Scheduling Aware Network Flow Models for Multi-hop Wireless Networks

Vinay Kolar and Nael B. Abu-Ghazaleh
CS. Department, Binghamton University
Binghamton, NY 13902–6000
{vinkolar,nael}@cs.binghamton.edu

Abstract

Network flow models have proven to be an effective tool in the analysis and optimization of networks. In addition, with some work, they have been used to develop stable and near-optimal distributed protocols. Critical to the success of these models in multi-hop wireless networks (MHWNs) is an accurate estimation of the effect of interference. While the existing models capture coarse grained estimates of interference, they do not account for the substantial impact of MAC scheduling. On the other hand, accurate models of throughput in CSMA networks exist. However, their complexity and some of their underlying assumptions make them unsuitable for use as part of a network flow formulation, which must explore a large number of candidate solutions. This paper contributes an efficient and constructive model to estimate the effect of scheduling on interfering links in general MHWN settings. We integrate this approach with a network flow routing model which works with aggregate estimates of capacity to improve the quality of the solution. Simulation results show that accounting for scheduling effects leads to large improvements in the quality of the solution.

I. Introduction

Network Flow Models of Multi-Hop Wireless Networks have important applications. MHWNs, including mesh networks, ad hoc networks, and some sensor networks, are emerging as important components of an increasingly ubiquitous and wireless world. Network flow models of MHWNs have recently been developed to allow analysis of capacity and optimization of routing [1]–[3]. Such models can be applied to analyze how far existing routing protocols, which are heuristic and greedy in nature, are from near optimal routing obtained with global knowledge and coordination among the connections. Further, they can be directly applied to traffic engineering and QoS for static MHWNs. Moreover, such models have been used in the context of wired networks (e.g., via exploring the dual problem or using decomposition [4]) to develop near optimal distributed protocols. Alternatively, the insight gained from these models can be heuristically used to better understand how to build effective protocols for more dynamic MHWNs.

However, existing models are limited, especially in how they model interference. While the initial efforts in this area make significant contributions, they make simplifying assumptions that limit their ability in finding effective solutions. Most important among the limitations is the approach to modeling interference, which does not take into account the effects that arise in a CSMA (Carrier Sense Multiple Access) MAC protocol. Instead, the existing models ignore the MAC effect or assume the presence of an omniscient scheduler. We show in Section III that MAC level interactions play an important, sometimes defining, role in determining link capacity, especially under high interference; ignoring their effect leads to inaccurate characterization of solution quality, which in turn produces inefficient solutions. Thus, the goal of this work is to improve the interference models by accounting for the MAC scheduling effects.

CSMA throughput models may offer a solution; however, they are not suitable in this context because of their high complexity and their iterative nature. The problem of estimating throughput for CSMA networks is well researched problem [5]–[9]. Available models typically estimate the link throughputs for a given network configuration (including the routes). Thus, they may be suitable for our purpose: as the solver considers candidate solutions, each may be evaluated using the CSMA models. Unfortunately, they are often computationally expensive, making it difficult to use them as part of an iterative optimization process that evaluates candidate solutions to converge on near optimal routing configurations. Importantly, the models are iterative in nature, estimating the performance, but not offering any insight into the processes that go into determining it. As a result, the structure of the problem is hidden and little insight into developing effective protocols is provided. Finally, many of the models use unrealistic simplifying assumptions. We discuss these models in more detail in Section II.

The paper contributes a scheduling aware model of interference and integrates it with network flow formulations, significantly improving the quality of the obtained solutions. The model, presented in Section IV is lightweight and constructive, allowing much faster solution time and making it feasible to integrate it with the network flow optimization framework. Further, the model improves on the scheduling component of the CSMA models in a number of ways: it uses Signal to Interference and Noise Ratio (SINR) physical model, rather than the simplified protocol model assumed by existing works [10]; and it does not use assumptions such as exponential distribution on packet transmission. The integration of the scheduling model with a network flow formulation provides for the first time accurate accounting for the effect of interference in network flow models (Section V). We show in Section VI that the scheduling estimates are accurate, and the derived routes from the integrated model achieve large improvements in performance over the those from the model using aggregate interference metrics only. Finally, Section VII presents concluding remarks.

II. Related work

In this section we overview related work, organized into two areas: (1) network flow models for MHWNs, and how our work improves on them; and (2) CSMA models of MHWNs and why they are not directly usable for improving network flow models.

A. Network Flow Models

Our work is motivated by recent MHWN network flow models, which differ from wired network models in that they account for interference [1]–[3]. These models are useful in the analysis and optimization of MHWNs. In addition, often the structure of the problem leads to the development of optimal distributed protocols [4]. However, existing models use inaccurate models of interference that do not account for the complex scheduling effects that arise in CSMA networks. The primary contribution of this work is to develop more accurate characterization of link quality under CSMA scheduling.

