traffic behavior in the backbone Internet [14], the traffic at an access point is significantly bursty due to the insufficient level of multiplexing. The above observations are consistent with the findings in [8].

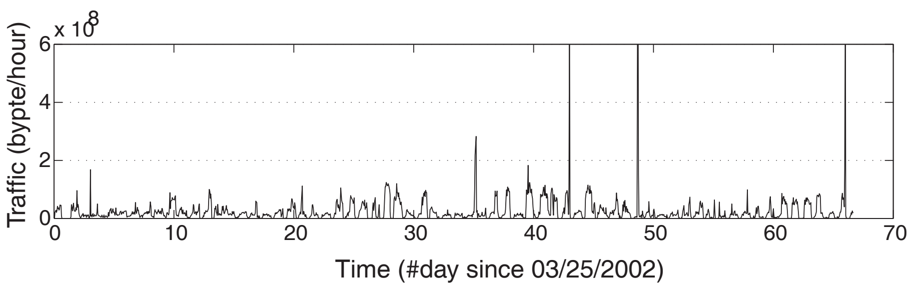


Fig. 2. Incoming Traffic Time Series of ResBldg97AP3 (March 25, 12am, 2002 - June 9, 11pm, 2002 EST).

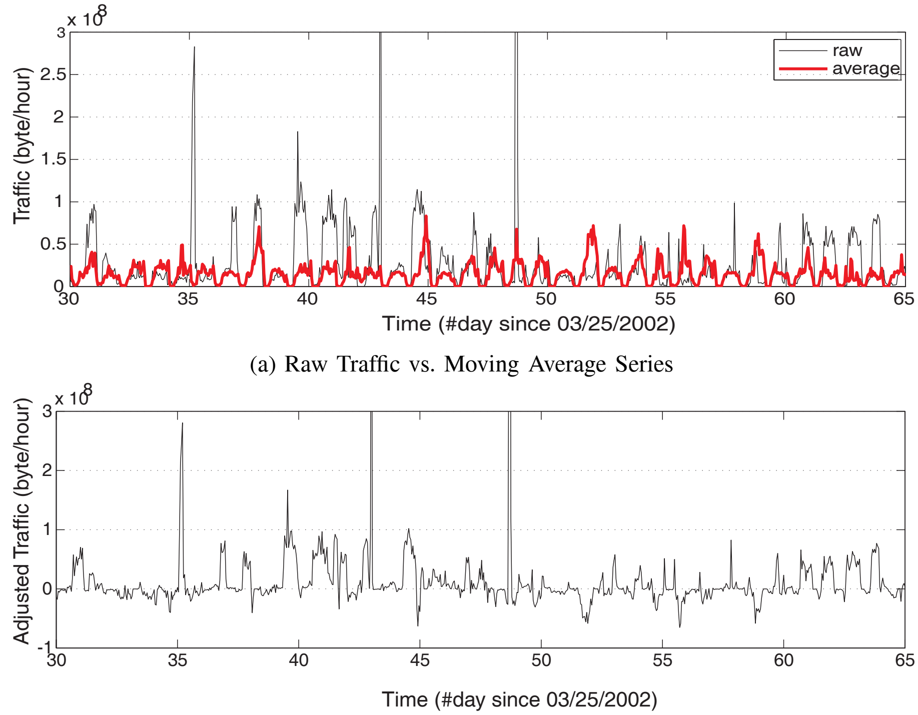
The first step of our analysis is to identify and remove the daily and weekly cyclic patterns in the time series. This requires us to calculate the weekly/daily cyclic average. Formally, let us denote $x(t)$ as the *raw traffic series*. We estimate the moving average of this series based on the same time of the same day of the week, *i.e.*,

$$\bar{x}(t) = \sum_{i=1}^{W} x(t - 24 \times 7 \times i)/W \tag{2}$$

where W is the size of moving window. To eliminate the effect of bursty traffic, we also filter out the spike traffic during the above averaging procedure. Fig. 3(a) plots the raw traffic as well as its moving average with $W = 5$. By removing the cyclic effect from the raw data, we derive the *adjusted traffic series* $z(t)$ as follows.

$$z(t) = x(t) - \bar{x}(t) \tag{3}$$

The adjusted series of the one shown in Fig. 3(a) is given in Fig. 3(b). This adjusted traffic exhibits short-term



(a) Raw Traffic vs. Moving Average Series

(b) Adjusted Traffic Series

Fig. 3. Traffic Series in 5 weeks

(a few hours) traffic correlations. We model the adjusted traffic series with an autoregressive process as follows[1].

$$z(t) = \beta_1 z(t-1) + \beta_2 z(t-2) + ... + \beta_K z(t-K) + \epsilon \tag{4}$$

where K is the process order. To apply this model for prediction, we estimate the parameters of this process. Given N observations $z_1, z_2, ..., z_N$, the parameters β_1, ..., β_K are estimated via least squares by minimizing:

$$\sum_{t=K+1}^{N} \left[z(t) - \beta_1 z(t-1) ... - \beta_K z(t-K) \right]^2 \tag{5}$$

Based on these parameters, we further derive the adjusted traffic prediction $\hat{z}(t)$ as follows:

$$\hat{z}(t) = \beta_1 z(t-1) + \beta_2 z(t-2) + ... + \beta_K z(t-K) \tag{6}$$

Fig. 4 illustrates the estimation results for the adjusted traffic series in Fig. 3(b), where $K = 2$, $\beta_1 = 0.531$, $\beta_2 = 0.469$. The figure plots the predicted series for the adjusted traffic as well as its raw data. In this figure, the number of observations used for parameter estimation is $N = 60$. The fitted traffic series is also plotted for the interval $[720, 779]$ for the purpose of comparison.

We now consider the errors involved in this prediction process. In particular, we define the adjusted traffic prediction error as follows.

[1]Ideally, $z(t)$ should have zero mean. In some cases, $z(t)$ has a small mean value which needs to be removed before fitting an autoregressive process.

$$\epsilon_z(t) = z(t) - \hat{z}(t) \tag{7}$$

Based on this definition, Fig. 5(a) plots the cumulative distribution function of the prediction error of the adjusted traffic series shown in Fig. 4. It is obvious that the error distribution fits the normal distribution with a mean close to zero.

