Abstract—Traffic routing plays a critical role in determining the performance of a wireless mesh network. To investigate the best 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 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. 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 traffic dynamics in mesh network routing.

I. INTRODUCTION

Wireless mesh networks 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. To alleviate the problem of location-dependent interference in wireless communication, mesh routers are usually equipped with multiple radios which enable them to transmit and receive simultaneously or transmit on multiple channels simultaneously.

Traffic routing and channel assignment jointly play 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 algorithms (e.g., [1]–[4]). 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 based on optimization methods (e.g., [5], [6]). The 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 [7] 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. This routing framework could work as a part of the joint routing and channel assignment solution in [5]. 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 [8], 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. 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 ($\lambda$) 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 next 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 $\lambda$ 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 $\lambda$ 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 a 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 modeled by their statistical distributions.

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

II. SYSTEM MODEL AND SOLUTION OVERVIEW
A. Network and Interference Model

In a multi-hop wireless mesh 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 gateways which are connected to the Internet. We use $w \in W$ to denote the set of gateways in the network. In the following discussion, local access point, gateway and mesh router are collectively called mesh nodes and denoted by set $V$ (Note that $W \subset V$). Further, we assume that node $v$ is equipped with $\kappa(v)$ radios. The network could use a set of orthogonal wireless channels denoted by $C$. For example, in the IEEE 802.11b standard, $|C| = 3$.

In a wireless network, packet transmissions in the same channel are subject to location-dependent interference. We assume that all mesh nodes have the uniform transmission range denoted by $R_T$. Usually the interference range is larger than its transmission range. We denote the interference range of a mesh node as $R_I = (1 + \Delta)R_T$, where $\Delta \geq 0$ is a constant. In this paper, we consider the protocol model presented in [9]. Let $r(u, v)$ be the distance between $u$ and $v$ ($u, v \in V$). In the protocol model, packet transmission from node $u$ to $v$ on channel $c \in C$ is successful, if and only if (1) the distance between these two nodes $r(u, v)$ satisfies $r(u, v) \leq R_I$; (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 on the same channel. If node $u$ can transit to $v$ directly on channel $c$, they form an edge $e(c)$. We denote the capacity of this edge as $\phi_c(c)$ which is the maximum data rate that can be transmitted. Let $E_c$ be the set of all edges $e(c)$. We say two edges $e(c), e'(c)$ interfere with each other, if they can not transmit simultaneously based on the protocol model. Further we define interference set $I_e(c)$ which contains the edges that interfere with edge $e$ and $e'$ itself.

Finally, we introduce a virtual node $w^*$ to represent the Internet. $w^*$ is connected to each gateway with a virtual edge $e' = (w^*, w), w \in W$. For simplicity, we assume that the link capacity in Internet is much larger than the wireless channel capacity, and thus the bottleneck always appears in the wireless mesh network. Under this assumption, the virtual edges could be regarded as having unlimited capacity. Note that all the virtual links do not interfere with any of the wireless transmissions.

B. Solution Overview

The performance of a multi-radio multi-channel wireless mesh network critically depends on the design of three interdependent components: scheduling, channel assignment, and routing. Their joint design has been studied in several existing works [5], [6]. In this paper, we adopt the same approach as in [5] which formulates this problem as an integer linear programming problem. To solve this problem, [5] first solves its LP (linear programming) relaxation and derives the routing solution based on the necessary conditions of channel assignment and schedulability. Then the channel assignment and post processing algorithms are designed to adjust the flows to yield a feasible solution.

We assume that the system operates synchronously in a time-slotted mode. The result we obtain will provide an upper bound for systems using IEEE 802.11 MAC. We further assume that the traffic between a local access point and
the Internet could be infinitesimally divided and routed over multiple paths to multiple gateways achieving the optimal load balancing and the least congestion.

Formally, let \( y_e(c) \) be the flow rate on edge \( e(c) \in E_c \), \( y \) be the link flow vector, \( \rho_e(c) = \frac{y_e(c)}{P_{\phi_e(c)}} \) be the utilization of channel \( c \) over link \( e \), and \( E(v) \) be the set of links that is adjacent to node \( v \). Based on the results presented in [5], the necessary conditions of channel assignment and scheduling are summarized in the following claim:

**Claim 1 (Necessary Condition of Channel Assignment and Schedulability).** For the multi-channel, multi-radio wireless mesh network, if a given link flow vector \( y \) does not satisfy the following inequalities:

\[
\sum_{e' \in I_e(c)} \rho_{e'}(c) \leq \gamma(\Delta); \forall e(c) \in E_c \tag{1}
\]

\[
\sum_{e \in C} \sum_{e' \in E(v)} \rho_e(c) \leq \kappa(v); \forall v \in V \tag{2}
\]

then \( y \) is not schedulable.

In particular, Inequality (1) is the congestion constraint over an individual channel. \( \gamma(\Delta) \) is a constant that only depends on the interference model. Inequality (2) gives the node radio constraint. Recall that a mesh node \( v \in V \) has \( \kappa(v) \) radios, and thus can only support \( \kappa(v) \) simultaneous communications.

