Robust Routing and Scheduling in Wireless Mesh Networks under Dynamic Traffic Conditions

Wei Wang, *Student Member, IEEE,* Xin Liu, *Member, IEEE,* and Dilip Krishnaswamy, *Member, IEEE*

Abstract—Joint routing-and-scheduling has been considered in wireless mesh networks for its significant performance improvement. While existing work assumes it, accurate traffic information is usually not available due to traffic dynamics, as well as inaccuracy and delay in its measurement and dissemination. In addition, the joint routing and scheduling usually requires a centralized controller to calculate the optimal routing and scheduling and distribute such policies to all the nodes. Thus, even if the accurate traffic information is always available, the central controller has to compute the routing and scheduling repeatedly because the traffic demands change continuously. This leads to prohibitive computation and distribution overhead. Therefore, in this paper, we propose a joint routing-scheduling scheme that achieves robust performance under traffic information uncertainty. In particular, it achieves worst-case optimal performance under a range of traffic conditions. This unique feature validates the use of centralized routing and scheduling in wireless mesh networks. As long as the traffic variation is within the estimation range, the routing and scheduling do not need to be recomputed and redistributed. Through extensive simulations, we show that our proposed scheme meets the objective (i.e., optimize the worst-case performance). Moreover, although it only guarantees the worst-case performance in theory, its average performance is also good. For example, our proposed scheme can perform better than a fixed optimal routing and scheduling scheme in more than 80% of 500 random traffic instances. Our scheme provides insights on the desired properties of multipath routing, namely, spatial reuse and load balancing.

Index Terms—robust, routing, scheduling, traffic-oblivious, wireless mesh.

1 Introduction

There has been a significant amount of interest in wireless mesh networking for its deployment flexibility and low cost. Joint routing and (TDM-based) scheduling has been considered in wireless mesh networks to improve performance [1], [2], [3], [4]. Most existing work in joint routing and scheduling assumes accurate traffic information, e.g., active source-destination (s-d) pairs and the traffic demand for each pair. However, accurate traffic information is hardly available because traffic is dynamic and its accurate measurement is rarely available. In addition, dissemination of traffic information incurs delay and signaling overhead.

In this paper, we propose a robust routing-and-scheduling scheme to handle traffic dynamics and traffic information uncertainty. To elaborate, the proposed scheme optimizes the worst-case performance (in terms of relative congestion level) for a range of traffic conditions and it performs under a whole spectrum of traffic information granularity. On the one end of the spectrum where prefect traffic information is available, the proposed scheme achieves the same performance as other optimal routing and scheduling schemes suggested in the literature. On the other end of the spectrum where there is no traffic information available, the proposed scheme achieves robust performance while other schemes cannot generate desired routing and schedule for the lack of information. In general, the amount of traffic information available is between the two extremes. For example, we may know that the traffic pattern is "single-sink", i.e., all nodes communicate with a common sink node (e.g., a portal or gateway in wireless mesh networks). In addition, the traffic demand between an s-d pair can be estimated to be within a certain range (e.g., 2.0Mbps-2.6Mbps) [1]. Such traffic information can be utilized in the proposed scheme. The more accurate the information, the better the performance of the proposed scheme. In addition, the scheme is robust even when the estimate deviates from the reality.

Routing has been extensively studied in multi-hop wireless networks [5], [6]. Most of the routing protocols are shortest path routing protocols, although the path length can be defined using metrics other than the number of hops or the actual distance [7]. Multipath routing has also been studied for fault tolerance and performance improvement [8], [9]. Routing only without the consideration of the interference can lead to poor performance in wireless networks. For example, [10]

- *Wei Wang and Xin Liu are with the Department of Computer Science, University of California, Davis, CA 95616. Email: {wangw, liu}@cs.ucdavis.edu.*
- *Dilip Krishnaswamy is with the Advanced Technology Research and Development, Chief Scientist Office, Qualcomm Corporation, San Diego, CA 92121. Email: dilip@ieee.org.*

1. In this paper, we use "traffic pattern" to denote the active source-destination pairs, and "traffic demand" to denote the actual throughput requirement for a particular source-destination pair.

Digital Object Indentifier 10.1109/TMC.2009.66 1536-1233/$25.00 © 2009 IEEE

shows that in a random topology with heavy traffic load, shortest path routing leads to heavy congestion, and correspondingly significant interference in the center of the network.

Therefore, joint routing and scheduling has been studied to achieve optimal performance under various scenarios. In [1], interference constraint is added into the multi-commodity flow formulation of the network to form a single linear program (LP). The active s-d pairs are required information for the formulation. It is implied that the actual demand for each s-d pair is saturated. However, traffic demand can be arbitrary in practice. Different demand vector may lead to different optimal routing and scheduling policies. Cruz et al. [2] considers joint routing, scheduling and power control. They assume every link in the network is active and thus do not consider end-to-end flows. In [4], a joint channel allocation, routing, and scheduling scheme is proposed for multi-channel multi-radio wireless networks. This formulation also requires similar information as in [1]. There are two problems which affect the practical deployment of the existing centralized routing and scheduling approaches. First, optimal routing and scheduling schemes assume accurate traffic information, which is often not available. Second, joint routing and scheduling optimization requires a central controller to collect the topology and traffic information of the whole network, compute the optimal routing and scheduling policies, and then distribute the policies to each individual node. Thus, even if the accurate traffic information is always available, the optimal policies have to be repeatedly computed and distributed since traffic demands vary over time. Although the centralized routing and scheduling approach can greatly improve the throughput performance of the network, such prohibitive *overhead* prevents it from being deployed.

Researchers have made great progress in traffic-oblivious routing. A traffic-oblivious routing scheme requires limited information on traffic patterns/demands and achieves worst-case optimal performance under the given range of traffic conditions. A recent seminal work of Räcke [11] shows that for any network there is an oblivious routing with a polylog competitive ratio with respect to congestion. In a follow-up work [12], the authors show how to compute the oblivious ratio in polynomial time. However, their formulation consists of polynomial number of variables, but infinite number of constraints. In [13], the authors further reduce the complexity to compute the oblivious ratio of a given network through a single LP with polynomial number of variables and constraints. In [14], the authors apply similar approaches to derive an energy-efficient traffic-oblivious routing scheme for wireless sensor networks. Our work is motivated by previous research. In the routing-only papers [11], [12], [13], [14], the main challenge is to handle the infinite number of possible traffic demands for optimal worst-case performance. Our contribution is to formulate a joint routing-and-scheduling scheme that takes into account of interference among links, which is critical and unique in a wireless network. The existing algorithm does not imply ours, as explained in more detail in Section 3.1.

Recently, Kashyap et.al.[15] proposed a joint routing, scheduling and power control framework that does not assume knowledge of the traffic matrix. However, they assumed a simplistic interference model where two links do not interfere with each other as long as no end point of the links is shared. While this interference model can largely reduce the computational complexity, it cannot capture the interference relationship in many practical wireless networks. Since the model is integrated into their framework, it is difficult to extend it to capture the more practical interference relationship. In our work, we consider both graph-based and SINR-based interference model, either of which captures the more complicated interference relationship in practical wireless networks. Thus, our work can be applied to a broader set of scenarios.

In this paper, we propose a robust routing-and-scheduling scheme under traffic dynamics and traffic information uncertainty. We show that the proposed scheme achieves optimal worst-case performance, as well as good average performance under traffic dynamics. It does not have the aforementioned two implementation problems, namely information availability and prohibitive overhead. First, it does not require accurate traffic information. Specifically, it only needs a rough estimation of the traffic demand, which can range from accurate information to no information at all. Second, the routing and scheduling policies need not be recomputed and redistributed often. As long as the actual traffic is within the estimation range, no update is needed. Therefore, such a scheme is feasible and with affordable overhead. In practice, centralized structure has also been considered in wireless mesh networks to provide better quality of service, especially for enterprise applications. For example, Cisco's unified wireless network architecture enables centralized control and include additional control and management functionalities [16]. Our scheme can be applied in such architecture to further improve the system performance.

We implement the joint routing and scheduling scheme in Qualnet [17]. Through extensive simulations, we show that our proposed scheme meets the objective (i.e., optimize the worst-case performance). Moreover, although it only guarantees the worst-case performance in theory, its average performance is also good. For example, our proposed scheme can outperform a fixed optimal routing and scheduling scheme in more than 80% of 500 random traffic instances.

