MIDAS: Inference and Mitigation of Hindering Transmissions in Managed 802.11 Wireless Networks

Eugenio Magistretti, Omer Gurewitz and Edward Knightly
Rice University Technical Report
TREE3610

***Abstract*—In 802.11 managed wireless networks, the manager can address under-served links by rate-limiting the conflicting nodes. In order to determine to what extent each conflicting node is responsible for the poor performance, the manager needs to understand the coordination among conflicting nodes' transmissions. In this paper, we present a management framework called MIDAS (Management, Inference, and Diagnostics using Activity Share). We introduce the concept of Activity Share which characterizes the coordination among any set of network nodes in terms of the time they spend transmitting simultaneously. Unfortunately, the Activity Share cannot be locally measured by the nodes. Thus, MIDAS comprises an inference tool which, based on a combined physical, protocol, and statistical approach, infers the Activity Share by using a small set of passively collected, time-aggregate local channel measurements reported by the nodes. MIDAS uses the estimated Activity Share as the input of a simple model that predicts how limiting the transmission rate of any conflicting node would benefit the throughput of the under-served link. The model is based on the current network conditions, thus representing the first throughput model using online measurements. We implemented our tool on real hardware and deployed it on an indoor testbed. Our extensive validation combines testbed experiments and simulations. The results show that MIDAS infers the Activity Share with an average normalized relative error as low as 4% in testbed experiments.**

I. Introduction

Managed enterprise WLANs and wireless mesh networks regularly encounter underperforming links, i.e., links with throughput below an acceptable value determined by the operator. A key corrective action available to the network manager is to throttle other nodes that may be hindering the underperforming link. However, to do so first requires identifying which node to throttle. While it is clear it should be a "neighbor," there may be a large set of candidate nodes for which throttling can have vastly different effects, including no effect on the under-served link. Moreover, it is not immediately evident how much throttling any node will increase the throughput of the targeted under-served link due to complex node interactions and coordination.[1]

In this paper, we design MIDAS, a framework that uses online measurements of network performance to infer the most hindering nodes which cause a target link to be under-served or to obtain poor performance. Moreover, MIDAS identifies effective management actions to increase the performance of the under-served link by appropriately limiting the transmission rates of the hindering nodes. Finally, we implement MIDAS on real hardware and investigate its performance in an indoor testbed and simulation.

MIDAS employs a methodology comprising three procedures: *i) measurement collection*, which gathers reports from each node consisting of a small set of passive time-aggregate measurements; *ii) inference*, which infers the coordination among the transmissions of different sets of nodes using the reported measurements; *iii) prediction*, which utilizes the inferred information to predict the throughput gain of any target link, corresponding to rate-limiting different conflicting nodes. In particular, our contributions are as follows.

First, we introduce the concept of *Activity Share* which characterizes the coordination and interference among any set of conflicting nodes. The throughput of a link is influenced by the sender busy time (i.e., the more the sender senses the medium busy the less it can transmit), and the collision probability (i.e., even if it can transmit, its transmissions are corrupted). Coordination is critical to understand how different nodes contribute to busy time and collision probability of each other. In fact, a sender's busy time is not simply the sum of the transmission times of its neighbors, as neighbors which are hidden from one another may transmit simultaneously. Analogously, link collisions are not the sum of the collisions with *each* hidden terminal, because multiple hidden terminals may collide with the same packet. Therefore, knowing which conflicting nodes are destructive to a link requires understanding their coordination. In order to capture node coordination, we define network state as a set of transmitting nodes; accordingly, in each time instant the network is in a unique state. We define *Activity Share* as the time share the network spends in each possible state in a given interval. That

E. Magistretti and E. Knightly are with the Department of Electrical and Computer Engineering, Rice University, Houston, TX 77005 USA (e-mail: {emagistretti,knightly}@ece.rice.edu). O. Gurewitz is with the Department of Communication Systems Engineering, Ben Gurion University of the Negev, Beer Sheva 84105, Israel (e-mail: gurewitz@cse.bgu.ac.il; gurewitz@gmail.com).