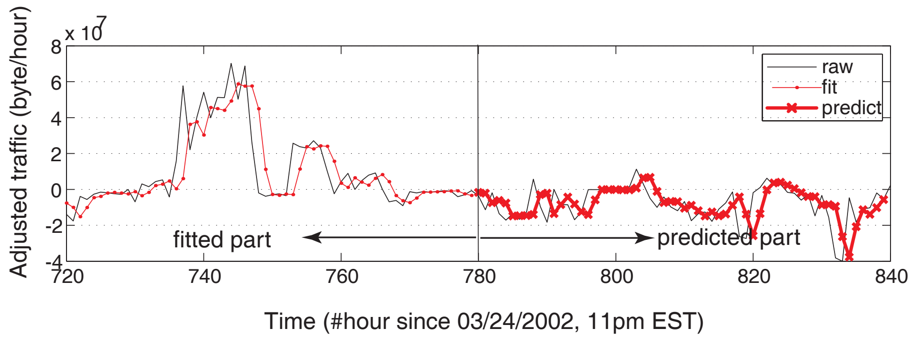
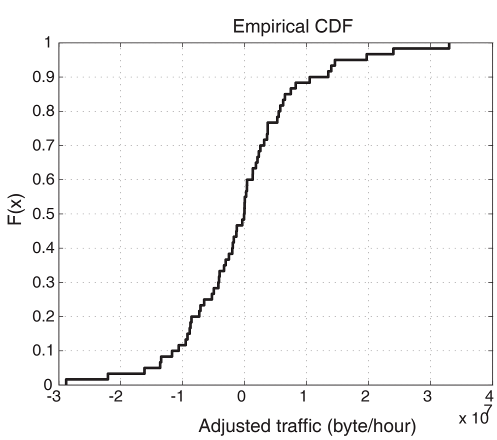


Fig. 4. Adjusted Traffic and Its Prediction

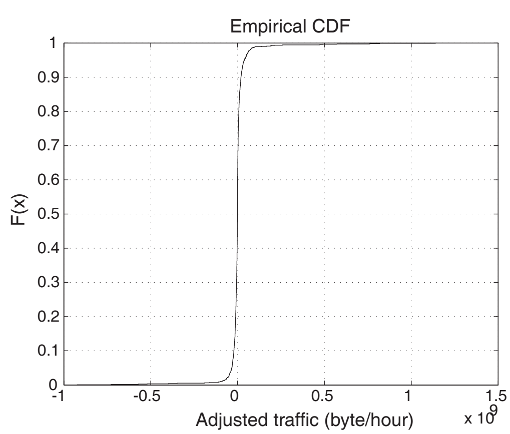
Finally, we define traffic prediction $\hat{x}$ as follows:

$$\hat{x}(t) = [\bar{x}(t) + \hat{z}(t)]^+ \tag{8}$$

where $[x]^+ = \max\{0, x\}$. Fig. 6 plots the predicted traffic series $\hat{x}(t)$ in comparison with the raw traffic. We can see the predicted traffic closely matches the real(raw) traffic. The cumulative distribution function of the prediction error $\epsilon_x(t)$, which is defined as $\epsilon_x(t) = x(t) - \hat{x}(t)$, is plotted in Fig. 5(b). It clearly shows that this distribution also fits the normal distribution with a near-zero mean.



(a) Prediction error for adjusted traffic



(b) Prediction error for entire series

Fig. 5. Cumulative Density Function of Prediction Error

We could consider the estimated traffic demand at time t as a random variable $X(t)$ which follows the normal distribution with mean $\hat{x}(t)$ and the same variance as ϵ_x. Fig. 7 shows the distribution of the predicted traffic demand of the 976th hour.

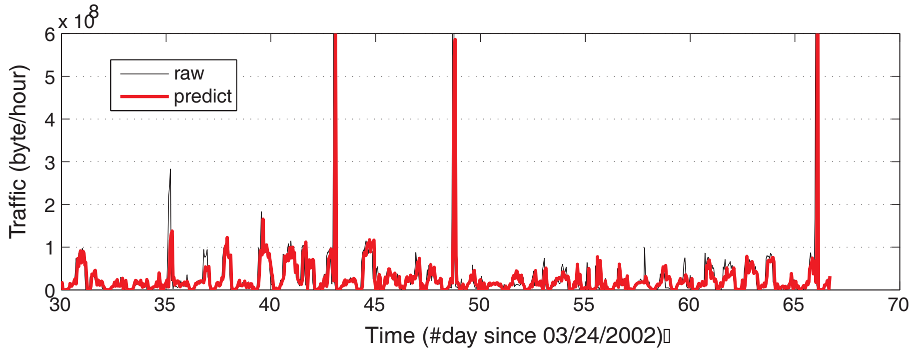


Fig. 6. Raw Traffic vs. Predicted Traffic

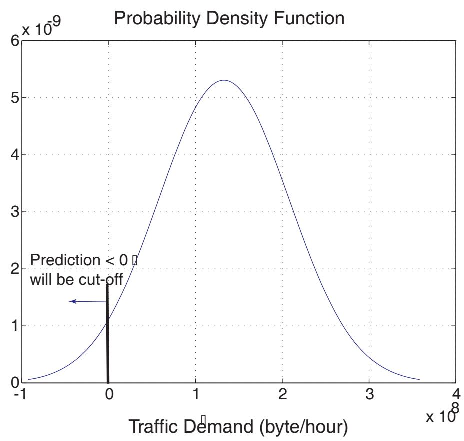


Fig. 7. Traffic Estimation Distribution

To summarize, the presented estimation method provides two prediction models: mean value and statistical distribution. These two traffic prediction models will serve as the inputs for the fixed-demand mesh network routing algorithm (**FMR**) and the uncertain-demand mesh network routing algorithm (**UMR**) which are presented in the next two sections.

IV. Fixed Demand Mesh Network Routing

This section investigates the optimal routing strategy for wireless mesh backbone network under fixed traffic demand. A common routing performance metric with respect to a fixed traffic demand is *resource utilization*. For example, link utilization is commonly used for traffic engineering in Internet [14], whose objective is to minimize the utilization at the most congested link. However, in a multihop wireless network, such as mesh backbone network, wireless link utilization may be inappropriate as a metric of routing performance due to the location-dependent interference.

On the other hand, the existing works on optimal mesh network routing [6] usually aim at maximizing the flow throughput, while satisfying the fairness constraints. In this formulation, traffic demand is reflected as the flow weight in the fairness constraints. In light of these results, we first outline the relation between the throughput optimization problem and the congestion minimization problem, and define the utilization (so-called *congestion*) of the adjusted interference set as the routing performance metric. We show that the solution derived from the throughput optimization could naturally leads to the routing scheme which balances the resource utilization and minimizes the network congestion under fixed traffic demand. We then present a fully polynomial time approximation algorithm, which finds an ϵ-approximate solution. The problem formulation and algorithm presented in this section will accept the mean-value traffic prediction as the input for routing. It also serves as the basis of uncertain demand routing discussed in Sec. V.

A. Problem Formulation

We first study the formulation of throughput optimization routing problem in a wireless mesh backbone network under the fixed traffic demand. Recall that $f \in F$ is the aggregated traffic flow between the local access points and the virtual 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 $\lambda \cdot d_f$ 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.

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 the 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

$$\sum_{e'\in S_e} \sum_{f:P\in\mathcal{P}_f \& e'\in P} x_f(P) \le c \tag{9}$$

To simplify the above equation, we define $A_{eP} = |S_e \cap P|$ as the number of wireless links path P passes in the adjusted interference set S_e. The throughput optimization routing with fairness constraint is then formulated as the following linear programming (LP) problem:

$$\begin{aligned} \mathbf{P_T}: \quad & \text{maximize} \quad \lambda && (10)\\ & \text{subject to} \quad \sum_{P\in\mathcal{P}_f} x_f(P) \ge \lambda \cdot d_f, \forall f \in F && (11)\\ & \sum_{f\in F} \sum_{P\in\mathcal{P}_f} x_f(P) A_{eP} \le c, \forall e \in E && (12)\\ & \lambda \ge 0, x_f(P) \ge 0, \forall f \in F, \forall P \in \mathcal{P}_f && (13) \end{aligned}$$

In this problem, the optimization objective is to maximize λ, such that at least $\lambda \cdot d_f$ units of data can be routed for each aggregated flow f with demand d_f. Inequality (11) enforces fairness by requiring that the comparative ratio of traffic routed for different flows satisfies the comparative ratio of their demands. Inequality (12) 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.