The focus of this paper is to investigate the optimal routing scheme under dynamic traffic based on the above necessary conditions of channel assignment and schedulability. Once the flow routes are derived, we simply apply the same method presented in [5] to adjust the flow routes and scale the flow rates to yield a feasible routing and channel assignment.

**III. OPTIMAL ROUTING**

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 virtual node \( w^* \) that connects to gateways 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. 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.

For simplicity, we call the aggregated traffic from a local access point to the Internet a flow and denote it as \( f \in F \), where \( F \) is the set of all aggregated flows. We also denote the rate of an aggregated flow \( f \in F \) as \( x_f \), and use \( x = (x_f, f \in F) \) to represent the aggregated flow rate vector.

**A. Fixed Demand Mesh Network Routing**

We first study the formulation of throughput optimization routing problem in a wireless mesh backbone network under the fixed traffic demand. We use \( d_f \) to denote the demand of flow \( f \) and \( d = (d_f, f \in F) \) 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 \( \lambda \) (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 \( P_f \) to denote the set of unicast paths that connect the source of \( f \) and \( w^* \). Let \( x_f(P) \) be the rate of flow \( f \) over path \( P \in P_f \). Obviously the link flow rate \( y_e(c) \) is given by \( y_e(c) = \sum_{f \in P \in P_f} \cdot k(c) \cdot x_f(P) \), which is the sum of the flow rates that are routed through paths \( P \) passing edge \( e(c) \in E_c \).

Based on the necessary conditions of scheduling and channel assignment in Claim 1 (Eq.(1) and Eq.(2)), we have that

\[
\sum_{e' \in I_e(c)} \frac{1}{\phi_e(c)} \sum_{f \in P \in P_f} \cdot k(c) \cdot x_f(P) \leq \gamma(\Delta); \forall e(c) \in E_c \tag{3}
\]

\[
\sum_{e \in C} \sum_{e' \in E(v)} \frac{1}{\phi_e(c)} \sum_{f \in P \in P_f} \cdot k(c) \cdot x_f(P) \leq \kappa(v); \forall v \in V \tag{4}
\]

To simplify the above equations, we define

\[
A_{e(c)}P = \sum_{e' \in I_e(c)} \cdot k(c) \cdot e' \cdot E \cdot P \cdot \frac{1}{\phi_e(c)} \text{ and } B_{eP} = \sum_{e \in C} \sum_{e' \in E(v), e' \cdot E \cdot P \cdot \phi_e(c)}.\]

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

\[
P_T: \text{ maximize } \lambda \text{ subject to } \sum_{P \in P_f} \sum_{f \in F} x_f(P) \geq \lambda \cdot d_f, \forall f \in F \tag{5}
\]

\[
\sum_{f \in F} \sum_{P \in P_f} x_f(P)A_{e(c)}P \leq \gamma(\Delta), \forall e(c) \in E_c \tag{6}
\]

\[
\sum_{f \in F} \sum_{P \in P_f} x_f(P)B_{eP} \leq \kappa(v), \forall v \in V \tag{7}
\]

\[
\lambda \geq 0, x_f(P) \geq 0, \forall f \in F, \forall P \in P_f \tag{8}
\]

In this problem, the optimization objective is to maximize \( \lambda \) such that at least \( \lambda \cdot d_f \) units of data can be routed for each aggregated flow \( f \) with demand \( d_f \). Inequality (6) enforces fairness by requiring that the comparative ratio of traffic routed for different flows satisfies the comparative ratio of their demands. Inequality (7) and (8) come from the necessary conditions of channel assignment and scheduling. This problem formulation follows the same form as the maximum concurrent flow problem.

Problem \( P_T \) could be solved by a LP-solver such as [10]. To reduce the complexity for practical use, we present a fully polynomial time approximation algorithm for problem \( P_T \), which finds an \( \epsilon \)-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 \( \mu_e \) to each set \( I_e(c) \) for \( e(c) \in E_c \) and a price \( \mu_v \) to each node \( v \in V \).

The objective is to minimize the aggregated price for all inter-
ference sets and all nodes. As the constraint, Inequality (11) requires that the price $\sum_{c(e) \in E_c} \gamma(c(e)) \cdot \mu_e + \sum_{v \in V} B_e \cdot \mu_v$ of any path $P \in \mathcal{P}_f$ for flow $f$ must be at least $\mu_f$, the price of flow $f$. Further, Inequality (12) requires that the weighted flow price $\mu_f$ over its demand $d_f$ must be at least 1.