[1] Network managers have a number of options for mitigation, including moving sets of APs or clients to alternate frequencies. MIDAS could equally be applied to such strategies (it would identify the best ones to move and could recompute the new throughputs). However, evaluation of such alternate mitigation schemes is beyond the scope of this paper.

is, the Activity Share is a vector including, for each possible set of nodes, the fraction of time they spend transmitting simultaneously. Note that the Activity Share depends not only on the topological relationships between the nodes as determined by carrier sensing and link interference, but also on the transmission rate and pattern of each node under the current traffic conditions. Furthermore, since the transmission pattern of any node depends on the transmission pattern of its neighbors, and the transmission pattern of its neighbors depends on the transmission pattern of their neighbors (and thus recursively of all nodes in the network), the Activity Share captures the effects of global network interactions that extend beyond node locality. In particular, the Activity Share captures the coordination among the transmissions of any set of conflicting nodes as determined by the current global network conditions. In contrast, alternative indicators, such as the individual node transmission rates, are insufficient to determine how conflicting nodes influence the target link, since they do not capture the coordination. For example, the conflicting node with the highest transmission rate might mostly transmit simultaneously with other conflicting nodes, such that limiting its rate may scarcely benefit the target link. We will show how the manager can utilize the Activity Share as a tool to understand the network behavior and to determine a strategy to change it, e.g., to increase the throughput of a congested or underperforming link. Unfortunately, the estimation of the Activity Share is challenging, because it cannot be locally measured by the nodes. In fact, during the reception of multiple overlapping packets, nodes cannot identify all senders, and thus recognize the network state.

Second, we design a tool to infer the Activity Share using a small set of *passively collected*, *time-aggregate* local channel measurements, reported by every node. Inferring the Activity Share requires computing the temporal distribution of the different network states, i.e., how long the network spent in each of them. We develop a technique to eliminate infeasible distributions by incorporating physical rules (e.g., the busy time of a node should coincide with the sum of the durations of the states in which its neighbors transmit and that node does not). Unfortunately, there can be an infinite number of temporal distributions that yield identical measurements. Consequently, we penalize unlikely distributions by incorporating protocol rules (e.g., the occurrence of states in which adjacent nodes simultaneously transmit is unlikely), and select a representative by using a statistical approach based on entropy considerations. To further limit the complexity of our problem, we propose a technique to reduce its dimensions, by actually *eliminating* the unlikely distributions.

Third, we develop a tool to predict the throughput increase achievable on the target link by rate-limiting the links formed by the target link's conflicting nodes. The Activity Share permits assessment of the current network conditions; however, it lacks predictive power to identify effective rate-limiting actions and to anticipate their outcomes. The challenge is to understand how changing the transmission time of a conflicting node affects the Activity Share, and subsequently how the new Activity Share affects the target link's throughput. We design a simple throughput prediction model that derives its inputs from the current network conditions, i.e., from the inferred Activity Share, thus representing the first throughput model based on online measurements.

Fourth, we extensively evaluate the accuracy of MIDAS by combining testbed experiments and simulations. We implemented MIDAS on real hardware and deployed it on an indoor testbed, where we investigated its sensitivity to different network settings under real channel conditions. The results show that MIDAS infers the Activity Share with high accuracy, i.e., with an average normalized relative error as low as 4%. In order to extend our validation to a broader set of scenarios, we performed numerous simulations. A key finding is that, by rate-limiting different conflicting nodes for the same fixed amount, the throughput of the target link can increase from 7% to 172% of the rate-limited quantity. We also validate the effectiveness of the Activity Share in supporting throughput prediction, and show that MIDAS anticipates the benefits of alternative rate-limiting actions with an error lower than 20% of the rate-limited quantity.

The remainder of the paper is organized as follows. In Section II we present MIDAS, and define the Activity Share. We develop a technique to infer the Activity Share in Section III. A throughput prediction tool using the Activity Share is described in Section IV. Section V presents testbed and simulation results. Finally, Section VI overviews related works, and Section VII concludes the paper.