The goal of the formulations by Jain [1] and Kodialam [2] is estimating the capacity of given scenarios. The above models assume an optimal scheduler, without the CSMA artifacts like hidden terminals. Further, their focus is on optimizing the overall network performance with cursory treatment to interaction between the connections. For these reasons, we extend our previously developed formulation which addresses these limitations [3]. It models a multiple-connection network using a multi-commodity flow framework, and allows derivation of near optimal routes with regards to a user defined objective function. However, like the other formulations, this model does not account for the CSMA scheduling.

B. CSMA Models of MHWNs

Modeling the operation of a CSMA based medium access protocol in a multi-hop wireless network is a classical problem (e.g., [5], [6]) and remains an area of active research (e.g., [7]–[9]). Existing models estimate the throughput of a network with a given routing topology –they cannot be used directly to derive routes. However, they could apply them as a fitness function to test solutions within an optimization problem that searches for effective routes. Unfortunately, the complexity of the existing models make them extremely computationally demanding, making the cost prohibitive. Thus, the goal of work is to come up with light-weight approach to account for the effect of scheduling.

Our approach builds on the classical model of the network introduced by Boorstyn et al [5]. In this model, groups of senders that can transmit concurrently are identified. A state model of the network is then constructed where each state represents the set of currently active senders. Throughput is computed based on the activation of these states.

The early models use unrealistic assumptions such as perfect capture (no collisions), Poisson traffic, packets are discarded if the medium is busy, as well as others. They also model a generic CSMA protocol that is quite different from bi-directional protocols such as IEEE 802.11. Recent studies [7]–[9] improve many aspects of the original model and tailor them towards IEEE 802.11. These models predict the link quality by considering the available capacity of the links *and* the scheduling effects. However, the core of the approach remains an iterative solution. In addition, some assumptions on the interference and traffic remain.

III. Motivation

CSMA protocols such as IEEE 802.11 are prone to collisions due to the hidden terminal problem [11], [12]. When collisions occur, packets are lost, and backoff values are increased, leading to underutilization of the channel. Since network flow formulations use coarse grained estimates of interference (effectively a function of the number of contenders), they ignore the impact of the

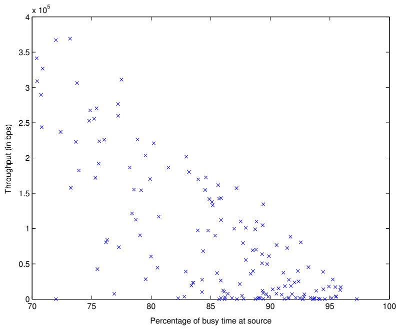


Fig. 1. Source busy time vs. Throughput

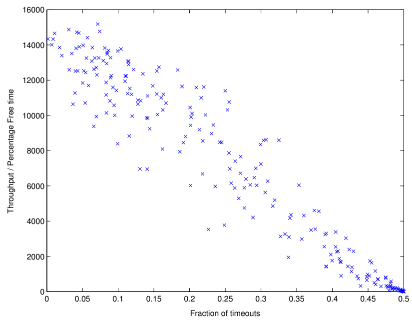


Fig. 2. Normalized Throughput vs. Percentage of MAC level timeouts.

scheduling effects (such as hidden terminals), and therefore over-estimate the link quality. In this section, we use a simulation study to demonstrate this effect. We also show that the normalized efficiency (achieved throughput relative to ideal throughput) correlates with the percentage of collisions suffered by each link. This observation motivates our approach, which estimates collision probability for each link and uses the probability to moderate the busy-time predicted capacity of the link.

We simulate different sets of 144 uniformly distributed nodes with 25 arbitrarily chosen *one-hop* CBR connections. Since the analysis targets MAC level interactions among interfering links, it does not require that the scenario be made up of multiple-hop connections–we are targeting link layer, rather than end-to-end phenomena. Modeling end-to-end effects such as chain self-interference and pipelining effects is a topic of future improvement.

Intuitively, under an ideal scheduler, the throughput of the link depends mainly on the available transmission time. Figure 1 plots the busy time[1] against the observed throughput of the link under moderate to high interference region. Its clear that link quality varies significantly for links with similar busy time values. Thus, using measures such as busy time result in inaccurate estimation of the link quality.

Next, we show that the reason for the variation in the observed link qualities for the same busy time is the scheduling effects which lead to packet drops. Let t_i be the throughput achieved by an ideal scheduler, which is proportional to the available transmission time. Let t_o be the observed throughput in the CSMA based scheduler in simulation. The ratio of $\frac{t_o}{t_i}$, called *normalized throughput* provides a measure of the scheduling efficiency relative to an ideal scheduler independently of the available transmission time. Figure 2 plots the normalized throughput against the percentage of MAC level transmissions that experience timeouts (due to collisions). It can be seen that as the fraction of frames experiencing timeouts increases, the normalized throughput decreases linearly. Thus *the reason for variations in observed capacity from the nominal capacity predicted by the interference metric are the scheduling effects which lead to collisions/timeouts.* As a result, our goal is to predict the effect of the scheduling behavior and use it to moderate the expected capacity obtained from aggregate interference metrics such as busy time.