$D_T : \begin{align*}
& \text{minimize} \quad \sum_{c(e) \in E_c} \gamma(\Delta) \cdot \mu_e + \sum_{v \in V} \kappa(v) \mu_v \quad (10) \\
& \text{subject to} \quad \sum_{c(e) \in E_c} \gamma(c(e)) \cdot \mu_e + \sum_{v \in V} B_e \cdot \mu_v \geq \mu_f, \quad \forall f \in F, \forall P \in \mathcal{P}_f \quad (11) \\
& \sum_{f \in P} \mu_f d_f \geq 1 \quad (12)
\end{align*}$

<table>
<thead>
<tr>
<th>FMR: Mesh Network Routing Under Fixed Demand</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. $\forall v \in E_c, \gamma \leftarrow \gamma(\Delta)$, $\mu_e \leftarrow \beta/\gamma$, $\mu_v \leftarrow \beta/\kappa(v)$</td>
</tr>
<tr>
<td>2. $x_f(P) \leftarrow 0$, $\forall P \in \mathcal{P}_f, \forall f \in F$</td>
</tr>
<tr>
<td>3. while $\sum_{c(e) \in E_c} \gamma(c(e)) \cdot \mu_e + \sum_{v \in V} \kappa(v) \mu_v &lt; 1$ do</td>
</tr>
<tr>
<td>4. for $\forall f \in F$ do</td>
</tr>
<tr>
<td>5. $d_f' \leftarrow d_f$</td>
</tr>
<tr>
<td>6. while $\sum_{c(e) \in E_c} \gamma(c(e)) \cdot \mu_e + \sum_{v \in V} \kappa(v) \mu_v &lt; 1$ and $d_f' &gt; 0$ do</td>
</tr>
<tr>
<td>7. $P \leftarrow$ lowest priced path in $\mathcal{P}_f$ using $\mu_e$ and $\mu_v$</td>
</tr>
<tr>
<td>8. $\delta \leftarrow \min{d_f', \min_{c(e) \in P} \frac{\gamma(c(e))}{\mu_e}, \min_{v \in V} \frac{\kappa(v)}{\mu_v}}</td>
</tr>
<tr>
<td>9. $d_f' \leftarrow d_f' - \delta$</td>
</tr>
<tr>
<td>10. $x_f(P) \leftarrow x_f(P) + \delta$</td>
</tr>
<tr>
<td>11. $\forall e(c) \in E_c$, s.t. $A_e(c) \cdot P \neq 0$, $\mu_e \leftarrow \mu_e (1 + \epsilon B_e \cdot \gamma(c(e))$</td>
</tr>
<tr>
<td>12. $\forall v \in V$, s.t. $B_v \neq 0$, $\mu_v \leftarrow \mu_v (1 + \epsilon \delta B_v / \kappa(v))$</td>
</tr>
<tr>
<td>13. end while</td>
</tr>
<tr>
<td>14. end for</td>
</tr>
<tr>
<td>15. end for</td>
</tr>
</tbody>
</table>

| **Table 1** ROUTING ALGORITHM UNDER FIXED DEMAND |

Based on the above dual problem $D_T$, our fast approximation algorithm is presented in Table 1. The algorithm design follows the idea of [11]. In particular, Line 1 and Line 2 initialize the algorithm. 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 the interference sets and the nodes appeared in this path based on the function defined in Line 11 and Line 12. We keep filling traffic to flow $f$ in the above fashion until all $d_f$ units are routed. This procedure is repeated until the weighted aggregated price of the interference sets and the nodes exceeds 1 (Line 3).

We formally analyze the properties of our algorithm in the following theorem. The proofs of the theorems in this paper are available in the Appendix.

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

### B. 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 $\Pr(D_f = d_f') = q_f'$, where $D_f = \{d_f', d_f^2, \ldots, d_f^n\}$ is the set of of values for $D_f$ with non-zero probabilities. Let $d = (d_f, d_f' \in D_f, f \in F)$ be a sample traffic demand vector, $D$ be the corresponding random variable, and $\Omega$ be the sample space. Thus the distribution of $D$ is given by the joint distribution of these random variables: $\Pr(D = d) = \Pr(D_f = d_f', f \in F)$.

Let us consider a traffic routing solution $(x_f(P), P \in \mathcal{P}_f, f \in F)$ that satisfies the capacity and node-radio constraints (Inequality (7) and (8)). It is obvious that $\beta$ is a function of $\lambda$ and is given by the joint distribution of these random variables: $\Pr(D = d) = \Pr(D_f = d_f', f \in F)$.

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

Obviously, the performance ratio is also a random variable under uncertain demand. We denote it as $\Omega$ which is a function of random variable $D$. Now we extend the wireless mesh network routing problem to handle such uncertain demand. Our goal is to maximize the expected value of $\Omega$, which is given by $E(\Omega) = \Pr(D = d) \times \frac{\lambda (d)}{\lambda (d)}$.

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

$P_U : \begin{align*}
& \text{maximize} \quad \sum_{d \in D} p(d) \frac{\lambda (d)}{\lambda (d)} \\
& \text{subject to} \quad \forall d \in D, where \quad d = (d_f, f \in F) \\
& \sum_{P \in \mathcal{P}_f} x_f(P) \leq \lambda(d) \cdot d_f, \forall f \in F \quad (14) \\
& \sum_{f \in F} \sum_{P \in \mathcal{P}_f} x_f(P) A_e(c) \cdot P \leq \gamma(\Delta), \forall e(c) \in E_c \\
& \sum_{f \in F} \sum_{P \in \mathcal{P}_f} x_f(P) B_{v} \cdot d_f \leq \kappa(v), \forall v \in V \quad (15) \\
& \lambda \geq 0, x_f(P) \geq 0, \forall f \in F, \forall P \in \mathcal{P}_f \quad (16)
\end{align*}$

Similar to problem $P_T$, the constraints of $P_U$ come from the fairness requirement and the wireless mesh network capacity. In particular, Inequality (14) enforces fairness for all demand $d \in D$, and Inequality (15) enforces capacity constraint as Inequality (7) in problem $P_T$. 


