Augmenting predictive with oblivious routing for wireless mesh networks under traffic uncertainty

Jonathan Wellons *, Liang Dai, Yuan Xue, Yui Cui

Department of Electrical Engineering and Computer Science, Vanderbilt University, Nashville, TN 37235-1679, United States

ABSTRACT

Traffic routing is central to the utility and scalability of wireless mesh networks. Many recent routing studies have examined this issue, but generally they have assumed that the demand is constant and given in advance. On the contrary, wireless traffic studies have shown that demand is highly variable and difficult to predict, even when aggregated at access points.

There are several approaches for handling volatile traffic. On one hand, traffic may be modeled in real-time with a dynamic routing based upon forecasted traffic demand. On the other hand, routing can be made with the focus towards maximally unbalanced demand, such that the worst-case performance is contained (known as oblivious routing). The first approach can perform competitively when traffic can be forecasted with accuracy, but may result in unbounded worst-case performance when forecasts go wrong. It is an open question how these two approaches would compare with each other in real networks and if possible at all, whether a benchmark could be defined to guide the selection of the appropriate routing strategy.

To answer the above open question, this paper conducts a systematic comparison study of the two approaches based on the extensive simulation study over a variety of network scenarios with real-world traffic trace. It identifies the key factors of the network topology and traffic profile that affect the performance of each routing strategy. A series of metrics are examined with varying powers of forecasting whether predictive routing or oblivious routing will perform better. Following the guidelines defined by these metrics, we present an adaptive strategy which augments the performance of the predictive routing with the worst-case bound provided by the oblivious routing through adaptive selection of routing strategies based on the degree of traffic uncertainty.

© 2009 Elsevier B.V. All rights reserved.

1. Introduction

Wireless mesh networks (e.g. [1,2]) which now offer a rapid and inexpensive solution to last-mile broadband Internet access, are attracting ever greater attention and widespread deployment. A wireless mesh network is composed of local access points and wireless mesh routers which form an organic backbone structure which forwards traffic between mobile clients and the Internet.

Traffic routing plays a critical role in determining the performance of a wireless mesh network and thus it has attracted extensive recent research. The key challenges come from the scarce wireless channel resource, high dynamic link quality, and the uncertain traffic demands. The proposed approaches address these challenges in different ways. On one end of the spectrum are the heuristic algorithms (e.g. [3–6]). 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 scarce channel resource is shared in an optimal and fair fashion). On the other end of the spectrum, there are theoretical studies that formulate mesh network routing as optimization problems (e.g. [7,8]). 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. Recent studies of wireless network traces [9], however, show that the traffic demand, even being aggregated at access points, is highly dynamic and hard to estimate. Such observations have significantly challenged the practicality of the existing optimization-based routing solutions in wireless mesh networks.

One natural approach to address the traffic uncertainty in network routing is predictive routing [10,11], which infers the traffic demand with maximum possibility based on history and optimizes the routing strategy for the predicted traffic demand. Underlying predictive routing is the assumption that past behavior is a good indicator of the future. The quality of a predictive algorithm is therefore tightly related to the traffic uncertainty and how readily this uncertainty can be modeled.

However, considering the high degree of uncertainty in wireless mesh networks, the following questions are still open issues: (1) how much benefit are we able to gain from the predictive routing; (2) under what circumstance does predictive routing perform competitively; (3) if the traffic variability is too high to predict, what is the worst-case performance of the predictive routing and could it could be controlled.

To answer the first question, this paper contrasts predictive routing with a radically different routing strategy - oblivious routing. Oblivious routing makes no assumptions about traffic demand. Instead, it considers all the possible traffic demands and selects the routing strategy where the worst-case network performance is optimized.

This paper then conducts a systematic comparison study of these two approaches based on the simulation study. To evaluate the performance of these two algorithms under a 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’s campus wireless networks. We develop a set of metrics which measure the unpredictability of the traffic demand – the traffic erraticity – and correlate them with the relative performances of the two algorithms. Our simulation results demonstrate that predictive routing performs better under consistent traffic demand compared to highly variable demand, as determined by our erraticity metrics. Furthermore, oblivious routing, being a stateless routing, is unaffected by erraticity.

This leads us to a discussion of a novel adaptive strategy which augments predictive routing with oblivious routing by automatically selecting the most fruitful strategy, given the erraticity of the traffic. Towards this goal, we formulate, implement and compare five different metrics to develop our intuition. The best performing metric is integrated into the augmented predictive algorithm, which uses the metric as a dynamic, self-adjusting cutoff to react to the algorithm’s own predictive quality. This algorithm compares favorably with either the original predictive or oblivious routing individually.

The original contributions of this work are threefold. First, it conducts detailed simulation studies of predictive and oblivious routing, identifies the key factors that affect the performance of each routing strategy and provides the evidence for the competitive performance of oblivious routing in highly-variable traffic networks. Second, it introduces a set of new erraticity metrics (in addition to one we briefly presented in [12]) to characterize the variability of traffic and examines their capabilities at inferring the relative performance of predictive routing and oblivious routing. Finally, it presents an augmented predictive algorithm, which reacts to the erraticity metrics by choosing the appropriate algorithm to employ. The algorithm gains the benefits of both oblivious and predictive routing and is shown to have the best performance of the three algorithms compared.

The remainder of this paper is organized as follows. Section 2 presents the network and system model. Section 3 reviews the optimal mesh network routing strategy. Section 4 describes the traffic prediction algorithm, and shows how the optimal framework could be integrated into predictive mesh network strategy. Section 5 presents the oblivious mesh network routing formulation and algorithm. Section 6 presents our preliminary performance evaluation. Section 7 describes the erraticity metrics, augmented predictive algorithm and its performance. Finally, Section 8 discusses the related work and Section 9 concludes the paper.

2. Model

2.1. Network and interference model

In a multi-hop wireless mesh network, local access points aggregate and forward traffic for the mobile clients which are associated with them. They communicate with each other and with the stationary wireless routers to form a multi-hop backbone network, which forwards the user traffic to the Internet gateways. We use \( w \in W \) to denote the set of gateways in the network and \( s \in S \) to denote the set of local access points that generate traffic in the network. Local access points, gateways and mesh routers are collectively called mesh nodes and denoted by the set \( V \).

In a wireless network, packet transmissions are subject to location-dependent interference. Here we consider the protocol model presented in [13]. We assume that all mesh nodes have the uniform transmission range denoted by \( R_t \). Usually the interference range is larger than the transmission range, which is denoted as \( R_i = (1 + \Delta)R_t \), where \( \Delta \geq 0 \) is a constant. For simplicity, in this paper we assume that each node is equipped with one radio interface which operates on the same wireless channel as the others. Let \( r(u, v) \) be the distance between two nodes \( u \) and \( v \in V \). 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 \( r(u, v) \leq R_u \); (2) any other node \( x \in \mathcal{V} \) within the interference range of the receiving node \( v \), i.e., \( r(x, v) \leq R_v \), is not transmitting. If node \( u \) can transmit to \( v \) directly, they form an edge \( e = (u, v) \). We assume that the maximum data rate that can be transmitted along an edge is the same for all edges, and denote it as \( c \) (also called channel capacity). Let \( E \) be the set of all edges. We say two edges \( e, e' \) interfere with each other, if they cannot transmit simultaneously based on the protocol model. Further we define interference set \( I(e) \) which contains the edges that interfere with edge \( e \) and \( e' \) itself.

Finally, we introduce a virtual node \( \mathcal{V} \) to represent the Internet. \( \mathcal{V} \) is connected to each gateway with a virtual edge \( e' = (\mathcal{V}, w), w \in \mathcal{W} \). Further, let \( E = E \cup \{e'\} \) and \( V = V \cup \{\mathcal{V}\} \). For simplicity, we assume that the link capacity in the 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, and they do not interfere with any of the wireless transmissions.