In case delay is a concern in the performance, we add a preprocessing stage to eliminate extra-long routes. In other words, we add the hop-count constraint (relative to the shortest-path routing) into our framework. The simulation result shows that the routing and scheduling with hop-count constraints can considerably improve the delay performance, especially when the traffic load is small. We consider the possibility to exploit the oblivious routing with the decentralized MAC in case the centralized scheduling is hard to implement (e.g., in wireless networks without centralized structure). Note that, it is different from generating the oblivious routing only as in the previous work. We actually generate the joint routing and scheduling. But we only use the routing with the 802.11 MAC. We study in detail the output from the scheme to understand the desirable properties of routing and scheduling that is robust to traffic variations. The routing generated in our framework is multi-path routing that *exploits spatial reuse* in the network and *provides tolerance to traffic dynamics.* Its tolerance to traffic dynamics is due to the load balancing effect of splitting flows to multiple paths. This is important in wireless networks because the load on a link interferes with its neighboring links. Therefore, good multi-path routing is important to achieve robust performance in a wireless network. Routes generated by our work provide insights to the design of decentralized routing and medium access schemes (i.e., sensing-based or distributed TDM schemes).

The rest of the paper is organized as follows. In Section 2, we introduce the system model and define routing and schedulability formally. In Section 3, we present our traffic-oblivious routing and scheduling (TORS) scheme. We first consider two cases: without traffic information and with some traffic information. Then we present how to incorporate hop-count constraints into our framework. Simulation results are presented in Section 4. Conclusions are in Section 6.

2 System Model

2.1 Interference

We note that various interference models have been proposed in the literature, including graph-based and SINR-based models. It is important to note that *none of our algorithms relies on the specifics of the interference model used.* The only input from the interference model required by our algorithm is the *sets of links that can be scheduled simultaneously*. Different interference models may result in different sets of links that can transmit simultaneously. As long as such an input is provided by an interference model, the proposed algorithm can be applied without modifications. We will show how different interference models can be incorporated into our framework. For simplicity, we will use graph model and SINR model to indicate graph-based interference model and SINR-based interference model, respectively.

2.1.1 Graph Model

There are several variants of the graph model [18], [19]. Usually the concept of transmission range and/or interference range are defined in graph model. A link exists between two nodes if they are within the transmission range of each other. The interference relationship defined between two links is binary, and is independent of other links. It can be generalized as a K-hop interference model where no two links within K hops can successfully transmit at the same time [19]. The mostly used model in the research community is 2-hop interference model [20], [21]. It is more realistic than the 1-hop model because the interference range of a node is usually larger than its transmission range in practice. Furthermore, it models the interference in the IEEE 802.11-based networks where the receiver also sends back the acknowledgment frame during the data transfer. Therefore, both the sender and the receiver may cause interference at other nodes.

We also use the 2-hop interference model when graph model is considered. Let l_{ij} be the link between nodes i and j. Links l_{ij} and l_{uv} interfere with each other if either of their transmissions can cause interference at the other. More precisely,

$$\begin{aligned} & l_{ij} \text{ and } l_{uv} \text{ interfere with each other} \\ & \text{if and only if} \\ & \quad i = u \text{ or } i = v \text{ or } j = u \text{ or } j = v \\ & \text{or at least one of the four links exist:} \\ & \quad l_{iu}, l_{iv}, l_{ju} \text{ and } l_{jv} \end{aligned} \tag{1}$$

We use the conflict graph to store the interference relationship among links. In the conflict graph, each vertex is a link in the original communication connectivity graph. Two vertices are connected with an edge in the conflict graph if the corresponding communication links cannot transmit simultaneously. Therefore, the conflict graph captures mutual interference between any two links. Due to space limit, please refer to Fig. 4 and Table 1 in [1] for an example of conflict graph. Links are scheduled based on maximal independent sets. In graph theory, an independent (vertex) set is a subset of vertices such that no two vertices in the subset represent an edge in the graph. A maximal independent set is therefore an independent set containing the largest possible number of vertices. The maximal independent sets are derived from the conflict graph. The vertices (the corresponding communication links in the connectivity graph) within a maximal independent set can transmit simultaneously without collision. Let I_i, $i = 1, \cdots, K$, be the maximal independent sets. The schedulability constraint under graph model can be formulated as:

$$\begin{aligned} & \sum_{i=1}^{K} \lambda_i \leq 1 \\ & g(l) \leq \sum_{l \in I_i} \lambda_i cap(l), \end{aligned} \tag{2}$$

where λ_i is the portion of time allocated to independent set I_i, $cap(l)$ is the capacity of link l, and $g(l)$ is the load on link l. A link can belong to multiple independent sets. Therefore, $\sum_{l \in I_i} \lambda_i$ is the total portion of time that link l can transmit. We denote $\lambda = \{\lambda_i\}$ the scheduling. In graph model, capacity of a link is usually considered as a constant.

It has been proved that when $K > 1$, the optimal scheduling based on K-hop interference model is NP-hard [19]. In our case, the hardness to perform optimal scheduling under 2-hop interference model is translated into finding all maximal independent sets, which is known to be NP-hard. We exploit the technique in [1] to find a subset of them within a given amount of *effort*. Thus, the scheduling on the subsets results in a lower-bound of the optimal performance. The effort needed to find the maximal independent sets is controllable and independent of the network size. Generally, the more the effort spent, the more close to the optimal performance the result will be. In our simulations, we set the number of efforts to be sufficiently large such that all maximal independent sets are found for the network we simulated. Fortunately, this is a one-time overhead. As long as the topology is not changed, the maximal independent sets will not be recomputed. In general, however, it is computationally infeasible to obtain all maximal independent sets for large networks. But [19] does show that, for the geometric graph (i.e., two nodes are connected as long as their distance are within a constant transmission range), the maximal scheduling policy can guarantee a constant factor approximation of the optimal performance under the graph model.

2.1.2 SINR Model

In SINR model, a message can be received successfully if the ratio of the received signal strength and the sum of the received interference from other senders plus the noise level is larger than a hardware-dependent threshold β. Formally, for link l_{ij}, the SINR at the receiver j is,

$$\begin{aligned} SINR(j) &= \frac{P_{ij}}{I_j + N} \\ &= \frac{P_{ij}}{\sum_{u \neq i} P_{uj} + N} \\ &\geq \beta, \end{aligned} \tag{3}$$

where P_{ij} is the power strength sent from node i and received at node j, I_j is the cumulated interference level which is the sum of received signal strength from other concurrent senders, and N is the noise level. P_{ij} is usually expressed as $P_{ij} = \frac{P}{d_{ij}^{\alpha}}$, where P is the transmission power, d_{ij} is the distance between node i and j, and α is the path-loss exponent, which usually ranges from 2 to 6.

The major difference between the SINR model and the graph model is the interference among links. The interference relationship between two links is no longer binary, and is dependent on other links. For example, link A and B may not interfere with each other when they are the only active links, neither does A and C. But if link A, B, and C are active at the same time, the cumulative interference of B and C may cause the SINR of A below the threshold. Thus, the interference exists between A and B, C, but not between A and B, or A and C. Thus, potentially all subsets of all links need to be examined to find the interference relationship, which is of exponential complexity. So it is not surprising that the optimal scheduling under SINR model is also NP-hard [22], [23].

We use a similar idea as in the graph model to obtain the performance lower-bound [1]. The conflict graph for graph model is a complete graph with weighted edges. The edge weight represents the amount of interference one end node contributes to the other. The same technique can be used to find the maximal independent sets.

The link capacity is not fixed under SINR model. It can be either expressed as the theoretical Shannon capacity, or the practical data rate after translating SINR into BER and taking into account the auto-rate algorithm used. In either case, the link capacity is a function of the received SINR. Since the schedule is performed on each maximal independent set, a link may have different SINRs when it is scheduled in different maximal independent sets. We use $cap(k, l)$ to denote the capacity of link l when it is scheduled in the independent set k. The schedulability constraint under SINR model can be formulated as:

$$\begin{aligned} &\sum_{i=1}^{K} \lambda_i \leq 1 \\ &g(l) \leq \sum_{l \in I_i} \left(\lambda_i cap(i, l) \right), \end{aligned} \tag{4}$$

Due to space limit, we will only present the framework under the graph model. It requires very minor changes to extend to the SINR model.

2.2 Traffic Patterns and Demands

We use a *traffic matrix* (TM) to represent the traffic pattern and demand. A traffic matrix is an $n \times n$ matrix where the diagonal entries are set to 0. A traffic matrix provides the traffic demand of each s-d pair. The ith row and jth column of the traffic matrix, d_{ij}, denotes the amount of traffic from the source node i to the destination node j. We use the traffic matrix to model different amount of traffic information available. We consider three cases: 1) no information is available; i.e., $0 \leq d_{ij} \leq \infty$ for all i, j; 2) known s-d pairs but no information on the corresponding traffic demand; i.e., $0 \leq d_{ij} \leq \infty$ for s-d pairs and $d_{ij} = 0$ for non-s-d pairs; and 3) known s-d pairs with a range of traffic demand; i.e., $a_{ij} \leq d_{ij} \leq b_{ij}$ for s-d pairs and $d_{ij} = 0$ for non-s-d pairs.

