Model-driven Optimization of Opportunistic Routing

Eric Rozner Mi Kyung Han Lili Qiu Yin Zhang
The University of Texas at Austin
{erozner,hanmi2,lili,yzhang}@cs.utexas.edu

ABSTRACT

Opportunistic routing aims to improve wireless performance by exploiting communication opportunities arising by chance. A key challenge in opportunistic routing is how to achieve good, predictable performance despite the incidental nature of such communication opportunities and the complicated effects of wireless interference in IEEE 802.11 networks. To address the challenge, we develop a model-driven optimization framework to jointly optimize opportunistic routes and rate limits for both unicast and multicast traffic. A distinctive feature of our framework is that the performance derived from optimization can be achieved in a real IEEE 802.11 network. Our framework consists of three key components: (i) a model for capturing the interference among IEEE 802.11 broadcast transmissions, (ii) a novel algorithm for accurately optimizing different performance objectives, and (iii) effective techniques for mapping the resulting solutions to practical routing configurations. Extensive simulations and testbed experiments show that our approach significantly outperforms state-of-the-art shortest path routing and opportunistic routing protocols. Moreover, the difference between the achieved performance and our model estimation is typically within 20%. Evaluation in dynamic and uncontrolled environments further shows that our approach is robust against inaccuracy introduced by a dynamic network and it also consistently out-performs the existing schemes. These results clearly demonstrate the effectiveness and accuracy of our approach.

Categories and Subject Descriptors

C.2.2 [**Computer-Communication Networks**]: Network Protocols—*Routing protocols*; C.2.1 [**Computer-Communication Networks**]: Network Architecture and Design—*Wireless communication*

General Terms

Algorithms, Experimentation, Measurement, Performance

Keywords

Opportunistic Routing, Wireless Mesh Networks, Wireless Network Model, Model-driven Optimization, Wireless Interference

1. INTRODUCTION

Wireless mesh networks are becoming a new attractive communication paradigm. Many cities have deployed or are planning to deploy them to provide Internet access to homes and businesses. Traditionally, a sender commits to a single node as the next hop to route towards its destination, and traffic makes progress only when it reaches the selected next hop. The high loss rates in wireless networks (*e.g.*, 20-40% as observed in several deployments [2, 38]) make traditional routing inefficient. To achieve better performance, opportunistic routing has been proposed to exploit communication opportunities that arise by chance due to the broadcast nature of the wireless medium. When a sender broadcasts its data, any node that hears the transmission may forward the data toward the destination. Although individual nodes may experience high loss rates, as long as there exists one forwarder that is closer to the destination and receives the transmission, the data can move forward. In this way, opportunistic routing can effectively combine multiple weak links into a strong link and take advantage of transmissions reaching unexpectedly near or unexpectedly far.

There are two key factors that determine the performance of opportunistic communication in wireless mesh networks: (i) routes (*i.e.*, for a given flow how much traffic node j should forward upon receiving a packet from another node i), and (ii) rate limits (*i.e.*, how fast each traffic source can inject traffic into the network). Routes determine how effectively we take advantage of communication opportunities and how efficiently we utilize network resources and exploit spatial reuse. Rate limits ensure that traffic sources do not send more than what paths can support. Without appropriate rate limits, the network throughput can degrade drastically under traditional shortest-path routing [23]. Rate limiting is even more critical for opportunistic routing due to its use of broadcast transmissions: (i) broadcast transmissions do not perform exponential backoff (*i.e.*, its contention window does not increase upon packet losses) and thus are more likely to cause network congestion; and (ii) broadcast transmissions preclude the use of 802.11's synchronous ACK mechanism, and receivers' feedback has to be sent above the MAC layer, which can easily get lost during network congestion and cause unnecessary retransmissions and serious throughput degradation.

In this paper, we jointly optimize routes and rate limits for opportunistic communication. We focus on static, 802.11-based, multihop networks, though we believe that the general methodology is applicable to other scenarios. We develop the first opportunistic routing protocol that can accurately optimize IEEE 802.11 end-to-end performance (*i.e.*, the performance derived from optimization can be realized in a real IEEE 802.11 multihop network). This distinctive feature is important given the wide deployment of IEEE 802.11 networks.

Challenges: Accurate optimization of opportunistic communication in an IEEE 802.11 network is challenging for the following four reasons. *First*, the dynamic and incidental nature of communication opportunities makes it difficult to estimate their impact on the resulting network performance. *Second*, optimization of opportunistic routing places stringent requirements on a network model: the model should (i) specify the region of feasible network configurations using a compact representation so that we can optimize the objective within the feasible region as defined by these constraints, (ii) accurately estimate performance on every link in the network (as opposed to only a small number of links on specified routes, as in [23], for the purpose of optimizing rate limiting alone), and (iii) be accurate across a wide range of traffic conditions, including high traffic load, which is common in opportunistic routing. *Third*, the non-convex interference relationships among different links and the huge search space of possible opportunistic routes and rate limits impose significant challenges on the optimization procedure itself.

SIGMETRICS'11, June 7–11, 2011, San Jose, California, USA.
Copyright 2011 ACM 978-1-4503-0262-3/11/06 ...$10.00.

Fourth, to be valuable in practice, the resulting optimization solution should be easy to implement, using only a small number of control knobs.

Approach and contributions: We address the above challenges using the following four steps:

1. *General optimization framework.* We develop a general framework to jointly optimize routes and rate limits for opportunistic communication (Section 3). The framework uses opportunistic constraints to probabilistically characterize the available communication opportunities. It can use different wireless interference models.
2. *Interference model for IEEE 802.11 broadcast traffic.* The complex interference, traffic, and MAC-induced dependencies in the network are often the underlying cause of unexpected behavior. We develop a new model to capture these dependencies for broadcast transmissions (Section 4). We use measurements from a given network to estimate link loss rate, carrier sense probability, and conditional collision loss probabilities to seed our model. Our model derives the relationships between sending rates, loss rates, and throughput to capture the effects of carrier sense and collisions. Our model involves only $O(E)$ constraints, where E is the total number of edges. Thus it can be easily incorporated into our optimization framework. Despite its simplicity, the model captures real-world complexities such as hidden terminals, non-uniform traffic, multihop flows, non-binary and asymmetric interference.
3. *Iterative procedure for non-convex optimization.* Since our model is non-convex, we develop an iterative optimization procedure to find a local optimal solution (Section 5). Our algorithm is flexible and can accommodate different performance objectives. For comparison, we explore an alternative approach that uses a widely used conflict-graph-based interference model [17] that is less accurate [23, 33], but convex, and thus allows global optimization. Our results show that our approach of combining a more accurate model with non-convex optimization yields better and more accurate performance.
4. *Practical installation of routes and rate limits.* We develop a practical opportunistic routing protocol that implements the opportunistic routes and rate limits optimized by our algorithm in real networks (Section 6). The mechanisms for installing routes and rate limits can support both unicast and multicast.

We implement our protocol in both the Qualnet simulator [34] and a 21-node wireless mesh testbed using Click [8] and the MadWiFi driver [27]. Extensive simulations and testbed experiments (Section 7– 9) show that our approach achieves high accuracy (*i.e.*, the difference between the achieved performance and our model estimation is within 20%) and significantly out-performs state-of-the-art shortest path and opportunistic routing protocols (*e.g.*, its total throughput is up to 14x ETX's throughput and 11x MORE's throughput). We further study the impact of dynamic and uncontrolled environments on accuracy and performance, and find that our approach is robust to inaccuracy in the input and it also consistently out-performs the existing schemes (Section 10).

2. RELATED WORK

We classify related work into three categories: (i) design of opportunistic routing protocols, (ii) analysis of opportunistic routing performance, and (iii) wireless network modeling.

Opportunistic routing protocols: ExOR [6] is a seminal opportunistic routing protocol. In ExOR, a sender broadcasts a batch of packets. Each packet contains a list of nodes that can potentially forward it. To maximize the progress of each transmission, the forwarding nodes relay data packets in the order of their proximity to the destination in terms of the ETX metric [9], which quantifies the number of transmissions required to deliver a packet from the forwarder to the destination. To avoid redundant transmissions, every forwarding node only forwards packets that have not been acknowledged by nodes with a smaller ETX to the destination.

Since then, several opportunistic routing protocols have been proposed (*e.g.*, [7, 20, 24, 25]). In particular, MORE [7] applies network coding to opportunistic routing. Since random linear coding generates linearly independent coded packets with high probability, the forwarding nodes in MORE require no coordination. Instead, each node computes how much traffic it should forward and independently generates random linear combinations of all the packets it has received from the current batch. By obviating the needs for strict coordination, MORE can out-perform ExOR. However, as we show in Section 9, the performance of MORE can degrade significantly when there are more than a few flows in the network. This is because (i) it lacks rate limiting and causes network congestion, and (ii) its routes only try to minimize the number of transmissions and do not take wireless interference into account. In comparison, we directly optimize end-to-end performance by computing interference-aware opportunistic routes and rate limits. The performance optimized by our approach can be realized in a real network and is significantly better than the existing schemes.

Theoretic analysis of opportunistic routing: There have been several studies analyzing the performance of opportunistic routing. For example, [46] develops a methodology for estimating the maximum throughput given forwarding paths and traffic demands, and [47] extends the work to multi-radio multi-channel wireless networks. Both works assume the opportunistic routes are given, where nodes only forward traffic that is not received by nodes closer to the destinations, so they cannot optimize routes. Note that such selected routes are not optimal since (i) a single path routing metric, such as ETX, does not capture the anycast performance in the opportunistic routes [10] and (ii) shortest path routes do not result in the highest throughput due to wireless interference.

A few studies (*e.g.*, [26, 36, 43, 45, 48]) propose optimization frameworks for opportunistic routing. These studies use a conflict-graph-like interference model, which significantly over-estimates the actual performance as shown in Section 8. Different from these works, we show that to achieve accurate optimization of network performance it is essential to use an effective network model that captures the non-convex relationship between the performance of different wireless links. This calls for a new wireless model and an efficient algorithm to search for a close-to-optimal solution, which we address in this paper. We further discuss the differences between the conflict graph interference model that these works use and our model in Section 4. In addition to a new interference model and model-based optimization, our work goes beyond theoretical analysis (which is the primary focus of the above works) by developing a practical routing protocol to realize the performance gains in a real IEEE 802.11 network.