[1]Similar results were obtained with other metrics such as busy time at destination and SINR at either source or destination. None of these metrics measure the impact of scheduling.

IV. Scheduling Effect in IEEE 802.11

In this section, we develop a lightweight model for estimating the effect of MAC scheduling on link quality by predicting the expected percentage of dropped packets. The network is represented as a graph $G(V, E)$ where V is the set of all the nodes and E is the set of active links. Let the gain matrix, (Θ_{ij}) represent the signal strength observed at node j for node i's transmission. The gain matrix can be obtained based geometric location of nodes or by measurements [13]. Let T_{RX} be the receiver sensitivity and T_{SINR} be the SINR threshold above which a signal is captured. W is Gaussian white noise.

In the scheduling analysis, we assume the following: (1) SINR physical layer model; (2) Transmissions are received instantaneously (zero propagation delay); and (3) The senders always have packets to send (saturated traffic). The last assumption is made to estimate the worst case impact of the scheduling. Note that this is not used directly as a measure of the link quality; rather, it used in combination with the coarse grained interference metric obtained from the network flow formulation. The integrated formulation can give appropriate weight for the effect of scheduling depending on the degree of interference.

The proposed approach builds on the Boorstyn model [5] which was reviewed in Section II. However, it differs from this work and others in literature [7]–[9] by considering a signal to interference and noise ratio

(SINR) model, as opposed to the idealized protocol model where every source has a fixed interference range. As a result, our model takes into account effects such as capture, and cumulative interference from multiple transmitters. Importantly, the model is also different because it is constructive; in contrast, existing models express behavior in terms of balance equations and solve them iteratively, incurring high computational cost, and hiding the nature of the processes that cause the observed performance.

We model the IEEE 802.11 protocol, with RTS-CTS. Although RTS-CTS packets are optional, it represents the more challenging case to model. We believe the approach –identifying and modeling the occurrence of collisions– generalizes to other MAC protocols. The problem is broken into the following parts: (1) Constructing the State Model via identification of the *Maximal Independent Contention Sets* (MICS); (2) Estimating the collision occurrence within each set; (3) Estimating the frequency of activation of each set; and (4) Combining the above estimates into a link quality estimate. In the remainder of this section, we discuss these steps in detail.

A. Constructing the State Model

This portion of the model is similar to the classical Boorstyn's Markov state model construction, with the exception that it is link-based rather than source-based. More specifically, each state in the original model represents the set of sources that can be concurrently active; in our model, states represent *the links* that can be concurrently active under a SINR based propagation model.

The first step in constructing the state model is to identify the set of edges that can be active concurrently, i.e which we call Independent Contention Sets (ICS). The sources of the edges in an ICS cannot hear to each others' transmission and hence, can initiate parallel transmissions. A Maximal ICS (MICS) is an ICS to which no other edge can be added without breaching independence of the sources (a maximal set of ICS).

Identifying all the MICS is an instance of the *Maximal Independent Sets* problem in graph theory and is known to be NP-hard. However, efficient approximate techniques exist for unit-disc graphs which arise in MHWNs [14]. We used an iterative heuristic [1] for finding independent sets. The states in the Markov model of the network consist of the power sets of the MICS.

Figure 3 shows a scenario of 4 one-hop connections. The MICS diagram is shown in the same figure. Each node in the figure represents an active edge in the scenario. The hidden terminals that cause packet timeouts are shown by a directed arrow. For example, in the above figure links (A, B), (G, H) and (E, F) are in the same MICS (denoted by MICS-1); and, (G, H) is a hidden terminal for the link (E, F) in MICS-1.

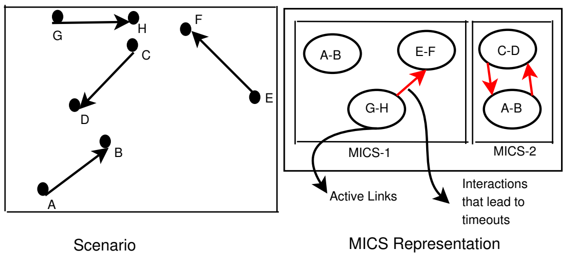


Fig. 3. An example of MICS

B. Estimating Timeouts in each MICS

The saturation traffic assumption allows us to focus on the MICS states, rather than all the states in the model (eliminating, ICS that are not MICS). Collisions occur due to interactions within an MICS; transmissions across different MICS' are prevented by the protocol, with the exception of concurrent transmissions that arise due to two sources sensing the medium to be idle and transmitting concurrently. Truly concurrent collisions are a minor effect in randomized protocols (less than 3% of collisions in our experiments (Section VI-A)); moreover, such collisions cannot be prevented by the routing protocol, so they are immaterial for our purposes.

In this component, we consider each MICS separately and attempt to estimate the packet timeout percentage given the physical layer model. In an 802.11 RTS/CTS handshake, the source may perceive a packet collision when: (1) it does not receive a CTS from the destination (an RTS timeout); or (2) it fails to get an ACK for the DATA frame (an ACK timeout).