2.3 Routing

A routing specifies how traffic of each s-d pair is routed across the network. We denote $f = \{f_{ij}(e)\}$ a routing, where $f_{ij}(e)$ specifies the *proportion* of demand from i to j that is routed on the edge e. When f is used to route a demand d_{ij}, the actual load of this demand on edge e

is $d_{ij} f_{ij}(e)$. Thus, a routing f is formally defined as

$$\forall \text{ edges } e, \forall \text{ nodes } i, j \neq i : f_{ij}(e) \geq 0 \quad (5)$$

$$\forall \text{ nodes } i, j \neq i, k \neq i, j : \sum_{e \in \text{out}(k)} f_{ij}(e) - \sum_{e \in \text{in}(k)} f_{ij}(e) = 0 \quad (6)$$

$$\forall \text{ nodes } i, j \neq i : \sum_{e \in \text{out}(i)} f_{ij}(e) - \sum_{e \in \text{in}(i)} f_{ij}(e) = 1, \quad (7)$$

where in(k) and out(k) denote the set of edges pointing to and from node k respectively, (6) is the flow conservation constraint at the intermediate node, i.e., if a node is neither the source nor the destination of the flow, then the amount of flow "in" equals the amount of flow "out", and (7) is the flow conservation constraint at the source node. Note that the flow conservation constraint at the destination node is redundant given (6) and (7). We should also note that arbitrary multi-path routing is considered in f by definition.

3 Traffic-Oblivious Routing and Scheduling

Motivated by existing work on traffic-oblivious routing in Internet, we propose a traffic-oblivious routing-and-scheduling (TORS) scheme that captures the interference constraints in wireless networks. The objective of a traffic-oblivious scheme is to handle uncertainty in traffic information and achieve worst-case optimal performance under the given range of traffic information. We first introduce the notations.

The performance metric of a given routing-and-scheduling pair, (f, λ), with respect to a TM D, is the maximum congestion level among all links via (f,λ), defined as follows:

$$\text{CONG}(f, \lambda, D) = \max_{m} \frac{\sum_{ij} f_{ij}(m) d_{ij}}{cap(m) \sum_{m \in I_i} \lambda_i}, \quad (8)$$

where $\sum_{ij} f_{ij}(m) d_{ij}$ is the aggregated load on link m for all s-d pairs, $\sum_{m \in I_i} \lambda_i$ is the aggregated portion of time available for link m to transmit from all independent sets it belongs to, $(\sum_{ij} f_{ij}(m) d_{ij})/(cap(m) \sum_{m \in I_i} \lambda_i)$ is ratio between the aggregated load and the available capacity on link m, i.e., the congestion level of link m. The congestion of the network is then defined as the maximum congestion among all links. An optimal routing and scheduling with respect to TM D minimizes the maximum congestion level:

$$\text{OPT}(D) = \min_{(f,\lambda)} \text{CONG}(f, \lambda, D).$$

In other words, OPT(D) is the minimum congestion level possible for the given D.

The *competitive ratio* of an (f,λ) on a given TM D measures the performance of the given (f,λ) with respect to that of the optimal one on TM D. It is defined as:

$$\text{PERF}(f, \lambda, D) = \frac{\text{CONG}(f, \lambda, D)}{\text{OPT}(D)}.$$

The competitive ratio is always at least 1. It is exactly 1 if and only if the routing and scheduling is optimal for D.

Let $\mathbf{D}$ includes all possible TMs. The *oblivious ratio* is defined as

$$\text{PERF}(f, \lambda, \mathbf{D}) = \max_{D \in \mathbf{D}} \text{PERF}(f, \lambda, D).$$

In other words, the oblivious ratio is the worst performance ratio of the given (f, λ) with respect to all TMs. A routing and scheduling with the minimum oblivious ratio is the optimal traffic-oblivious routing-and-scheduling. It optimizes the worst-case performance.

3.1 Routing and Scheduling with No Traffic Knowledge

Our objective is to find a traffic-oblivious routing-and-scheduling to minimize the oblivious ratio. In other words, we want a fixed routing and scheduling pair (f, λ) such that it achieves the worst-case minimum congestion level for all possible TMs under consideration, among all fixed routing-scheduling pairs. Formally, the optimal (f, λ) is defined as

$$\underset{(f,\lambda)}{argmin} \ \text{PERF}(f, \lambda, \mathbf{D}),$$

where $\mathbf{D}$ includes all the possible TMs under consideration. We first consider the case with no traffic information.

The optimal oblivious ratio r^* and the corresponding (f^*, λ^*) can be computed by solving the following problem:

$$\min_{(f,\lambda)} r \quad (9)$$
$$\sum_{i=1}^{K} \lambda_i \leq 1$$
$$f \text{ is a routing}$$
$$\forall \text{ links } m, \forall \text{ TMs } D \text{ with OPT}(D) = 1 :$$
$$\frac{\sum_{ij} f_{ij}(m) d_{ij}}{cap(m) \sum_{m \in I_i} \lambda_i} \leq r.$$

In this problem, if we scale a TM D by a constant, the performance ratio remains unchanged. So we only need to consider the D's with OPT(D) $= 1$ when we compute the oblivious ratio, as other D's are scaled versions of such matrices. The problem is interpreted as follows: consider any given TM D, if it takes its corresponding optimal routing and scheduling scheme one unit of time to serve the demand, then it takes (f^*, λ^*) *at most* r^* unit of time to serve the same demand.

We note that the last constraint is not a linear constraint. To solve the problem, we introduce an alternative variable $\beta_i = r\lambda_i$. We transform Eq. (9) to the following

LP problem.

$$\min_{(f,\beta)} r \tag{10}$$
$$\sum_{i=1}^{K} \beta_i \leq r$$
$$f \text{ is a routing}$$
$$\forall \text{ links } m, \forall \text{ TMs } D \text{ with OPT}(D) = 1:$$
$$\sum_{ij} f_{ij}(m) d_{ij}/cap(m) \leq \sum_{m \in I_i} \beta_i. \tag{11}$$

We call this problem the master LP. After obtaining the optimal solution, we can obtain λ^* by $\lambda_i^* = \beta_i^*/r^*$. With some abuse of notation, we also call $\beta = \{\beta_i\}$ scheduling.

The master LP has infinite number of constraints because there are infinite number of D's with OPT$(D) = 1$. Next, we transform the LP formulation with infinite number of constraints to a LP formulation with finite number of constraints. Note that for a given link m, routing and scheduling pair (f,β), the constraint in Eq. (11) is satisfied if the maximum value of the following slave LP is less than 0.

$$\max_{g,\mu,d} \left(\sum_{ij} f_{ij}(m) d_{ij}/cap(m) - \sum_{m \in I_i} \beta_i \right) \tag{12}$$
$$g \text{ is a flow of demand } d_{ij} \tag{13}$$
$$\sum_{i=1}^{K} \mu_i \leq 1 \tag{14}$$
$$\forall \text{ links } l : \sum_{ij} g_{ij}(l) \leq \sum_{l \in I_i} \mu_i cap(l) \tag{15}$$
$$\forall \text{ demands } i \to j\text{: } d_{ij} \geq 0, \tag{16}$$

where both $g_{ij}(e)$ and μ_i are new variables introduced in the slave LP to restrict D. Note that g is the flow, defined as follows:

$$\forall \text{ edges } e, \forall \text{ nodes } i, j \neq i : g_{ij}(e) \geq 0 \tag{17}$$
$$\forall \text{ nodes } i, j \neq i, k \neq i, j :$$
$$\sum_{e \in \text{out}(k)} g_{ij}(e) - \sum_{e \in \text{in}(k)} g_{ij}(e) = 0$$
$$\forall \text{ nodes } i, j \neq i :$$
$$\sum_{e \in \text{out}(i)} g_{ij}(e) - \sum_{e \in \text{in}(i)} g_{ij}(e) = d_{ij},$$

and μ_i is the portion of time the maximal independent set i can be active. With the two intermediate variables g and μ, only D's with OPT$(D) \leq 1$ satisfy the constraints defined in Eqs. (13-16). As stated in [12], at least one of the maximum of a linear objective function over a polyhedron are obtained on a vertex of the polyhedron and that polynomial time LP algorithms can obtain such a vertex maximum. Thus, the maximum objective value must be obtained at a certain D with OPT$(D) = 1$. Therefore, the slave LP can be used to check the constraint Eq. (11) in the master LP problem.