Wireless network modeling: Significant research has been done on wireless network modeling. One class of work focuses on asymptotic performance bounds (*e.g.*, [15, 16, 22]). These models provide useful insights as a network scales, but cannot be applied to a specific network. Another large class of models predict performance for a given scenario (*e.g.*, [5, 12, 14, 18, 33, 37]). They differ in their generality: some assume that everyone is within communication range of each other [5, 12, 14, 21], while others assume restricted traffic demands (*e.g.*, a single flow [12, 14], two flows [37], sending to a single neighbor [13], adding one new flow at a time [40], or one-hop demands [18, 33]). Moreover, models in this class predict performance under a given scenario and cannot support optimization without enumerating all possible network configurations, which is prohibitive due to a huge search space. To facilitate optimization, we need a model that can specify the entire region of feasible network configurations using a compact set of constraints, which can then be incorporated into the optimization procedure to optimize the desired objective within the feasible

$Flows$	the set of unicast or multicast flows
$src(f)$	source of flow f
$dest(f,d)$	d-th destination of flow f
$Demand(f)$	traffic demand of flow f, *i.e.*, the amount of traffic f desires to send
$G(f)$	throughput of flow f
$T(f,i)$	node i's sending rate for flow f
$Y(f,d,i,j)$	information receiving rate along link $i \rightarrow j$ for d-th destination in flow f ($d = 1$ for unicast)
$P(i,j)$	loss rate of link $i \rightarrow j$ (including both collision and inherent wireless medium loss)
$\mathcal{N}(i)$	a subset of i's neighbors
$S(i, \mathcal{N}(i))$	success rate from node i to i's neighbor set $\mathcal{N}(i)$

Table 1: Notations for optimizing opportunistic routing.

$\triangleright\, Input: \ Flows, \ Demand(f)$
$\triangleright\, Output: \ T(f,i), \ Y(f,d,i,j)$
maximize: $\sum_{f \in Flows} G(f) - \beta \sum_{f,i} T(f,i)$
subject to:
[C1] $G(f) \le Demand(f)$ $(\forall f)$
[C2] $G(f) \le \sum_k Y(f,d,k,dest(f,d))$ $(\forall f, d)$
[C3] $Y(f,d,k,src(f)) = 0$ $(\forall f, d, k)$
[C4] $Y(f,d,dest(f,d),k) = 0$ $(\forall f, d, k)$
[C5] $\sum_k Y(f,d,k,i) \ge \sum_j Y(f,d,i,j)$
$(\forall f, d, i : i \ne src(f) \text{ and } i \ne dest(f,d))$
[C6] $S(i,\mathcal{N}(i))T(f,i) \ge \sum_{k \in \mathcal{N}(i)} Y(f,d,i,k)$ $(\forall f, i, \mathcal{N}(i))$
[C7] $interference\ constraints\ on\ T_i \triangleq \sum_f T(f,i)$

Figure 1: Problem formulation to optimize multicast throughput of opportunistic routing.

region. Two existing models are in this category: (i) the conflict-graph model [17], and (ii) the unicast interference model [23]. We discuss why they are insufficient for optimizing opportunistic routing in Section 4.1.

3. OPTIMIZATION FRAMEWORK

Overview: We develop a general framework for jointly optimizing opportunistic routes and rate limits. Our formulation assumes the use of network coding, which prevents nodes from forwarding redundant information without requiring fine-grained coordination among different nodes. Without loss of generality, we focus on multicast flows, since unicast flows are a special case of multicast with one receiver in each multicast group. The main design issue becomes how fast each traffic source should send traffic and how much traffic an intermediate node should forward to achieve high performance. This can be formulated as an optimization problem that maximizes total network throughput subject to information conservation constraints, opportunistic constraints, and interference constraints. Figure 1 shows the resulting formulation, and Table 1 specifies the variables in the formulation.

Optimization objective: Given the set of unicast or multicast flows $Flows$, and the traffic demands $Demand(f)$, our optimization outputs traffic sending rates $T(f,i)$ and information receiving rates $Y(f,d,i,j)$, which will be converted to opportunistic routing configurations using a credit-based scheme described in Section 6. As shown in Figure 1, the first term in the objective, $\sum_{f \in Flows} G(f)$, reflects the primary goal of maximizing the total throughput over all flows. The second term in the objective, $-\beta \sum_{f,i} T(f,i)$ represents the total amount of wireless traffic. Including both terms reflects the goals of (i) maximizing total throughput and (ii) preferring the least amount of traffic among all solutions that support the same total throughput (*e.g.*, avoiding loops and unnecessary traffic). Since the first objective is more important, we use a small weighting factor $\beta = 10^{-5}$ for the second term just for tie breaking (*i.e.*, only when the first objective is the same, we prefer the one with the least traffic).

To compute the first term, for a unicast flow f, $G(f)$ is its throughput. For a multicast flow f, $G(f)$ is the throughput of the bottleneck receiver. Note that there are many other ways to define the objective in multicast setting [42]. Here we use one of the metrics as an example. Our optimization framework can support other multicast objectives, such as total throughput over all receivers in the multicast group or other weighted versions. Moreover, while we focus on total throughput, our framework can be directly applied to optimizing other objectives. For example, our evaluation also considers optimizing a linear approximation of proportional fairness, defined as $\sum_{f \in Flows} \log G(f)$, which strikes a good balance between fairness and throughput [35]. We can also maximize total revenue if the revenue of a flow is a function of its throughput.

Throughput constraints: To ensure $G(f)$ is the throughput of flow f, it has to satisfy constraints (C1) and (C2) in Figure 1. Constraint (C1) indicates that the throughput of a flow should be no more than its traffic demand (*i.e.*, total amount of information a source desires to send). Constraint (C2) ensures that $G(f)$ is no more than the total amount of information delivered from all links incident to the destination of flow f. For a multicast flow f, $G(f)$ should be no more than the total amount of information delivered to each destination in the flow f. Note that we do not need a lower bound on $G(f)$ since the objective is to maximize $G(f)$.

Information conservation constraints: To handle lossy wireless links, we distinguish traffic and information sent along a link. A feasible routing solution should satisfy information conservation. This property is given by constraints (C3–C5) in Figure 1. Constraint (C3) ensures no incoming information to a traffic source, constraint (C4) ensures no outgoing information from a destination, and constraint (C5) represents flow conservation at an intermediate node i, *i.e.*, the total amount of incoming information is no less than the total amount of out-going information.

Opportunistic constraints: Opportunistic routing exploits the wireless broadcast medium by having different nodes extract information from the same transmission. We formally capture this notion using opportunistic constraints, which relate traffic volume to the amount of information delivered.

For ease of explanation, we first consider one sender sending to two receivers, and then generalize it to an arbitrary number of receivers. Consider a sender s, and denote the link loss rates from s to its neighbors r_1 and r_2 as $P(s, r_1)$ and $P(s, r_2)$, respectively. It is evident that for a given flow the amount of information delivered to a neighbor is bounded by the product of the sending rate and link delivery ratio. Therefore we have $(1 - P(s,r_1))T(f,s) \ge Y(f,d,s,r_1)$ and $(1 - P(s,r_2))T(f,s) \ge Y(f,d,s,r_2)$. In addition, since there is overlap between the information delivered to r_1 and r_2, we are only interested in the non-overlapping information (*i.e.*, when redundant information is delivered to both nodes, it should only count once). The total non-overlapping information delivered to r_1 and r_2 should satisfy the following constraints:

$$(1 - P(s,r_1)P(s,r_2))T(f,s) \ge \sum\nolimits_{i \in \{1,2\}} Y(f,d,s,r_i),$$

where the left hand-side represents the total amount of traffic successfully delivered to at least one of the receivers, and the right hand-side represents the total non-overlapping information delivered to the receivers.

Now we consider a general setting, where a sender s has N neighbors. We enumerate all possible subsets of its neighbors. For each neighbor set $\mathcal{N}(i)$, we require:

$$S(i, \mathcal{N}(i))T(f,i) \ge \sum\nolimits_{k \in \mathcal{N}(i)} Y(f,d,i,k), \qquad (1)$$

where $S(i, \mathcal{N}(i))$ denotes the delivery probability from i to at least one node in $\mathcal{N}(i)$. When delivery rates of different links are independent, which holds for some networks [44], $S(i, \mathcal{N}(i)) = 1 - \prod_{k \in \mathcal{N}(i)} P(i,k)$. When the link delivery rates are correlated, we can empirically measure $S(i, \mathcal{N}(i))$. Equation 1 indicates the total traffic successfully delivered to at least one neighbor in $\mathcal{N}(i)$ should be no less than the total non-overlapping information delivered to $\mathcal{N}(i)$. This results in (C6) in Figure 1. When i has many (say, K) neighbors, we limit the number of such constraints by only enumerating neighbor sets of size 1, 2, and K (*i.e.*, we enumerate only O(K^2) instead of O(2^K) neighbor sets).

Interference constraints: Wireless interference has a significant impact on wireless network performance. In particular, nearby senders carrier sense and defer to each other. Moreover, since carrier sense is not perfect, there may be multiple overlapping nearby transmissions that can cause collisions. These effects can further constrain the amount of traffic on each link and introduce strong inter-dependency between sending rates, loss rates, and throughput. We address this issue in Section 4 by developing the constraints that capture the relationships between $T(f, i)$ and $P(i, j)$.

4. BROADCAST INTERFERENCE MODEL

4.1 Motivation for a Better Model

Despite significant research on modeling the impact of wireless interference, none of the existing models directly fulfills our need for optimizing opportunistic routing. To support optimization, we need a model that specifies the feasible region of network configurations using a compact representation. The following two existing models fall into this category.

Conflict-graph model: The first model, proposed in [17], is a conflict-graph model that represents wireless links as vertices and draws a conflict edge between two vertices if the corresponding wireless links interfere. Based on this definition, it is clear that links corresponding to an independent set in the conflict graph can be active simultaneously. Therefore, the interference constraints are the schedule restrictions imposed by the independent sets, which can be expressed as a set of linear constraints.

There are two limitations in applying the conflict-graph model for optimizing opportunistic routing. First, the model in [17] assumes perfect scheduling, *i.e.*, packet transmissions at different nodes can be precisely controlled and it over-estimates the performance in real networks as we will show in Section 8. Second, the conflict-graph model is a link-based model, while opportunistic routing uses broadcast transmissions and requires a node-based broadcast model. Existing broadcast extensions of the conflict-graph model provide only an aggregate answer of whether two broadcast transmissions interfere or not. For example, some extensions [36, 41, 46, 48] conservatively consider two broadcast transmissions to interfere if any one of their receivers is interfered by the other transmission, while other extensions [46] consider broadcast transmissions to interfere if all of their receivers are interfered by the other transmission. A single aggregate answer on whether broadcast transmissions interfere does not fully characterize the impact of interference on different receivers and is therefore inadequate for use in optimizing opportunistic routing.

IEEE 802.11 unicast model: The other model, proposed in [23], models interference among unicast transmissions in IEEE 802.11. Since opportunistic routing uses broadcast traffic, we need to develop interference models for broadcast transmissions. Furthermore, as broadcast transmissions does not perform binary backoff to limit the sending rate (*i.e.*, its contention window does not increase even under packet losses), it is necessary to have an accurate model even for high traffic load and channel occupancy, which induces high collision losses, and the linear approximation used in [23] becomes inaccurate under high collision losses. In addition, [23] is used for rate limiting unicast transmissions when given specified routes. Therefore it suffices to accurately estimate the sending rates and loss rates on a small number of links used for routing. In contrast, for the purpose of route optimization, we need to accurately estimate the performance for *all* receivers of a given sender, which is much more challenging.