Now we consider the dual problem $D_U$ of $P_U$. Similar to $D_T$, the objective of $D_U$ is to minimize the aggregated price for all adjusted interference sets. However, in Inequality (20), for each sample demand vector $d$, the aggregated price of all flows weighted by their demand needs to be larger than its probability.

\[
D_U: \text{ minimize } \sum_{e \in E} \gamma(\Delta) \cdot \mu_e + \sum_{v \in V} \kappa(v) \mu_v \\
\text{ subject to } \sum_{e \in E} A_{e(c)} p \mu_e + \sum_{v \in V} B_v p \mu_v \geq \mu_f, \\
\forall f \in F, \forall P \in \mathcal{P}_f, \\
\sum_{f \in F} \mu_f d_f \geq \frac{p(d)}{\lambda^*(d)}, \forall d \in \mathcal{D} \\
\text{ where } d = (d_f, f \in F)
\]

**UMR: Mesh Network Routing Under Uncertain Demand**

1. $\forall e \in E, \gamma(\Delta), \mu_e = \beta/\gamma, \mu_v = \beta/\kappa(v)$
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. $P \leftarrow$ lowest priced path in $\mathcal{P}_f$ using $\mu_e, \mu_v$
6. $\mu_f \leftarrow \sum_{e \in E} A_{e(c)} p \mu_e + B_v p \mu_v$
7. end for
8. for $\forall d \in \mathcal{D}$ do
9. $\mu_d \leftarrow \sum_{f \in F} \mu_f d_f \lambda^*(d)$
10. end for
11. $\mu_{\min} \leftarrow \min_{d \in \mathcal{D}} \mu_d$
12. $d_{\min} \leftarrow \arg \min_{d \in \mathcal{D}} \mu_{\min}$
13. if $d_{\min} \geq 1$
14. return
15. for $\forall f \in F$ do
16. $d_f^{\prime} \leftarrow d_f^{\min}$
17. while $d_f^{\prime} > 0$ do
18. $P \leftarrow$ lowest priced path in $\mathcal{P}_f$ using $\mu_e, \mu_v$
19. $\gamma \leftarrow \min\{d_f^{\min}, \min_{e \in E} \mathcal{F}_{e(c)} p \mu_e, \min_{v \in V} \mathcal{F}_{v} p \mu_v\}$
20. $d_f^{\prime} \leftarrow d_f^{\prime} - \gamma$
21. $x_f(P) \leftarrow x_f(P) + \gamma$
22. $\forall e \in E$ s.t. $A_{e(c)} p \mu_e \neq 0, \mu_e \leftarrow (1 + \epsilon) A_{e(c)} p \mu_e$
23. $\forall v \in V$ s.t. $B_v p \mu_v \neq 0, \mu_v \leftarrow (1 + \epsilon) B_v p \mu_v$
24. end while
25. end for
26. end loop

**TABLE II**

<table>
<thead>
<tr>
<th>Routing Algorithm Under Uncertain Demand</th>
</tr>
</thead>
<tbody>
<tr>
<td>$D_U$: minimize $\sum_{e \in E} \gamma(\Delta) \cdot \mu_e + \sum_{v \in V} \kappa(v) \mu_v$</td>
</tr>
<tr>
<td>subject to $\sum_{e \in E} A_{e(c)} p \mu_e + \sum_{v \in V} B_v p \mu_v \geq \mu_f$, $\forall f \in F, \forall P \in \mathcal{P}_f$</td>
</tr>
<tr>
<td>$\sum_{f \in F} \mu_f d_f \geq \frac{p(d)}{\lambda^*(d)}, \forall d \in \mathcal{D}$</td>
</tr>
<tr>
<td>where $d = (d_f, f \in F)$</td>
</tr>
</tbody>
</table>

Now we present an approximation algorithm for $P_U$ in Table II. This algorithm (UMR) has the same initialization as the algorithm for problem $P_T$ (FMR). Then we march into the iteration, in which we find $d_{\min}$, 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 (19) and (20) would be satisfied for all demand. Otherwise, we will increase the price of $d_{\min}$ by routing more traffic through its node pairs. This procedure (Lines 16 to 23) 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 $\beta = (|E_c| + |V|)/(1 - \epsilon)$, then the final flow generated by UMR is at least $(1 - 3\epsilon)$ times the optimal value of $P_U$. The running time is $O((\frac{1}{\epsilon^2}) \log (|E_c| + |V|)(2|D||F| \log |F| + |E_c| + |V|) + \log U)) \cdot T_{mp}$, where $U$ is the length of the longest path in $G$, $T_{mp}$ is the running time to find the shortest path.

**IV. 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 are presented in Sec. III-A and Sec. III-B.

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 [8]. By analyzing the sump 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.

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. 1. 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 [13], 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 [7].