To further reduce the complexity, we combine the master LP with the slave LP. First, for the given m and (f, β) pair, we derive the dual form of the slave LP (12):

$$\min_{s,\pi,p(m)} \left(p(m) - \sum_{m \in I_i} \beta_i \right) \tag{18}$$
$$\forall \text{ nodes } i, j \neq i : \frac{f_{ij}(m)}{cap(m)} \leq s_{ij}(m, j)$$
$$\forall \text{ nodes } i, j \neq i, \forall \text{ edges } e = i' \to j' :$$
$$\pi(m, \text{link-of}(e)) + s_{ij}(m, i') - s_{ij}(m, j') \geq 0$$
$$\forall \text{ maximal independent sets } k \in [1, K] :$$
$$\sum_{l \in I_k} \pi(m, l) cap(l) \leq p(m)$$
$$\forall \text{ links } l : \pi(m, l) \geq 0$$
$$\forall \text{ nodes } i, j \neq i, \forall \text{ nodes } k :$$
$$s_{ij}(m, k) \geq 0$$
$$\forall \text{ nodes } i, j \neq i : s_{ij}(m, i) = 0$$
$$p(m) \geq 0,$$

where link-of(e) denotes the link that edge e belongs to. Since we assume bidirectional link in our study, $l_{ij} =$ link-of(e_{ij}) $=$ link-of(e_{ji}). The dual variable $s_{ij}(m, j)$ corresponds to the flow conservation constraint of d_{ij}, $\pi(m, l)$ to constraint (15), and $p(m)$ to constraint (14).

Based on the duality theorem [24], the dual LP has the same objective value as the primal LP problem. We then combine the master LP (10) with the dual form of the slave LP (18) to form a single LP:

$$\min r \tag{19}$$
$$f \text{ is a routing}$$
$$\sum_{i=1}^{K} \beta_i \leq r$$
$$\forall \text{ links } m : p(m) - \sum_{m \in I_i} \beta_i \leq 0$$
$$\forall \text{ links } m, \forall \text{ nodes } i, j \neq i :$$
$$\frac{f_{ij}(m)}{cap(m)} \leq q_m(i, j)$$
$$\forall \text{ links } m, \forall \text{ nodes } i, \forall \text{ edges } e = j \to k :$$
$$\pi(m, \text{link-of}(e)) + q_m(i, j) - q_m(i, k) \geq 0$$
$$\forall \text{ links } m, \forall \text{ maximal independent sets } k \in [1, K] :$$
$$\sum_{l \in I_k} \pi(m, l) \leq p(m)$$
$$\forall \text{ links } m, l : \pi(m, l) \geq 0$$
$$\forall \text{ links } m : p(m) \geq 0$$
$$\forall \text{ links } m, \forall \text{ nodes } i : q_m(i, i) = 0$$
$$\forall \text{ links } m, \forall \text{ nodes } i, j \neq i : q_m(i, j) \geq 0$$
$$\forall \text{ maximal independent sets } k \in [1, K] : \beta_k \geq 0.$$

In this LP, there are finite number of variables and finite number of constraints. Thus, it can be solved using standard LP solvers.

While motivated by previous work, our problem is different because we need to explicitly address schedulability. In [11], [12], [13], Internet routing is considered, and thus there is no interference or schedulability issue. We note that we cannot directly apply the schedulability constraint defined in Eq. (2) to the slave LP in [13] to set additional constraints on $g_{ij}(e)$. Because schedulability constraint is not considered in the master LP of [13], some qualified D's (i.e., with OPT(D) $= 1$) may not be included if (2) is directly applied. As a consequence, the maximum value of the slave LP could be obtained at some D's with OPT(D) $\neq 1$, which would result in incorrect solutions.

In [14], the authors propose an energy-efficient traffic-oblivious routing scheme in wireless sensor networks. The authors also discuss the schedulability issue and suggest additional constraints to be placed in a slave LP formulation. In [14], the definition of OPT(D) is based on power consumption instead of link congestion, and thus does not need to take into account schedulability. Therefore, the schedulability constraint can be directly applied to the slave LP as suggested in the discussion in [14]. However, applying schedulability constraint in the slave LP implies that schedulability is guaranteed only with perfect traffic information, which deviates from the original motivation of being traffic-oblivious. We note that schedulability is not a major concern in wireless sensor networks due to the low traffic demand. In comparison, because our objective is to maximize network utilization, schedulability constraint has to be defined in OPT(D). Therefore, the approach in [14] does not apply to our case. In addition, our scheduling is also traffic-oblivious while it is not in [14]. In summary, all previous works consider traffic-oblivious *routing* and our framework considers traffic-oblivious *routing-and-scheduling*, which addresses the unique constraint in wireless networks.

3.2 Routing and Scheduling with Some Traffic Information

The LP in (19) is derived without any traffic information. As discussed in Section 2.2, in some cases we could also have a certain amount of traffic information, i.e., the active s-d pairs or the s-d pairs with the range of the traffic demand. Such information can be modelled by introducing constraints on d_{ij}; i.e., $a_{ij} \leq d_{ij} \leq b_{ij}$ for s-d pairs, and $d_{ij} = 0$ for non-s-d pairs. Similar to [13], we introduce slack variables $s_m^-(i,j)$ and $s_m^+(i,j)$ for the lower and upper bounds on d_{ij}. Following similar approaches in Section 3.1, we can find the optimal routing-scheduling by solving the following finite LP formulation:

$$\min_{f,\beta} r \tag{20}$$

$$f \text{ is a routing}$$

$$\sum_{i=1}^{K} \beta_i \leq r$$

$$\forall \text{ link } m : p(m) - \sum_{m \in I_i} \beta_i \leq 0$$

$$\forall \text{ link } m, \forall \text{ node } i, j \neq i :$$

$$\frac{f_{ij}(m)}{cap(m)} - s_m^+(i,j) + s_m^-(i,j) = q_m(i,j)$$

$$\forall \text{ link } m, \forall \text{ node } i, \forall \text{ edge } e = j \rightarrow k :$$

$$\pi(m, \text{link-of}(e)) + q_m(i,j) - q_m(i,k) \geq 0$$

$$\forall \text{ link } m, \forall \text{ maximal independent sets } k \in [1, K] :$$

$$\sum_{l \in I_k} \pi(m,l) \leq p(m)$$

$$\forall \text{ link } m : \sum_{ij} \left(b_{ij} s_m^+(i,j) - a_{ij} s_m^-(i,j) \right) \leq 0$$

$$\forall \text{ links } m, l : \pi(m,l) \geq 0$$

$$\forall \text{ link } m : p(m) \geq 0$$

$$\forall \text{ link } m, \forall \text{ node } i : q_m(i,i) = 0$$

$$\forall \text{ link } m, \forall \text{ node } i, j : q_m(i,j) \geq 0$$

$$\forall \text{ maximal independent sets } k \in [1, K] : \beta_k \geq 0$$

$$\forall \text{ link } m, \forall \text{ node } i, j : s_m^-(i,j) \geq 0$$

$$\forall \text{ link } m, \forall \text{ node } i, j : s_m^+(i,j) \geq 0.$$

Note that, if $a_{ij} = b_{ij} = d_{ij}$ for all i, j (i.e., there is no uncertainty on traffic information), the oblivious ratio is one, which means that the routing and scheduling outputted from our framework is the optimal routing and scheduling.

3.3 Routing and Scheduling with Delay Considerations

Delay is an important issue in real-time applications, such as audio, video, and gaming. In a wireless network, packet delay is affected by many factors, including hop count, link quality, congestion level, and scheduling. In a TDM-scheduled network, hop count determines the *minimum* delay of a packet: if a route has n hops, then the packet delay is *at least* n time slots. Therefore, it is necessary to eliminate extra-long candidate routes for each s-d pair when delay is a concern. In this section, we discuss p-TORS, a modified TORS with preprocessing to eliminate extra-long routes.

Let D_{min}^{ij} be the shortest distance (in terms of hop count) from node i to node j. Let R_{th} be a threshold such that the maximum hop count of a route from i to j must be smaller than $R_{th} * D_{min}^{ij}$. This defines the hop count constraint. For example, we set $R_{th} = 2$ in our simulations. Consider an edge e from k to m. We set $f_{ij}(e) = 0$ in the LP formulation if

$$\max_{l \in \{k,m\}} \left(D_{min}^{il} + D_{min}^{lj} \right) \geq R_{th} \times D_{min}^{ij}. \tag{21}$$

In other words, we eliminate routes through edges that will definitely violate the hop count constraint in the preprocessing. We conduct the preprocessing for all s-d pairs. The results are used in the LP formulation. The preprocessing disables extra-long routes. It also reduces the number of (unknown) variables in the LP formulation and thus helps speed up the computation. Thus early elimination of longer paths based on hop-count helps in improving the scalability of the LP solution.

Two things are worth noting before we proceed. First, our heuristic cannot guarantee a strict hop count relative to the shortest path length. Instead, it tries to reduce the hop count by pruning out the links which apparently lead to extra long routes. Second, the hop count constraint cannot guarantee an upper bound of the delay. It can reduce the delay when hop count is the major cause of the delay. It does not work well when other factors (e.g., congestion) dominate. More details are provided in Section 4.3.