1) Identifying RTS Timeouts: An RTS timeout occurs for one of three reasons: (1) an RTS collision; (2) a receiver not responding to the RTS due to its physical or virtual carrier sensing being active; or (3) a CTS collision. In IEEE 802.11, CTS collisions are improbable since the channel at the sender is idle before the RTS, and because CTS gets priority over other transmissions by virtue of a shorter inter-frame separation period before transmission. Hence, we focus on the first two causes.

We denote the set of links that can cause an RTS timeout to a given link (s, d) during the activation of MICS C as $\mathrm{U}^{C}_{(s,d)}$. In a given MICS C, an *unsafe link* (s_1, d_1) for a link (s, d) is a link whose transmission at source (s_1) may create a busy channel at the receiver d (the first condition of Equation 1) or cause an RTS collision (the second condition). These two cases correspond to the two causes for RTS timeouts discussed above.

$$\mathrm{U}^C_{(s,d)} = \Big\{(s_1, d_1) \mid (s_1, d_1) \in C,$$
$$\Theta_{s_1 d} \geq \mathrm{T}_{RX} - W \text{ or } \frac{\Theta_{sd}}{\Theta_{s_1 d} + W} < \mathrm{T}_{SINR}\Big\} \qquad (1)$$

2) Identifying ACK Timeouts: ACK timeouts, which are caused by collisions affecting DATA or ACK packets, are costly because they indicate the loss of the potentially long DATA packet. For the same reasons that CTS losses are improbable, ACK losses are also improbable. ACK timeouts are mostly due to DATA packet collisions.

The basis for modeling ACK timeouts is to determine the links (s_1, d_1) that can corrupt an ongoing DATA packet by a transmission from s_1 to d_1 (RTS or DATA packets) or from d_1 to s_1 (CTS or ACK packets). DATA collisions due to interfering DATA packets are rare since such transmissions only happen after successful RTS-CTS exchange. The CTS and ACK transmission have to be considered separately as the CTS transmission happens after sensing that the channel is not busy (Clear Channel Assessment (CCA)), whereas the ACK transmission happens without a CCA.

We define a set of links in a MICS C that can corrupt the DATA transmission of the link (s, d) by initiating an RTS, CTS or ACK by $\mathrm{A}^{(s,d)}_C$. These set of links are derived in a fashion similar to derivation of the *unsafe links*. Capturing the DATA packet collisions in a given MICS requires additional consideration of other intricate effects such as eliminating the links that experience hidden terminals, accounting for the "Virtual Carrier Sensing (VCS)". A detailed derivation of the ACK timeout estimate and related metrics is omitted due to space; it can be found in the technical report [15].

A DATA packet can also be dropped due to cumulative interference from multiple links (the longer duration of DATA packet transmission creates a higher possibility of such drops). We are able to detect this effect due to the SINR physical model we use; other models use a fixed interference range and cannot capture this effect [5], [8], [9].

C. Estimating the Probability of MICS Activation

In this step of the model, we need to estimate the frequency of activation of each MICS. From the previous step, we have a ranking of each link for each MICS it belongs to. Thus, we use the MICS probability to create a weighted average of the rating of each link across the MICS to which it belongs.

We use the saturation assumption to focus on MICS. Let m_C represent the probability that a MICS C is currently active. We observe experimentally that the relative MICS activation frequency correlates with the number of sources that belong to the MICS. The intuition is that with more contending sources, a MICS has a higher chance of one of the sources becoming active first and as a result, blocking MICS to which the active link does not belong from contention. Thus, we approximate the probability of the MICS activation by Equation 2. Informally, the amount of time a MICS will be active is proportional to the number of sources present in the MICS. However, with collisions and IEEE 802.11 backoff mechanism, the precise calculation of m_C is challenging. Refining this probability estimate is an area of future refinement of the model discussed in Section IV-E.

$$m_C = \frac{|C|}{\sum_{C' \in \mathrm{M}} |C'|} \qquad (2)$$

where $|C|$ notation is used to represent the number of links in MICS C.

D. Quantifying Link Quality

In this paragraph, we use the estimates developed in the previous sections to quantify the susceptibility of the links to RTS timeouts and ACK timeouts.

1) RTS Timeout Metric: Recall from Section IV-B.1 that the *unsafe links* cause RTS timeouts by two primary events: an RTS collision or a busy channel at the receiver. It can be seen that both these events are mainly caused by the DATA transmission from an unsafe link (we ignore the effect of smaller sized RTS/CTS/ACK packets). If the unsafe link has not yet started its DATA transmission, then an RTS timeout will not occur. Also, a larger number of unsafe links will result in higher chances of RTS timeout. Hence, the above two conditions – the ordering of the link transmissions and the number of unsafe links – need to be accounted for calculating RTS timeout metric.

We define $p_u(k)$ to be the probability that at least one of the k unsafe links of the given link (s, d) will initiate transmission before (s, d). It can be shown that $p_u(k) = \frac{1}{k+1} \sum_{i=1}^{k} (-1)^{(i-1)} \binom{k}{i}$ (please refer to technical report [15] for derivation).