### 2.2. Traffic model and schedulability

This paper studies the routing strategies for wireless mesh backbone networks. Thus it only considers the aggregated traffic between the local access points and the Internet gateways. Here we call the aggregated traffic in (or out) a local access point a flow and denote it as \( f \in \mathcal{F} \), where \( \mathcal{F} \) is the set of all aggregated flows. All flows will take \( \mathcal{V} \) as their source (or destination). We denote the traffic demand of flow \( f \) as \( d_f \) and use vector \( d = (d_f, f \in \mathcal{F}) \) to denote the demand vector consisting of all flow demands.

Now we proceed to study the flow rate constraint. Let \( y = (y(e), e \in E) \) denote the edge rate vector, where \( y(e) \) is the aggregated flow rate along edge \( e \). Edge rate vector \( y \) 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 edge rate vector \( y \).

The edge rate schedulability problem has been studied in several existing works, which lead to different models [14–16]. In this paper, we adopt the model in [15], which is also extended in [7] for a multi-radio, multi-channel mesh network. In particular, [15] presents a sufficient condition under which an edge scheduling algorithm is given to achieve stability with bounded and fast approximation of an ideal schedule. [7] presents a scheme that can adjust the flow routes and scale the flow rates to yield a feasible routing and channel assignment. Based on these results, we have the following claim as a sufficient condition for schedulability.

**Claim 1 (Sufficient condition of schedulability).** The edge rate vector \( y \) is schedulable if the following condition is satisfied:

\[
\forall e \in E, \sum_{e' \in \partial(e)} y(e') \leq c. \tag{1}
\]

### 3. Optimal mesh network routing

This section presents the problem formulation of optimal mesh network routing under fixed traffic demand. The existing works on optimal multihop wireless network routing [7,8,14] usually formulate it as a throughput optimization problem which maximizes the flow throughput, while satisfying the fairness constraints. In this formulation, traffic demand is fixed and reflected as the flow weight in the fairness constraints. Recall that \( f \in \mathcal{F} \) is the aggregated traffic flow between the local access points and the virtual gateway (i.e., the Internet) and \( d = (d_f, f \in \mathcal{F}) \) is 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 \( d_f \). The goal of throughput maximization routing is to maximize \( \lambda \) (the scaling factor) where at least \( \lambda \cdot d_f \) amount of throughput can be routed per flow \( f \).

To balance the traffic load, flow \( f \) could be routed over multiple paths, let \( \mathcal{P} \) be the set of unicast paths that could route flow \( f \), and \( x_f(P) \) be the rate of flow \( f \) over path \( P \in \mathcal{P} \). Obviously the aggregated flow rate \( y \), along edge \( e \in \mathcal{E} \) is given by \( y = \sum_{e \in \mathcal{E}} x_f(P) \), which is the sum of the flow rates that are routed through paths \( \mathcal{P} \) passing edge \( e \). Based on the sufficient condition of schedulability in Claim 1 (Eq. (1)), we have that

\[
\sum_{e \in \partial V, f \in \mathcal{F}, e \in \mathcal{P}} x_f(P) \leq c. \tag{2}
\]

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

\[
P_f: \begin{align*}
\text{maximize} & \quad \lambda \\
\text{subject to} & \quad \sum_{P \in \mathcal{P}} x_f(P) \geq \lambda \cdot d_f, \quad \forall f \in \mathcal{F}, \\
& \quad \sum_{f \in \mathcal{F}} x_f(P) \leq A_{\mathcal{P}}, \quad \forall e \in \mathcal{E}, \\
& \quad \lambda \geq 0, \quad x_f(P) \geq 0, \quad \forall f \in \mathcal{F}, \forall P \in \mathcal{P}.
\end{align*} \tag{3}
\]

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 (3) enforces fairness by requiring that the comparative ratio of traffic routed for different flows satisfies the comparative ratio of their demands. Inequality (4) enforces the capacity constraint by requiring the traffic aggregation of all flows passing wireless link \( e \in \mathcal{E} \) satisfy the sufficient condition of schedulability. This problem formulation follows the classical maximum concurrent flow problem.

While the above throughput maximization routing problem formulation is widely used in designing optimal mesh network routing strategies under known demands, it is not suitable to study the routing performance under dynamic and uncertain traffic demand. Here we consider a formulation based on another routing performance metric – network congestion (or utilization). In the Internet, link
utilization is commonly used for traffic engineering [17], whose objective is to minimize the utilization at the most congested link under a given traffic demand. However, link utilization cannot be straightforwardly applied to multi-hop wireless networks, such as mesh backbone network, as a metric of routing performance due to the location-dependent interference. In what follows, we define the network congestion based on the utilization of the interference set as the routing performance metric and outline the relation between the formulation of the throughput optimization problem and the congestion minimization problem.

Let \( x_f(P) \) be the rate of flow \( f \) on path \( P \) under traffic demand \( d_f \). It is obvious that \( \sum_{P \in F_f} x_f(P) = d_f \). The traffic being routed within the interference set \( I_e \) is then given by \( \sum_{f \in F} \sum_{P \in F_f} x_f(P)A_{eP} \). Formally, the congestion of an interference set \( I_e \) is defined as its utilization (i.e., the ratio between its load and the channel capacity) and denoted as \( \theta_e \):

\[
\theta_e = \frac{\sum_{f \in F} \sum_{P \in F_f} x_f(P)A_{eP}}{C_e}.
\]

Further, the network congestion is defined as the maximum congestion among all the interference sets, i.e.,

\[
\theta = \max_{e \in E} \theta_e.
\]

The network congestion minimization routing problem is then formulated as follows:

\[
P_C : \begin{align*}
\text{minimize} & \quad \theta \\
\text{subject to} & \quad \sum_{P \in F_f} x_f(P) \geq d_f, \quad \forall f \in F, \\
& \quad \sum_{f \in F} \sum_{P \in F_f} x_f(P)A_{eP} \leq c \cdot \theta_e, \quad \forall e \in E,
\end{align*}
\]

\[
\theta \geq 0, \quad x_f(P) \geq 0, \quad \forall f \in F, \quad \forall P \in F_f.
\]

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

\[
P_C : \begin{align*}
\text{minimize} & \quad \frac{1}{\lambda} \\
\text{subject to} & \quad \frac{1}{\lambda} \sum_{P \in F_f} x_f(P) \geq d_f, \quad \forall f \in F, \\
& \quad \frac{1}{\lambda} \sum_{f \in F} \sum_{P \in F_f} x_f(P)A_{eP} \leq c \cdot \theta, \quad \forall e \in E,
\end{align*}
\]

\[
\lambda \geq 0, \quad x_f(P) \geq 0, \quad \forall f \in F, \quad \forall P \in F_f.
\]

which is obviously equivalent to the throughput optimization problem \( P_T \).

If the demand vector \( d \) is known, both problem \( P_T \) and \( P_C \) could be solved by a LP-solver such as [18,19]. To reduce the complexity for practical use, the work of [10] also presents a fully-polynomial time approximation algorithm for problem \( P_T \), which finds an \( \epsilon \)-approximate solution.

4. Predictive mesh network routing

The fixed-demand optimal routing strategy presented in Section 3 needs the traffic demands from local access points as inputs. When the demand inputs \( d \) reflect the real network traffic, the computed routing results will achieve optimal network performance (i.e., minimize the network congestion). Under dynamic traffic conditions, however, the traffic demand vector \( d \) usually cannot be known a priori. To address this problem, the predictive mesh network routing needs to estimate the traffic demand based on the analysis of its dynamic behavior. This section first presents several traffic estimation methods that are used in our comparison study. Then it discusses how to handle the estimation uncertainty in routing optimization.

4.1. Traffic prediction

To illustrate the traffic prediction method, we use the traffic traces (snmp log) collected at the campus wireless LAN network of Dartmouth College in Spring 2002 [20]. It is argued 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. Here we choose one of the access points (ResBldg97AP3) as an example, whose incoming traffic series is plotted in Fig. 1. We call this traffic series the raw traffic series and use \( x(t) \) to denote it. The predicted traffic series will be denoted as \( \hat{x}(t) \) (t in hours).