4 Performance Evaluation

4.1 Simulation Setup

We use Qualnet [17] for performance evaluation. Qualnet is a commercial wireless-network simulator that provides good PHY layer modeling as well as upper-layer protocol implementations. We use 802.11b PHY layer. The data rate is 11 Mbps. Every node uses the same transmission power. The channel model is the 2-ray path loss model with slow fading. The conflict graph is measured after the network is setup. For a link A, we enable all the other links one by one, and measure the throughput. If the throughput degrades after link B is enabled, A and B interfere with each other. CBR is used as the application for each active s-d pair. We use the maximum possible payload size (i.e., 1500 bytes MAC layer payload). The traffic demand varies, which will be explained later in this section. For a multipath routing, a node splits its traffic among multiple paths based on the corresponding ratio on each path. We implement the scheduling using TDMA with fixed length super frames where each time slot is allocated to a maximal independent set. Total number of time slots allocated to a maximal independent set i is proportional to its time allocation λ_i. Note that the length of the super frame is carefully chosen to fit the precision of the scheduling λ_i. For example, if the λ_i is rounded to the second decimal place, we will use 100 time slots in each super frame. Within a super frame, we schedule the transmission as fair as possible. For instance, consider two maximal independent sets A and B, A is allocated 60 time slots while B is allocated 40 time slots. In the first 80 time slots, we schedule A and B in a round-robin manner, then A transmits in the remaining 20 time slots.

To make the scheduling overhead comparable to the 802.11 CSMA-based MAC protocol, the slot duration is carefully chosen such that for the simplest scenario where one node is always sending packets to the other node with maximum possible payload size, both 802.11 and scheduling lead to the same throughput. In other words, the slot duration includes all the MAC and physical overhead of 802.11 except for the collision overhead. This enables us to distinguish the scheduling gain (i.e., avoiding collisions) from joint routing-and-scheduling gain. In practice, the slot time can be further reduced, and TORS can achieve a better performance than that shown here.

We have conducted simulation in a variety of topologies, including 9-node grid, 16-node grid, chain, and random networks. We have also conducted simulation under different traffic patterns, including single-sink, double-sink, random peer-to-peer, and all s-d pairs. Due to space limitation, we present the simulations in a 16-node grid with single-sink and double-sink traffic patterns unless otherwise specified. Other simulations yield similar conclusions. In the single-sink traffic pattern, there is one sink node in the network and every other node communicates with the sink node. In the double-sink pattern, there are two sink nodes in the network, every other node choose the closer sink (in terms of the number of hops) to transport its data. These two traffic patterns represent wireless mesh networks where there are mesh portals providing Internet access to the other nodes in the network. Furthermore, single-sink pattern leads to a more congested network condition than double-sink pattern. We compare the performance of four routing and scheduling schemes. Since TORS and its extension p-TORS do not incur any *network* overhead, we use a pre-computed shortest path routing instead of practical distributed shortest path routing such as AODV to ensure fair comparison.

- *SP+802.11*: The routing protocol is a pre-computed shortest path routing, and the MAC is 802.11 CSMA-based MAC.
- *SP+schd*: The pre-computed shortest path routing is combined with an optimal scheduling for a given TM D. The optimal scheduling is calculated as follows:

$$\min \sum_{i=1}^{K} \beta_i \tag{22}$$
$$\forall \text{ links } m: \quad \sum_{ij} f_{ij}(m) d_{ij}/cap(m) \le \sum_{i:m\in I_i} \beta_i,$$

  where $f_{ij}(m)$ is pre-computed.
- *Estimate*: We assume a *base* TM, which is our best estimate of the actual traffic. The actual traffic pattern and demand may vary from the estimation. Then we compute the *optimal* routing and scheduling based on this estimated traffic matrix.
- TORS or p-TORS: We use the oblivious routing and scheduling proposed in Section 3. We assume the same base TM as in *Estimate*, but takes into account traffic information uncertainty.

We assume a base traffic demand for each s-d pair. The actual traffic demand fluctuates around the base traffic demand when we run the simulation. In order to model the traffic fluctuation as well as traffic information inaccuracy, we consider the error margin w. An actual TM, $D' \in \mathbf{D'}$, satisfies $d_{ij}/w \le d'_{ij} \le w d_{ij}$. For example,

TABLE 1
Throughput performance of four routing and scheduling schemes when $w = 1$

Pattern	*SP+802.11*	*SP+schd*	*Estimate*	TORS
Single	2.59	3.02	4.07	4.07
Double	4.83	5.02	5.76	5.76

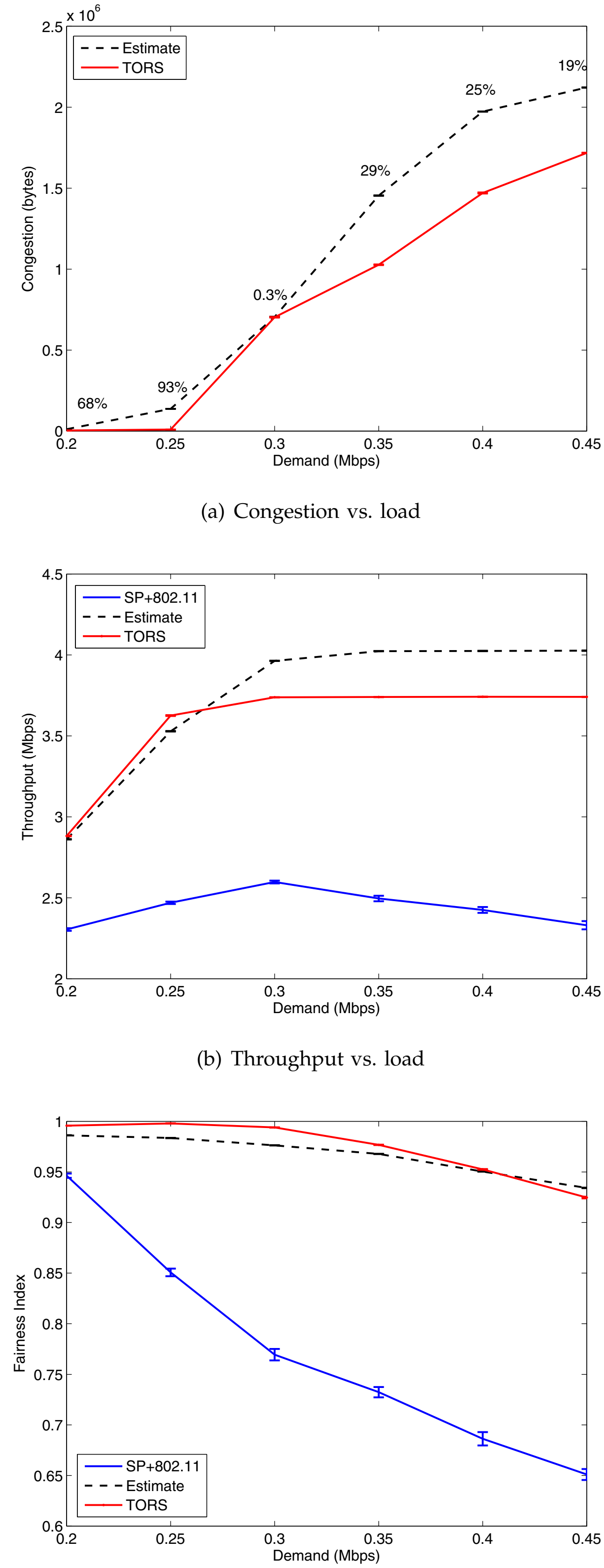
if $w = 1.5$, the upper bound on traffic demand is 225% of the lower bound. For each set of simulations, the actual demand is generated randomly within $[d_{ij}/w, d_{ij} * w]$. We do not use the range $[d_{ij} - w, d_{ij} + w]$, to ensure that the lower bound is always larger than zero. As we explained in Section 3.1, the oblivious ratio remains the same if we scale TM D by a constant. Thus, once the lower bound is zero, no matter what the upper bound is, it is always equivalent to no information at all.

4.2 Traffic-Oblivious Routing and Scheduling

We first compare the throughput performance of the four routing and scheduling schemes discussed in Section 4.1. Although the objective of our framework is not to maximize the throughput, we still expect a large throughput gain over heuristic-based protocols (i.e., shortest path routing and 802.11 scheduling). The results are tabulated in Table 1. We only consider $w = 1$ here. As mentioned earlier, when $w = 1$, there is no variation in the traffic demand, which means that perfect traffic information is given. In this case, TORS and *Estimate* both achieve the same optimal performance. Through this simple case, we want to show how optimal spatial reuse and scheduling can improve the performance. Benefiting from optimal scheduling only, *SP+schd* achieves 17% and 4% higher throughput than *SP+802.11* under single-sink and double-sink pattern, respectively. Such gains increase to 57% and 19% under *Estimate* and TORS, as they benefit from both the optimal spatial reuse and scheduling.