The probability of an RTS Timeout for the link (s, d) in a MICS C is the conditional probability of the unsafe links transmitting before the given link ($p_u(|\mathrm{U}^C_{(s,d)}|)$), given that the MICS C is active The RTS Timeout metric of a link (s, d) is the sum of such conditional probabilities over all the MICS C that the link belongs (Equation 3). Note that m_C is the probability of occurrence of MICS C.

$$\mathcal{R}_{(s,d)} = \sum_{C \in \mathrm{M}, (\mathrm{s,d}) \in \mathrm{C}} p_u(|\mathrm{U}^C_{(s,d)}|) m_C \qquad (3)$$

2) ACK Timeout Metric: An estimate for the amount of ACK Timeouts is obtained by accounting for the links that corrupt DATA packet by RTS, CTS and ACK. ACK

Timeouts due to cumulative interference is also accounted by estimating the interference by multiple links of the same MICS. Let $\nu_C^{(s,d)}$ be the probability that the interference from the links of the MICS C corrupts the DATA packet. We calculate the $\nu_C^{(s,d)}$ based on observing the cumulative noise from the sources of the MICS (not shown due to space; please refer to the technical report [15]).

The fraction of ACK timeouts depend upon the probability of the link (s,d) winning the contention (represented by $w_{(s,d)}$). This can be directly derived by considering the set of all MICS that (s,d) belongs. Equation 4 represents the rating that a link (s,d) is susceptible to an ACK timeout considering the above factors.

$$\mathcal{D}_{(s,d)} = \frac{1}{w_{(s,d)}} \sum_{\forall C \in \mathrm{M}} \left(m_C \; p_u(|\mathrm{A}_C^{(s,d)}|) + \nu_C^{(s,d)} \right) \qquad (4)$$

We henceforth refer to the timeout ratings ($\mathcal{R}$ and $\mathcal{D}$) as *Interaction based Link Rating (IBLR)* metrics. IBLR metrics along with the unsafe links (U and A) will be used in the routing model to identify and constrain destructive link interference.

E. Discussion

A limitation of the current formulation is that we do not account for the effect of RTS timeouts due to the virtual carrier sense (VCS) being set at the receiver. Empirical evaluation in Section VI-A shows around 17% of observed timeouts were due to this effect. Modeling this effect requires estimating: (1) Packet capture ability of the node under a given MICS (which cannot be assumed to be a constant "reception threshold" as done in Protocol Model of interference); (2) Distinguishing the timeouts due to "False VCS" where an RTS packet turns on the VCS at a node and fails to followup with the DATA packet (due to RTS timeout). This effect is referred as "Gagged Node Situation" in [16]. A deeper study of VCS related effects is an area of extension of the model. We also note that for cases when RTS/CTS is not enabled, this effect is not present, and more accurate modeling of the quality is possible.

Precise formulation of probability of occurrence of a MICS (m_C) is a challenging problem due to the dynamic nature of MICS activation which depends on several factors.We are pursuing a more accurate constructive model of MICS probabilities based on the probability of link activation. However, a more accurate model is more complex; the current approximate solution allows faster searching of the optimization space.

V. Scheduling-aware Routing Formulation

Algorithm 1 Algorithm for SAR

```
 1: while iterationCount ≤MAX_ITER AND constraintsAdded
    = true AND Interference metric is acceptable do
 2:    Calculate routes by Network Flow (Netflow routes)
 3:    Calculate Interference metric for ALL the routes
 4:    Evaluate the routes by IBLR metric
 5:    Calculate Link quality metric link qualities considering
       routing behavior
 6:    constraintsAdded = Check for mutual excluding of con-
       flicting links and add constraints
 7:    CIM = Weighted measure of Interference metric and Link
       quality metric
 8:    if Netflow routes is the best route seen then
 9:       best_routes = Update the best routes
10:    end if
11: end while
12: return best_routes
```

In this section, we integrate the scheduling model into our previously developed multi-commodity network flow formulation [3]. Algorithm 1 briefly explains how the integrated framework works. We use the network flow model to provide an initial interference aware routing configuration, which we refer to as *Interference-Aware Routes* (IAR). The scheduling component evaluates these routes and finds poor quality links, and the links which cause them to have collisions (line 5 in Algorithm 1). This information is fed back into the network flow model in the form of additional constraints, allowing the solver to avoid using conflicting links (line 6 in Algorithm 1). The updated network flow problem is solved to obtain another candidate configuration (*Netflow routes*) and the procedure is repeated.

At each iteration, the overall quality of this routing configuration *Configuration Interaction Metric (CIM)* is computed. CIM serves as the fitness function of each configuration, which can be compared against other routing configurations. To compute CIM, the route quality obtained from the network flow optimization problem are combined updated with the scheduling derived link quality metric to obtain the overall quality. In addition, since drops at links closer to the destination are more costly than drops closer to the source, we bias the link quality to attempt to select more stable links as we get closer to the destination.