Modeling goals and strategy: We develop our model specifically for IEEE 802.11 broadcast traffic. We observe that wireless interference affects IEEE 802.11 traffic in two important ways: (i) nearby senders cannot transmit simultaneously due to carrier sense, and (ii) transmissions may sometimes result in collisions due to imperfect carrier sense. We model these effects by developing the relationships between sending rates, loss rates, and throughput, which can be incorporated into our optimization framework and facilitate model-driven optimization. While this paper applies the model to optimizing opportunistic routing, the model is useful in other contexts (*e.g.*, optimizing network topology and network planning). Our model is general and captures real-world complexities (*e.g.*, hidden terminals, multihop flows, non-binary interference, and heterogeneous traffic), which is confirmed by simulation and testbed experiments using multihop networks in Section 8. Compared with [23], both our sender model (Section 4.3.1) and loss model (Section 4.3.2) are much more refined and do not involve any linear approximation. Thus, our model can more accurately estimate the loss rates for all receivers even under heavy traffic loads, which is essential for the optimization of opportunistic routing.

4.2 Background and Assumptions

We first review the broadcast transmissions as specified by the IEEE 802.11 standard [32]. Before transmission, a sender first checks to see if the medium is available using carrier-sensing. A sender determines the channel to be idle when the total energy received is less than the clear-channel assessment threshold. In this case, a sender may begin transmission using the following rule: If the medium has been idle for longer than a distributed inter-frame spacing time (DIFS) period, transmission can begin immediately. Otherwise, a sender waits for DIFS and then waits for a random backoff interval uniformly chosen between $[0, CW_{\min}]$, where $CW_{\min}$ is the minimum contention window.

Our model strikes balance between realism and simplicity in order to support effective model-driven optimization. We make the following assumptions, which help simplify our model:

$A1$) It assumes pairwise interference, *i.e.*, the interference relationship between two links is independent of activities on other links. Previous works show that pairwise interference is a good approximation in real networks [1, 31]. Hence this assumption is widely used in the literature (*e.g.*, [5, 12, 14, 23, 37]). Moreover, for optimizing the routing of multihop wireless networks, it is often more important to capture the interference relationship among links that are not too far apart. For these links, the pairwise interference relationship is likely to be an even better approximation.

$A2$) It assumes that inherent wireless medium loss (*i.e.*, loss under no interfering traffic) and collision loss are independent, which has been commonly used (*e.g.*, [23, 33]).

$A3$) The inherent wireless medium losses at different nodes are independent, which is experimentally validated in [28, 29, 37].

$A4$) Inter-packet delays from a node follow an exponential distribution, as assumed in [14, 23, 33]. This assumption is only needed for deriving overlapping probabilities between two transmissions.

While some of these assumptions do not always hold (*e.g.*, [44] shows that loss rates of different wireless links may be correlated for some networks), our evaluation results show that our model-driven optimization yields accurate performance estimates despite such simplifications. With these assumptions, we develop a tractable model with $O(E)$ constraints, where E is the number of links.

4.3 Our New Model

We develop a simple interference model for multihop wireless networks to capture the interdependency between broadcast sending rates, loss rates, and throughput. Such interdependency can be captured using $O(E)$ constraints, where E is the total number of edges in the network. These constraints can then be incorporated into the optimization problem as interference constraints [C7] shown in Figure 1. We present methods to measure the input parameters of the model in Section 6.

Our model consists of two main components: (i) a *sender model* that captures the effects of carrier-sensing on a sender's sending rate, and (ii) a *loss model* that captures both inherent loss (*i.e.*, packet loss under no interference) and the effects of overlapping packet transmissions on the collision loss rates for different links.

4.3.1 Broadcast Sender Model

Modeling the effects of carrier sense on traffic rates: We divide time into *variable-length slots* (VLS) for each sender i. A variable-length slot may last for either IEEE 802.11 slot time T_{slot} or the transmission time of a packet followed by a DIFS duration. The former occurs when i senses a clear channel but either has no data to transmit or has data but cannot transmit due to a non-zero backoff counter. The latter occurs when i either transmits a packet or waits for a transmission from another sender to complete.

Let τ_i be the probability for i to start a new packet transmission in a variable-length slot. Clearly, τ_i depends on (i) how often i has data to send, and (ii) the random backoff interval (*i.e.*, $CW_{\min}$). As derived in [5], when i has saturated traffic demand (*i.e.*, it always has data to transmit), on average i performs one transmission every $CW_{\min}/2+1$ variable-length slots (since there is no exponential backoff for broadcast traffic, we have $CW_{\min}/2$ slots for backoff plus 1 slot for the transmission). Therefore, the transmission probability τ_i is bounded by the following *feasibility constraint*:

$$\tau_i \leq \tau_{\max} \triangleq \frac{1}{CW_{\min}/2+1} \quad (\text{for } \forall i). \tag{2}$$

Under the pairwise interference model (*i.e.*, A1), whether sender i carrier-senses (and thus defers to) an ongoing transmission of sender j only depends on nodes i and j and is independent of if other senders are transmitting. Let D_{ij} be this carrier sense probability (*i.e.*, probability for node i to defer to node j when node j is transmitting). For convenience, let $D_{ii}=1$. Let T_i be sender i's sending rate over all flows ($T_i = \sum_f T(f,i)$), VLS_i be its expected VLS duration, and P_i^{idle} be the idle probability of node i. T_i, VLS_i and τ_i have the following approximate relationship, called the *throughput constraints*:

$$\begin{aligned} T_i &= (EP \times \tau_i)/VLS_i, & (3)\\ VLS_i &= T_{\text{slot}} P_i^{\text{idle}} + (T_{\text{xmit}} + T_{\text{DIFS}})(1-P_i^{\text{idle}}) & \\ &= T_{\text{slot}} + (T_{\text{xmit}} + T_{\text{DIFS}} - T_{\text{slot}})(1-P_i^{\text{idle}}), & (4)\\ P_i^{\text{idle}} &= \textstyle\prod_j \left(1 - D_{ij} \times \tau_j \times \frac{VLS_i}{VLS_j}\right), & (5) \end{aligned}$$

where EP is the expected packet payload size, EH is expected header size, $T_{\text{xmit}} = (EP+EH)/rate$ is the expected packet transmission time, and T_{slot} is an IEEE 802.11 slot time. Eq. (3) computes throughput as the total amount of payload transmitted during one VLS divided by the expected VLS duration. Eq. (4) computes expected VLS duration as idle probability times an idle slot duration plus transmission (including collision) probability times a transmission duration. Finally, Eq. (5) gives the probability that i finds the medium is idle, where $\tau_j \times \frac{VLS_i}{VLS_j}$ is the probability for j to start a transmission in i's VLS, $D_{ij} \times \tau_j \times \frac{VLS_i}{VLS_j}$ is the probability that i defers to j's transmission, and $\prod_j (1 - D_{ij} \times \tau_j \times \frac{VLS_i}{VLS_j})$ is the probability that i does not defer to any node in the network including its own transmission (*i.e.*, i senses the medium is idle).

Eliminating model parameters $\{\tau_i\}$ and $\{P_i^{\text{idle}}\}$: To better facilitate model-driven optimization, we eliminate model parameters $\{\tau_i\}$ and $\{P_i^{\text{idle}}\}$ and transform (2)–(5) into the following equivalent constraints, which apply directly to the traffic rates $\{T_i\}$.

- *Feasibility constraint.* According to Eq. (3), we have: $\tau_i = \frac{T_i \times VLS_i}{EP}$. As a result, Eq. (2) is equivalent to:

$$\frac{T_i}{EP} \leq \frac{\tau_{\max}}{VLS_i} \quad (\text{for } \forall i). \tag{6}$$

- *Throughput constraint.* With $\tau_i = \frac{T_i \times VLS_i}{EP}$, Eq. (5) becomes: $P_i^{\text{idle}} = \prod_j \left(1 - \frac{D_{ij} \times T_j \times VLS_i}{EP}\right)$. So Eq. (4) becomes:

$$\begin{aligned} VLS_i = T_{\text{slot}} + \ & (T_{\text{xmit}} + T_{\text{DIFS}} - T_{\text{slot}}) \times \\ & \left[1 - \textstyle\prod_j \left(1 - \frac{D_{ij} \times T_j \times VLS_i}{EP}\right)\right]. \end{aligned} \tag{7}$$

Eq. (6) and (7) fully capture the relationships in (2)–(5) but have fewer variables. Moreover, note that when traffic rates $\{T_j\}$ are given as inputs, Eq. (7) contains only a single variable: VLS_i. This allows us to numerically derive VLS_i and partial derivatives $\frac{\partial VLS_i}{\partial T_j}$ from the given $\{T_j\}$ (as described in Section 5.2). We will therefore use (6) and (7) in our model-driven optimization.

4.3.2 Broadcast Loss Model

Integrating inherent loss and collision loss: To estimate loss rates $P(i,j)$ from traffic rates T_i, we distinguish between two types of loss: inherent wireless medium loss (*i.e.*, loss rate under no interference) and collision loss. The former is denoted as $P^{\text{raw}}(i,j)$ for link $i \rightarrow j$ and can be periodically measured. The latter depends on two factors: (i) how often transmissions from different nodes overlap and (ii) how often such overlapping transmissions result in a collision. To capture the first effect, we introduce $O(i,k)$ to denote the probability for an i's transmission to overlap with a k's transmission (conditioned on i's transmission) and derive its value based on the carrier sense probability. To capture the second effect, we observe that the pairwise interference model indicates there is a constant conditional collision loss probability L_{ij}^k (*i.e.*, the probability that a transmission on link $i \rightarrow j$ collides with an overlapping transmission from node k). We assume that inherent wireless medium loss and collision loss are independent, which has been commonly used (*e.g.*, [23, 33]). We then compute $P(i,j)$ as:

$$P(i,j) = 1 - (1 - P^{\text{raw}}(i,j)) \times \prod_{k \neq i} \left[1 - L_{ij}^k \times O(i,k)\right]. \tag{8}$$

This is because a packet is delivered when it is not lost due to either inherent loss or collision loss. To ensure no collision, the packet should not collide with any node's transmission. Since $L_{ij}^k \times O(i,k)$ is the collision loss probability with node k's transmission, $\prod_{k \neq i} \left[1 - L_{ij}^k \times O(i,k)\right]$ is the probability that the link has no collisions with any other node in the network.

Estimating overlap probabilities: We next estimate the overlap probability $O(i,j)$, which depends on whether i and j can carrier sense each other. Our model has two salient features: (i) it supports both symmetric and asymmetric deferral (*e.g.*, node i defers to node j but not vice versa), and (ii) it handles non-binary deferral (*e.g.*, node i sometimes defers to j and sometimes does not).