Now we proceed to study the congestion minimization routing. Let $x'_f(P)$ be the rate of flow f on path P under traffic demand d_f. It is obvious that $\sum_{P\in P_f} x'_f(P) = d_f$. The traffic being routed within the adjusted interference set S_e is given by $\sum_{f\in F}\sum_{P\in\mathcal{P}_f} x'_f(P)A_{eP}$. We define the *congestion* of an adjusted interference set S_e using its utilization (*i.e.*, the ratio between its load and the channel capacity) and denote it as θ_e:

$$\theta_e = \frac{\sum_{f\in F}\sum_{P\in\mathcal{P}_f} x'_f(P)A_{eP}}{c} \tag{14}$$

Further, we define $\theta = \max_{e\in E}\theta_e$ as the maximum congestion among all the adjusted interference sets. The congestion minimization routing problem is then formulated as follows:

$$\mathbf{P_C}: \quad \text{minimize} \quad \theta \tag{15}$$

$$\text{subject to} \quad \sum_{P\in\mathcal{P}_f} x'_f(P) \geq d_f, \forall f \in F \tag{16}$$

$$\sum_{f\in F}\sum_{P\in\mathcal{P}_f} x'_f(P)A_{eP} \leq c\cdot\theta, \forall e \in E \tag{17}$$

$$\theta \geq 0, x'_f(P) \geq 0, \forall f \in F, \forall P \in \mathcal{P}_f \tag{18}$$

To reveal the relation between $\mathbf{P_T}$ and $\mathbf{P_C}$, we let $\theta = \frac{1}{\lambda}$ and $x'_f(p) = \frac{x_f(p)}{\lambda}$. Problem $\mathbf{P_C}$ is then transformed to:

$$\mathbf{P'_C}: \quad \text{minimize} \quad \frac{1}{\lambda} \tag{19}$$

$$\text{subject to} \quad \frac{1}{\lambda}\sum_{P\in\mathcal{P}_f} x_f(P) \geq d_f, \forall f \in F \tag{20}$$

$$\frac{1}{\lambda}\sum_{f\in F}\sum_{P\in\mathcal{P}_f} x_f(P)A_{eP} \leq c\cdot\theta, \forall e \in E \tag{21}$$

$$\lambda \geq 0, x_f(P) \geq 0, \forall f \in F, \forall P \in \mathcal{P}_f \tag{22}$$

which is obviously equivalent to the throughput optimization problem $\mathbf{P_T}$.

B. Algorithm

Both problems $\mathbf{P_T}$ and $\mathbf{P_C}$ could be solved by a LP-solver such as [15]. To reduce the complexity for practical use, we present a fully polynomial time approximation algorithm for problem $\mathbf{P}_T$, 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. We assign a price μ_e to each set S_e for $e \in E$. The objective is to minimize the aggregated price for all adjusted interference sets. As the constraint, Inequality (24) 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 (25) requires that the weighted flow price μ_f over its demand d_f must be at least 1.

$$
\begin{aligned}
\mathbf{D_T}: \quad & \text{minimize} \quad \sum_{e \in E} c \cdot \mu_e && (23)\\
& \text{subject to} \quad \sum_{e \in E} A_{eP}\mu_e \geq \mu_f, \forall f \in F, \forall P \in \mathcal{P}_f && (24)\\
& \qquad\qquad\quad \sum_{f \in F} \mu_f d_f \geq 1 && (25)
\end{aligned}
$$

	FMR: Mesh Network Routing Under Fixed Demand
1	$\forall e \in E$, $\mu_e \leftarrow \beta/c$
2	$x_f(P) \leftarrow 0$, $\forall P \in \mathcal{P}_f$, $\forall f \in F$
3	**while** $\sum_{e \in E} c \cdot \mu_e < 1$
4	**for** $\forall f \in F$ **do**
5	$d'_f \leftarrow d_f$
6	**while** $\sum_{e \in E} c \cdot \mu_e < 1$ **and** $d'_f > 0$ **do**
7	$P \leftarrow$ lowest priced path in $\mathcal{P}_f$ using μ_e
8	$\delta \leftarrow \min\{d'_f, \min_{e \in P} A_{eP}\}$
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 II
ROUTING ALGORITHM UNDER FIXED DEMAND

Based on the above dual problem $\mathbf{D_T}$, our fast approximation algorithm is presented in Table II. The algorithm design follows the idea of [16]. To start with, we initialize the price on each adjusted interference set S_e as β/c (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 adjusted interference sets 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 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, since capacity c is the same for all the adjusted interference sets, its value does not affect the routing solution. Third, as one might see, the algorithm in fact routes more traffic than its actual demand, Therefore, a scaling procedure is needed to scale down the routed traffic so it fits its actual demand. In particular, $x_f(P)$ will be scaled as follows

$$
x'_f(P) = x_f(P) \cdot \frac{d_f}{\sum_{P \in \mathcal{P}_f} x_f(P)} \tag{26}
$$

We formally analyze the properties of our algorithm in the following theorem.

Theorem 1: If $\beta = (|E|/(1-\epsilon))^{-1/\epsilon}$, then the final flow generated by **FMR** is at least $(1-3\epsilon)$ times the optimal value of $\mathbf{P}$. The running time is $O(\frac{1}{\epsilon^2}[\log|E|(2|F|\log|F|+|E|)+\log U)]) \cdot T_{mp}$, where U is the length of the longest path in G, and T_{mp} is the running time to find the shortest path.

V. 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

$$Pr(D_f = d_f^i) = q_f^i \tag{27}$$

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, $\boldsymbol{D}$ be the corresponding random variable, and $\mathcal{D}$ be the sample space. We further 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.

$$Pr(\boldsymbol{D} = \boldsymbol{d}) = Pr(D_f = d_f^i, f \in F) = \Pi_{f \in F} q_f^i \tag{28}$$

Let us consider a traffic routing solution $(x_f(P), P \in \mathcal{P}_f, f \in F)$ that satisfies the capacity constraint (Inequality (12)). It is obvious that λ is a function of $\boldsymbol{d}$:

$$\lambda(\boldsymbol{d}) = \min_{f \in F}\{\frac{x_f}{d_f}\} \tag{29}$$

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 II. We denote the optimal value of λ as $\lambda^*(\boldsymbol{d})$. We further define the *performance ratio* ω of routing solution $(x_f(P), P \in \mathcal{P}_f, f \in F)$ as follows:

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

Obviously, the performance ratio is also a random variable under uncertain demand. We denote it as Ω. Ω 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 Ω, which is given as follows.

$$E(\Omega) = Pr(\boldsymbol{D} = \boldsymbol{d}) \times \frac{\lambda(\boldsymbol{d})}{\lambda^*(\boldsymbol{d})} \tag{30}$$

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 for wireless mesh backbone network under uncertain traffic demand as follows.