Next, we show how TORS is robust to traffic variations. We compare the throughput and fairness performance under *SP+802.11*, *Estimate* and TORS. More importantly, we compare the network congestion between *Estimate* and TORS, as minimizing congestion is our objective. Note that, the maximum *link* congestion is defined as the network congestion in our framework. However, there is no direct way to measure link congestion in the simulation. Thus, we measure an alternate metric: *node* congestion, which is represented as the average queue length of a node throughout the simulation. This is reasonable because queue length is directly related to link congestion. In order to minimize the impact of limited queue capacity, we assume a large queue size so that no packet is lost due to queue overflow throughout the simulation.

Both single-sink and double-sink patterns exhibit the same performance trend. We show the performance under single-sink to save space. We assume every s-d pair

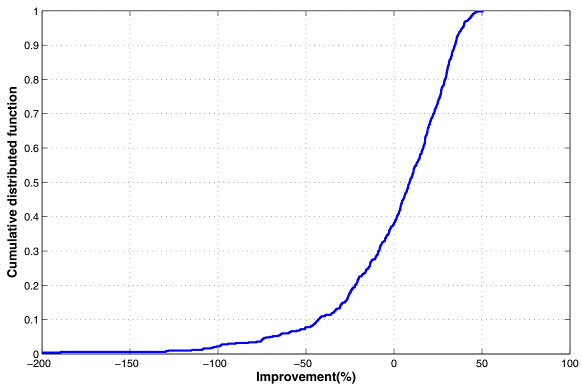


(a) Congestion vs. load

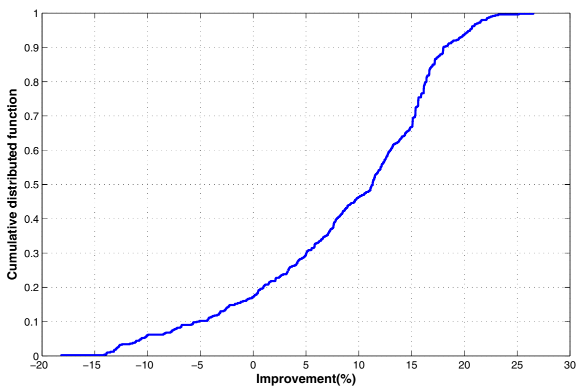
(b) Throughput vs. load

(c) Fairness vs. load

Fig. 1. Performance of three schemes under a simple variation model



(a) Single sink



(b) Double sink

Fig. 2. Cumulative distribution of improvement (TORS over *Estimate*) for 500 random traffic demand instances

(totally 15 pairs in the single-sink pattern) has the same demand d. The base demand is set as 0.3 Mbps. TORS is computed with $w = 1.5$, which corresponds to the variation range of [0.2, 0.45]. We increase d from 0.2 to 0.45, at a step size of 0.05 Mbps. This is an oversimplified "variation" model since every s-d pair always has the same demand. But it gives us insights on how TORS works.

Fig. 1(a) compares the congestion performance between *Estimate* and TORS. To better illustrate the difference, we also print the percentage of improvement of TORS over *Estimate*, i.e., $\frac{CONG_{Estimate} - CONG_{TORS}}{CONG_{Estimate}}$. As can be seen, the difference between *Estimate* and TORS is almost zero when the actual demand is exactly the base demand (i.e., 0.3 Mbps). This is expected because *Estimate* is the optimal routing and scheduling for the base demand. However, TORS has a considerably better congestion performance than *Estimate* as long as the actual demand deviate (smaller or larger) from the base demand. This is more than expected. Theoretically, TORS only guarantees the worst-case performance. But the simulation results clearly show that it can do better than that.

Fig. 1(b), 1(c) show the throughput and fairness performance. The fairness index is computed as $f = \frac{(\sum_i T_i)^2}{N * \sum_i T_i^2}$ [25]. As expected, both *Estimate* and TORS achieve a better throughput and fairness performance than *SP+802.11*. In addition, fairness performance of *Estimate* and TORS decreases slower with traffic load than *SP+802.11*. The reason is the hidden-node problem. As the traffic load increases, hidden-node becomes the major reason to cause collisions. In the single-sink traffic pattern, nodes farther away from the sink suffer more from hidden-node collision losses than nodes closer to the sink, which in turn leads to a larger unfairness in the throughput distribution. On the other hand, *Estimate* and TORS completely eliminate collisions since the scheduling always performs on independent sets. Their fairness performance is thus much better than *SP+802.11* under high traffic demand.

To verify our observation in Fig. 1(a), we adopt a more realistic variation model. We run 500 simulations. Each simulation lasts for 60 seconds. In each simulation, each s-d pair independently chooses a random traffic demand from the variation range. We measure the congestion of *Estimate* and TORS for each simulation. Among 500 simulations, the worst-case congestion of TORS is 24% and 15% better than that of *Estimate* under single-sink and double-sink pattern, respectively. In addition, we plot the cumulative density function of the improvement of TORS over *Estimate* in Fig. 2. The improvement is defined as $\frac{CONG_{Estimate} - CONG_{TORS}}{CONG_{Estimate}}$. The x-axis is the improvement in percentage. The y-axis is the cumulative density. The negative improvement indicates that TORS leads to a worse congestion than *Estimate* for this instance. So we are interested in how often the improvement can be positive. Under single-sink, more than 60% of all simulations have positive improvement. Under double-sink, such percentage is larger than 80%. That is, for majority of the time, TORS has a better congestion performance than *Estimate*. This is a good news for the applicability of TORS in practice. Its average performance is good enough despite the fact that it only guarantees the worst-case performance in theory.

4.3 TORS with Hop Count Constraints

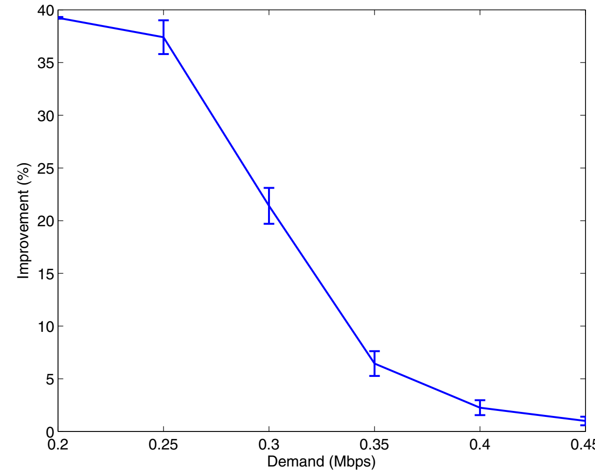
In Table 2, we compare TORS and p-TORS with $w = 1.0$. We set $R_{th} = 2$ for both traffic patterns. Recall that $R_{th} = \alpha$ means the maximum hop count for each s-d pair under p-TORS cannot exceed α times the hop count under shortest path routing. In both cases, shortest path routing always has the minimum average hop count. The average hop count of p-TORS is considerably smaller than that of TORS. On the other hand, the throughput of p-TORS is lower than TORS, because the hop count constraint limits candidate routes, which limits spatial reuse. The average number of edges pruned out per s-d pair is smaller in single-sink pattern than that in double-sink pattern. This is because the average shortest path length in single-sink pattern is larger than that in

TABLE 2
TORS with hop-count constraint

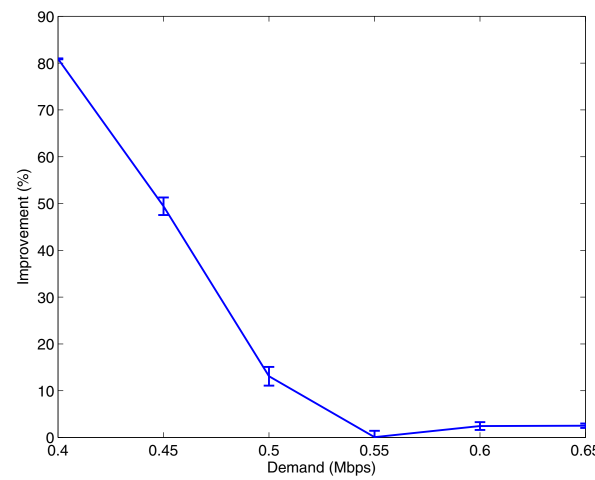
Single	*SP+802.11*	p-TORS	TORS
Throughput (Mbps)	2.57	3.03	4.04
Avg. hop count	3.2	4.9	8.8
Avg. # of edges pruned	\	18.4	\
Double	*SP+802.11*	p-TORS	TORS
Throughput (Mbps)	5.24	6.32	6.43
Avg. hop count	2.3	2.7	4.8
Avg. # of edges pruned	\	28.3	\

double-sink pattern. Thus, more links will be pruned out in double-sink pattern.