To provide both features, our modeling strategy is to divide time into regions to which one of the following four cases applies:

- Case 1: i and j can both carrier sense each other;
- Case 2: neither i nor j can carrier sense each other;
- Case 3: i can carrier sense j but j cannot carrier sense i; and
- Case 4: i cannot carrier sense j but j can carrier sense i.

Let $Q_c(i,j)$ be the probability for Case c to occur. Let $O_c(i,j)$ be the probability for a transmission of i to overlap with any transmission of j under Case c. We then have:

$$O(i,j) = \sum_{c=1}^{4}(Q_c(i,j) \times O_c(i,j)). \quad (9)$$

Assuming whether i can carrier sense j is independent of whether j can carrier sense i, we can simply compute $Q_c(i,j)$ as:

$$\left\{\begin{array}{lcl} Q_1(i,j) & = & D_{ij} \times D_{ji}, \\ Q_2(i,j) & = & (1-D_{ij}) \times (1-D_{ji}), \\ Q_3(i,j) & = & D_{ij} \times (1-D_{ji}), \\ Q_4(i,j) & = & (1-D_{ij}) \times D_{ji}. \end{array}\right. \quad (10)$$

In our technical report [39], we derive $O_c(i,j)$ as follows.

$$\left\{\begin{array}{lcl} O_1(i,j) & = & \tau_j = \frac{T_j \times VLS_j}{EP}, \\ O_2(i,j) & = & 1-(1-\theta_j)\exp\left[-T_{\mathrm{xmit}}/IPD_j\right], \\ O_3(i,j) & = & 1-\exp\left[-T_{\mathrm{xmit}}/IPD_j\right], \\ O_4(i,j) & = & \frac{\theta_j}{\theta_j+(1-\theta_j)\exp\left[-T_{\mathrm{xmit}}/IPD_j\right]}, \end{array}\right. \quad (11)$$

where $\theta_j = \frac{T_j}{rate} \times \frac{EP+EH}{EP}$ is the fraction of time j is transmitting (either payload or header) and $IPD_j \triangleq \frac{1-\theta_j}{\theta_j} \times T_{\mathrm{xmit}}$ is j's expected inter-packet delay.

4.3.3 Model Initialization

Our model has the following input parameters: (i) inherent wireless link loss rates P_{ij}^{raw}, (ii) carrier sense probabilities D_{ij}, and (iii) conditional collision loss probabilities L_{ij}^{k}. For simplicity, we estimate these parameters by conducting pairwise broadcast measurements [1, 23], but our model can just as easily use the inputs inferred by more scalable approaches (*e.g.*, [3, 4]).

1. We first let node a send broadcast traffic alone. The other nodes record the receiving rates from a. We then estimate $P^{\mathrm{raw}}(a,b) = 1-$($b$'s receiving rate from a)/(a's sending rate).
2. We next let two nodes a and b send broadcast traffic simultaneously and measure their sending rates T_a and T_b. Since neither a nor b has any rate limit, we have $\tau_a = \tau_b = \tau_{\max} = \frac{1}{CW_{\min}/2+1}$. From Eq. (3), we can then compute $VLS_a = (EP \times \tau_a)/T_a$ and $VLS_b = (EP \times \tau_b)/T_b$. Applying Eq. (7) to the case with only two senders a and b, we have:

   $$VLS_a = T_{\mathrm{slot}} + (T_{\mathrm{xmit}} + T_{\mathrm{DIFS}} - T_{\mathrm{slot}}) \times \\ \left[1-(1-\tfrac{D_{aa}\times T_a \times VLS_a}{EP})\left(1-\tfrac{D_{ab}\times T_b \times VLS_a}{EP}\right)\right)\right]. \quad (12)$$

   Note that $D_{aa} = 1$. So linear equation (12) has only a single unknown D_{ab}. We can therefore estimate D_{ab} by solving (12).
3. Finally, when both a and b are sending broadcast traffic, the other nodes record their receiving rates from a and b. For any node $c \notin \{a,b\}$, we can compute the loss rate $P(a,c) = 1-$ (c's receiving rate from a)/T_a. Moreover, given T_a, T_b, D_{ab} and D_{ba}, we can compute the overlapping probability $O(a,b)$ according to Eq. (9)–(11). Applying Eq. (8) to the case in which there are only two senders a and b, we obtain:

   $$P(a,c) = 1-(1-P^{\mathrm{raw}}(a,c)) \times (1-L_{ac}^{b} \times O(a,b)). \quad (13)$$

   We can then estimate L_{ac}^{b} by solving linear equation (13), which has only a single unknown L_{ac}^{b}.

5. MODEL-DRIVEN OPTIMIZATION

5.1 Iterative Model-driven Optimization

The interference constraints [C7] in Figure 1 consist of Eq. (6)–(11), which capture the inter-dependency between $\{T_i\}$, $\{VLS_i\}$ and $\{P(i,j)\}$. A key challenge in optimization is that these relationships are non-convex. To address this challenge, we perform optimization in an iterative fashion, as illustrated in Figure 2. To decouple the non-linear inter-dependency between $\{T_i\}$, $\{VLS_i\}$, and $\{P(i,j)\}$, we perform the following steps in each iteration:

```
   ▷ $\mathbf{T}$: traffic rates, $\mathbf{Y}$: information, $\mathbf{P}$: loss rates
 1 initialization: $\mathbf{T}^* = \mathbf{0}$, $\mathbf{Y}^* = \mathbf{0}$, $thruput^* = 0$
 2 for $k = 1$ to $KMAX$
 3     $\mathbf{P}^* =$ estimate_loss($\mathbf{T}^*$)
 4     [$\mathbf{VLS}^*$, $\frac{\partial \mathbf{VLS}^*}{\partial \mathbf{T}^*}$] = estimate_VLS_and_partial_derivatives($\mathbf{T}^*$)
 5     derive linearized interference constraints in Eq. (14) using $\mathbf{VLS}^*$ and $\frac{\partial \mathbf{VLS}^*}{\partial \mathbf{T}^*}$
 6     construct a linear program (LP$_k$) from Figure 1 by adding linearized
       interference constraints (14), and fixing loss rates $\mathbf{P} = \mathbf{P}^*$ as constants
 7     solve (LP$_k$); let ($\mathbf{T}^{\mathrm{opt}}$, $\mathbf{Y}^{\mathrm{opt}}$) be the optimal solution
 8     $\alpha = \alpha_{\max}$; $succ =$ false
 9     while ($\alpha \geq \alpha_{\min}$) and ($succ =$ false)  // line search for a better solution
10         $\mathbf{T} = (1-\alpha) \times \mathbf{T}^* + \alpha \times \mathbf{T}^{\mathrm{opt}}$
11         $feasible =$ test_traffic_rates_feasibility($\mathbf{T}$)
12         if ($feasible$)
13             [$thruput$, $\mathbf{Y}$] = compute_OR_thruput_from_traffic_rates($\mathbf{T}$)
14             if ($thruput > thruput^*$)
15                 $thruput^* = thruput$; $\mathbf{T}^* = \mathbf{T}$; $\mathbf{Y}^* = \mathbf{Y}$;
16                 $succ =$ true; break
17             end
18         end
19         $\alpha = \alpha/2$
20     end
21     if ($succ =$ false), break; end
22 end
23 return ($thruput^*$, $\mathbf{T}^*$, $\mathbf{Y}^*$)
```

Figure 2: Iterative optimization of opportunistic routing.

1. We first fix traffic rates $\{T(f,i)\}$ to their values $\{T^*(f,i)\}$ obtained in the previous iteration and estimate the loss rates $\{P^*(i,j)\}$ as described in Section 4.3.2.
2. We then numerically compute VLS_i^* and partial derivatives $\frac{\partial VLS_i^*}{\partial T_k^*}$ from $\{T_j^*\}$ according to Eq. (7). The key observation we leverage is that when $\{T_j\}$ are given, Eq. (7) only contains a single variable, *i.e.*, VLS_i. We present the details of this step later in Section 5.2.
3. We then approximate the non-linear interference constraints given in Eq. (6) and (7) using linear constraints. This can be achieved by computing the first-order approximation to the R.H.S. of (6) as a Taylor expansion at the current T_i^*. Specifically, we use the following linearized interference constraints:

   $$\frac{T_i}{EP} \leq \frac{\tau_{\max}}{VLS_i^*} - \frac{\tau_{\max}}{(VLS_i^*)^2}\sum_k \frac{\partial VLS_i^*}{\partial T_k^*} \times (T_k - T_k^*), \quad (14)$$

   where VLS_i^* and $\frac{\partial VLS_i^*}{\partial T_k^*}$ are computed in step 2.
4. We then treat loss rates $P^*(i,j)$ as constants in Figure 1. We also add the linearized interference constraints given in Eq. (14) to the formulation in Figure 1, yielding a linear program (LP_k) that can be solved efficiently by LP solvers like `cplex`.
5. Since the linearized interference constraints are only an approximation to the true interference constraints, the optimal solution to (LP$_k$) may be infeasible under IEEE 802.11. We therefore perform a line search between the old solution and the optimal solution to (LP$_k$) to find a new set of traffic rates that are both feasible and improves the total throughput. During the line search, we need two capabilities: (i) to test whether a set of traffic rates are feasible under 802.11 (line 11 in Figure 2), and (ii) to find the maximum total throughput of opportunistic routing under such traffic rates. The former is performed as described in Section 5.2. The latter can be achieved by treating $T(f,i)$ as constants while solving the problem formulated in Figure 1.

The iterative process continues until it reaches a solution that cannot be further improved upon after enough attempts. Since the total throughput will strictly increase over each iteration, the process is guaranteed to converge. In our experiments, we conservatively limit the maximum number of iterations to 30. Our experience suggests that typically the iteration stops much earlier.

5.2 Technical Details

Our model-driven optimization framework above makes use of the following three key capabilities: (i) estimating VLS_i from traffic rates $\{T_j\}$, (ii) testing the feasibility of given traffic rates $\{T_j\}$, and (iii) computing partial derivatives $\frac{\partial VLS_i}{\partial T_k}$. Below we present details on how to support these capabilities using our model.

Estimating VLS_i from traffic rates $\{T_j\}$: To numerically derive VLS_i from given traffic rates $\{T_j\}$, let $f_i(x) \triangleq x - T_{\text{slot}} - (T_{\text{xmit}} + T_{\text{DIFS}} - T_{\text{slot}}) \times \left[1 - \prod_j \left(1 - \frac{D_{ij} \times T_j \times x}{EP}\right)\right]$. According to Eq. (7), $x = VLS_i$ is a root of $f_i(x)$. Moreover, we need $x \in \left[0, \frac{EP}{\max_j(D_{ij} \times T_j)}\right]$ to ensure $1 - \frac{D_{ij} \times T_j \times x}{EP} \geq 0$ in Eq. (7). In our technical report [39], we prove that when $x \in \left[0, \frac{EP}{\max_j(D_{ij} \times T_j)}\right]$, $f_i(x)$ is convex and has at most one root. Therefore, we can apply any univariate root-finding algorithm (*e.g.*, Matlab's fzero function) to numerically compute the root of $f_i(x)$ over interval $x \in \left[0, \frac{EP}{\max_j(D_{ij} \times T_j)}\right]$ and let the solution be VLS_i (if a root exists).