The iterative routing and link quality estimation process terminates when either no constraints can be added (since all the links have the IBLR rating lesser than a given threshold) or when the quality of the interference based metrics alone drops below a certain threshold (currently we use 20%) of the initial IAR routes. The configuration with the best CIM is chosen to derive the **Scheduling-Aware Routes (SAR)**.

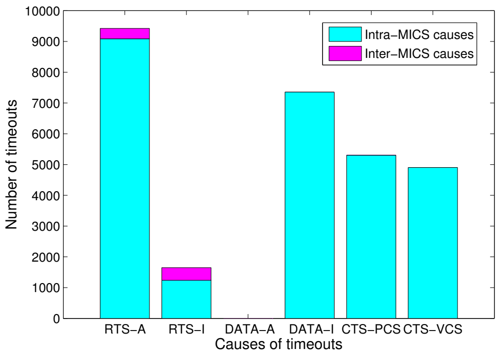


Fig. 4. Breakdown of Collision Causes

VI. Simulation study

We use the QualNet simulator [17], which is a commercial simulator with state of the art physical propagation and MAC models. We modify the simulator to account for the *SINR Threshold* propagation model. The transmission power and receiver sensitivity is set to 24.5 dB and -73 dB respectively. The data rate is set to 2 Mbps and the standard IEEE 802.11 MAC parameters are used. We first validate the scheduling formulation, in terms of accuracy and complexity, and then evaluate the performance of *Scheduling-Aware Routing (SAR)* formulation.

A. Empirical analysis of scheduling model

Detailed simulation results are used to empirically analyze the causes of the timeouts, which also aids in validating certain assumptions made during the derivation of RTS and CTS timeouts. Later, we evaluate the success of IBLR metrics in estimating timeouts. We use a scenario with random placement of 144 nodes in a 1600m$\times$1600m area with 25 one-hop connections. We use 10 different scenarios for a total of 250 active links.

Figure 4 compares the simulation results of the causes for packet timeouts to the average number of instances. *RTS-A* and *RTS-I* stands for the RTS packet collision during the *arrival* and *intermediate stage* of the RTS packet (DATA-A and DATA-I are similarly defined). *CTS-PCS* and *CTS-VCS* denotes the CTS packet not being sent by the destination as a result of physical and virtual carrier sensing , respectively. *Intra-MICS* and *Inter-MICS* are interactions observed within a MICS and between two MICS, respectively. As our assumptions stated, Inter-MICS collisions are quite rare (2.67% in our results) with most occurring in the *RTS-A* and *RTS-I* stage. It can be also observed that DATA packets generally experience collisions at intermediate reception stage and foreign transmissions

	$\mathcal{R}$	Interference Level	Busy Time	SINR
R	0.819	-0.121	0.655	-0.059
ρ	0.872	0.183	0.529	-0.404

TABLE I. Correlation of RTS Timeouts

	$\mathcal{D}$	Interference Level	Busy Time	SINR
R	0.916	-0.071	0.262	-0.061
ρ	0.899	0.110	0.320	-0.558

TABLE II. Correlation of ACK Timeouts

conflicting with DATA packet are not initiated in between the CTS transmission and the DATA reception (since DATA-A is approximately zero). These results validate a majority of the of the assumptions in capturing the occurrence of timeouts; a significant proportion that is not captured is the CTS-VCS (around 17% of the timeouts).

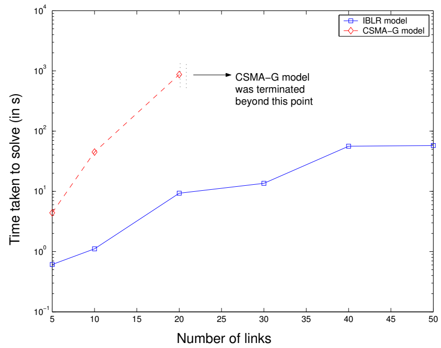


Fig. 5. Timing comparison

Tables I and II use correlation (R) and Spearman's Rank Coefficient (ρ) to measure how successfully different interference metrics predict timeouts [2]. We evaluate the IBLR RTS Timeout ($\mathcal{R}$) and ACK Timeout ($\mathcal{D}$) along with the aggregate metrics in Tables I and II. *Interference Level* was computed by measuring the interference at the destination of each link by all the active sources. The *Busy Time* is calculated as the amount of time the destination of a link was busy due to interfering traffic (measured in simulation). The ratio of the signal strength to the cumulative interference by all the other sources was used to calculate *SINR* metric. There is a strong correlation

[2]Correlation(R) is a statistical technique to measure the relationship between a pair of variables. Spearman's Rank Coefficient (ρ) is used to measure the ranking order correlation which is especially helpful in the absence of a linear relationship. R and ρ can take any real values between -1 and 1. A value of 1 indicates a perfect correlation and a value of 0 indicates independence of the two values. Negative values indicate inverse relationship.

between IBLR and simulation result while the aggregate interference metrics which are used in existing models [1]–[3], correlate poorly; they cannot predict the effect of scheduling.