$$\mathbf{P_U}: \quad \text{maximize} \quad \sum_{\boldsymbol{d}\in\mathcal{D}} p(\boldsymbol{d}) \frac{\lambda(\boldsymbol{d})}{\lambda^*(\boldsymbol{d})} \tag{31}$$

$$\text{subject to} \quad \forall \boldsymbol{d} \in \mathcal{D}, \text{where } \boldsymbol{d} = (d_f, f \in F)$$

$$\sum_{P\in\mathcal{P}_f} x_f(P) \geq \lambda(\boldsymbol{d}) \cdot d_f, \forall f \in F \tag{32}$$

$$\sum_{f\in F} \sum_{P\in\mathcal{P}_f} x_f(P) A_{eP} \leq c, \forall e \in E \tag{33}$$

$$\lambda \geq 0, x_f(P) \geq 0, \forall f \in F, \forall P \in \mathcal{P}_f \tag{34}$$

Similar to problem $\mathbf{P_T}$, the constraints of $\mathbf{P_U}$ come from the fairness requirement and the wireless mesh network capacity. In particular, Inequality (32) enforces fairness for all demand $\boldsymbol{d} \in \mathcal{D}$, and Inequality (33) enforces capacity constraint as Inequality (12) in problem $\mathbf{P_T}$.

Now we consider the dual problem $\mathbf{D_U}$ of $\mathbf{P_U}$. Similar to $\mathbf{D_T}$, the objective of $\mathbf{D_U}$ is to minimize the aggregated price for all adjusted interference sets. However, in Inequality (37), for each sample demand vector $\boldsymbol{d}$, the aggregated price of all flows weighted by their demand needs to be larger than its probability.

$$\mathbf{D_U}: \quad \text{minimize} \quad \sum_{e\in E} c \cdot \mu_e \tag{35}$$

$$\text{subject to} \quad \sum_{e\in E} A_{eP}\mu_e \geq \mu_f, \forall f \in F, \forall P \in \mathcal{P}_f \tag{36}$$

$$\sum_{f\in F} \mu_f d_f \geq \frac{p(\boldsymbol{d})}{\lambda^*(\boldsymbol{d})}, \forall \boldsymbol{d} \in \mathcal{D} \tag{37}$$

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

Now we present an approximation algorithm for $\mathbf{P_U}$ in Table III. Note that since the channel capacity c will not affect the final result of the algorithm, we simply omit it here. This algorithm (**UMR**) has the same initialization as the algorithm for problem $\mathbf{P_T}$ (**FMR**). 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 (36) and (37) 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 Lines 4 to 11 of **FMR** algorithm. Following the same proving sequence for **FMR**, we are able to prove the similar properties with **UMR**.

Theorem 2: If $\beta = (|E|/(1-\epsilon))^{-1/\epsilon}$, then the final flow generated by **UMR** is at least $(1-3\epsilon)$ times the optimal value of $\mathbf{P_U}$. The running time is $O(\frac{1}{\epsilon^2}[\log|E|(2|\mathcal{D}||T_{fmr}||F|\log|F| + |E|) + \log U)]) \cdot T_{mp}$, where U is the length of the longest path in G, T_{mp} is the running time to find the shortest path, and T_{fmr} is the running time of the **FMR** algorithm.

UMR: Mesh Network Routing Under Uncertain Demand

$\forall e \in E$, $\mu_e \leftarrow \beta$
$x_f(P) \leftarrow 0$, $\forall P \in \mathcal{P}_f$, $\forall f \in F$
loop
  for $\forall f \in F$ **do**
    $\bar{P} \leftarrow$ lowest priced path in $\mathcal{P}_f$ using μ_e
    $\mu_f \leftarrow \sum_{e \in E} A_{e\bar{P}} \mu_e$
  end for
  for $\forall \boldsymbol{d} \in \mathcal{D}$ **do**
    $\mu_{\boldsymbol{d}} \leftarrow \sum_{f \in F} \mu_f d_f \frac{\lambda^*(\boldsymbol{d})}{p(\boldsymbol{d})}$
  end for
  $\mu^{\min} \leftarrow \min_{\boldsymbol{d} \in \mathcal{D}} \mu_{\boldsymbol{d}}$
  $\boldsymbol{d}^{\min} \leftarrow \arg\min_{\boldsymbol{d} \in \mathcal{D}} \mu^{\min}$
  if $\mu^{\min} \geq 1$
    return
  for $\forall f \in F$ **do**
    $d'_f \leftarrow d^{\min}_f$
    while $d'_f > 0$ **do**
      $P \leftarrow$ lowest priced path in $\mathcal{P}_f$ using μ_e
      $\delta \leftarrow \min\{d'_f, \min_{e \in P} \frac{1}{A_{eP}}\}$
      $d'_f \leftarrow d'_f - \delta$
      $x_f(P) \leftarrow x_f(P) + \delta$
      $\forall e$ s.t. $A_{eP} \neq 0$, $\mu_e \leftarrow \mu_e(1 + \epsilon\delta A_{eP} \times \frac{\lambda^*(\boldsymbol{d}^{\min})}{p(\boldsymbol{d}^{\min})})$
    end while
  end for
end loop

TABLE III
ROUTING ALGORITHM UNDER UNCERTAIN DEMAND

VI. SIMULATION STUDY

A. Simulation Setup

AP	31AP3	34AP5	55AP4	57AP2	62AP3	62AP4	82AP4	94AP1	94AP3	94AP8
Node ID	22	18	57	5	55	20	53	3	56	27
AP	27AP3	3AP3	21AP2	23AP4	33AP2	62AP2	82AP3	84AP1	90AP2	97AP2
Node ID	9	23	25	33	19	35	58	42	6	48

TABLE IV
OVERVIEW OF TRAFFIC DEMAND

We evaluate the performance of our algorithms via simulation study. The tool we used for the simulation study is a self-developed simulator package which provides flow-level routing simulation in wireless mesh networks. The simulator follows the protocol model and the scheduling algorithm presented in [12] to resolve the wireless interference at the MAC layer. In the simulated wireless mesh network, 60 mesh nodes are randomly deployed over a $1000 \times 2000m^2$ region. The simulated network topology is shown in Fig. 8. In the basic setting, 10 nodes at the edge of this network are selected as the local access points (LAP) that forward traffic for clients. Four nodes

(1, 31, 50 and 51) in the center of the deploy region are selected as the gateway access points. Aside from this basic setting, we have also evaluated the performance of our algorithms with different configurations of LAPs and gateways, which we will show at the later part of this section. Each mesh node has a transmission range of $250m$ and an interference range of $500m$. The channel capacity c is set as 11 Mbps.

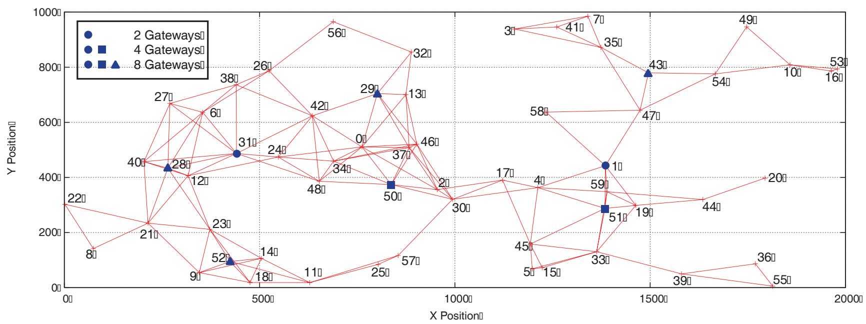


Fig. 8. Mesh Network Topology.