Testing the feasibility of traffic rates $\{T_j\}$: To test whether traffic rates $\{T_j\}$ are feasible, we first numerically compute VLS_i from Eq. (7) by finding a root of $f_i(x)$ over $x \in \left[0, \frac{EP}{\max_j(D_{ij} \times T_j)}\right]$ as described above. If no solution is found or if the solution VLS_i violates Eq. (6), then traffic rates $\{T_j\}$ are infeasible. Otherwise, $\{T_j\}$ are feasible.

Computing partial derivatives $\frac{\partial VLS_i}{\partial T_k}$: Eq. (7) also allows us to compute the partial derivatives $\frac{\partial VLS_i}{\partial T_k}$ for given traffic rates $\{T_j\}$, which allows us to linearize the non-linear interference constraints (see Section 5.1). Specifically, we have $\frac{\partial VLS_i}{\partial T_k} = \frac{N_{ik}}{1 - M_i}$, where $M_i \triangleq (T_{\text{xmit}} + T_{\text{DIFS}} - T_{\text{slot}}) \times P_i^{\text{idle}*} \times \sum_j \frac{D_{ij} T_j}{EP - D_{ij} T_j VLS_i}$, $N_{ik} \triangleq (T_{\text{xmit}} + T_{\text{DIFS}} - T_{\text{slot}}) \times P_i^{\text{idle}*} \times \frac{D_{ik} VLS_i}{EP - D_{ik} T_k VLS_i}$, and $P_i^{\text{idle}*} \triangleq \prod_j (1 - \frac{D_{ij} T_j VLS_i}{EP})$.

6. PROTOCOL IMPLEMENTATION

Overview: We develop a practical opportunistic routing protocol to install the opportunistic routes and rate limits computed by our optimization algorithm. It is built on top of MORE [7], which sits between the IP and 802.11 MAC layers. It differs from MORE in that it uses interference modeling and optimization to derive rate limits and opportunistic routes for a given performance objective. As in MORE, it leverages intra-flow network coding to carry out the derived routes (*i.e.*, an intermediate forwarder transmits random linear combinations of the packets it receives for a given flow at the rate derived from our optimization).

As most opportunistic routing protocols, we target medium to large file transfers. A traffic source divides data packets into batches, and broadcasts a random linear combination of the original packets at the rate computed according to Figure 2. Upon receiving encoded packets, an intermediate node generates a random linear combination of all the innovative packets it has from the current batch. Each intermediate node uses the algorithm described in Figure 2 to determine how much traffic it should forward. After receiving enough innovative packets, the destination extracts the original data packets and sends an end-to-end ACK using MAC-layer unicast. When the source receives the ACK, it moves to the next batch. Below we describe several key steps in our protocol: (i) measuring inputs to seed our interference model, (ii) computing opportunistic routes and rate limits for each flow, (iii) routing traffic according to the derived sending rates and routes, (iv) supporting multicast, and (v) enhancing the reliability of end-to-end ACKs.

Measuring input parameters: Our model-driven optimization framework has the following input parameters: (i) traffic demands, (ii) carrier sense probabilities, (iii) conditional collision loss probabilities, and (iv) inherent wireless link loss rates. As reported in [11, 23], wireless traffic exhibits temporal stability and we can estimate current traffic demands based on previous demands. In our evaluation, we also test the sensitivity to the demand estimation error. We conduct pairwise broadcast measurements [1] and compute the carrier sense probabilities $\{D_{ab}\}$ and conditional collision loss probabilities $\{L_{ac}^b\}$ as described in Section 4.3.3. The pairwise broadcast measurements takes $O(N^2)$ time for an N-node network. In our 21-node testbed, each pair of nodes broadcasts for 30 seconds, and the entire measurement takes around 2 hours. To minimize measurement overhead, we conduct pairwise broadcast measurement infrequently, around once a week. Note that recent works have developed efficient online techniques to measure interference when a network is in use (*e.g.*, [3, 4]). These techniques can be incorporated into our implementation to further reduce measurement overhead. In addition, we conduct per-node broadcast measurements at the beginning of each experiment to measure the inherent wireless link loss rates. The latter is based on more frequent measurements because it is more light-weight (only requiring $O(N)$ measurements) and existing routing protocols, such as [6, 7, 9], all use frequent loss measurements for route selection.

Deriving opportunistic routes and rate limits: Note that since our optimization problem is non-convex, existing techniques developed for distributed convex optimization (*e.g.*, [19]) are not directly applicable. Instead, we optimize opportunistic routes and rate limits at a central location and then distribute the optimized results to the other nodes. We use this approach in our implementation. The amount of information to distribute is very small compared to data traffic: the optimization input is around 2 KB per node and the optimization output is within a 100 bytes per node. Alternatively, the computation can also be done in a fully distributed fashion, similar to link-state protocols like OSPF, where every node implements the same algorithm over the same data to arrive at the same results. Such computation happens once every several minutes. For instance, default SNMP polling intervals are typically 5 minutes, so the optimization can rerun when the traffic demands and network topology change. The optimization is fairly efficient (*e.g.*, it takes around 3 seconds to optimize routes and rate limits for 16 flows under the 5×5 grid topologies used in our simulation).

Enforcing derived routes and rate limits: An intermediate node enforces its forwarding strategy according to the derived $T(f, i)$ and $Y(f, d, i, j)$ using the following credit-based scheme. When node j receives a packet from node i, it increments its credit, which denotes how many packets j should transmit for each received packet. If its credit is at least 1, j generates and transmits a random linear combination of the packets from the current batch buffered locally, and decrements the credit by 1. This process repeats until the credit goes below 1. The credit computation in our protocol differs from MORE in two main aspects. First, our protocol computes credit to ensure the traffic and information sending rates conform to the derived T and Y. Second, unlike MORE, which treats all transmissions equally if coming from nodes with larger ETX to the destination, our protocol differentiates transmissions coming from different neighbors as follows. Upon receiving a packet from i, j increments its credit by $C \times R$, where C reflects the fraction of useful information contained in each packet received from i and R reflects the amount of redundancy j should include to compensate for loss to its neighbors. Specifically, we have $C = \frac{Y(f,d,i,j)}{T(f,i)(1-P(i,j))}$, and $R = \frac{T(f,j)}{\sum_k Y(f,d,j,k)}$. For example, when j receives a packet from a downstream node i, $C = 0$ to prevent j from sending non-innovative packets; when receiving a packet from an upstream node i, j updates its credit according to how much new information is involved in the packet and its loss rate to its forwarders.

Supporting multicast extension: Our previous description applies to the unicast case. A few modifications are required to support multicast. First, since a single packet carries a different amount of information for different destinations in the same multicast group, a node j increments its credit by $C \times R$, where $C = \frac{\max_d Y(f,d,i,j)}{T(f,i)(1-P(i,j))}$ and $R = \frac{T(f,j)}{\sum_k \max_d Y(f,d,j,k)}$. Second, when some destinations receive enough innovative packets, the encoded packets from the current batch should only be delivered to those who have not received all packets. To adapt to the changes in the set of destinations that need the packets, we dynamically re-adjust credit increment based on the remaining receivers who have not finished.

Enhancing ACK reliability: The destination sends an end-to-end ACK to the source upon receiving enough innovative packets for decoding so that the source can move on to the next batch. To ensure the reliability of ACKs, we keep retransmitting ACKs until they are received. To expedite ACK transmissions, ACKs do not perform binary backoff so that they have higher priority over retransmitted data. For fair comparison, we apply the same optimizations to MORE. Finally, since there is only one ACK for a batch of data packets, ACK overhead is negligible in opportunistic routing.

7. EVALUATION METHODOLOGY

We evaluate our approach using extensive simulation and testbed experiments. Our evaluation consists of four parts. First, we compare the fidelity of the conflict-graph (CG) model and our new model by quantifying their under-prediction and over-prediction errors. We use a conservative CG model, which considers two broadcast transmissions to interfere if any one of their receivers is interfered by the other transmission.

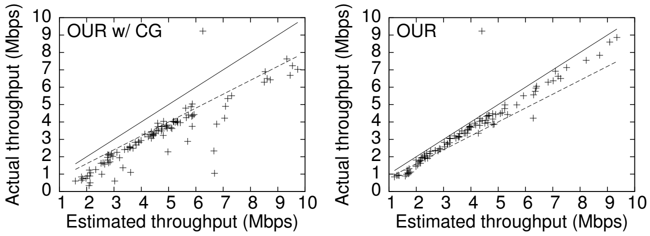
Second, we compare the performance of our opportunistic routing protocol using either the CG model or the new broadcast interference model with the following existing routing protocols: (1) shortest-path routing using the ETX routing metric, which minimizes the total number of expected transmissions from a source to its destination [9], (2) shortest-path routing with rate limit optimization as developed in [23], and (3) MORE, a state-of-art opportunistic routing protocol.

We compare total network throughput under 1–16 simultaneous flows. We also compare in terms of the proportional fairness metric [19], which is defined as: $\sum_{f \in Flows} \log G(f)$, where $G(f)$ is flow f's throughput. This metric strikes a balance between increasing network throughput and maintaining fairness among the flows. Higher values are more desirable. Unless otherwise noted, each flow sends saturated CBR traffic.

Third, we evaluate the multicast performance of one multicast group with a varying group size, and measure the average throughput of the bottleneck receiver. As in [7], we extend shortest path routing to support multicast by generating a multicast tree as a union of shortest paths towards all destinations and sending one copy of traffic along the links that are shared by multiple destinations. It saves the traffic on shared point-to-point links as in wireline multicast routing but does not leverage the broadcast nature of wireless links (*e.g.*, a node still needs to send traffic separately to reach each of its next hops). Shortest path with rate limit [23] takes a routing matrix R as part of the input, where R_{id} is the fraction of flow d that traverses link i. To support multicast, we derive a multicast routing tree R, where $R_{ig} = 1$ if link i appears in multicast group g's routing tree.

Fourth, we evaluate the sensitivity of our protocol against (i) errors in the input traffic demands, (ii) unknown external interference, and (iii) errors in link loss estimation.

For simulation, we implement all protocols in Qualnet 3.9.5 [34]. For testbed experiments, we use the shortest path routing and MORE implementations publicly available [30]. In particular, the shortest path routing is the Click implementation released as part of MORE source code. We calculate ETX according to [9] and configure the link weight accordingly. The shortest path with rate limiting is based on the shortest path code but the rate limit of each flow is computed using the algorithm in [23]. We extend MORE to implement our protocol as described in Section 6. Both MORE and our protocol use 64 packets as the batch size for network coding. All these routing protocols are implemented using Click [8] and the MadWiFi driver [27] in the testbed.