The argument against using using existing accurate CSMA models for estimating the impact of scheduling is their high run-time. Figure 5 compares the time taken to evaluate a given network by the IBLR formulation and a recent accurate CSMA throughput model proposed by Garetto et al [9] (*CSMA-G*). CSMA-G estimates the expected throughput of each link using an iterative numerical process. As a result, its runtime grows very quickly with the size of the problem (note that the y-axis is log-scale). CSMA-G model was terminated if it failed to complete within 2 hours. In contrast, IBLR computation is lightweight, making it suitable for use in a network flow optimization framework.

B. Scheduling Aware Routing (SAR) evaluation:

In the next experiment, we evaluate the effectiveness of scheduling aware routes (SAR) obtained using the approach proposed in this paper. First, we compare SAR against the routes obtained by a conventional routing protocol (DSR [18]). Since SAR uses static routes, to provide a fair comparison that eliminates routing overhead and dynamic effects such as false disconnections [19], we allowed DSR to use static routes (selecting the most commonly used route found by DSR). We refer to such routes as *S-DSR*. This main objective of this evaluation is to demonstrate the vulnerability of Interference-only-aware routing models (IAR) and the effectiveness of SAR under IEEE 802.11. The scheduling-unaware IAR formulation is prone to RTS/ACK timeouts, thus leading to poor performance, while SAR has to pick the links that is both interference-aware and scheduling-aware.

We first demonstrate the effectiveness of SAR in a 10×10 grid topology in a 1000×1000 area. The transmission range is around 390m. Two connections are placed on the opposite edges of the grid.

Figure 6(a) compares the average throughput of the connection for varying packet sending rates. The scheduling effects play a less important role under lower traffic as concurrent transmission between unsafe links is less likely. Under high packet sending rates, it can be observed that the throughput of the scheduling aware scheme is significantly higher than the other schemes. The average throughput of the S-DSR is lesser than the IAR and SAR routes. The throughput of the DSR and S-DSR fluctuates widely over different seeds. In around 30% of the cases, the best routes from S-DSR performed better than the IAR routes. DSR treats packet drops (after 4 ACK timeouts, or 7 total timeouts for the same packet) as an indication of broken

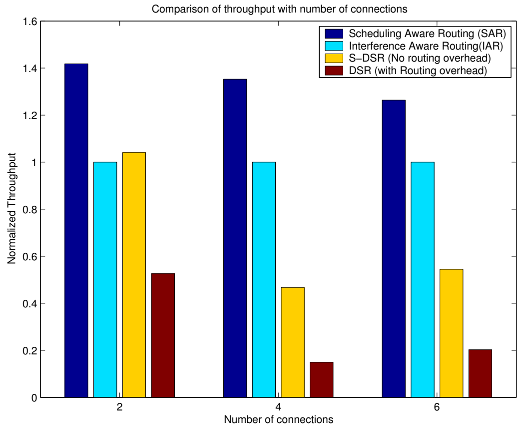


Fig. 7. Effect of multiple connections

path and switches to another path. Thus, the most used path in S-DSR are likely to have relatively few timeouts which will allow them to be used for a long time without packet drops.

In this regular setting, SAR found routes with no destructive scheduling interactions, leading to zero packet timeouts (Figure 6(b)). An improvement of around 50% was observed in end-to-end delay.

Next, we analyze the effect of cross connection interference. The number of connections in the above grid topology is increased, thus increasing the number of competing links. Although SAR cannot find a routes with perfect scheduling, it chooses the routes with low scheduling conflicts while maintaining interference separated routes. The normalized throughput is shown in the Figure 7.

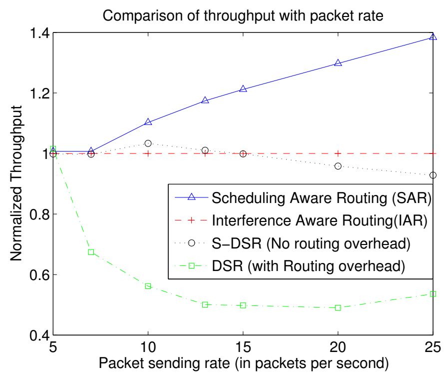
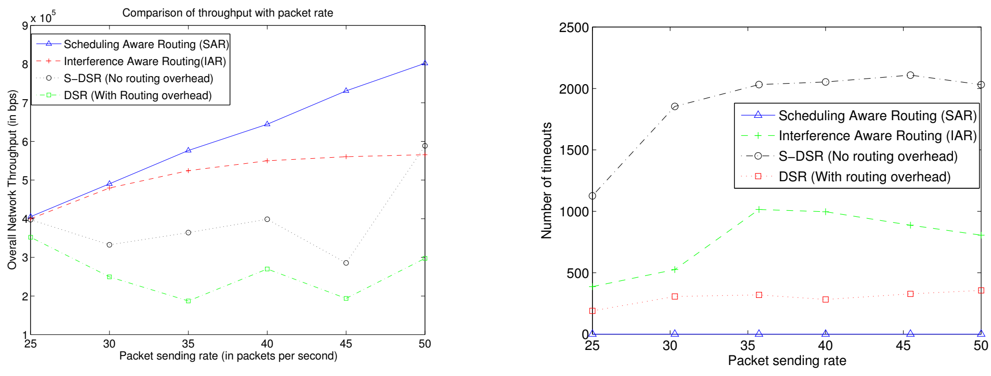