4.1.1. Weighted average of recent hours

One intuitive way to predict the future traffic demand is to summarize the most recent traffic demand history. This could be achieved by taking a weighted average of the traffic demand in recent hours. Formally, the traffic estimate \( \hat{x} \) can be represented as follows:

\[
\hat{x}(t) = \sum_{i=1}^{K} \alpha x(t - i),
\]

which is obviously equivalent to the throughput optimization problem \( P_T \).

![Fig. 1. Incoming Traffic Time Series of ResBldg97AP3 (March 25, 12 a.m., 2002 – June 9, 11 p.m., 2002 EST).](image-url)
where \( z_i \) is the weight factor that satisfies \( \sum_{i=1}^{K} z_i = 1 \) and \( K \) is the history window size. This method characterizes the short-term dynamics in the traffic demand and adapts agilely to the dynamic traffic demand. However, it does not characterize the hourly difference in the traffic demand. For example, the traffic demand observed at 3 p.m. is usually different from the one observed at 6 p.m. on the same day due to the work schedule. On the other hand, we may expect similar traffic demand at 6 p.m. on consecutive days. One way to characterize this intuition is to look at the long-term dependency in the traffic demand.

### 4.1.2. Weighted average of the same hour of the day

This method characterizes the long-term dependency in the traffic series by taking a weighted average of the same hours of previous days. Formally, \( \hat{x}(t) \) is computed as follows:

\[
\hat{x}(t) = \sum_{j=1}^{W} \delta_j x(t - j \times 24),
\]

where \( \delta_j \) is the weight factor that satisfies \( \sum_{j=1}^{W} \delta_j = 1 \) and \( W \) is the history window size. Since traffic will exhibit different behavior over weekend days and work days, this method can also be refined to differentiate weekend/work days.

### 4.1.3. Time-series analysis

The time-series-based prediction method integrates the above intuitive considerations together and determines the parameters in a precise way \([11,21]\). In what follows, we briefly review this prediction method. The first step of the analysis is to identify and remove the daily and weekly cyclic patterns in the time series by taking the moving average of this series based on the same hour of the day:

\[
\hat{x}(t) = \frac{\sum_{t=1}^{W} x(t - j \times 24)}{W},
\]

where \( W \) is the size of the moving window. After removing the cyclic effect from the raw data, the adjusted traffic series \( z(t) \) is derived as \( z(t) = x(t) - \hat{x}(t) \). The adjusted traffic series contains the short-term (a few hours) traffic correlations which can be modeled as an autoregressive process.

\[
z(t) = \beta_1 z(t - 1) + \beta_2 z(t - 2) + \cdots + \beta_K z(t - K) + \epsilon,
\]

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

\[
\sum_{t=K+1}^{N} [z(t) - \beta_1 z(t - 1) - \cdots - \beta_K z(t - K)]^2.
\]

Based on the estimated parameters, the adjusted traffic prediction \( \hat{z}(t) \) is given as

\[
\hat{z}(t) = \beta_1 z(t - 1) + \beta_2 z(t - 2) + \cdots + \beta_K z(t - K).
\]  

Fig. 2 plots the predicted series for the adjusted traffic, in comparison with its raw data and the fitted traffic series over the interval [720,779].

The traffic demand \( \hat{x}(t) \) is then predicted as follows:

\[
\hat{x}(t) = [x(t) + \hat{z}(t)]^+, \quad \text{where } [x]^+ = \max(0,x).
\]

The predictive routing strategy uses the predicted traffic demand \( \hat{x}(t) \) as the input to the fixed-demand mesh network routing problem \( P_f \) and derives the routing solution.

### 4.2. Handling prediction uncertainty

We now consider the errors involved in the above prediction process. In particular, in the time-series-based approach, we define the prediction error \( \epsilon(t) \) as

\[
\epsilon(t) = x(t) - \hat{x}(t).
\]

Its cumulative distribution function is plotted in Fig. 3. It clearly shows that this distribution fits the normal distribution with a near-zero mean.

Thus 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 \( \epsilon \).

Let us denote the uncertain traffic demand of an aggregated flow \( f \in F \) using a random variable \( D_f \). Since the optimal mesh network routing problem \( P_f \) can only take fixed demands as input, we need to extend it in order to handle the uncertain demand. To make the problem tractable, we first discretize the demand distribution by sampling the following values of the normal random variable \( D_f \), 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 have a probability smaller than 5%. Let \( Pr(D_f = d_f) = q_f \) be the discrete probability distribution for random variable \( D_f \) and \( \mathcal{D}_f = \{d_{f1}, d_{f2}, \ldots, d_{fn}\} \) be the set of values for \( D_f \) with non-zero probabilities. Let \( d = (d_f, d_i \in \mathcal{D}_f, f \in F) \) be a sample traffic demand vector, \( D \) be the corresponding random variable, and \( \mathcal{X} \) be the sample space. Thus the distribution of \( D \) is given by the

![Fig. 2. Adjusted traffic and its prediction.](image-url)
joint distribution of these random variables: \( Pr(D = d) = Pr(D = d, f \in F) \). We abbreviate \( Pr(D = d) \) as \( p(d) \). It is clear that \( \sum_{d \in \mathcal{D}} p(d) = 1 \).

It is obvious that \( \lambda \) is a function of \( d \),\( \lambda(d) = \min_{i \in F} \{ x_i \} \), where \( x_i = \sum_{P \in \mathcal{P}, \mathcal{I}(P)} \mathcal{I}(P) \). Let us denote the optimal value of \( \lambda \) as \( \lambda(d) \). We define the throughput performance ratio \( \omega \) of a routing solution \( \mathcal{X}(P), P \in \mathcal{P}, f \in F \) as \( \omega(d) = \frac{\lambda(d)}{\frac{c}{e} d_f} \).

Obviously, the throughput performance ratio is also a random variable under uncertain demand. We denote it as \( \Omega \). Now we extend the wireless mesh network routing problem to handle such uncertain demand. Our goal is to maximize the expected value of \( \lambda \), which finds an

\[
P_0: \quad \text{maximize } \sum_{d \in \mathcal{D}} p(d) \frac{\lambda(d)}{\frac{c}{e} d_f} \\
\text{subject to } \forall d \in \mathcal{D}, \text{ where } d = (d_f, f \in F), \text{ (20)}
\]

\[
\sum_{P \in \mathcal{P}} x_i(P) \geq \lambda(d) d_f, \forall f \in F, \text{ (21)}
\]

\[
\sum_{f \in F} \sum_{P \in \mathcal{P}} x_i(P) A_{ep} \leq c, \forall e \in E, \text{ (22)}
\]

\[
\lambda \geq 0, \quad x_i(P) \geq 0, \forall f \in F, \forall P \in \mathcal{P}. \text{ (23)}
\]

Problem \( P_0 \) optimizes the expected performance of the network under uncertain traffic demand. Our work of [10] presents a fast approximation algorithm for this problem, which finds an \( \epsilon \)-approximate solution. Using the estimated traffic demand distribution as the input to problem \( P_0 \), the predictive routing strategy could derive a routing solution that considers the traffic uncertainty and prediction errors.

5. Oblivious mesh network routing

In contrast to the predictive routing which establishes traffic models based on time-series analysis and optimizes towards the traffic demands with maximum possibility, oblivious routing makes no assumptions on the traffic model. Rather it considers all traffic demand possibilities and optimizes towards the worst-case scenario. To formally study oblivious routing strategy, we need a performance metric that could characterize the worst-case congestion under all possible traffic demand.

First, let’s examine the formal description of routing, which specifies how traffic of each flow is distributed across the network. In the previous formulation (\( P_C \)), a routing is characterized through the traffic load distribution along different paths \( (i.e., x_i(P)) \). This description of a routing depends on the traffic demand of each flow. When we have to consider all possible traffic demands, it becomes infeasible. In fact, a routing strategy could be modeled independently of the traffic demand, which is the core of the oblivious routing problem formulation.