(a) CG model (b) Our model

Figure 3: Actual vs. estimated throughput in simulation (25-node random topologies).

Qualnet simulation: In simulation, we use 802.11a with a fixed MAC rate of 6 Mbps. The communication range is 230 meters, and interference range is 253 meters. These are the default values in Qualnet under transmission power of 10dBm, and we use them in the CG model to determine if two nodes interfere. As in [23], we seed the new interference model by having two senders broadcast simultaneously and measuring the resulting sending rates and receiving rates. Unless noted otherwise, we use saturated UDP traffic with 1024-byte payloads.

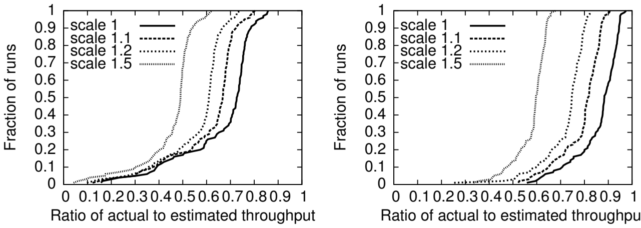
For each scenario, we conduct 20 random trials. In each trial, flow sources and destinations are picked randomly and the simulation time is 20 seconds. We extend Qualnet to generate directional inherent packet losses, which are uniformly distributed between 0 and 90%. We consider two types of topologies: 5×5 grid and 25-node random topologies, each occupying a 750×750 m^2 area.

Testbed experiments: Our testbed consists of 21 nodes located on two floors inside an office building. Each node runs Linux and is equipped with a NetGear WAG511 NIC. Unless otherwise specified, we use 802.11a to minimize interference with campus wireless LAN traffic, which uses 802.11g. This allows us to evaluate in a controlled environment. We use 20 mW transmission power and 6 Mbps transmission rate so that the network paths are up to 7 hops. Among the node pairs that have connectivity, 47.8% of them have links with loss $\leq$ 20%. All the routing protocols require estimation of link loss rates, which are measured by having one sender broadcast at a time and the other nodes measure the receiving rates. The loss measurements were collected before the experiments. In addition, our protocol and shortest path with rate limiting require interference measurement, which we collected once per week. As in simulation, we conduct 20 random trials for each scenario. Each trial lasts one minute. Other settings are consistent with the simulation. Finally, in Section 10, we further evaluate using 802.11b, which competes with campus WLAN traffic, in order to assess the sensitivity against unknown external interference.

8. MODEL VALIDATION

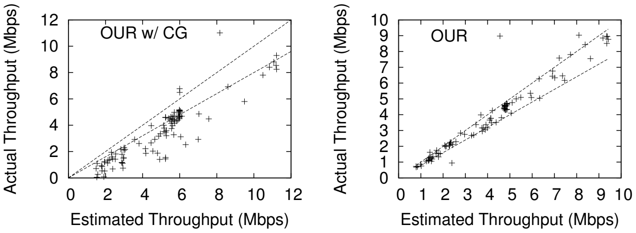
We adopt the evaluation methodology presented in [23] to quantify the accuracy of our model. In particular, to evaluate the over-prediction of our model, we install the estimated throughput to the network to see if it can be satisfied. To evaluate the under-prediction error, we uniformly scale each flow throughput by the same factor and check if the scaled demand is achievable. If the scaled demand is achieved in the network, it indicates that the under-prediction error is at least the scaling factor. We vary the scaling factor from 1.1, 1.2, 1.5, corresponding to a load increase of 10%, 20%, and 50%, and vary the number of flows from 1 to 16.

Simulation results: We first evaluate how often the models over-predict. In Figure 3, we plot the estimated throughput versus the actual throughput using the CG model and our model in 25-node



(a) CG model (b) Our model

Figure 4: CDF of ratios between actual and estimated throughput in simulation (25-node random topologies).



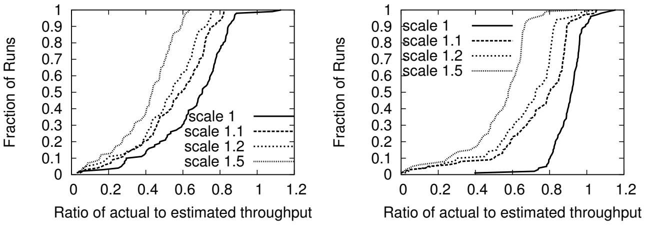
(a) CG model (b) Our model

Figure 5: Actual vs. estimated throughput in the testbed.

random topologies. For reference, we plot lines $y = x$ and $y = 0.8x$. Here, the CG model significantly over-predicts the actual throughput obtained, whereas the actual performance under our model is mostly within 80% of the estimated throughput. The CG model experiences significantly higher over-prediction errors since it assumes perfect scheduling, whereas our model explicitly models the interference between broadcast transmissions in IEEE 802.11, thereby achieving higher accuracy. Moreover, the amount of over-estimation by CG heavily depends on the network topology (*e.g.*, whether the network has hidden terminals) and simply scaling down the performance estimated by CG by a constant factor does not work. Both CG and our models have part of their over-prediction errors coming from the delay in end-to-end ACKs, during which time the source keeps retransmitting the current batch. This effect is not modeled. The use of a larger batch size can reduce the gap between the model estimation and actual performance at the cost of a larger header size and longer delay.

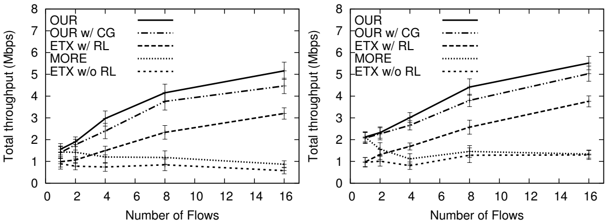
Next we quantify under-prediction errors. In Figures 4(a) and (b), we plot CDFs of the ratios between actual and estimated throughput in random topologies for the CG model and our model, respectively. Consistent with the scatter plots, the CG model mostly over-predicts, and virtually none of the scaled demands are satisfied. In comparison, using our model with a scale factor of 1, 80% of the runs have actual throughput within 80% accuracy of the estimation. Increasing the scale factor to 1.1 causes 65% of the actual throughput to be within 80% accuracy. After a further increase of the scale factor to 1.2, only 11% of actual throughput falls into 80% accuracy. This indicates that the demands scaled up by 20% can rarely be satisfied and shows our model has low under-prediction errors.

Testbed results: Next we validate our model and the CG model using testbed experiments. Figures 5(a) and (b) show the scatter plots of the CG model and our model, respectively. Figures 6(a) and (b) plot the CDFs of the ratios between actual and estimated throughput using different scale factors. As in simulation, the scatter plots from testbed experiments show a good match between actual and estimated throughput using our model and a significant over-estimation in the CG model. Scaling the demands by 1.1 leads to only 50% of the demands being satisfied and scaling the demands by 1.2 leads to only 29% of the demands being satisfied. These results indicate low over-prediction and under-prediction error. There are a few points in the testbed results where the actual throughput is higher than the estimated throughput. These cases arise from loss fluctuation: we use loss measurements to seed our model and derive opportunistic routes and rate limits, but the actual link loss rates in the experiment improve and support higher throughput.



(a) CG model (b) Our model

Figure 6: CDF of ratios between actual and estimated throughput in the testbed.



(a) 5×5 grid (b) 25-node random topology

Figure 7: Total unicast throughput in simulation (25-node random topologies).

Summary: The simulation and testbed results demonstrate that our model rarely over-estimates or under-estimates performance by more than 20%. In comparison, the CG model consistently over-predicts network throughput due to its assumption of perfect scheduling. These results highlight the importance of model fidelity on performance predictability.

9. PERFORMANCE COMPARISON

In this section, we compare the performance of different routing protocols using simulation and testbed experiments.

9.1 Simulation Results

Total throughput of unicast flows: Figures 7(a) and (b) show the total throughput for 5×5 grid and 25-node random topologies, respectively. The error bars on the graph show the standard deviation of the sample mean.

We make several observations. First, in all cases our protocol using our model yields the best performance. It out-performs ETX by 76%-799% in the grid topology and by 117%-327% in the random topologies. Its gain over ETX with rate limiting ranges from 57%-99% in the grid topology and 46%-117% in the random topology. Its gain over MORE increases rapidly with the number of flows, ranging from 34% (2 flow) to 146% (4 flows) to 501% (16 flows) in the grid topology, and from 50% (2 flows) to 169% (4 flows) to 311% (16 flows) in random topologies. It out-performs the protocol with CG, the second best performing protocol by up to 24% in the grid topologies and 16% in the random topologies. Its performance benefit comes from three main factors: (i) taking advantage of opportunistic transmissions to cope with lossy wireless links, (ii) using interference-aware rate limiting to avoid network congestion, and (iii) using interference-aware opportunistic routing to maximize spatial reuse.

Second, comparing MORE against shortest path rate limiting, we observe that MORE out-performs the latter under 1 or 2 flows by leveraging opportunistic transmissions to recover losses. As the number of flows increases, the performance of MORE degrades and becomes significantly worse than shortest path with rate limiting due to lack of rate limiting. The impact of rate limiting on opportunistic routing is even higher than shortest path routing because opportunistic routing uses broadcast transmissions, which do not

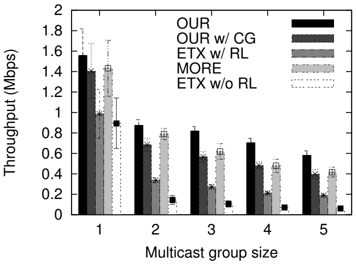
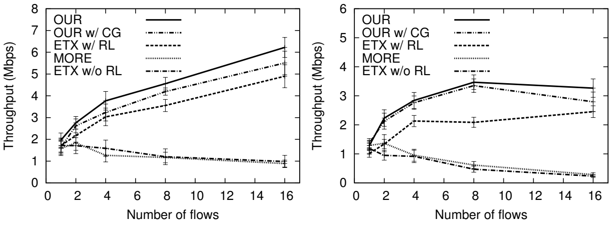


Figure 8: Multicast throughput in a 5×5 grid.



(a) Random flow selection (b) Flows with ETX > 1.25

Figure 9: Total unicast throughput in the testbed.

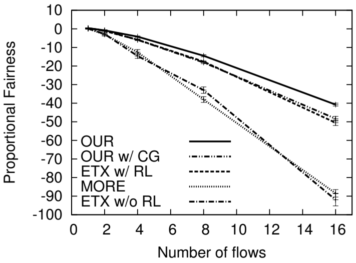


Figure 10: Unicast proportional fairness in the testbed.