![Fig. 1. Incoming Traffic Time Series of ResBldg97AP3 (March 25, 12am, 2002 - June 9, 11pm, 2002 EST).](image-url)
its moving average with \( W = 5 \). By removing the cyclic effect from the raw data, we derive the \textit{adjusted traffic series} \( z(t) \) as \( z(t) = x(t) - \bar{x}(t) \).

The adjusted series of the one shown in Fig. 2(a) is given in Fig. 2(b). This adjusted traffic exhibits short-term (a few hours) traffic correlations. We model the adjusted traffic series with an autoregressive process as follows\(^1\),

\[
z(t) = \beta_1 z(t - 1) + \beta_2 z(t - 2) + \ldots + \beta_K z(t - K) + \epsilon \quad (21)
\]

where \( K \) is the process order. To apply this model for prediction, we estimate the parameters of this process. Given \( N \) observations \( z_1, z_2, \ldots, z_N \), the parameters \( \beta_1, \ldots, \beta_K \) are estimated via least squares by minimizing:

\[
\sum_{t=K+1}^{N} \left[ z(t) - \beta_1 z(t - 1) - \ldots - \beta_K z(t - K) \right]^2 \quad (22)
\]

Based on these parameters, we further derive the adjusted traffic prediction \( \hat{z}(t) \) as \( \hat{z}(t) = \beta_1 z(t - 1) + \beta_2 z(t - 2) + \ldots + \beta_K z(t - K) \). Fig. 3 illustrates the estimation results for the adjusted traffic series in Fig. 2(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 \( \epsilon_x(t) = z(t) - \hat{z}(t) \). Based on this definition, Fig. 4(a) plots the cumulative distribution function of the prediction error of the adjusted traffic series shown in Fig. 3. It is obvious that the error distribution fits the normal distribution with a mean close to zero.

Finally, we define traffic prediction \( \hat{x} \) 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.
nodes are randomly deployed over a $1000 \times 2000 m^2$ region. 20 nodes at the edge of this network are selected as the local access points (LAP) that forward traffic for clients. 4 nodes in the center of the deploy region are selected as the gateway access points. The simulated network topology is shown in Fig. 6. Each mesh node has a transmission range of $250m$ and an interference range of $500m$, which means $\Delta = 2$. The channel capacity $\phi_e(c)$ is the same for all links $e$ and channels $c$, which is set as 54 Mbps. In the basic setting, each mesh node is equipped with 3 radio interfaces. And there are 3 orthogonal channels in the network. Aside from this basic setting, we have also evaluated the performance of our algorithms with different configurations of radio and channel numbers, which we will show in the later part of this section.

In the next hour, but also its distribution. It runs the UMR algorithm (presented in Tab. II) with the predicted traffic demand distribution as its input. Since UMR only accepts discrete probability distribution, we need to discretize the demand distribution by sampling the following values, the mean value $\mu$, 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.

Note that the flows derived from the above routing strategies will be adjusted by the channel assignment, post processing and flow scaling algorithms in [5]. We denote the final rate of flow $f$ along path $P$ as $x_f^A = \sum_{P \in \mathcal{P}_f} x_f^A(P)$. This is the maximum flow throughput under the fairness constraint weighted by the traffic demand, which maximizes the scaling factor $\lambda$. However, for performance study, $\lambda$ is not a suitable performance metric. First, we are more interested in the network performance (i.e., congestion) incurred by the given traffic demand, instead of the achievable throughput. Second, the absolute value of $\lambda$ could be misleading, especially when the actual demand is not the same as the predicted demand which is being used for routing.

Now we proceed to define the performance metric we use in the simulation study. First, we scale the achievable flow rate $x_f^A$ derived from the routing and channel assignment process by its actual traffic demand $d_f$:

$$x_f(P) = x_f^A(P) \cdot \frac{d_f}{x_f^A}$$

$x_f(P)$ is the actual traffic load that is imposed on path $P$ under our routing and channel assignment scheme. Thus the traffic being routed within the interference set $I_e(c)$ over channel $c$ is given by $\sum_{f \in \mathcal{F}} \sum_{P \in \mathcal{P}_f} x_f^A(P) A_e(c) P$. We define the congestion of an interference set $I_e(c)$ using its utilization and denote it as $\theta_e^c(c) = \sum_{f \in \mathcal{F}} \sum_{P \in \mathcal{P}_f} x_f^A(P) A_e(c) P$. Then $\theta_e^c = \max_{e(c) \in E_v} \theta_e^c(c)$ is the maximum congestion among all the interference sets. We further consider the congestion at a single mesh node incurred by the traffic from all channels. The congestion of a node $v$ is defined as $\theta_v^{rd} = \sum_{f \in \mathcal{F}} \sum_{P \in \mathcal{P}_f} x_f^A(P) B_{v,P}$. And $\theta_v^{rd} = \max_{e(v) \in V} \theta_e^{rd}$. Finally, the network congestion $\theta$ is defined as $\theta = \max\{\theta_v^{rd}, \theta_v^{ch}\}$.

B. Simulation Results

We experiment with the above routing strategies along the time range [108, 1108], a 1000-hour period excerpted from the trace. Note that all the simulation results presented in this

\[^2\text{Note that the beginning part of the trace [0, 107] is used as training data, thus is not included in the simulation result.}\]
section are using 108 as the zero point.