Formally, we define a routing by the fraction of each flow that is routed along each edge \( e \in E \). We use \( \phi_f(e) \) to denote the fraction of demand of flow \( f \) that is routed on the edge \( e \in E \). Thus, a routing could be specified by the set \( \phi = \{ \phi_f(e), f \in F, e \in E \} \). Recall that the demand of flow \( f \in F \) is denoted by \( d_f \). Therefore, the amount of traffic demand of \( f \) that needs to be routed over \( e \) in routing \( \phi \), denoted by \( y_f(e) \), is given as follows:

\[
y_f(e) = d_f \phi_f(e). \text{ (24)}
\]

Thus the congestion \( \theta_e \) of an interference set \( I(e) \) is given by

\[
\theta_e = \sum_{e \in I(e)} \sum_{f \in F} \frac{y_f(e)}{c} = \sum_{e \in I(e)} \sum_{f \in F} \frac{d_f \cdot \phi_f(e)}{c}. \text{ (25)}
\]

We further use \( \theta(\phi, d) = \max_{e \in I(e)} \theta_e(\phi, d) \) to denote the network congestion under a certain routing \( \phi \) and traffic demand vector \( d \).

Now we proceed to study the performance metric that could characterize a “good” routing solution under all
possible traffic demands. We start with the \textit{optimal routing} \( \phi_{\text{opt}} (d) \) for a certain demand vector \( d \), which would give the minimum congestion under this demand, \( i.e., \)
\[ \theta_{\text{opt}} (d) = \min_{\phi} \theta (\phi, d). \]  

(26)

Now we define the \textit{congestion performance ratio} \( \gamma (\phi, d) \) of a given routing \( \phi \) on a given demand vector \( d \) as the ratio between the network congestion under the routing \( \phi \) and the minimum congestion under the optimal routing, \( i.e., \)
\[ \gamma (\phi, d) = \frac{\theta (\phi, d)}{\theta_{\text{opt}} (d)}. \]  

(27)

Performance ratio \( \gamma \) measures how far \( \phi \) is from being optimal on the demand \( d \). Now we extend the definition of performance ratio to handle uncertain traffic demand. Let \( D \) be a set of traffic demand vectors. Then the performance ratio of a routing \( \phi \) on \( D \) is defined as the worst-case performance ratio for all demands in \( D \), \( i.e., \)
\[ \gamma (\phi, D) = \max_{d \in D} \gamma (\phi, d). \]  

(28)

A routing \( \phi_{\text{opt}} \) is optimal for the traffic demand set \( D \) if and only if
\[ \phi_{\text{opt}} = \arg \min_{\phi} \gamma (\phi, D), \]  

(29)

which means \( \phi_{\text{opt}} \) minimizes the performance ratio under the worst-case scenario. When the set \( D \) includes all possible demand vectors \( d \), we refer to the performance ratio as the \textit{oblivious performance ratio}. The oblivious performance ratio is the worst performance ratio a routing obtains with respect to all possible demand vectors. To study the optimal routing strategy under uncertain traffic demand, we are interested in the \textit{optimal oblivious routing} problem which finds the routing that minimizes the oblivious performance ratio. We call this minimum value the \textit{optimal oblivious performance ratio}.

It is worth noting that the performance ratio \( \gamma \) is invariant to scaling. Thus to simplify the problem, we only consider traffic demand vectors \( D \) that satisfy \( \theta_{\text{opt}} (d) = 1 \), instead of considering all possible traffic vectors. In this case,
\[ \gamma (\phi, D) = \max_{d \in D} \theta (\phi, d). \]  

(30)

Thus the goal of oblivious routing is given by
\[ \min_{\phi} \max \theta (\phi, d), \]  

(31)

Traffic into and out of a mesh node must be conserved. In \( P_c \), a \textit{path representation} of the routing is being used \((X_f(P))\), which implicitly formulates the flow conservation. Here, because we use an edge representation of the routing \((\phi_f(e))\), the flow conservation has to be explicitly formulated. In particular, for the node \( v \in V' \) that only relays for flow \( f \) (\( i.e., \) neither source or destination), we have the following relations:
\[ \forall f \in F, \sum_{e \in (u,v)} y_f^v (e) = \sum_{e \in (v,u)} y_f^v (e) = 0 \]  

(32)

if \( v \) is a relay of \( f \).

If \( v \) is the source node of flow \( f \), then we have
\[ \forall f \in F, \sum_{e \in (u,v)} y_f^v (e) = - d_f \]  

(33)

if \( v \) is the source node of \( f \).  

Summarizing the above discussion, the \textit{oblivious mesh network routing} problem is formulated as follows.

\begin{align*}
\textbf{P_0 :} & \quad \text{minimize } \theta \\
& \text{subject to } \sum_{e \in (u,v)} y_f^v (e) - \sum_{e \in (v,u)} y_f^v (e) = d_f, \\
& \quad \forall f \in F, \forall v \in V', \text{ if } v \text{ is a relay of } f, \\
& \quad \sum_{e \in (u,v)} y_f^v (e) - \sum_{e \in (v,u)} y_f^v (e) = 0, \\
& \quad \forall f \in F, \forall v \in V', \text{ if } v \text{ is the source node of } f, \\
& \quad \forall f \in F, \forall v \in V', y_f^v (e) = d_f \cdot \phi_f (e) \geq 0, \\
& \quad 0 \geq 0, \quad \forall d \text{ with } \theta_{\text{opt}} (d) = 1. \\
\end{align*}

(34)

Note that the flow conservation is relaxed to \( \leq - d_f \) to allow more flow than demanded. Different from \( P_c \), the oblivious mesh routing problem \( P_0 \) cannot be solved directly, because it is taken over all demand vectors, and \( \theta_{\text{opt}} (d) \) is an embedded maximization in the minimization problem.

Here we use a similar method as in [22], which provides a \textit{LP formulation} of the oblivious routing problem. The key insight is to look at the dual problem of the slave LPs of the original oblivious routing problem. Given a routing \( \phi_f (e) \), the constraints (34) can be tested by solving, for each interference set \( I(e) \), the following \"slave LP\", and testing if the objective is \( \leq 0 \) or not.

\[ \max \sum_{e \in I(e)} \sum_{f \in \phi} d_f \cdot \phi_f (e) \]  

subject to \( \phi_f (e) \) is a routing; constraints (35) and (36).

(38)

In the dual formulation, we first introduce interference set weights \( \pi_e (e') \) for every pair of interference sets \( e, e' \). Each \( \pi \) variable can be thought of as a dual multiplier on the capacity constraint. There are three essential properties shown in \textit{Theorem 1}.

\textbf{Theorem 1.} A routing \( \phi \) has oblivious ratio \( \leq 0 \) if and only if there exist weights \( \pi_e (e') \), for every pair of interference set \( I(e), I'(e'), e, e' \in E \), such that

\begin{align*}
P_1 & \quad \forall e, e' \in E, \sum_{e \in I(e)} \pi_e (e') \leq 0; \\
P_2 & \quad \forall \text{paths } h, \forall f \in F, \forall e \in E \\
& \sum_{e \in I(e)} \phi_f (e') \leq c \cdot \sum_{a \in E} \pi_e (a) |I(a) \cap h|; \\
P_3 & \quad \forall \text{interference sets } I(e), I'(e), \pi_e (e') \geq 0.
\end{align*}
Proof.  The proof applies duality to the slave problem. Requirements P1–P3 are equivalent to stating that the slave LP's have dual objective values $\leq 0$.