To realistically simulate the traffic demand at each LAP, we employ 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 [9], which is the best set that resembles the traffic condition of a wireless mesh network, given the present availability of wireless network traces. By analyzing the *snmp* log trace at each access point, we are able to derive its 1847-hour incoming and outgoing traffic volume since 12:00AM, March 25, 2002 EST. We argue that the LAPs 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 the access points from the Dartmouth campus wireless LAN and assign their traffic traces to the LAPs in our simulation. The traffic assignment is given in Table IV. In the basic setting, the 10 access points in the first row of the table are used.

We evaluate and compare different traffic prediction and routing strategies for this simulated network. In particular, we consider the following strategies.

- *Oracle Routing (OR)*. In this strategy, the traffic demand is known a priori. It runs the **FMR** algorithm (presented in Tab. II) based on this demand. By default, this solution runs every hour based on the up-to-date traffic demand from the trace and returns the optimal set of routes. This ideal strategy is designed to return the benchmark result, which the rest of the practical strategies compare to.
- *Mean-Value Prediction Routing (MVPR)*. This strategy does not know the traffic demand a priori. Instead, it only predicts the traffic demand based on its historical data. In particular, it employs the mean value prediction model and runs the **FMR** algorithm based on this predicted demand. This solution also runs every hour to provide the set of routes for the next hour by default.
- *Statistical-Distribution Prediction Routing (SDPR)*. Similar to *MVPR*, this strategy also relies on traffic prediction. It predicts not only the mean-value of the traffic demand in the next hour (by default), but also its distribution. It runs the **UMR** algorithm (presented in Tab. III) with the predicted traffic demand distribution as its input. Since **UMR** only accepts discrete probability distribution, we need to discretize the demand distribution as follows.

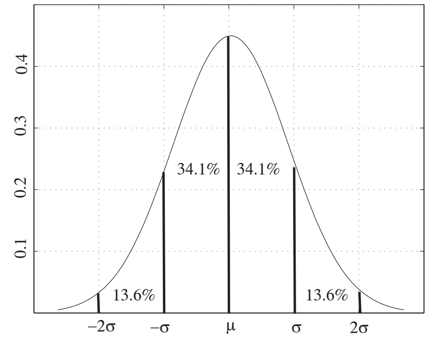


Fig. 9. Discretization of Traffic Distribution

As illustrated in Fig. 9, we sample the following values, the mean value μ, and values $\mu - \sigma$, $\mu + \sigma$, $\mu - 2\sigma$, and $\mu - 2\sigma$. Since about 95% of all traffic demand values fall within the range $[\mu - 2\sigma, \mu + 2\sigma]$, we ignore the values which has a probability smaller than 5%.

- *Shortest-Path Routing (SPR)*. This strategy is agnostic of traffic demand, and returns fixed routing solution purely based on the shortest distance (number of hops) from each mesh node to the gateway. The purpose to evaluate this strategy is to quantitatively contrast the advantage of our traffic-predictive routing strategies.

B. Simulation Results

We experiment with the above routing strategies along the time range $[108, 1847]$, a 1740-hour period excerpted from the trace[2]. We mainly study the congestion θ of each routing strategy under the given traffic demands.

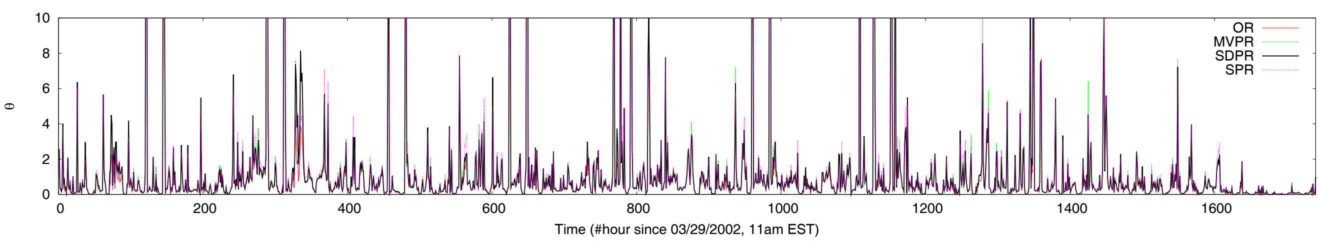


Fig. 10. Overview of All Strategies

We start by presenting the congestion achieved by all strategies (*OR*, *MVPR*, *SDPR*, and *SPR*) during the entire 1740-hour simulation period. As seen in Fig. 10, *OR* constantly achieves the minimum worst-case congestion among others, due to its unrealistic capability to know the actual traffic demand. We note that the burstiness of θ applies to all strategies including *OR*. Such observation comes from the burstiness of the traffic load in the *snmp* log trace, which is caused by the insufficient level of traffic multiplexing at wireless local access points.

To show the effectiveness of prediction-based routing, we take a closer look of the comparison. We calculate the congestion ratio between *MVPR* and *SPR* $\frac{\theta_{MVPR}}{\theta_{SPR}}$ at all time instances and sort the result in Fig. 11. From the figure, we could see that *MVPR* outperforms *SPR* in 70.6% of all time instances (test cases).

To understand how competitive each routing strategy is, in Fig. 12(a), we normalize θ achieved by each routing strategy by the same value of *OR*. Since *OR* always achieves the minimum θ among others, this ratio will end up

[2]Note that the beginning part of the trace $[0, 107]$ is used as training data, thus is not included in the simulation result.

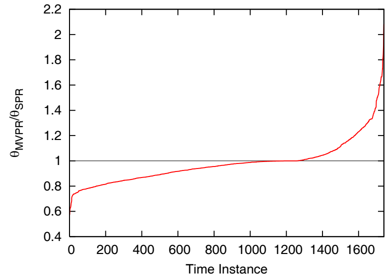


Fig. 11. $\frac{\theta_{MVPR}}{\theta_{SPR}}$ Sorted View

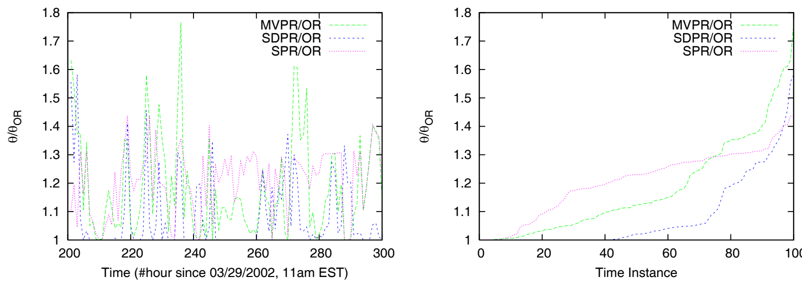
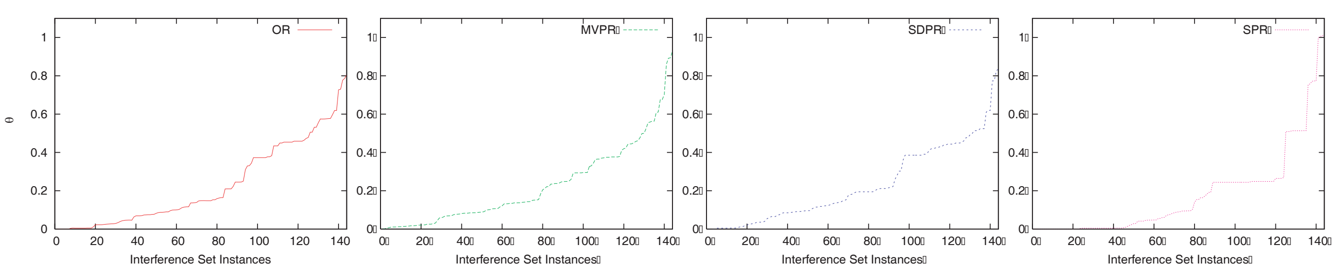