![Fig. 7. Overview of All Strategies](image)

We start by presenting the congestion achieved by all strategies (OR, MVPR, SDPR, and SPR) during the entire 1000-hour simulation period. As seen in Fig. 7, 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 $\theta$ 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.

![Fig. 8. (a) Congestion Ratio ($\theta_{OR}$) and (b) Sorted View](image)

To filter out the noise caused by traffic burstiness, in Fig. 8(a), we normalize $\theta$ achieved by other strategies by the same value of OR. Since OR always achieves the minimum $\theta$ among others, this ratio will end up at least 1. Also we take a close-up look during the hour range [190, 290]. Here, the MVPR and SDPR strategies achieve less than 2 times of the optimal congestion in most cases, while the SPR strategy can only achieve 4–7 times of the optimal performance. The above observations get clearer when we sort out the normalized congestion ratio for the three strategies in Fig. 8(b). It is clear that our MVPR and SDPR strategies which integrate the traffic prediction with the optimal routing outperform the SPR strategy which is agnostic about the traffic demand. Further, SDPR achieves lower congestion than MVPR in most of the time due to more comprehensive representation of the traffic demand estimation. However, in a few cases (less than 10% of the time), the worst-case congestion of SDPR is higher than MVPR. This problem can be mostly attributed to the inaccuracy of traffic prediction.

![Fig. 9. Adjusted Interference Set Sorted By Congestion](image)

Next, we take a closer look at each strategy’s ability to balance the traffic within the mesh network. In Fig. 9, we unfold a single time instance at hour 271 and exhibit the congestion $\theta_{ch}$ at each interference set $I(c)$ 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 a balance, which resulted in the best $\theta$ value 0.65. SPR has the highest $\theta$ value as more than 2. The results for MVPR and SDPR are 0.8 and 0.7 respectively. We also observe that the distribution of $\theta_{ch}$ under the SDPR strategy closely matches the OR strategy.

![Fig. 10. Impact of Number of Radio Interfaces](image)

In what follows, we alter our simulation configurations to examine the abilities of different strategies at adapting various network settings, such as radio interface numbers and channel numbers. Here, we focus on the traffic prediction strategies, namely, MVPR and SDPR. Also we plot their performances by the congestion ratio $\theta/\theta_{OR}$ normalized by the OR routing results. We first vary the number of radio interfaces from 2 to 4 and study the congestion $\theta$ during the time interval [190, 290]. Fig. 10 plots the sorted normalized congestion $\frac{\theta}{\theta_{OR}}$ of the two strategies. Comparing these two figures, we could see that the SDPR strategy performs slightly better than the MVPR strategy. The improvement of both strategies over the...
OR strategy increases (i.e., normalized congestion decreases) with the radio number.

Finally, Fig. 11 plots the normalized congestion under different radio and channel numbers at a single time instance 271 for these two strategies. The results show that the improvement of both strategies over the OR strategy decreases with the channel number. This is because when the network has more channels, the algorithms are likely to find more paths and the prediction error is more likely to be magnified.

VI. RELATED WORK

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

The problem of wireless mesh network routing, channel assignment, and the joint solution of these two 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 [2], such as expected packet transmission time [1], channel cost metric (CCM) which is the sum of expected transmission time weighted by the channel utilization [4]. Joint solutions for channel allocation and routing are explored in [14] using a centralized algorithm and in [3] 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 works of [5], [6] 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 [15] presents a rate limiting scheme to enforce the fair-share bandwidth allocation schemes to achieve maximum throughput and lexicographical max-min fairness respectively. Further, the work of [16] 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 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. Distributed algorithms have been 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. For example, [7] statistically characterizes both static flows and roaming flows in a large campus wireless network. 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 [13] in Internet, which also consider 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.

VII. CONCLUSION

This paper studies the optimal routing strategies for wireless mesh networks. Different from existing works which implicitly assume traffic demand as static and known a priori, this work considers the traffic demand uncertainty. It studies the dynamic behavior of wireless network traffic, establishes two prediction models based on time series analysis, and extends the classical maximum concurrent flow problem with statistical demand input. Simulation study is conducted based on the traffic demand from the real wireless network traces. The results show that our problem formulation and algorithm could effectively incorporate the traffic demand dynamics.

VIII. 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 \( P_T \) and \( D_T \), and \( OPT' \) to represent the solution derived from FMR algorithm.