*if* direction: Let $\phi$ be a routing, and $\pi_{L}(e')$ satisfy requirements P1–P3. Let $y$ be a flow of demand $d$ with maximum utilization of 1, and $y_{j}(h)$ be the amount of flow $f$ routed on path $h$. From P2, we have that

$$\forall h, \sum_{e' \in I(h)} \phi_{y}(e') \cdot y_{j}(h) \leq c \cdot \sum_{a \in E} \pi_{L}(a) |l(a) \cap h| \cdot y_{j}(h). \quad (39)$$

Summing over all $h$ for flow $f$, we have

$$\sum_{e' \in I(a)} \phi_{y}(e') \sum_{h} y_{j}(h) \leq c \cdot \sum_{a \in E} \pi_{L}(a) |l(a) \cap h| \cdot y_{j}(h). \quad (40)$$

Since $\sum_{h} y_{j}(h) = d_{f}$, we have that

$$\sum_{e' \in I(a)} \phi_{y}(e') \cdot d_{f} \leq c \cdot \sum_{a \in E} \pi_{L}(a) \sum_{h} |l(a) \cap h| \cdot y_{j}(h). \quad (41)$$

By interchanging the order of the summations, we have

$$\sum_{e' \in I(a)} \phi_{y}(e') \cdot d_{f} \leq c \cdot \sum_{a \in E} \pi_{L}(a) \sum_{h} \sum_{e' \in I(h)} y_{j}(h). \quad (42)$$

Let $y_{j}(a')$ be the amount of flow $f$ routed over edge $a'$. Then $\sum_{e' \in I(h)} y_{j}(h)$ is the amount of flow routed in interference set $l(a)$. We also have

$$\forall a, \sum_{h} |l(a) \cap h| \cdot y_{j}(h) = \sum_{h} y_{j}(a'). \quad (43)$$

Thus

$$\sum_{e' \in I(a)} \phi_{y}(e') \cdot d_{f} \leq c \cdot \sum_{a \in E} \pi_{L}(a) \sum_{a' \in I(a)} y_{j}(a'). \quad (44)$$

Summing over all $h$, we have

$$\sum_{f} \sum_{e' \in I(e)} \phi_{y}(e') \cdot d_{f} \leq c \sum_{a \in E} \pi_{L}(a) \sum_{a' \in I(a)} y_{j}(a') \quad (45)$$

$$= c \cdot \sum_{a \in E} \pi_{L}(a) \sum_{f} \sum_{a' \in I(a)} y_{j}(a') \quad (46)$$

$$\leq c \cdot \sum_{a \in E} \pi_{L}(a) \cdot c. \quad (47)$$

The last inequality follows because $\sum_{a' \in I(a)} y_{j}(a') \leq c$ from Eq. (38) in the slave problem. Finally by property P1, we have $\sum_{a \in E} \pi_{L}(a) \cdot c \leq 0$. Thus

$$\sum_{f} \sum_{e' \in I(e)} \phi_{y}(e') \cdot d_{f} \leq c \cdot 0. \quad (48)$$

This states that the congestion on any interference set $l(e)$ is at most $\theta$ for demand vector $d$ which can be routed with maximum congestion 1.

*only if* direction: Let a routing $\phi$ have oblivious ratio $\leq \theta$. The dual of the slave LP for an interference set $l(e)$ is

$$\min \sum_{e' \in E} c \cdot \pi_{L}(e'), \quad \forall u, v \in V \setminus \{u \neq v\} : \quad \zeta_{L}(u, v) \geq \frac{\sum_{h \in I(a)} \phi_{y}(e')}{c}, \quad (49)$$

$$\forall u, \forall a = (u', v') : \quad \sum_{a' \in I(a)} \pi_{L}(a') + \zeta_{L}(u, u') - \zeta_{L}(u, v') \geq 0, \quad (50)$$

\begin{equation}
\forall a \in E : \quad \pi_{L}(a) \geq 0. \quad (51)
\end{equation}

\begin{equation}
\forall u, w \in V \setminus \{u \neq w\} : \quad \zeta_{L}(u, w) \geq 0, \quad (52)
\end{equation}

\begin{equation}
\forall u \in V \setminus \{u \neq u\} : \quad \zeta_{L}(u, u) = 0. \quad (53)
\end{equation}

Because $\phi$ has oblivious ratio $\leq \theta$, which means the primal slave LP for any interference set $l(e)$ must have optimum $\leq \theta$, and thus the dual slave LP for interference set $l(e)$ must have optimum $\leq \theta$. Hence the $\phi_{y}(e')$ must satisfy P1. Obviously, they also satisfy property P3. Now consider flow $f : u \rightarrow v$ and path $h = (a_{1}, a_{2}, \ldots, a_{k})$ from $u$ to $v$. Summing up constraint (50) over all edges $a_{1}, a_{2}, \ldots, a_{k}$, we have

$$\sum_{i=1}^{k} \sum_{a' \in I(a_{i})} \pi_{L}(a') + \zeta_{L}(u, u) - \zeta_{L}(u, v) \geq 0, \quad (54)$$

which gives

$$\sum_{a \in E} |l(a) \cap h| \pi_{L}(a) + \zeta_{L}(u, u) - \zeta_{L}(u, v) \geq 0. \quad (55)$$

Since $\zeta_{L}(u, u) = 0$, by Eq. (49) we have

$$\sum_{a \in E} |l(a) \cap h| \pi_{L}(a) \geq \zeta_{L}(u, v) \geq \frac{\sum_{e' \in I(a)} \phi_{y}(e')}{c} \quad (56)$$

Thus, P2 is satisfied by definition. □

The number of paths between any two nodes grows exponentially with the size of the network (in P2). In order to retain polynomial solvability, we introduce variable $\zeta_{L}(u, v)$ for each edge $e$ and node pair $u$ and $v$, which is the length of the shortest path from $u$ to $v$ based on interference set weights $\pi_{L}(e')$. The introduction of these variables allows us to replace the exponential number of constraints in P2 with a polynomial number of constraints. Summarizing the above discussions, the LP formulation of problem $P_{0}$ is given as follows:

**P_{LP}**: minimize $\theta$

$\phi$ is a routing.

$$\forall e, e' \in E : \quad \sum_{e' \in E} c \cdot \pi_{L}(e') \leq \theta, \quad (57)$$

$$\forall e \in E : \quad \forall f : u \rightarrow v \in F : \quad \sum_{e' \in I(e)} \phi_{y}(e') / c \leq \zeta_{L}(u, v), \quad (58)$$

$$\forall u \in V : \quad a = (v, w) \in E : \quad \sum_{e' \in I(a)} \pi_{L}(a') + \zeta_{L}(u, v) - \zeta_{L}(u, w) \geq 0, \quad (59)$$

$$\forall u \in V : \quad \zeta_{L}(u, u) = 0, \quad (60)$$

$$\forall u, v \in V : \quad \zeta_{L}(u, v) \geq 0, \quad (61)$$

$$\forall e, e' \in E : \quad \pi_{L}(e') \geq 0. \quad (62)$$

In the above formulation, Eq. (57) can be explained by property P1. Property P2 and the shortest interference set paths account for Eq. (58), and finally property P3 appears at Eq. (62). The dual problem is a single polynomial-size LP instance, which can be solved with any LP solver. Our choice of LP solver was $lp\_solve$ [19], an open source Mixed Integer Linear Programming (MILP) solver.
6. Performance evaluation

6.1. Algorithm overview

Table 1 gives an overview of the key differences between Predictive and Oblivious Routing algorithms. We briefly explain the rows here: “Worst-Case Bound” refers to a provable bound on the congestion ratio for a given routing, as defined in Section 5. “Routes Over” indicates which sets of demands are accounted for by the routing algorithm. The “Most Useful With” row refers to the demand profile which is best suited for that algorithm. “Times Calculated” is the number of times the algorithm needs to be run for a given network. Finally, “Formulation” refers to the variation that is possible within the algorithm. Many different models could be used for predictive algorithms, including many others besides the parameterizations described in Section 4. By contrast, oblivious routing has a fixed definition and a unique set of flows for a network.

6.2. Performance overview

In this section, we simulate each of the oblivious, and predictive (with and without distributions (i.e., uncertainty-awareness)) routing strategies over a variety of mesh network setups. We also employ the oracle routing and the shortest-path routing strategies as baselines to provide performance bounds. Our purpose is to evaluate and compare their performances and precisely characterize the conditions under which each strategy performs better. This will help to define metrics on the topology and traffic pattern that will best aid the choice of the best routing strategy under different network environments. The list below describes the routing strategies that are used in the simulation study.