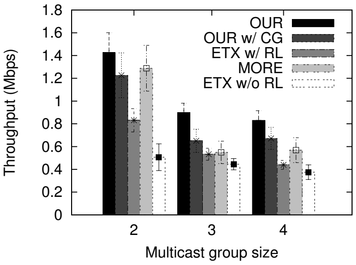


Figure 11: Multicast throughput in the testbed.

have binary backoff and are more likely to cause network congestion. Further, congestion on the data path may corrupt end-to-end ACKs in opportunistic routing and lead to unnecessary retransmissions and throughput degradation. In contrast, shortest path routing uses unicast transmissions, whose MAC-layer ACKs are given higher priority and hence more reliable.

Multicast flows: Figure 8 shows the throughput of the bottleneck receiver in a multicast group as we vary the group size from 1 to 5. As in unicast flows, our protocol consistently out-performs the alternatives. It improves the protocol with CG by 10%-46%, MORE by 8%-47%, shortest path rate limiting by 58%-232%, and shortest path by 74%-894%. The larger performance gain over both versions of shortest path is because our protocol effectively exploits the broadcast nature of the wireless medium to reduce the number of transmissions. When sending to multiple neighbors, it uses one broadcast transmission to reach all the neighbors. In comparison, while shortest path routing uses a multicast tree to compress the traffic on a shared link, the links from one sender to different neighbors are considered different and multiple transmissions are required to reach them. For the same reason, MORE consistently out-performs both versions of shortest path routing. Our protocol still out-performs MORE and the protocol with CG by using a more accurate model to optimize rate limit and opportunistic routes.

9.2 Testbed Results

Throughput of unicast flows: Figure 9(a) shows the total throughput of different protocols in the testbed, which has up to 7 hops. The relative rankings of the routing schemes are consistent with the simulation. As before, our protocol yields the best performance. The links in our testbed tend to be binary: either low loss or close to no connectivity. Among the node pairs that have network connectivity, 47.8% of them have loss rate within 20%. So the benefit of opportunistic routing is smaller in the testbed than in simulation. MORE performs close to shortest path routing, and significantly worse than shortest path with rate limiting; similarly, the gap between our protocol and shortest path routing also becomes smaller. These results confirm the intuition that opportunistic routing is most useful under lossy wireless medium.

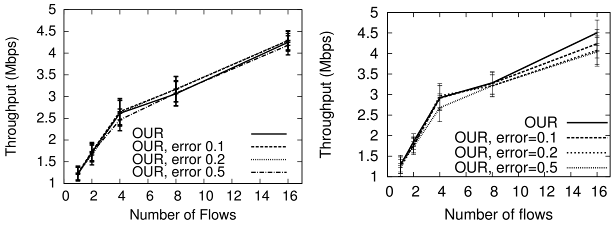
To understand how opportunistic routing performs under more lossy wireless medium, we conduct another set of experiments where we pick only flows whose ETX between source and destination is at least 1.25. Figure 9(b) summarizes the results. In this case, the throughput of our protocol is 1.09-14.0x that of shortest path without rate limiting and 1.26-1.67x that of shortest path with rate limiting. Its throughput is similar to MORE under 1 flow and 11.47x MORE's throughput under 16 flows. Furthermore, MORE yields low throughput: its performance is worse than shortest path with rate limiting as the number of flows reaches 4 or higher. These results are consistent with the simulation, and highlight the importance of jointly optimizing rate limits and opportunistic routes.

Proportional fairness of unicast flows: Next we consider maximizing proportional fairness. Since this objective is non-linear, in order to optimize it, we first approximate it using a piecewise linear, increasing, convex function as follows. We select s points on $\log(x)$, and approximate $\log(x)$ using s line segments, each connecting two adjacent points. We perform two different point selections and observe similar performance. In the interest of space, below we present results from only one selection: $x = 0.001$, 0.01, 0.1, $\sqrt{0.1}$, 1, $\sqrt{10}$, 10. When a flow's throughput is 0, its log value is undefined, so we set its throughput to 1 Kbps. Figure 10 shows the proportional fairness as we vary the number of unicast flows in the testbed. The three routing schemes that support rate limiting significantly out-perform MORE and shortest path without rate limiting since the latter two can easily cause starvation. Among those that support rate limit, our protocol performs the best due to its opportunistic routing and high-fidelity model.

Multicast flows: Finally, we evaluate the performance of multicast in our testbed. Figure 11 shows the throughput of the bottleneck multicast receiver in one multicast group, where the multicast group size is varied from 2 to 4. Our protocol performs the best. It out-performs the protocol with CG by 16%-38%, MORE by 10%-63%, shortest path with rate limiting by 68%-89%, and shortest path routing by 101%-181%. In addition, by leveraging the broadcast wireless medium, all types of opportunistic routing, including MORE, out-perform both versions of shortest path routing. These results suggest opportunistic routing is even more useful to multicast, and the effective optimization of multicast routes and rate limiting continues to be important.

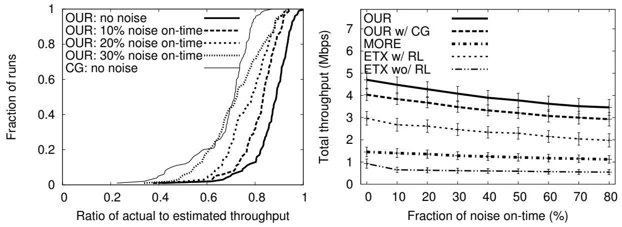
9.3 Summary of Performance

The simulation and testbed results show that our protocol consistently out-performs the alternatives. By leveraging opportunistic transmissions and effective route optimization, it significantly out-performs state-of-the-art shortest path routing protocols. By using a high fidelity network model to jointly optimize rate limits and opportunistic routes, it significantly out-performs state-of-the-art opportunistic routing protocols. These benefits suggest that all the



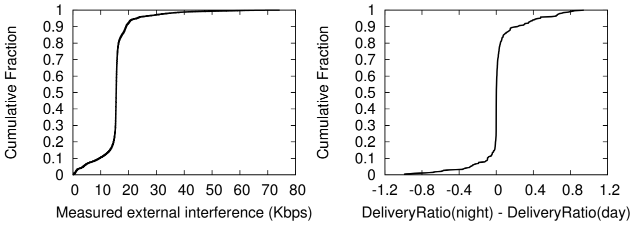
(a) 5×5 grid in simulation (b) 802.11a Testbed

Figure 12: Throughput under inaccurate traffic estimates.



(a) CDF of ratios between actual and estimated throughput (b) Total throughput of 8 flows

Figure 13: Simulation results under 2 noise sources with varying on-time in 25-node 802.11a random topologies.



(a) Amount of external traffic (b) CDF of loss fluctuation

Figure 14: Amount of external traffic from the campus network and loss fluctuation in our 802.11b testbed.

design components in our protocol, including opportunistic routing, network model, and joint rate limit and route optimization, are essential and help improve the performance.

10. EVALUATION OF SENSITIVITY

10.1 Impact of Inaccurate Traffic Demand

Methodology: We first evaluate the performance under inaccurate traffic demand estimation, since in practice traffic demands fluctuate and may not be known exactly. The actual traffic demands are uniformly distributed between 0 and the maximum link throughput. To simulate demand estimation error, we inject errors into the actual demands and feed the salted demands to our optimization framework while imposing the actual demands to the network for evaluation. The error injected is uniformly distributed between 0-10%, 0-20%, and 0-50%. To protect against estimation error, our protocol slightly over-provisions by scaling the derived sending rates from the optimization output by a factor of 1.1.

Simulation: Figure 12(a) shows the total throughput versus the number of flows. We see similar performance across different error ranges. This indicates that our protocol is fairly robust against demand estimation errors, because for the purpose of performance optimization, the spatial traffic demand distribution is more important than the exact demand values.

Testbed: Figure 12(b) shows the performance of our protocol when we feed inaccurate traffic demands as input to our optimization. As in simulation, it is robust to the inaccuracy in traffic demand estimation in testbed. Its performance under no error is close to that under the relative error of 0.5.

10.2 Impact of Unknown External Interference and Loss Fluctuation

10.2.1 Simulation

Methodology: We create external interference by randomly placing two external noise sources in 25-node random topologies. All protocols compute routes and rate limits without considering the external noise, and we measure the throughput of using the derived routes and rate limits under external noise. The noise sources have uniformly distributed on and off time, where the average on-time is 0.25 second and the total simulation time is 20 seconds. We vary the average off-time so that every noise source is on 10% to 80% of time. During on-time, each noise source broadcasts 802.11 packets (with 1024-byte payload) as fast as possible.

Model validation: First, we compare actual throughput under external noise versus estimated throughput derived without considering the noise sources. As shown in Figure 13(a), the accuracy of our protocol degrades gracefully as we increase the on-time of each noise source. The fractions of runs that achieve within 30% error are 99% under 10% noise on-time, 76% under 20% noise on-time, and 56% under 30% noise on-time. Moreover, even with 30% noise on-time, it achieves much higher predictability than the protocol with CG model under no external noise.

Performance comparison: As shown in Figure 13(b), the ranking of different protocols remains the same across all noise levels. Our protocol consistently out-performs all other protocols. Even when every noise source is active 80% of time, it out-performs the one with CG by 18%, shortest path with rate limiting by 75%, MORE by 209%, and shortest path without rate limiting by 535%. Moreover, the performance of different protocols degrades smoothly as the on-time of each noise source increases.

10.2.2 Testbed

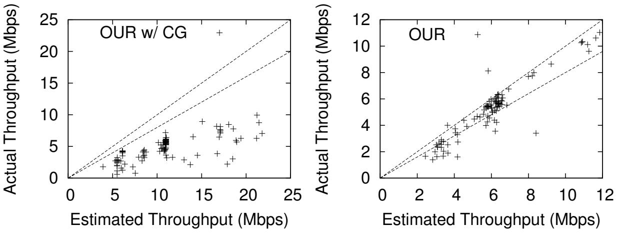
Methodology: We also evaluate the sensitivity in an 802.11b testbed consisting of 22 nodes. As before, we randomly select flows in our network. As common practice, we run the link loss measurements at night, which has low network activity. Then we run all evaluation during the day. This allows us to evaluate the sensitivity against unknown external interference and loss fluctuation. In particular, our building has an active 802.11g campus network, whose traffic directly interferes with our wireless mesh traffic. We treat traffic from the campus network as unknown external interference. Figure 14(a) plots the CDF of the average campus network traffic measured by all mesh nodes in promiscuous mode every 30 seconds. The median and mean are both 15.5 Kbps. Moreover, loss fluctuates from nights to daytime. Figure 14(b) plots a CDF of $DeliveryRatio(night) - DeliveryRatio(day)$ over all links that have $\geq 5\%$ delivery rates. We observe loss fluctuation, because during the day time (i) more people sit near mesh nodes and cause more attenuation, and (ii) more people move around and close/open doors and cause frequent changes to the RF environment.