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

Proof: We first make the following denotations. Regarding a set of price assignments \( \mu_e \) for \( e(c) \in E_c \), \( \mu_v \) for \( v \in V \), the objective function of \( D_T \) is \( L'' \). Let \( P''(f) \) be the minimum path of the flow \( f \in F \) using \( \mu_e \) and \( \mu_v \). \( \mu(P''(f)) \triangleq \sum_{e(c) \in E_c} A_e(c)P''(f)\mu_e + \sum_{v \in V} B_vP''(f)\mu_v \) is the aggregated price of \( P''(f) \). 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 \( \mu_{e, (i,j)}^{(o)} \) to denote the price of \( e(c) \in E_c \), \( \mu_{v, (o,j)}^{(i)} \) to denote the price of \( v \in V \) after the \( j \)th iteration...
of the $i$th phase. Regarding $\mu^{(i)(j)}_c$ and $\mu^{(i)(j)}_v$, we simplify the notation $L^{(i)(j)}_c$ into $L^{(i)(j)}_c$, $P^{(i)(j)}_c$ into $P^{(i)(j)}_c$, and $\mu(P^{(i)(j)})$ into $\mu(P^{(i)(j)})$. Based on the price update function (Line 11 in Tab. I), we have

\[
\begin{align*}
L^{(i)(j)}_c & = \sum_{e' \in E_c} \mu^{(i)(j-1)}_c + \epsilon \sum_{e' \in P^{(i)(j-1)}} A_{e'c} P^{(i)(j-1)}_c \mu^{(i)(j-1)}_c \Delta d_{f_j} \\
& + \sum_{v \in V} \mu^{(i)(j)}_v + \epsilon \sum_{v \in P^{(i)(j-1)}} B_{vP^{(i)(j-1)}} \mu^{(i)(j)}_v \Delta d_{f_j} \\
& = L^{(i)(j-1)}_c + \Delta d_{f_j} \mu(P^{(i)(j-1)})
\end{align*}
\]

The price assignment at the start of the $(i+1)$th phase are the same as that at the end of the $i$th phase, i.e., $\mu^{(i+1)(0)} = \mu^{(i)(F)}$. The price of any interference set $e(c)$ is initialized as $\mu^{(1)(0)}_c = \mu^{(0)(F)}$, and the price of any node $v$ is initialized as $\mu^{(1)(0)}_v = \mu^{(0)(F)}_v = \beta/k(v)$. Hence,

\[
L^{(i)(F)} \leq L^{(i)(0)} + \epsilon \sum_{j \geq 1} d(f_j) \mu(F^{(i)(F)})
\]

since $\mu_c$ and $\mu_v$ are monotonically increasing.

Let us define $\mu^{(i)(F)} = \sum_{F \in F} d(f_j) \mu(P^{(i)(F)})$. Then the objective of $D_T$ 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 prices of all inference sets and nodes by $1/\mu^{(i)(F)}$. So $D_T$ is equivalent to finding a set of inference set lengths, such that $\sum_{F \in F} d(f_j)$ is minimized. Thus the optimal value of $D_T$ is $OPT = \min_{\mu^{(i)(F)}} L^{(i)(0)} \mu^{(i)(F)}$.

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

\[
L^{(i)(F)} \leq \frac{L^{(0)(F)}}{1 - \epsilon e^{OPT - 1}}
\]

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

\[
L^{(i)(F)} \leq \frac{\beta(\|E_c\| + |V|)}{(1 - \epsilon e^{OPT - 1})}
\]

\[
= \frac{\beta(\|E_c\| + |V|)}{1 - \epsilon OPT - \epsilon^{OPT - 1}}
\]

\[
\leq \frac{\beta(\|E_c\| + |V|)}{1 - \epsilon OPT}
\]

\[
\leq \beta(\|E_c\| + |V|) e^{\epsilon OPT - 1}
\]

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

\[
1 \leq L^{(t)(F)} \leq \frac{\beta(\|E_c\| + |V|)}{1 - \epsilon e^{\epsilon OPT - 1}}
\]

which implies

\[
OPT \leq \frac{\epsilon}{(1 - \epsilon) \ln \frac{1}{\beta(\|E_c\| + |V|)}}
\]

Now consider an intersection set $e(c)$. For every $\gamma(\Delta)$ units of flow routed through $e(c)$, we increase its price by at least a factor $(1 + \epsilon)$. Initially, its length is $\beta/\gamma(\Delta)$ and after $t - 1$ phases, since $L^{(t)(F)} < 1$, the price of $e(c)$ satisfies $\frac{(1 + \epsilon^{-1})}{\gamma(\Delta)} < 1/\gamma(\Delta)$. Therefore the total amount of flow through $e(c)$ in the first $t - 1$ phases is strictly less than $\log_{1+\epsilon} \frac{1}{\gamma(\Delta)} = \log_{1+\epsilon} 1/\beta$ times its capacity. The same procedure applies for any node $v$. 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 $OPT' = \frac{1}{t-1} \log_{1+\epsilon} 1/\beta$.

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 $\beta = (\|E_c\| + |V|)/(1 - \epsilon)^{-1}$.

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

\[
\frac{OPT}{OPT'} < \log_{1+\epsilon} 1/\beta
\]

Substituting the bound on $OPT/(t - 1)$ from Inequality (25), we get

\[
\frac{OPT}{OPT'} < \frac{\epsilon \log_{1+\epsilon} 1/\beta}{(1 - \epsilon) \ln \frac{1}{\beta(\|E_c\| + |V|)}} = \frac{\epsilon}{(1 - \epsilon) \ln(1 + \epsilon) \ln \frac{1}{\beta(\|E_c\| + |V|)}}
\]

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

\[
\frac{OPT}{OPT'} < \frac{\epsilon}{(1 - \epsilon)^2 \ln(1 + \epsilon)} \leq \frac{1}{(1 - \epsilon)^3} \leq \frac{1}{(1 - 3\epsilon)}
\]