- **Oracle Routing (OR)**. This strategy assumes that the traffic demand is known *a priori* and produces optimal routings based on this demand. This strategy provides a upper bound on the routing performance.
- **Shortest-Path Routing (SPR)**. This strategy produces fixed routings which do not change over time with the demand. As a result, it is calculated only once. This agnostic routing is based on the shortest path (by hop count here) from mesh nodes to the Internet. This routing may perform poorly because shortest paths take no account of congestion. However, its simplicity makes it appealing for practical deployment. Many mesh network routing heuristics resemble this strategy by taking a link-state-aware path metric other than hop count.
- **Predictive Routing (PR)**. This strategy forecasts the future demand based on the historical traffic and produces a fixed prediction as described in Section 4.1. PR calculates the optimal routing using the optimal routing method described in Section 3. The routing for this strategy is updated every hour.
- **Predictive Routing with Distribution (PD)**. Similar to PR, the PD strategy also forecasts the future demand based on the historical traffic. The difference is that it produces the traffic prediction as a random variable distribution and calculates the optimal routing using the uncertainty-aware routing method described in Section 4.2. The routing for this strategy is also updated every hour.
- **Oblivious Routing (OBR)**. This is also a demand-agnostic strategy. Unlike SPR, which optimizes for path length, this strategy applies the oblivious routing algorithm described in Section 5 where the worst-case congestion ratio is minimized. It is enlightening to note that OBR has the same spirit as the most “oblivious” case of PD. When the predictive model cannot sensibly formulate a forecast for the traffic profile, it must optimize for a wide range of possible demands, effectively approaching $[0, \infty)$. Of course, the resulting routing and the mechanics will be different, because PD always operates on a PDF over a bounded range, whereas OBR operates on a truly unbounded, non-probabilistic range.

It is worthwhile to observe that SPR and OBR compute the traffic routes only once and use them during the entire simulation time, while OR and PR need to compute and update the routes every hour.

To realistically simulate the traffic demand at each AP, we employ the traces collected in a campus wireless LAN network. The network traces used in this work are collected in Spring 2002 at Dartmouth College and provided by CRAWDAD [20]. By analyzing the *snmp* log trace at each access point, we are able to derive its 1108-hour incoming and outgoing traffic volume beginning 12:00 a.m., March 25, 2002 EST. We select the access points from the Dartmouth campus wireless LAN and assign their traffic traces to the APs in our simulation. The traffic assignment given in Table 2 shows the mapping of trace APs to the APs in the simulated network shown in Fig. 4.

We experiment with the above routing strategies on the time range [108,1108], a 1000-h period excerpted from the trace. Note that all the simulation results presented in this section use 108 as the zero point.

We start by presenting the congestion achieved by the OR, PR, SPR, and OBR strategies during the entire 1000-hour simulation period. Over the whole range, OR achieves the minimum worst-case congestion, due to its unrealistic capability to know the actual traffic demand. We note that the burstiness of $\theta$ applies to all strategies including OR. This 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.

---

Table 1

<table>
<thead>
<tr>
<th></th>
<th>Predictive Routing</th>
<th>Oblivious Routing</th>
</tr>
</thead>
<tbody>
<tr>
<td>Worst-Case bound</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Routes over</td>
<td>Expected profile</td>
<td>All profiles</td>
</tr>
<tr>
<td>Most useful with</td>
<td>Predictable traffic</td>
<td>Erratic traffic</td>
</tr>
<tr>
<td>Times calculated</td>
<td>Each timestep</td>
<td>Once</td>
</tr>
<tr>
<td>Formulation</td>
<td>Many parameters</td>
<td>Unique</td>
</tr>
</tbody>
</table>

1 Note that the beginning of the trace [0,107] is used as training data, thus it is not included in the simulation result.
To filter out the noise caused by traffic burstiness, in Fig. 5a, 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, PR, SPR, and OBR achieve less than two times of the optimal congestion in most cases. The above observations get clearer when we sort out the normalized congestion

![Fig. 4. Mesh network topology.](image)

![Fig. 5. (a) Congestion ratio $(\frac{\theta}{\theta_{OR}})$; (b) Sorted view.](image)
ratio for the three strategies in Fig. 5b. It is clear that both PR and OBR which integrate the traffic prediction with the optimal routing, outperform the SPR strategy which is agnostic about the traffic demand. Further, PR achieves lower congestion than OBR for many time points due to more comprehensive representation of the traffic demand estimation. However, in other cases (less than 10% of the time), the worst-case congestion of PR is substantially higher than OBR. This problem can be largely attributed to the fundamental inaccuracy of traffic prediction.

6.3. Impact of network topology

We investigate the performance of PR and OBR in representative random topologies with Internet gateways near the perimeter. For a more complete picture, we investigate cases with two gateways and with four gateways. Each of these topologies has a total of 64 nodes, including 10 access points receiving traffic from mobile clients, the gateways, and the remaining nodes forwarding traffic on behalf of the Internet and the access points. The points are distributed at random over a simulation square 1000 m on the side, with an interference range of 155 m. For simplicity, the transmission range is equal to the interference range.

In both the 4-gateway and the 2-gateway scenarios, we run PR, OBR and SPR using the demand data from the Dartmouth trace. Fig. 6 plots the congestion ratios of PR over SPR and OBR over SPR. In both pictures, OBR and PR outperform SPR in more than 50% of the cases. During the time when they are inferior to SPR, the worst-case ratio is bounded by 2. Also when we increase the number of gateways from 2 to 4, both ratios decrease. Obviously, SPR takes advantage of this topology change, due to the fact that more gateways will diversify the shortest paths from access points to nearest gateways, and also shrink the lengths of the paths.

To study the impact of path length to our strategies, we create seven random topologies whose shortest paths from access points to the nearest gateways have different hop counts. The average hop count ranges from 2.5 to 5.5. In Fig. 7, OBR, PR, and SPR consistently deliver a better congestion ratio when the hop count decreases. Among the three, SPR has the worst congestion ratio except in the case when the average hop count is 4. Moreover, the performance of SPR worsens at an accelerated speed when the average hop count grows further, coinciding with what is observed in Fig. 6. Comparatively, the congestion ratio of OBR and PR degrades gracefully, with OBR consistently outperforming PR by less than 0.1.

Figs. 6 and 7 collectively demonstrate that OBR and PR are more advantageous than the simple shortest path routing scheme in large and complicated topologies with long paths from access points to gateways.

6.4. Impact of prediction method

We study the performance of the predictive algorithm under different prediction methods. In particular, we consider the following methods described in Section 4.1: (1) Weighted-average-of-recent-hours with $K = 3$; (2) Weighted average of the-same-hour-of-a-day with $W_1 = 1$ and $W_2 = 2$; (3) Time-series-based method (used on PR routing strategy). Fig. 8 plots the ratio of the first three methods to the time-series-based method. The results show that the time-series-based method has better performance.
than other prediction method. We also find that in this time period, the traffic demands that are closer in the time sequence have stronger correlation. Thus the prediction methods that consider more recent data produce better prediction and better routing performance.

6.5. Impact of traffic demand

To study the impact of traffic dynamics on the each of the routing strategies, we compare their performance over each timestep. Each of the Figs. 9–14 shows the 1000 timesteps. The horizontal axis is the congestion ratio achieved at that timestep by one given algorithm and the vertical axis is the ratio for the corresponding compared algorithm. The congestion ratio of the optimal routing is 1 by definition, so it would be redundant to illustrate in a graph. Because there are 4 algorithms, there are six pairs to compare. The graphs in Figs. 9–14 shows case-by-case comparisons at each timestep between each pair of algorithms. Figs. 15 and 16 show the congestion ratios attained by each of the four algorithms over all timesteps.

In each of the figures, some points lie on both sides of this line. This shows that no algorithm strictly dominates another. Furthermore each of these graphs, which show all points, depict a range from approximately 1 up to approximately 2.5. In all six cases, both algorithms are in somewhat close agreement for many points at the low end and at the high end. One conclusion from this is that
although there are factors that cause one algorithm to greatly outperform another, there are also factors that cause any pair of algorithms to succeed or fail together. Fig. 10 shows that predictive routing is better than shortest path routing in large number of cases. Fig. 9 is similar to Fig. 10, but shows a slightly larger smear of points above the line of equality (the main diagonal). Fig. 11 shows the strongest and most focused correlating line, showing that predictive and distribution predictive routings often agree nearly exactly, which does not happen for other pairs of algorithms. This result is to be expected based on the mechanics of these two algorithms. The second takeaway from this figure is that the spread is approximately equal on both sides of this diagonal. This illustrates the fundamental tradeoff between increasing the routing’s flexibility by accommodating many possible sets of demands, and being able to optimize for a narrow range. We would expect the difference in the density of points between the two sides of this diagonal to vary according to the erraticity of the traffic as well as parameters of distribution.