Fig. 12. (a) Congestion Ratio (b) Sorted Overview

at least 1. Also we take a close-up look during the hour range $[201, 300]$. Here, all three strategies (*MVPR*, *SDPR*, and *SPR*) achieve less than 2 times of the optimal congestion. Although *MVPR* has the worst performance at a few occasions, *SPR* causes greater congestion than others in most of the time, revealing the disadvantage of overlooking the varying traffic demand. *SDPR* constantly achieves lower congestion than *MVPR* due to more comprehensive representation of the traffic demand estimation.

The above observations get clearer when we sort out the normalized congestion ratio for the three strategies in Fig. 12(b). Interestingly, although *SPR* is inferior than the traffic-prediction strategies generally, its worst-case congestion is lower than *MVPR* in 25% of the time, and *SDPR* in less than 5% of the time. This problem can be mostly attributed to the inaccuracy of traffic prediction. When traffic variation is too dramatic and irregular, the prediction error can cause routing solutions worse than being agnostic about it. However, more sophisticated prediction technique (*SDPR*) can greatly reduce its occurring probability than the simple one (*MVPR*).



(a) *OR* (b) *MVPR* (c) *SDPR* (d) SPR

Fig. 13. Adjusted Interference Set Sorted By Congestion

Next, we take a closer look at each strategy's ability to balance the traffic within the mesh network. In Fig. 13, we unfold a single time instance at hour 1521 and exhibit the congestion at each adjusted interference set resulted from each strategy. In order to achieve the lowest worst-case congestion, a good strategy should maximally even out the traffic routed through all interference sets. Obviously, *OR* achieves such optimality, which resulted in the best θ value 0.8. *SPR* has the highest θ value as more than 1. The results for *MVPR* and *SDPR* are 0.9 and 0.8 respectively. In *MVPR*, 120 out of 140 interference sets have their congestions less than 0.4. Comparatively, in *SDPR*, about 100 interference sets are below this threshold, whereas the number is below 100 in *OR*, and above 120 in *SPR*. This observation keeps consistent when we repeat with different threshold in θ. The balanced traffic in the network also leads to higher throughput (weighted over their demands) as shown in Fig. 14. From the figure, we could see that the *SDPR* strategy has a flow throughput close to the *OR* routing strategy and also slightly outperforms *MVPR*. The *SPR* has the worst performance due to the fact that it is unable to balance the traffic load.

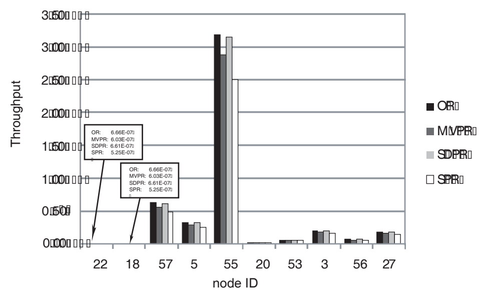
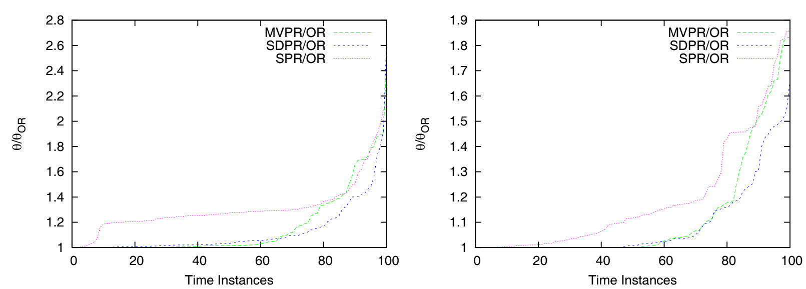


Fig. 14. Flow Throughput

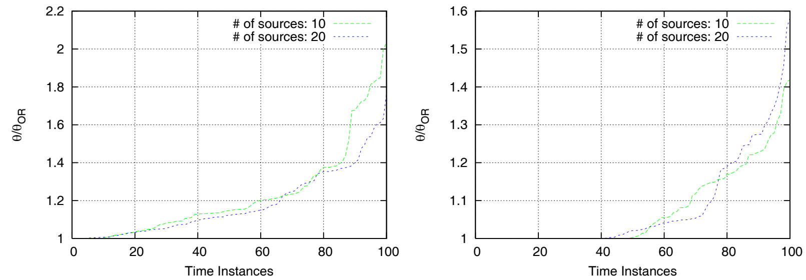


(a) 2 Gateways (b) 8 Gateways

Fig. 15. Impact by Number of Gateways

In what follows, we alter our simulation configurations to examine the abilities of different strategies at adapting various network settings and evaluate the impact of network parameters on these routing strategies.

Since deploying multiple gateways is a commonly-used solution to improve mesh network throughput and avoid hot spots, we first evaluate the impact of number of gateways in the network. In the experiment, we change the default number of gateways from 4 to 2 and 8 respectively as shown in Fig. 8. We show the performances of *MVPR*, *SDPR*, and *SPR* using their congestion ratios θ/θ_{OR} (normalized by the *OR* routing results) in Fig. 15. In Fig. 15, we observe that the highest congestion ratios by both strategies drop linearly as we increase the number of gateways, i.e., 2.6 at 2 gateways and 1.8 at 8 gateways. Also *SDPR* consistently outperforms *MVPR* at approaching the optimal *OR* strategy. In case of 2 gateways, more than 80% of the time, the performance of *SDPR* is within 20% of optimal congestion, whereas the same value is 40% for *MVPR*. In addition, in case of 8 gateways, *SDPR* achieves within 60% of optimal congestion of all times, 20% less than *MVPR*. In all cases, *SPR*'s performance is generally 20% worse than the *MVPR* and *SPDR* strategies.



(a) *MVPR* (b) *SDPR*

Fig. 16. Impact by Number of Sources

Now, we test our solutions' adaptability to traffic demand by doubling the number of LAPs to 20. The traffic assignments of these 20 LAPs are shown in Table IV. Here, we observe that both solutions show good and stable approximation to the optimal *OR* strategy. In fact, more intensive traffic (with increasing multiplexing) makes our solutions approximate closer to the optima one. Compared to the case of 10 LAPs, the worst-case performance of *MVPR* moves closer from within 100% optimality to 80%. For *SDPR*, it reduces from within 60% to 40%, consistently outperforming *MVPR* by 40%. In addition, *SDPR* achieves the same congestion with *OR* in 50% of the times, a 10% increase from the case of 10 LAPs.