Fig. 3 plots the delay improvement of p-TORS over TORS under different traffic demands. Both TORS and p-TORS is computed with $w = 1.5$. As shown in the figure, the improvement in the average delay can achieve up to 40% under single-sink and 80% under double-sink, when traffic demand is small. But such improvement decreases significantly when the demand is high. Under light traffic demand, delay is mainly determined by the number of hops. Thus, p-TORS has a big advantage (in terms of delay) over TORS because p-TORS tried to reduce the number of hops in the framework. Under heavy traffic load, however, the delay is mainly caused by congestion (channel access delay, queuing delay, etc.). In this case, p-TORS cannot significantly outperform TORS in terms of delay.



(a) Single sink



(b) Double sink

Fig. 3. Delay improvement of p-TORS over TORS under different traffic demands

4.4 Oblivious Routing with 802.11 MAC

Centralized routing is easier to implement than centralized scheduling because scheduling requires synchronization. Therefore, a natural step is to integrate centralized routing with distributed scheduling or sensing-based medium access scheme. Our heuristic is to generate the joint routing and scheduling using TORS and then extract the routing and combine it with a decentralized MAC (e.g., 802.11). The results are shown in Table 3. w is fixed at 1.5. We generate 100 random traffic demands within the range and measure the average throughput. In single-sink traffic pattern, oblivious routing with 802.11 has comparable throughput as *SP+802.11*. But in double-sink traffic pattern, oblivious routing with 802.11 outperforms *SP+802.11*. In general, we observe that the performance of the heuristic depends on traffic patterns. If the number of s-d pairs is relatively small (as in double-sink pattern), the proposed heuristics outperforms *SP+802.11* significantly. It is due to the spatial reuse introduced by the routing. Under more "congested" traffic pattern (as in single-sink pattern), the heuristic using oblivious routing may not outperform *SP+802.11* due to the existence of collisions and the lack of optimal scheduling support. We do not have a way to give a quantitative measure on the traffic patterns under which oblivious routing with 802.11 outperforms *SP+802.11*. We consider it as our future work.

TABLE 3
Oblivious routing with 802.11 MAC

	SP+802.11	Oblivious routing+802.11
Single	2.58	2.55
Double	5.94	6.51

4.5 Insights of TORS

Fig. 4 shows the oblivious routing (left-hand side) and scheduling (right-hand side) when there is only one

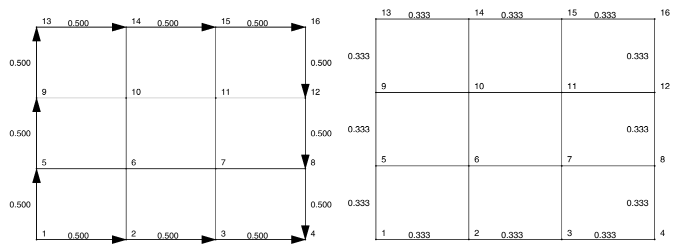


Fig. 4. Traffic-oblivious routing and scheduling for the s-d pair (1, 4)

TABLE 4
The oblivious ratio under different w's

w	1.0	1.5	2.0	2.5	3.0	4.0
obl. ratio	1.00	1.24	1.52	1.64	1.71	1.79

s-d pair (1, 4). The routing is of multiple paths: half of the flow goes through the shortest path, the other half goes through a much longer path. Such routing is quite counter-intuitive, especially the longer path. It seems that such a long path may consume much more bandwidth than the shortest path, which should lead to a lower performance. However, we will show that this oblivious routing does require less scheduling time than the shortest path routing. Given the routing f and the TM D, the optimal scheduling can be obtained using (22).

According to (22), the optimal scheduling time for the shortest path routing is 3, for oblivious routing is 1.5. It means that oblivious routing does lead to a higher throughput than shortest path routing under the optimal scheduling. The intuition is that, although oblivious routing consists of a longer path which consumes more bandwidth, such a path also balances the traffic load to exploit the spatial reuse of the network, with the help of the scheduling.

In order to better understand TORS, we measure the oblivious ratio obtained under different levels of traffic variations. The results are shown in Table 4. The first row is w. The second row is the corresponding oblivious ratio for TORS. As w increases, TORS has to cope with more variation in traffic demands, which leads to a higher oblivious ratio. But even if $w = 4.0$ (i.e., upper bound of traffic variation is 1600% of the lower bound), the oblivious ratio is only 1.79, which guarantees a satisfactory worst-case performance.

TORS is also robust under false traffic information. In Table 5, we compare TORS and *Estimate*. Both calculate routing and scheduling assuming the all-pair traffic pattern (i.e., every node transmits to all the other nodes.), and TORS is calculated with $w = 1.5$. However, in the simulations, the actual traffic patterns are single-sink, random (i.e., we randomly pick half of the nodes as senders. Each node randomly chooses a destination.), and all-pair, respectively. We generate 100 random traffic demands within the estimation range for each traffic pattern. For each entry in the table, the left is the average throughput, and the right is the worst-case throughput.

TABLE 5
Performance under inaccurate estimation of the traffic

	Single-sink	Random	All-pair
Estimate	1.15 / 1.02	2.50 / 1.98	4.86 / 3.97
TORS	1.73 / 1.54	3.26 / 2.75	4.74 / 4.48

We observe that, when the estimation is correct, i.e., the all-pair traffic pattern, TORS achieves comparable average performance and better worst-case performance. In addition, when the actual traffic patterns are single-sink and random (i.e., largely deviate from the estimation), the performance of TORS is robust and always better than *Estimate*. It indicates the robustness of TORS even under false traffic information.

We study in detail the output of TORS to understand the desirable properties of robust routing and scheduling. The routing generated by TORS is multi-path routing that *exploits spatial reuse* in the network and *tolerance to network dynamics.* In general, TORS results in a more even distribution of flows in the network and more balanced utilization of network resources. Its tolerance to traffic dynamics is due to the load balancing effect of splitting flows among multiple paths. In addition, TORS tends to allocate overlapping routes to different flows; i.e., an edge can be used by subflows from multiple flows. Such an allocation provides tolerance to traffic dynamics due to the smoothing effect of statistic multiplexing. TORS makes it easier to admit new flows because it adapts to demand changes and distributes resource relatively evenly.

5 Discussions

5.1 Implementation Issues

Contemporary infrastructure-based WLAN network designs employ a *centralized*, lightweight WLAN architecture [16]. In such an architecture, a WLAN controller conducts most management functionalities, and lightweight access points offer services, monitor channel conditions, and send feedback information to the controller. Similar monitoring and control functions are needed in general multi-hop mesh networks. For example, Intel's enterprise OverMesh [26] is hierarchical and implements a lightweight control-and-management plane [27]. The control plane is an overlay network logically, which consists of a portion of mesh nodes to monitor the network, collect traffic information, and send feedback information to control nodes. Therefore, in the implementation, we assume that a (logical) control plane is available and periodically sends traffic and channel information to the control node.

To implement the schemes suggested in this paper, a control node could compute the desired routing and scheduling based on available information, and subsequently distribute the routing and scheduling information through the network. The network would operate in cycles. Each cycle would consist of a short routing-scheduling update frame and a super transmission frame. In the beginning of the transmission cycle, the controller broadcasts the routing/scheduling in the update frame using a precomputed broadcasting tree. The length of the update frame depends on the size of the network. We consider that 3-8 time slots are reasonable for a medium size network. The super transmission frame follows the update frame. In the transmission frame, each node performs routing and scheduling following the broadcasted routing and scheduling information. The length of a time slot is the same for all nodes and when using a schedule, one can assume that these times slots are synchronized. The number of packets transmitted in a time slot varies depending on link conditions.

The length of the transmission frame should be larger than the time for information feedback and computing routing and scheduling. Traffic and channel information can be collected in the network in the order of a few hundred ms through lightweight overlays, to achieve relatively stable measurements. Therefore, the length of the transmission frame can be a few hundred time slots, before the routing and scheduling are updated. In general, cross-layer adaptation in the system needs to happen fast enough so as to be proactive, and yet slow enough so as to allow for link condition variations to stabilize and for the overhead of such overlays and cross-layer optimization to be minimal.

5.2 The path to full decentralization without scheduling

While a schedule is suggested, it is not required that the schedule needs to be followed. A controller node could recompute the LP periodically. However, the node will broadcast the new routing only policies for the flows, and choose to not broadcast any schedule at all. Not implementing the optimal schedule will result in some performance degradation. However, not requiring the broadcast of a schedule will result in reduced complexity of implementation, and will not require any synchronization between nodes. It is not required that all nodes receive the route updates for the flows at the same time. Source nodes change the routes whenever they receive the updated routes. Likewise, intermediate nodes can start forwarding packets on new routes for each flow whenever they receive suggestions for updates to their routing tables. Thus the entire optimization with overlays and data transmissions in the network can proceed completely asynchronously. In general, it is expected that this would result in the nodes contending for access and transmitting data packets in maximal independent sets simultaneously resulting in a schedule that could be expected to produce a performance relatively similar to what could have been expected if the schedule obtained from the LP had actually been implemented. In addition, based on the information in the network, each of the source nodes in the network could independently solve the LP assuming requirements for other flows have not changed. In general, asynchronous computation and decentralized decision-making without constraints of fixed schedules will result in the least complex mesh network system implementations and help in scalability in future mesh networks.