Figs. 13 and 14 show some interesting behavior. Fig. 13 is more “filled-in”. However, because the oblivious points are the same, this means that predictive routing points have migrated only up or down. In particular, points from the lower end of the diagonal in Fig. 14 have moved upwards (where oblivious routing is performing well) and points at the upper end moved downwards. This suggests that predictive routing with distributions is less coupled to oblivious routing than predictive routing is. Thus, adding the predictive ranges is most advantageous in cases where oblivious routing is not performing well.

Finally, Fig. 15 shows the sorted congestion ratios for each algorithm over all timesteps. Because each algorithm’s ratios are sorted point-to-point comparisons are not possible here. Rather this figure is to be interpreted by the overall shape of each curve, which shows whether a routing algorithm produces lower congestions ratios on the whole. As it shows, predictive distribution routing tends to have as many favorably low congestion points as predictive routing does at low congestion values. Similarly, predictive distribution routing does no worse than oblivious routing at high levels of congestion (with the exception of the maximal congestion ratios, where oblivious routing is superior to each other algorithm as expected.) Apparently, predictive distribution routing achieves a good compromise between oblivious and predictive routing over all but fairly unbalanced traffic.

7. Erraticity metrics and augmented predictive algorithm

To further understand the factors in traffic behavior that cause an algorithm to perform better or worse, we next explore correlated factors between the traffic profile and the algorithms’ relative performance. There are two quantities to keep track of at time $t$. First, there is the actual choice of which algorithm is best, which cannot be known in advance and which we will measure with the real-valued variable $E(t)$. Second, there is an estimate $e(t)$ that is calculated exactly from previous demands and is based on a

---

2 Notice that if Oracle routing was shown on this chart, it would be identically one over the whole domain. Near the zero end of the scale some points are less than one because of numerical instabilities and rounding.
model of erraticity. The purpose of this section is to use \( e(t) \) to predict \( E(t) \). Both of these variables require further definition and discussion.

For exactness, we say the algorithmic discrepancy at a time \( t \) is

\[
E(t) = \theta_{\text{OBR}}(t) - \theta_{\text{PR}}(t).
\]

(63)

Here, \( \theta_{\text{OBR}}(t) \) and \( \theta_{\text{PR}}(t) \) are the congestions oblivious and predictive routing would incur at time \( t \), respectively. In effect, \( E(t) \) measures the favorability of using oblivious routing during timestep \( t \). If the demands at \( t \) were known in advance, \( E(t) \) could be calculated and the algorithm choice made directly. However, the demands are not available and \( E(t) \) must be estimated based on the trace of previously seen demands.

To calculate the estimate \( e(t) \), we must measure the variation in demand across a set of APs over time. Therefore, we are required to condense the multidimensional data set of demands over one or more timesteps into a single number. The choice of metric is not a precise science, because of the dependence on demand profile and the quality of the predictive algorithm. Because the ideal metric cannot be analytically derived \textit{a priori} due to this ambiguity and uncertainty in its definition, we next develop a series of metrics and culminate in one that directly takes into account the quality of the predictive algorithm. In the following list of erraticity metrics, say that \( e_d \) is the demand imposed on AP \( a \) at time \( t \) and that \( p_d \) was the value of that demand predicted by the predictive algorithm in Section 4.

7.1. Sum-based erraticity

Although the congestion in a network is invariant to scaling in demand, it is fair to assume that when total demand is changing rapidly, the distribution of demand across APs is also changing. The simplest metric to arise from this is \( e_s \), which we initially define as the fraction change in total demand from one timestep to the next. The metric would be in the range \( [0, \infty] \), which is not ideal because extreme erraticities would exert excessive influence on statistical properties. For this reason, and for consistency, all erraticities will be scaled to the range \( [0, 1] \), Eq. (64) transforms the range \( [0, \infty] \) into \( [0, 1] \) while preserving ordering. Define \( e_s(t) \) as follows:

\[
e_s(t + 1) = \frac{e_s'(t + 1)}{1 + e_s'(t + 1)},
\]

(64)

where

\[
e_s'(t + 1) = \frac{|\sum_{a|\text{AP}} d_a t - 1 - \sum_{a|\text{AP}} d_a t|}{\sum_{a|\text{AP}} d_a t - 1}
\]

7.2. Maximum-based erraticity

It is also fair to assume that the most-congested interference set is disproportionately likely to route traffic from the AP imposing the highest demand. In particular, that the metric should be sensitive to the changes affecting this AP. Taken to its logical extreme, this metric analyzes only the most heavily loaded AP. With \( m \) the demand on the most heavily loaded AP at time \( t \) is

\[
e_m(t + 1) = \frac{e_m'(t + 1)}{1 + e_m'(t + 1)},
\]

(66)

where

\[
e_m'(t + 1) = \frac{|m_{t - 1} - m_t|}{m_{t - 1}}
\]

7.3. Piecewise-based erraticity

The most obvious weakness in the simple approaches taken above is that they do not account for global changes in the demand profile. Every access point has the potential to make a contribution to \( E(t) \). We can accommodate this appeal by calculating the change in each access point’s demand divided by the maximum of the before and after demand, and averaging these resulting numbers. These do not need to be scaled because the use of the maximum leaves them in the correct range.

\[
e_p(t + 1) = \frac{\sum_{a|\text{AP}} |d_{a t - 1} - d_{a t}|}{\text{Number of APs}}.
\]

(68)
7.4. Relative erraticity

The next improvement that could be made is to account for the relative sizes of the demands. This can be done by calculating the total change in individual AP demands and scaling by the sum of the maxima of the demands before and after, taken one AP at a time. This has several favorable properties. First, this metric is sensitive to any perturbation in any of the APs’ demands. Second, $e_d$ is sensitive to changes in an AP’s demand in proportion to that AP’s total contribution to the demand. Finally, $e_d$ is symmetric. This metric has the property that for any demands the two transitions $d_a \rightarrow d_b$ and $d_b \rightarrow d_a$ would be equally erratic. This metric was presented and briefly analyzed in our work [12].

$$e_d(t + 1) = \frac{\sum_{a:\text{AP}}|d_{a,t} - d_{a,t-1}|}{\sum_{a:\text{AP}} \max(d_{a,t-1}, d_{a,t})}.$$  

(69)

7.5. Reactive erraticity

The most sophisticated metric takes into account the predictions made. This erraticity compares the most recent predictions to the actual demand that occurred, discerning whether the predictive mechanism is currently predicting well or predicting poorly. It is defined as follows:

$$e_r(t + 1) = \frac{\sum_{a:\text{AP}}|d_{a,t} - p_{a,t}|}{\sum_{a:\text{AP}} \max(d_{a,t-1}, p_{a,t})}.$$  

(70)

In particular, this metric can clearly accommodate any predictive model and it will be a generally reasonable predictor as long as the model has intermittent periods of relative accuracy.

As noted above, the question of which metric is superior is not subject to proof, only to experiment and argument. To evaluate the quality of the metrics, we plot them against the actual demand that occurred, discerning whether the predictive mechanism is currently predicting well or predicting poorly. It is defined as follows:

$$e_r(t)$$ vs. $E(t)$ does not lie on a line, high values of one are associated with high values of the other, and low values with low values. Fig. 17 confirms our intuition that the reasoning behind $e_r(t)$ is theoretically sound. For the remainder of the paper, we will use $e_r(t)$ as a general-purpose metric, but it must be clarified that depending on the network parameters, it will always be possible to find customized metrics which perform arbitrarily well.