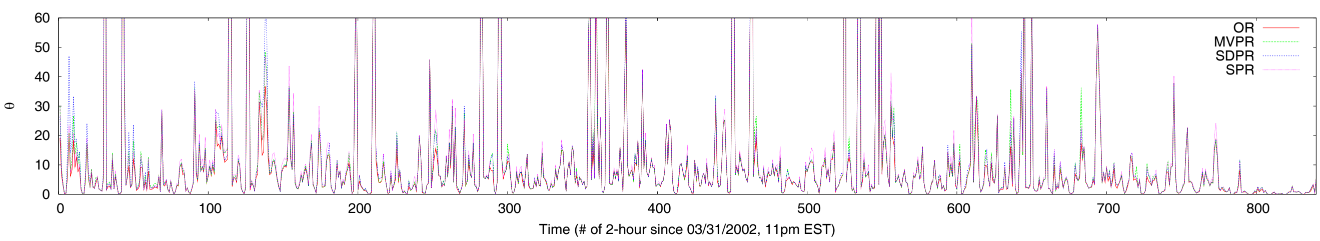


Fig. 17. Overview of All Strategies Based on 2-Hour Prediction

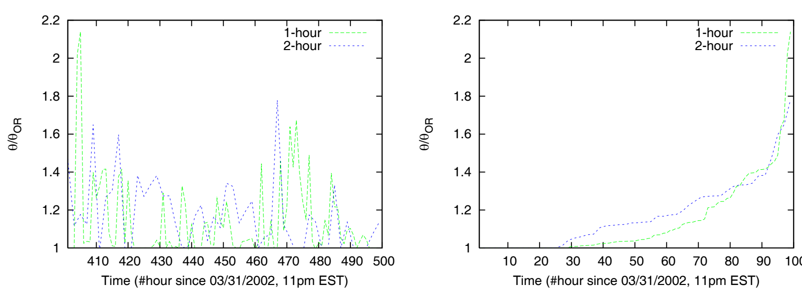


Fig. 18. (a) Congestion Ratio (b) Sorted Overview

In the default setting, the *OR*, *MVPR* and *SDPR* strategies are executed every hour based on the updated/predicted traffic demands. We now evaluate the impact of this prediction period and study how it could be selected appropriately. To do so, we run our simulation under different prediction periods. Fig. 17 shows the congestion of all strategies by performing the traffic prediction and the routing update every two hours. A closer comparison of the performance of the *SDPR* strategy based on 1-hour and 2-hour prediction respectively is shown in Fig. 18. From the figure we can see that the *SDPR* strategy performs better under the 1-hour traffic prediction for most of the time instances, while the result based on the 2-hour traffic prediction has a relatively better worst-case performance. This is mainly because the 2-hour traffic prediction produces relatively more stable prediction results than the 1-hour traffic prediction, yet it is not as agile as 1-hour prediction to timely track the traffic changes. We have also tested several other time periods and had similar observations – while shorter prediction intervals may track the traffic changes more timely, it is also unstable and sometimes deviates larger from the actual value.

It is also worth noting that both routing strategies (*MVPR* and *SDPR*) need to be implemented in a centralized fashion. To implement these algorithms, a mesh node in the network should be responsible for periodically collecting the traffic load information from all local access points, predicting the traffic demands, and calculating the routing paths. Once the routing paths are computed, this node also disseminates the new routing information to all the other mesh nodes, which will update their routing tables. It is obvious that shorter prediction period also leads to higher computational load, more frequent route updates and higher message overhead. So in order to balance the traffic prediction agility and stability, and control the overhead, we choose one hour as the default prediction period in our simulation study. In general, this period needs to be selected based on the traffic dynamics so that it is short enough to track the traffic dynamics; at the same time long enough to produce stable prediction and minimize the route update overhead.

VII. 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 [4], such as expected packet transmission time [3], channel cost metric (CCM) which is the sum of expected transmission time weighted by the channel utilization [17]. Joint solutions for channel allocation and routing are explored in [18] using a centralized algorithm and in [5] in a distributed fashion. These heuristic solutions are designed to adapt to the dynamic network condition. However, 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) under their routing schemes.

There are also theoretical studies that formulate these network planning decisions into optimization problems. For example, the work of [11] formulates the wireless network routing problem into a linear programming problem. The works of [6], [19] study the optimal solution of joint channel assignment and routing for maximum throughput under a multi-commodity flow problem formulation and solve it via linear programming. The work of [7] presents bandwidth allocation schemes to achieve maximum throughput and lexicographical max-min fairness respectively. Further, the work of [20] 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. However, they may not be unsuitable for practical use under highly dynamic traffic situation. Different from these existing works, our work explicitly incorporates traffic behavior analysis and prediction into the routing optimization, thus better fits the routing need in the dynamic wireless mesh networks. It is also worth mentioning that although our problem formulation is similar to the one in [11], this problem is solved by a LP solver in [11], while we provide a polynomial approximation algorithm (*FMR*) for it. The design of *FMR* is tailored to the mesh network routing problem and can be extended to *UMR* which handles the uncertain demands, whereas a general LP solver can not.

Distributed algorithms are presented for joint scheduling and routing in [21], and for joint channel assignment, scheduling and routing in [22]. 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 [23], [24] analyze a campus-wide wireless network traffic and its changes in two years. [8] statistically characterizes both static flows and roaming flows in a large campus wireless network. In [25], 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 [14] and oblivious routing [26] in Internet, which also considers the impact of demand uncertainty in making routing decisions. The major difference between our work and these existing works lies in the different network and traffic models of wireless mesh network and Internet. For example, traffic engineering tries to minimize the congestion (utilization) of the wireline 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.

VIII. Conclusion

This paper studies the optimal routing strategies for single-radio single-channel wireless mesh networks under dynamic and uncertain traffic demand. It establishes two prediction models based on time series analysis, and extends the classical maximum concurrent flow problem with statistical demand input. Based on the traffic demand from the real wireless network traces, the simulation study shows that our problem formulation and algorithm could effectively incorporate the traffic demand dynamics.

References

[1] “Seattle wireless,” http://www.seattlewireless.net.

[2] “Mit roofnet,” http://www.pdos.lcs.mit.edu/roofnet/.

[3] Richard Draves, Jitendra Padhye, and Brian Zill, “Routing in multi-radio, multi-hop wireless mesh networks,” in *Proc. of ACM Mobicom*, 2004, pp. 114–128.

[4] Sanjit Biswas and Robert Morris, “Exor: opportunistic multi-hop routing for wireless networks,” in *Proc. of ACM SIGCOMM*, 2005, pp. 133–144.

[5] 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.

[6] Mansoor Alicherry, Randeep Bhatia, and Li Li, “Joint channel assignment and routing for throughput optimization in multi-radio wireless mesh networks,” in *Proc. of ACM MobiCom*, 2005.

[7] 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.

[8] Xiaoqiao (George) Meng, Starsky H. Y. Wong, Yuan Yuan, and Songwu Lu, “Characterizing flows in large wireless data networks,” in *Proc. of ACM MobiCom*, 2004.

[9] “A community resource for archiving wireless data at dartmouth,” http://crawdad.cs.dartmouth.edu/.