6 Conclusion

In this paper, we propose TORS, a joint routing-and-scheduling scheme, that achieves robust performance under traffic information uncertainty. To elaborate, the proposed scheme optimizes the worst-case performance (in terms of relative congestion) for a range of traffic conditions. The proposed scheme works under the whole spectrum of traffic information uncertainty from perfect traffic information to no traffic information. The performance of TORS adapts to the granularity of the traffic information available. The more accurate the information is, the better the performance will be. We show that TORS achieves optimal worst-case performance, as well as good average performance. The performance of TORS is robust under traffic dynamics, or even false traffic information.

TORS generates multipath routes that exploit spatial reuse and provide more balanced utilization of network resources. These properties also provide insights into the design of joint routing and MAC schemes in a distributed manner.

Acknowledgment

The work was in part supported by NSF through CAREER Award #0448613 and Grant #0520126, and by Intel through a gift grant.

References

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

[2] R. Cruz and A. Santhanam, "Optimal routing, link scheduling and power control in multihop wireless networks," in *IEEE INFOCOM*, 2003.

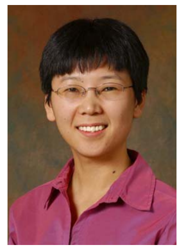
[3] J. Zhang, H. Wu, Q. Zhang, and B. Li, "Joint routing and scheduling in multi-radio multi-channel multi-hop wireless networks," in *IEEE Broadband Networks*, 2005.

[4] M. Kodialam and T. Nandagopal, "Characterizing the capacity region in multi-radio multi-channel wireless mesh networks," in *MobiCom '05: Proceedings of the 11th annual international conference on Mobile computing and networking*. New York, NY, USA: ACM Press, 2005, pp. 73–87.

[5] T. Larsson and N. Hedman, "Routing protocols in wireless ad-hoc networks - a simulation study," Master's Thesis, LuLea Tekniska University, 1998.

[6] J. Bicket, D. Aguayo, S. Biswas, and R. Morris, "Architecture and evaluation of an unplanned 802.11b mesh network," in *MobiCom '05: Proceedings of the 11th annual international conference on Mobile computing and networking*. New York, NY, USA: ACM Press, 2005, pp. 31–42.
[7] R. Draves, J. Padhye, and B. Zill, "Comparison of routing metrics for static multi-hop wireless networks," in *SIGCOMM '04: Proceedings of the 2004 conference on Applications, technologies, architectures, and protocols for computer communications*. New York, NY, USA: ACM Press, 2004, pp. 133–144.
[8] S. Mueller, R. P. Tsang, and D. Ghosal, "Multipath routing in mobile ad hoc networks: issues and challenges," in *Invited paper in Lecture Note in Computer Science*, 2004.
[9] W. Lou, W. Liu, and Y. Zhang, "Performance optimization using multipath routing in mobile ad hoc and wireless sensor networks," in *Combinatorial optimization in communication networks*, 2005.
[10] Y. Ganjali and A. Keshavarzian, "Load balancing in ad hoc networks: single-path routing vs. multi-path routing," in *IEEE Infocom*, 2004.
[11] H. Räcke, "Minimizing congestion in general networks," in *IEEE FOCS*, 2002.
[12] Y. Azar, E. Cohen, A. Fiat, H. Kaplan, and H. Racke, "Optimal oblivious routing in polynomial time," in *ACM STOC*, 2003.
[13] D. Applegate and E. Cohen, "Making intra-domain routing robust to changing and uncertain traffic demands: understanding fundamental tradeoffs," in *ACM SIGCOMM*, 2003.
[14] Y. Li, J. Harms, and R. Holte, "Optimal traffic-oblivious energy-aware routing for multihop wireless networks," in *IEEE INFOCOM*, 2006.
[15] A. Kashyap, S. Sengupta, R. Bhatia, and M. Kodialam, "Two-phase routing, scheduling and power control for wireless mesh networks with variable traffic," in *ACM Sigmetrics*, 2007.
[16] Cisco, "Using radio resource management to deliver secure and reliable wlan services," White Paper.
[17] I. Scalable Network Technologies, "Qualnet 4.0 user's guide," 2006.
[18] P. Stuedi and G. Alonso, "Log-normal shadowing meets sinr: A numerical study of capacity in wireless networks," in *IEEE SECON*, 2007.
[19] G. Sharma, R. R. Mazumdar, and N. B. Shroff, "On the complexity of scheduling in wireless networks," in *ACM Mobicom*, 2006.
[20] J. Tang, G. Xue, and W. Zhang, "Interference-aware topology control and qos routing in multi-channel wireless mesh networks," in *ACM MobiHoc*, 2005.
[21] M. Burkhart, P. von Rickenbach, R. Wattenhofer, and A. Zollinger, "Does topology control reduce interference," in *ACM MobiHoc*, 2004.
[22] G. Brar, D. M. Blough, and P. Santi, "Computationally efficient scheduling with the physical interference model for throughput improvement in wireless mesh networks," in *ACM Mobicom*, 2006.
[23] O. Goussevskaia, Y. A. Oswald, and R. Wattenhofer, "Complexity in geometric sinr," in *ACM MobiHoc*, 2007.
[24] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms, Second Edition*. The MIT Press, 2001.
[25] R. Jain, A. Durresi, , and G. Babic, "Throughput fairness index: an explanation," in *ATM Forum Document Number: ATM Forum/990045*, 1999.
[26] J. Vicente, S. Rungta, G. Ding, D. Krishnaswamy, W. Chan, and K. Miao, "OverMesh: Network centric computing," *IEEE Communication Magazine, to appear, (invited).*
[27] G. Ding, J. Vicente, S. Rungta, D. Krishnaswamy, W. Chan, and K. Miao, "Overlays on wireless mesh networks: Implementation and cross-layer searching," in *9th IFIP/IEEE International Conference on Management of Multimedia and Mobile Networks and Services*, 2006.

Wei Wang received his M.Phil. degree in information engineering from Chinese University of Hong Kong in 2004, and his B.Eng. degree in electrical engineering from University of Science and Technology of China in 2002. He is working towards his Ph.D. degree in computer science at University of California, Davis.



Xin Liu received her Ph.D. degree in electrical engineering from Purdue University in 2002. She is currently an associate professor in the Computer Science Department at the University of California, Davis. Before joining UC Davis, she was a postdoctoral research associate in the Coordinated Science Laboratory at UIUC. Her research is on wireless communication networks, with a focus on resource allocation and dynamic spectrum management. She received the Best Paper of year award of the Computer Networks Journal in 2003 for her work on opportunistic scheduling. She received NSF CAREER award in 2005 for her research on "Smart-Radio-Technology-Enabled Opportunistic Spectrum Utilization". She received the Outstanding Engineering Junior Faculty Award from the College of Engineering, University of California, Davis in 2005.

Dilip Krishnaswamy received the B.Tech. degree in Electronics and Communication Engineering in 1991 from the Indian Institute of Technology, Madras, the M.S. degree in Computer Science in 1993 from Syracuse University where he was a University Fellow, and the Ph.D. degree in Electrical Engineering in 1997 from the University of Illinois at Urbana-Champaign. He presently is a senior staff researcher in the office of the chief scientist at Qualcomm in San Diego. He is currently the associate editor-in-chief of the IEEE Wireless Communications Magazine. He also serves as the vice-chair of the IEEE Communications Society emerging technical committee on Applications of Nanotechnologies in Communications. He received the best paper award for the 1997 IEEE VLSI Test Symposium. He was a Platform Architect in the Mobility Group at Intel Corporation where he worked on projects related to the Pentium4 processor development, high-level architectural modeling, system-on-chip mobile platform architectures, wireless multimedia optimizations in the digital home, cross-layer optimized 802.11a/n wireless communications, and cross-layer cross-overlay mesh networking architectures. He was the architect for Intel's first Cellular processor (Manitoba/PXA800F) consisting of a heterogeneous multicore system-on-chip architecture. He taught courses related to computer architecture, parallel computer architecture, and digital systems design at the University of California, Davis, where he is presently a member of the Industrial Board of Advisors for the ECE department. He chaired the Design and Developer's Forum and Tutorials and Workshops at IEEE Globecom 2006, and chairs the D&D Forum for IEEE Globecom 2009. He co-chaired the 2007 IEEE MMNS conference and co-chaired the Access Business Forum at IEEE Globecom 2007. His research interests include heterogeneous wireless communications processing systems, heterogeneous wireless networking architectures, distributed cooperative processing systems, wireless network economics, cross-layer optimized wireless multimedia, parallel processing, non-linear optimization, and game theory.