Model validation: Figure 15 shows the scatter plot of actual versus estimated throughput from the 802.11b testbed. We also plot $y = x$ and $y = 0.8x$ for reference. Our protocol continues to exhibit high predictability: 78% of runs have within 20% error.

Performance comparison: As shown in Figure 16, our protocol continues to perform the best. Different from simulation, MORE sometimes performs worse than shortest path without rate limiting because the network congestion in MORE is more severe in a dense network like our 802.11b testbed.

11. CONCLUSION

In this paper, we present the first protocol that can accurately optimize the performance of opportunistic routing in IEEE 802.11 networks. Our framework consists of three key components: (i) a simple yet effective wireless network model to support optimiza-



(a) CG model (b) Our model

Figure 15: Actual vs. estimated throughput in 802.11b testbed under unknown external interference and loss fluctuation.

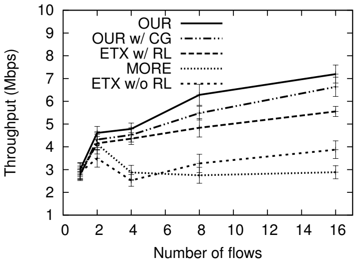


Figure 16: Unicast throughput in our 802.11b testbed under unknown external interference and loss fluctuation.

tion, (ii) a novel algorithm for optimizing different performance objectives, and (iii) an opportunistic routing protocol that effectively maps solutions resulted from our optimization into practical routing configurations. Through testbed implementation and simulation, we show that the performance of our protocol is close to our estimation, and is much better than state-of-the-art shortest path routing and opportunistic routing protocols. Moreover, it is robust against inaccuracy introduced by a dynamic network and it also consistently out-performs the existing schemes. To further enhance the robustness against traffic and topology variations, in the future we plan to extend the robust traffic engineering techniques developed in the Internet to optimize wireless networks. In particular, a traffic engineering system usually collects a set of traffic matrices and uses their convex combination to cover the space of common traffic patterns for optimization. These new demand constraints are compact and can be easily incorporated into our framework. We plan to extend this technique to cope with both traffic and topology variations in wireless networks.

Acknowledgments: This research is supported in part by NSF grants CNS-0916106, CNS-0546755, and CNS-0546720. We thank Sem Borst and anonymous reviewers for their valuable comments.

12. REFERENCES

[1] S. Agarwal, J. Padhye, V. N. Padmanabhan, L. Qiu, A. Rao, and B. Zill. Estimation of link interference in static multi-hop wireless networks. In *Proc. of IMC*, 2005.

[2] D. Aguayo, J. Bicket, S. Biswas, G. Judd, and R. Morris. Link-level measurements from an 802.11b mesh network. In *Proc. of SIGCOMM*, 2004.

[3] N. Ahmed, U. Ismail, S. Keshav, and K. Papagiannaki. Online estimation of RF interference. In *Proc. of ACM CoNext*, Dec. 2008.

[4] N. Ahmed and S. Keshav. SMARTA: A self-managing architecture for thin access points. In *Proc. of ACM CoNEXT*, Dec. 2006.

[5] G. Bianchi. Performance analysis of the IEEE 802.11 distributed corrdination function. *IEEE Journal on Selected Areas in Communications*, Mar. 2000.

[6] S. Biswas and R. Morris. ExOR: Opportunistic multi-hop routing for wireless networks. In *Proc. of ACM SIGCOMM*, Aug. 2005.

[7] S. Chachulski, M. Jennings, S. Katti, and D. Katabi. Trading structure for randomness in wireless opportunistic routing. In *Proc. of SIGCOMM*, 2007.

[8] Click. `http://pdos.csail.mit.edu/click/`.

[9] D. D. Couto, D. Aguayo, J. Bicket, and R. Morris. A high-throughput path metric for multi-hop wireless routing. In *Proc. of ACM MobiCom*, 2003.

[10] H. Dubois-Ferrier, M. Grossglauser, and M. Vetterli. Least-cost opportunistic routing. In *Proc. of Allerton*, Sept. 2007.

[11] H. Feng, Y. Shu, S. Wang, and M. Ma. SVM-based models for predicting WLAN traffic. In *Proc. of IEEE ICC*, 2006.

[12] Y. Gao, J. Lui, and D. M. Chiu. Determining the end-to-end throughput capacity in multi-hop networks: Methodolgy and applications. In *Proc. of ACM SIGMETRICS*, Jun. 2006.

[13] M. Garetto, T. Salonidis, , and E. Knightly. Modeling per-flow throughput and capturing starvation in CSMA multi-hop wireless networks. In *Proc. of IEEE INFOCOM*, Mar. 2006.

[14] M. Garetto, J. Shi, and E. Knightly. Modeling media access in embedded two-flow topologies of multi-hop wireless networks. In *Proc. of ACM MobiCom*, Aug. - Sept. 2005.

[15] M. Grossglauser and D. N. C. Tse. Mobility increases the capacity of ad hoc wireless networks. In *Proc. of IEEE INFOCOM*, Apr. 2001.

[16] P. Gupta and P. R. Kumar. The capacity of wireless networks. *IEEE Transactions on Information Theory*, 46(2), Mar. 2000.

[17] K. Jain, J. Padhye, V. N. Padmanabhan, and L. Qiu. Impact of interference on multi-hop wireless network performance. In *Proc. ACM MobiCom*, 2003.

[18] A. Kashyap, S. Das, and S. Ganguly. A measurement-based approach to modeling link capacity in 802.11-based wireless networks. In *Proc. of ACM MobiCom*, Sept. 2007.

[19] F. P. Kelly, A. K. Maulloo, and D. K. H. Tan. Rate control in communication networks: Shadow prices, proportional fairness and stability. *Jounal of the Operational Research Society*, 1998.

[20] D. Koutsonikolas, C.-C. Wang, and Y. C. Hu. CCACK: Efficient network coding based opportunistic routing through cumulative coded acknowledgments. In *Proc. of IEEE INFOCOM*, 2010.

[21] A. Kumar, E. Altman, D. Miorandi, and M. Goyal. New insights from a fixed point analysis of single cell IEEE 802.11 wireless LANs. In *Proc. of IEEE INFOCOM*, Mar. 2005.

[22] J. Li, C. Blake, D. S. J. D. Couto, H. I. Lee, and R. Morris. Capacity of ad hoc wireless networks. In *Proc. of MobiCom*, Jul. 2001.

[23] Y. Li, L. Qiu, Y. Zhang, R. Mahajan, and E. Rozner. Predictable performance optimization for wireless networks. In *Proc. of ACM SIGCOMM*, Aug. 2008.

[24] Y. Lin, B. Li, and B. Liang. CodeOR: Opportunistic routing in wireless mesh networks with segmented network coding. In *Proc. of IEEE ICNP*, Oct. 2008.

[25] Y. Lin, B. Liang, and B. Li. SlideOR: Online opportunistic network coding in wireless mesh networks. In *Proc. of IEEE INFOCOM*, Mar. 2010.

[26] D. Lun, M. Medard, and R. Koetter. Network coding for efficient wireless unicast. *International Zurich Seminar on Communications*, 2006.

[27] MadWiFi. `http://madwifi.org`.

[28] A. K. Miu, H. Balakrishnan, and C. E. Koksal. Improving loss resilience with multi-radio diversity in wireless networks. In *Proc. of ACM MobiCom*, 2005.

[29] A. K. Miu, G. Tan, H. Balakrishnan, and J. Apostolopoulos. Divert: Fine-grained path selection for wireless LANs. In *Proc. of ACM MobiSys*, 2004.

[30] MORE source code. `http://people.csail.mit.edu/szym/more/README.html`.

[31] D. Niculescu. Interference map for 802.11 networks. In *Proc. of IMC*, 2007.

[32] L. M. S. C. of the IEEE Computer Society. Wireless LAN medium access control (MAC) and physical layer (PHY) specifications. *IEEE Standard 802.11*, 1999.

[33] L. Qiu, Y. Zhang, F. Wang, M. K. Han, and R. Mahajan. A general model of wireless interference. In *Proc. of ACM MobiCom*, Sept. 2007.

[34] The Qualnet simulator from Scalable Networks Inc. `http://www.scalable-networks.com/`.

[35] B. Radunovic and J. Y. L. Boudec. Rate performance objectives of multihop wireless networks. In *Proc. of IEEE INFOCOM*, Apr. 2004.

[36] B. Radunovic, C. Gkantsidis, P. Key, and P. Rodriguez. An optimization framework for opportunistic multipath routing in wireless mesh networks. In *Proc. of IEEE INFOCOM*, Apr. 2008.

[37] C. Reis, R. Mahajan, M. Rodrig, D. Wetherall, and J. Zahorjan. Measurement-based models of delivery and interference. In *Proc. of ACM SIGCOMM*, 2006.

[38] M. Rodrig, C. Reis, R. Mahajan, D. Wetherall, and J. Zahorjan. Measurement-based characterization of 802.11 in a hotspot setting. In *Proc. of E-WIND*, Aug. 2005.

[39] E. Rozner, M. K. Han, L. Qiu, and Y. Zhang. Model-driven optimization of opportunistic routing. Technical Report TR-11-12, The University of Texas at Austin, Dept. of Computer Science, Austin, TX, 2010.

[40] T. Salonidis, M. Garetto, A. Saha, and E. Knightly. Identifying high throughput paths in 802.11 mesh networks: A model-based approach. In *Proc. of IEEE ICNP*, Oct. 2007.

[41] S. Sengupta, S. Rayanchu, and S. Banerjee. An analysis of wireless network coding for unicast sessions: The case for coding-aware routing. In *Proc. of IEEE INFOCOM*, Apr. 2007.

[42] J. K. Shapiro, D. Towsley, and J. Kurose. Optimization-based congestion control for multicast communications. *IEEE Communication Magazine*, 2002.

[43] F. Soldo, A. Markopoulou, and A. Toledo. A simple optimization model for wireless opportunistic routing with intra-session network coding. In *Proc. of IEEE NetCod*, Jun. 2010.

[44] K. Srinivasan, M. Jain, J. I. Choi, T. Azim, E. S. Kim, P. Levis, and B. Krishnamachari. The kappa factor: Inferring protocol performance using inter-link reception correlation. In *Proc. of ACM MobiCom*, 2010.

[45] J. J. T. Ho and H. Viswanathan. On network coding and routing in dynamic wireless multicast networks. In *Proc. of Workshop on Information Theory and its Applications*, 2006.

[46] K. Zeng, W. Lou, and H. Zhai. On end-to-end throughput of opportunistic routing in multirate and multihop wireless networks. In *Proc. of IEEE INFOCOM*, Apr. 2008.

[47] K. Zeng and Z. Y. W. Lou. Oportunistic routing in multi-radio multi-channel multi-hop wireless networks. In *Proc. of IEEE INFOCOM*, Mar. 2010.

[48] X. Zhang and B. Li. Optimized multipath network coding in lossy wireless networks. In *Proc. of IEEE ICDCS*, 2008.