[10] P. Gupta and P.R. Kumar, “The capacity of wireless networks,” *IEEE Trans. on Information Theory*, pp. 388–404, 2000.

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

[12] V. S. Anil Kumar, M. V. Marathe, S. Parthasarathy, and A. Srinivasan, “Algorithmic aspects of capacity in wireless networks,” in *Proc. of ACM SIGMETRICS*, 2005, pp. 133–144.

[13] Yuan Xue, Baochun Li, and Klara Nahrstedt, “Optimal resource allocation in wireless ad hoc networks: A price-based approach,” *IEEE Transactions on Mobile Computing*, vol. 5, no. 4, pp. 347–364, April 2006.

[14] Hao Wang, Haiyong Xie, Lili Qiu, Yang Richard Yang, Yin Zhang, and Albert Greenberg, “Cope: traffic engineering in dynamic networks,” in *Proc. of ACM SIGCOMM*, 2006.

[15] “Ilog cplex mathematical programming optimizers,” http://www.ilog.com/products/cplex.

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

[17] H. Wu and F. Yang and K. Tan and J. Chen and Q. Zhang and Z. Zhang, “Distributed Channel Assignment and Routing in Multi-radio Multi-channel Multi-hop Wireless Networks,” *IEEE JSAC special issue on multi-hop wireless mesh networks*, vol. 24, no. 11, pp. 1972–1983, 2006.

[18] A. Raniwala, K. Gopalan, and T. Chiueh, “Centralized channel assignment and routing algorithms for multi-channel wireless mesh networks,” *Mobile Computing and Communications Review*, vol. 8, no. 2, pp. 50–65, 2004.

[19] Murali Kodialam and Thyaga Nandagopal, “Characterizing the capacity region in multi-radio multi-channel wireless mesh networks,” in *Proc. of ACM Mobicom*, 2005.

[20] Violeta Gambiroza, Bahareh Sadeghi, and Edward W. Knightly, “End-to-end performance and fairness in multihop wireless backhaul networks,” in *Proc. of ACM MobiCom*, 2004.

[21] Emilio Leonardi, Marco Mellia, Marco Ajmone Marsan, and Fabio Neri, “Joint optimal scheduling and routing for maximum network throughput,” in *INFOCOM*, 2005.

[22] Xiaojun Lin and Shahzada Rasool, “A distributed joint channel-assignment, scheduling and routing algorithm for multi-channel ad hoc wireless networks,” in *INFOCOM*, 2007.

[23] David Kotz and Kobby Essien, “Analysis of a campus-wide wireless network,” in *ACM MobiCom*, 2002, pp. 107–118.

[24] Tristan Henderson, David Kotz, and Ilya Abyzov, “The changing usage of a mature campus-wide wireless network,” in *Proc. of ACM MobiCom*, 2004, pp. 187–201.

[25] Nathanael Thompson, Guanghui He, and Haiyun Luo, “Flow scheduling for end-host multihoming,” in *Proceedings of IEEE INFOCOM*, 2006.

[26] Yossi Azar, Edith Cohen, Amos Fiat, Haim Kaplan, and Harald Racke, “Optimal oblivious routing in polynomial time,” *J. Comput. Syst. Sci.*, vol. 69, no. 3, 2004.

IX. Appendix

A. *Proof for* **Theorem 1**

The proof to **Theorem 1** is precluded by a sequence of lemmas. We first make the following denotations. We use OPT to represent the optimal solution of both $\mathbf{P_T}$ and $\mathbf{D_T}$, and OPT' to represent the solution derived from **FMR** algorithm.

Lemma 1 : If $OPT \geq 1$, scaling the final flow by $\log_{1+\epsilon} 1/\beta$ yields a feasible primal solution of value $OPT' = \frac{t-1}{\log_{1+\epsilon} 1/\beta}$, t being the number of phases the algorithm takes to stop.

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 $L^{\mu_e} \triangleq \sum_{e\in E} c \cdot \mu_e$. 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. II), 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 $\mu_e^{(1)(0)} = \mu_e^{(0)(|F|)} = \beta/c$. Hence,

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

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 $\mathbf{D_T}$ is to minimize $L^{(i)(|F|)}$, subject to the constraint that $\mu^{(i)(|F|)} \geq 1$. This constraint can be easily satisfied if we scale the length of all inference sets by $1/\mu^{(i)(|F|)}$. So $\mathbf{D_T}$ 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 $\mathbf{D_T}$ is $OPT \triangleq \min_{\mu^{(i)(|F|)}} \frac{L^{(i)(|F|)}}{\mu^{(i)(|F|)}}$.

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

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

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

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

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

Therefore,

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

which implies

$$\frac{OPT}{t-1} \le \frac{\epsilon}{(1-\epsilon)\ln\frac{1-\epsilon}{\beta|E|}} \tag{38}$$

Now consider an interference set S_e. For every c units of flow routed through S_e, we increase its price by at least a factor $(1+\epsilon)$. Initially, its length is β/c and after $t-1$ phases, since $L^{(t)(|F|)} < 1$, the price of S_e satisfies $\mu_e^{(t-1)(|F|)} < 1/c$. Therefore the total amount of flow through S_e in the first $t-1$ phases is strictly less than $\log_{1+\epsilon}\frac{1/c}{\beta/c} = \log_{1+\epsilon} 1/\beta$ 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 $OPT' = \frac{t-1}{\log_{1+\epsilon} 1/\beta}$.

■

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

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

$$\frac{OPT}{OPT'} < \log_{1+\epsilon} 1/\beta \tag{39}$$

Substituting the bound on $OPT/(t-1)$ from In Equality (38), we get

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

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

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

■

Lemma 3: If $OPT \ge 1$ and $\beta = (|E|/(1-\epsilon))^{-1/\epsilon}$, **Algorithm I** terminates after at most $t = 1 + \frac{OPT}{\epsilon}\log_{1+\epsilon}\frac{|E|}{1-\epsilon}$ phases.

Proof: From In Equality (39) and weak-duality, we have

$$1 \le \frac{OPT}{OPT'} < \log_{1+\epsilon} 1/\beta$$

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 $t \le 1 + \frac{OPT}{\epsilon}\log_{1+\epsilon}\frac{|E|}{1-\epsilon}$ ■

These lemmas require that $OPT \ge 1$. 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_i be the maximum traffic value of flow f_i when all other flows have zero traffic. Let $\zeta = \min_i \frac{\zeta_i}{d(f_i)}$. Since at best all single commodity maximum flows can be routed simultaneously, ζ is an upper bound on OPT'. On the other hand, routing $1/|F|$ fraction of each flow of value ζ_i is a feasible solution, which implies that $\zeta/|F|$ is a lower bound on OPT. To ensure that $OPT \ge 1$, 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.

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 OPT'. 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 $O(\frac{\log|E|}{\epsilon^2}(2|F|\log|F| + |E|)) \cdot T_{mp}$.

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 $1+\epsilon$. d_e has initial value β/c and value at most $1/c$ before the final step of the algorithm. Otherwise, the stop criterion of the algorithm, $\sum_{e\in E} c \cdot d_e \geq 1$, 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. ∎

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

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. ∎

B. Proof for **Theorem 2**

The proof for **Theorem 2** follows the same sequence as the proof to **Theorem 1**, 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. II), we have

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

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.