II. The MIDAS Framework

A link can be considered under-served due to a discrepancy between the network manager's targeted link throughput and the actual throughput. The network manager's policy for setting target throughputs (incorporating factors such as fairness, QoS, pricing, offered load, etc.) is beyond the scope of this paper. The objective of MIDAS is to determine the causes of the poor performance and design corrective actions.[2] While local node observations can point out problematic links, in general the causes of the low throughput cannot be locally inferred. For instance, in the case of high packet drop rate, the local measurements can seldom determine the hindering nodes. MIDAS helps improving the problematic link by inferring the impact of hindering transmitters and by rate-limiting the most destructive flows.

The severity of link hindering interactions mainly depends on three factors: *i)* network topology: nodes' pairwise relations, as determined by carrier sensing and interference, determine the form of interaction, e.g., hidden terminals are responsible for transmission corruptions, while carrier sensed nodes affect deferral; *ii)* link transmission rates: nodes that transmit few packets are less likely to interfere with a target link; and *iii)* link transmission coordination: the number of link packets corrupted by a hidden terminal depends on how frequently a link sender and hidden terminal transmit simultaneously. Note that transmission rates and coordination strongly depend on the traffic load of each node.

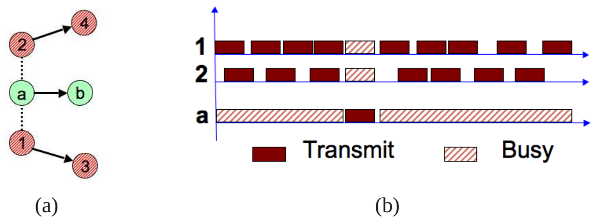
[2] In this paper, we only consider 802.11 MAC issues, e.g., we do not address throughput losses due to TPC dynamics, or low received signal strength.

In this section, we introduce a novel metric termed *Activity Share* which captures the coordination between any possible set of nodes, by measuring the fraction of time they transmit simultaneously. Even though the Activity Share does not directly measure the interference between nodes, it reflects node interactions. Thus, the Activity Share is affected by node interference relationships, traffic load, MAC protocol, etc. We will show how MIDAS can utilize the Activity Share to evaluate the potential effects of alternative corrective actions (see Section IV). We will also show that the Activity Share cannot be locally observed by the network nodes and describe how it can be inferred from measurements collected by the nodes. Note that, in contrast to the Activity Share, alternative indicators that evaluate *pairwise* conflicts between interfering links taking into consideration only *topological* information (e.g., the conflict graph [10]) miss the important dynamic information about the coordination of the transmissions of multiple nodes.

A. The Activity Share: Fundamental Element of Network Observation

As previously explained, our management framework aims to identify the originating causes of under-served links and to increase their throughput by rate-limiting conflicting nodes. In this study, we consider 802.11 stationary multihop wireless networks, including enterprise WLANs and mesh networks. In such networks, nodes can affect the throughput attained on a link (sender-receiver pair) by two key means: *i)* reducing the time the medium is perceived as free by the sender, thereby forcing the sender to defer; *ii)* corrupting the packet reception at the receiver end, i.e., colliding. In multi-hop topologies, despite the use of the carrier sensing mechanism, several nodes that are in conflict with a specific transmitter can potentially transmit simultaneously. Hence, it is challenging to anticipate the benefits of rate-limiting conflicting links on the sender busy time or collisions of the target link, and thus on its throughput. Even knowing the exact packet transmission rate of each node in conflict with the link of interest is not sufficient, because the throughput gain mainly depends on the coordination among the conflicting nodes as illustrated in the example below.