Fig. 8. Throughput study in random scenarios

We now examine the performance of these schemes in 10 random scenarios with 64 nodes placed in $1000\text{m}\times1000\text{m}$ area. Six connections were randomly selected such that the source and destinations are approxi-



(a) Throughput

(b) Timeouts

Fig. 6. Study of connection parameters v/s packet sending rate in a Grid topology

mately 3 or 4 hops away. Figure 8 shows the throughput improvement of SAR over other protocols (normalized with respect to IAR throughput). An improvement of 33% over IAR and a greater improvement over the DSR protocol under high loads was observed. Notable improvements in other metrics like end to end delay and packet drops were also observed. Detailed results is available in the technical report [15].

VII. Concluding Remarks

Aggregate metrics of link capacity such as channel busy time do not account for the effect of scheduling. It is known that in CSMA networks, scheduling plays a critical role in how interference is manifested. Thus, it is desirable to avoid links that experience mutually destructive interaction. We used this observation to develop a methodology for rating link quality based on their interactions with other links. We show that these estimates correlate strongly with packet drops. We integrate the developed model in a network flow formulation of traffic engineering in static MHWNs. We show that capturing the scheduling effects leads to considerable improvement in performance of the derived routes. Our future work includes continued refinement of the model, which uses coarse approximations in several points.

References

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

[2] M. Kodialam and T. Nandagopal, "The Effect of Interference on the Capacity of Multi-hop Wireless Networks," in *Bell Labs Technical Report*, 2003.

[3] V. Kolar and N. B. Abu-Ghazaleh, "A Multi-Commodity Flow Approach to Globally Aware Routing in Multi-Hop Wireless Networks," in *IEEE PerCom*, 2006.

[4] M. Chiang, S. H. Low, A. R. Calderbank, and J. C. Doyle, "Layering as optimization decomposition: A mathematical theory of network architectures," in *Proceedings of IEEE*, 2007.

[5] R. R. Boorstyn, A. Kershenbaum, B. Maglaris, and V. Sahin, "Throughput analysis in multihop csma packet radio networks," *IEEE Trans. on Communication*, 1987.

[6] F. A. Tobagi and J. M. Brazio, "Throughput analysis of multihop packet radio network under various channel access schemes," *IEEE INFOCOM*, 1983.

[7] M. M. Carvalho and J. J. Garcia-Luna-Aceves, "A scalable model for channel access protocols in multihop ad hoc networks," in *MobiCom*, 2004.

[8] Y. Gao, D.-M. Chiu, and J. C. Lui, "Determining the end-to-end throughput capacity in multi-hop networks: methodology and applications," *SIGMETRICS Perform. Eval. Rev.*, 2006.

[9] M. Garetto, T. Salonidis, and E. W. Knightly, "Modeling per-flow throughput and capturing starvation in CSMA multi-hop wireless networks," in *IEEE INFOCOM*, 2006.

[10] P. Gupta and P. Kumar, "The Capacity of Wireless Networks," in *IEEE Trans. on Info. Theory*, 2000.

[11] V. Bharghavan, A. Demers, S. Shenker, and L. Zhang, "MACAW: A Media Access Protocol for Wireless LANs," in *SIGCOMM*, 1994.

[12] M. Garetto, J. Shi, and E. W. Knightly, "Modeling media access in embedded two-flow topologies of multi-hop wireless networks," in *MobiCom '05*, 2005, pp. 200–214.

[13] C. Reis, R. Mahajan, M. Rodrig, D. Wetherall, and J. Zahorjan, "Measurement-based models of delivery and interference in static wireless networks," in *Proc. of SIGCOMM 2006*, 2006, pp. 51–62.

[14] R. Gupta and J. Walrand, "Approximating maximal cliques in ad-hoc networks," in *Personal, Indoor and Mobile Radio Communications (PIMRC)*, 2004.

[15] V. Kolar and N. Abu-Ghazaleh, "The effect of scheduling on link capacity in multi-hop wireless networks," in *CoRR, arXiv:cs.NI/0608077 v1*, 2006.

[16] M. Durvy and P. Thiran, "Understanding the gap between the IEEE 802.11 protocol performance and the theoretical limits," in *IEEE SECON*, 2006.

[17] "Qualnet network simulator, version 3.6," http://www.scalable-networks.com/.

[18] D. B. Johnson, D. A. Maltz, Y.-C. Hu, and J. G. Jetcheva, "The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks (DSR)," 2002.

[19] S. Xu and T. Saadawi, "Revealing the problems with 802.11 medium access control protocol in multi-hop wireless ad hoc networks," *Comput. Networks*, vol. 38, no. 4, pp. 531–548, 2002.