7.6. The augmented predictive algorithm

We will use the capability of $e_r(t)$ in estimating $E(t)$ to derive an adaptive algorithm that adjusts its choice of routing strategy based on the observed connection between these two quantities. It is essential that these adjustments are dynamically made based on the specifics of the network and demand profile, rather than encoded in advance. This is because a network is likely to have a unique traffic profile and congestion dynamics. This algorithm augments the performance of the original predictive routing with the worst-case bound of oblivious routing through the guidance of the traffic erraticity metrics. We will refer to this as Augmented Predictive Routing and abbreviate it APR where needed in the remainder of the paper.

In particular, the algorithm relies on a saved history of past predicted performance data to decide which algorithm to use at each timestep, based on a critical cutoff parameter $c$. The derivation of $c$ can be explained with the following idea: APR maintains a set of pairs $(e_r(t), E(t))$ which have been observed and saved to memory. These pairs are expected to be roughly distributed along a line as shown in Fig. 18. The $e_r(t)$ values are positive and the $E(t)$ are positive and negative, so that when fit with a trendline, this line will cross the x-axis at the point $c$. Fig. 18 is based on a slice of trace samples which is very small for clarity of the algorithm. $c$ is an estimate of the $e_r(t)$ value at which $E(t)$ would be equal to zero. If a calculated value of $e_r(t)$ is less than $c$, $E(t)$ is likely to be negative, and so original predictive routing is advised. Otherwise, oblivious routing is likely to be more successful. This is the fundamental idea that guides the algorithm’s choices.

At the beginning of each timestep, $e_r(t)$ is calculated and compared with $c$ to determine which algorithm will be
After each timestep, when the true demands are known, it is possible to forensically determine which algorithm actually was the better choice (i.e., calculate $E(t)$). This is paired with $e_t(t)$ and the two are added to the memory. Because of the uncertainty about future demands, the cutoff value is updated with the new trendline, and the cycle begins again. The mechanics are shown more specifically in Algorithm 1.

### Algorithm 1. The augmented predictive algorithm

- $c = \frac{1}{2} \{\text{The initial cutoff for } e_t(t)\}$
- $OR = OR(G) \{\text{One-time Oblivious Routing computed from network } G\}$
- $M = \emptyset \{\text{Set of } (e_t(t), E(t)) \text{ pairs}\}$

for Each Timestep $t$ do
  if Not enough history exists for estimation then
    Apply routing $OR$ to network
  else
    Compute $e_t(t)$ by Eq. (70)
    if $e_t(t) < c$ then
      Apply routing $PR(t - 1)$ to network \{Use original predictive routing based on history\}
    else
      Apply routing $OR$ to network
    end if
  end if
  Wait until timestep is over \{Actual demands become available\}
  $D \leftarrow$ Demands presented in this timestep
  $\theta_{PR} \leftarrow PR(t - 1)(D) \{\text{Apply calculated } PR(t - 1) \text{ to actual demands}\}$
  $\theta_{OR} \leftarrow OR(D) \{\text{Apply } OR \text{ to actual demands}\}$
  $E(t) \leftarrow \theta_{OR} - \theta_{PR} \{\text{Determine which algorithm would have been superior}\}$
  $M \leftarrow M \cup \{e_t(t), E(t)\}$
  if $|M| \geq 2$ then
    \{The next two lines are illustrated in Fig. 18\}
    $t$ $\leftarrow$ trendline applied to $M$
    $c$ $\leftarrow$ intersection of $t$ with $x$-axis.
  end if
end for

Depending on how readily demand data becomes available, the statistical steps could be executed asynchronously, or at the end of the timestep. The time needed for the calculations is quite small compared to standard overheads in network routing algorithms. The memory set will lag the decision making process by one timestep, which would have minimal cost when a handful of points become available for the least-squares fitting.

As noted earlier, OBR is similar in spirit to PD when the traffic is not easily subject to prediction. APR exploits this connection to improve and formally bound the performance of Predictive Routing when it appears that the predictions are not successful. As a result, the worst case for APR is not when the predictions are weak, but when a prediction is abruptly weak, following a period of high accuracy.

We conduct a simulation similar to the studies in Section 6. Fig. 19a shows the sorted performance of Augmented Predictive Routing compared to Oblivious and Predictive Routing over a representative slice of the time series. Note that from the far right of this figure, we see that Augmented Predictive Routing has a lower worst case congestion than the original predictive routing in this trace. Fig. 19b shows that Augmented Predictive Routing has a lower average congestion ratio compared to the other benchmark algorithms over the full trace simulation.

8. 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 problems 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], and the channel cost metric (CCM) which is the sum of expected transmission time weighted by the channel utilization [6]. Joint solutions for channel allocation and routing are explored in [23] using a centralized algorithm and in [5] in a distributed fashion. These heuristic solutions are designed to adapt to dynamic network
conditions. However, they lack the theoretical basis to determine the network’s global performance (e.g., whether the network bandwidth is completely utilized and whether the conflicting demands 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 [7,24] study the optimal solution of joint channel assignment and routing for maximum throughput under a multi-commodity flow problem formulation and solves it via linear programming. The work of [8] presents bandwidth allocation schemes to achieve maximum throughput and lexicographical max–min fairness respectively. Further, the work of [25] 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.


Oblivious routing [26,27] has been studied for traffic engineering on the Internet. Our work [28] studies oblivious routing in the context of mesh networks and considers the wireless communication interference.

Trace analysis has been used to study the behavior of wireless networks in many recent works. For example, [9] statistically characterizes both static flows and roaming flows in a large campus wireless network. Our work is also related to dynamic traffic engineering [17] in the 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 underlying network and traffic models for wireless mesh networks compared to the Internet.

This work extends our preliminary conference publication [12] with in-depth theoretical analysis for both predictive and oblivious routing, a survey of metrics and the construction and analysis of a novel augmented predictive algorithm.

9. Conclusion

This first part of this paper covers optimal routing strategies for wireless mesh networks with attention to traffic demand uncertainty over time and provable robustness. A predictive algorithm is used which adapts to traffic dynamics and an oblivious algorithm is tested which offers provable worst-case bounds. This paper then compares the performance of both routing algorithms in an extensive simulation study over a variety of network scenarios driven by real traffic traces. The goal is to identify the key factors that affect their performance. This paper further defines a set of erraticity metrics with the powers of estimating whether predictive routing or oblivious routing will perform better. Following the guidelines of these metrics, a novel augmented predictive algorithm is presented which can accommodate the changing conditions in the predictability of the traffic and has a superior overall performance to either oblivious or predictive routing in isolation.

References

Jonathan Wellons received his B.A. degree in Computer Science from Grinnell College in Iowa. Currently he is a Ph.D. candidate at the Department of Electrical Engineering and Computer Science of Vanderbilt University. His research interests include algorithms, distributed systems and computer networks with a focus on wireless mesh networks. He is a student member of IEEE and Comsoc.

Liang Dai received his B.E. degree in Software Engineering from Zhejiang University, China in 2004. Currently he is a Ph.D. candidate at the Department of Electrical Engineering and Computer Science of Vanderbilt University. His research interests include distributed systems and computer networks with a focus on wireless mesh networks and peer-to-peer systems. He is a student member of IEEE.

Yuan Xue received her B.S. in Computer Science from Harbin Institute of Technology, China in 1998 and her M.S. and Ph.D. in Computer Science from the University of Illinois at Urbana-Champaign in 2002, and 2005. Currently she is an assistant professor at the Department of Electrical Engineering and Computer Science of Vanderbilt University. Her research interests include wireless and sensor networks, peer-to-peer and overlay systems, QoS support and network security.

Yi Cui received his B.S. and M.S. degrees in 1997 and 1999, from Department of Computer Science, Tsinghua University, China, and his Ph.D. degree in 2005 from the Department of Computer Science, University of Illinois at Urbana-Champaign. Since then, he has been with the Department of Electrical Engineering and Computer Science at Vanderbilt University, where he is currently an assistant Professor. His research interests include peer-to-peer system, multimedia system, and wireless mesh network.