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

Proof: From Inequality (26) and weak-duality, we have

\[
1 \leq \frac{OPT}{OPT'} < \log_{1+\epsilon} 1/\beta
\]

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

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 $\zeta$ be the maximum traffic value of flow $f_i$ when all other flows have zero traffic. Let $\zeta = \min_{F} \frac{\zeta}{\gamma(\Delta)}$. Since at best all single commodity maximum flows can be routed simultaneously, $\zeta$ is an upper bound on $OPT'$. On the other hand, routing $1/|F|$ fraction of each flow of value $\zeta$ 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 $T = \frac{1}{t} \log_{1+\epsilon} \frac{1}{\|E_c\| + |V|}$ 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 ζ = \min_f ζ_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 ζ/|F| is a lower bound on OPT. To ensure that OPT \geq 1, we can scale the original demands so that ζ/|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 \( T = \frac{1}{8} \log_2(1 + \frac{|E| + |V|}{|E| - |V|}) \) 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 1 is \( O\left(\log\left(\frac{|E|_v + |V|}{\epsilon}\right) (2|F| \log|F| + |E_c| + |V|)\right) \cdot T_{mp} \).

**Proof:** The above demand-doubling procedure is repeated for at most \( \log |F| \) times. Thus, the total number of phases is at most \( T \log |F| \). Since each phase contains |F| iterations, the algorithm runs for at most \( |F| T \log |F| \) 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 price of some edge interference set or node by \( 1 + \epsilon \). \( \mu_e \) has initial value \( 1/\gamma(\Delta) \) and value at most \( 1/\gamma(\Delta) \) before the final step of the algorithm. The same condition applies for nodes \( v \in V \). Otherwise, the stop criterion of the algorithm would have been reached. This means that the price of an edge interference set or node can be updated in at most \( \log_2(1 + \frac{E_v + |V|}{E_v - |V|}) \) steps. Therefore, the algorithm contains at most \( \frac{|E|_v + |V|}{\epsilon} \log_2(1 + \frac{E_v + |V|}{E_v - |V|}) \) steps. Since each phase contains \( |F| T \log |F| \) steps, we get \( |F| \log |F| + |E|_v + |V| \) steps. Each step contains a minimum overlap spanning tree operation.

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

**Proof:** Computing \( \zeta_i \) corresponds to the maximum flow problem, where \( f_i \) is the only commodity. The running time of getting \( \zeta_i \) is \( O\left(\frac{|E|_v + |V|}{\epsilon^2} \log U\right) \cdot T_{mp} \), 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 \( d^{(i)} \) as the demand vector chosen at the ith phase.

Based on the price update function (Line 11 in Tab. 1), we have

\[
L^{(i)}(j) = L^{(i-1)} + d(f_i) \mu(P_{(i)(j-1)}) \frac{\lambda^*(d^{(i)})}{p(d^{(i)})}
\]

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., \( \mu_e^{(i+1)(0)} = \mu_e^{(i)(1)(0)} = \mu_e^{(1)(0)} = \mu_e^{(0)(F)^{\ast}} = \beta / k(v) \).

Hence,

\[
L^{(i)(F)} = L^{(i)(0)} + \epsilon \sum_{j=1}^{F} d(f_i) \mu(P_{(i)(j-1)}) \frac{\lambda^*(d^{(i)})}{p(d^{(i)})}
\]

since the edge lengths are monotonically increasing.

Let us define \( \mu^{(i)(F)} = \sum_{j=1}^{F} d(f_i) \mu(P_{(i)(j-1)}) \frac{\lambda^*(d^{(i)})}{p(d^{(i)})} \).

Then the objective of \( D \) is to minimize \( L^{(i)(F)} \), subject to the constraint that \( \mu^{(i)(F)} \geq 1 \), i.e., \( \mu^{(i)(F)} \geq OPT \).

The rest of the proof follows the same as the original proof to **Lemma 1**. The proofs to **Lemma 2**, **3** remain the same. In the proof of **Lemma 4**, the total number of phases is changed from at most \( T \log |F| \) to \( |D| \log |F| \). The proof of **Theorem 2** follows these results.

**REFERENCES**


[10] “Ilog cplex mathematical programming optimizers,”

commodity flow and other fractional packing problems,” in Proc. of

[12] L. Dai, Y. Xue, B. Chang, Y. Cao, and Y. Cui, “Inte-
grating traffic estimation and routing optimization for multi-
radio multi-channel wireless mesh networks,” in Vanderbilt
Technical Report http://eecs.vanderbilt.edu/people/yuanxue/publication-

“Cope: traffic engineering in dynamic networks,” in Proc. of ACM
SIGCOMM, 2006.

ment and routing algorithms for multi-channel wireless mesh networks,”
Mobile Computing and Communications Review, vol. 8, no. 2, pp. 50–

bandwidth allocation in multi-channel wireless mesh networks,” in Proc. of

and fairness in multihop wireless backhaul networks,” in Proc. of ACM

[17] X. Lin and S. Rasool, “A distributed joint channel-assignment, schedul-
ing and routing algorithm for multi-channel ad hoc wireless networks,”