Example. The following example shows that node coordination is the key to understand the effectiveness of rate-limiting conflicting nodes to improve the throughput of an under-served link. Let us consider the simple wireless network depicted in Figure 1(a), where a dotted line connecting two nodes indicates that the two nodes are within carrier sensing range. The link (a, b) is identified as under-served; the goal of the network manager is to assess how decrementing the transmission rates of the conflicting links formed by nodes 1 and 2 can benefit the throughput of link (a, b). Since nodes 1 and 2 are not coordinated by carrier sensing, they can transmit simultaneously. Figure 1(b) depicts a typical timeline of the transmissions of the three nodes. The continuous deferral is the cause of the performance issue of link (a, b); in fact, (a, b) can transmit only when both nodes 1 *and* 2 are silent. Thus, decreasing the transmission rate of only one of them will produce a minimal benefit to (a, b); this is because only a small portion of the released airtime will result in free airtime for (a, b). The analysis of the coordination between the conflicting nodes 1 and 2, and in particular of the large overlap between their transmissions, can promptly lead to this conclusion. Obviously, this is only a simple case, where the large overlap between the transmissions of 1 and 2 is not surprising; however, in more complex topologies with several conflicting nodes, it is not clear how to determine node coordination and its effect.



(a) (b)

Fig. 1. Example of transmission alignment due to (lack of) carrier sensing.

Network State and Activity Share. The key to understanding how conflicting nodes affect an underserved link is to determine the time they spend transmitting simultaneously. For instance, in the example in Figure 1, the transmissions of nodes 1 and 2 mostly overlap, anticipating a small gain in free airtime perceived by the link (a, b), from the reduction of the transmission times of either one. Furthermore, the higher the number of nodes in conflict with a target link which can potentially transmit simultaneously, the lower the gain from limiting the transmission time of a single node. For instance, if in the example instead of two uncoordinated nodes in conflict with link (a, b), there were three or more such nodes, the free airtime gained by rate limiting a single node would be even lower.

Let us consider an N-node network. To formalize the concept of simultaneous transmission of a set of nodes, we define *network state* and *Activity Share* as follows.

Definition 1: The *Network State* $\vec{D}$ denotes the transmission status of each node in the network. $\vec{D}$ is an N-dimensional vector comprising an entry for each node that indicates whether the node is transmitting or idle in the state. $\vec{D} = (d_1, d_2, \dots d_N), \quad d_i \in \{0, 1\}$, where $d_i = 1, 0$ indicates that node i is transmitting or not, respectively. Note that each network state is univocally identified by the set of transmitting nodes.

Since there are N nodes in the network there are 2^N possible states denoted by $\vec{D}_1, \vec{D}_2, \dots, \vec{D}_{2^N}$. The network transitions in time through a succession of network states. The *Instantaneous Network State* at time t_0, $\vec{D}(t_0)$, is the state of the network at time t_0, i.e., $\vec{D}(t_0) = \vec{D}_j$ *iff* the network state at time t_0 is $\vec{D}_j$.

Next, we define the *Activity Share* which is the time share the network spends in each state per time unit.

Definition 2: The *Activity Share of the network state* $\vec{D}_j$, denoted by $AS(L, \vec{D}_j)$, is the fraction of time during the interval $[0, L]$ for which the network was in state $\vec{D}_j$, i.e., $AS(L, \vec{D}_j) = \frac{1}{L}\int_0^L \mathbf{1}_{[\vec{D}(t)=\vec{D}_j]}(t)dt$, where $\mathbf{1}_{[\vec{D}(t)=\vec{D}_j]}(t)$ denotes the indicator function such that $\mathbf{1}_{[\vec{D}(t)=\vec{D}_j]}(t) = 1$ if the network state at time t is $\vec{D}_j$, and 0 otherwise. The sum of $AS(L, \vec{D}_j)$ over all possible states adds to one:

$$\sum_{j=1}^{2^N} AS(L, \vec{D}_j) = 1 \qquad \forall L \tag{1}$$

We separately denote as *Activity Share*, $\vec{AS}$, the distribution of time among all states that the network visited during the time interval $[0, L]$, i.e., $\vec{AS} = \{AS(L, \vec{D}_j), \forall \vec{D}_j\}$. Note that if the network is stationary $\lim_{L\to\infty} AS(L, \vec{D}_j)$ is the probability that the network at any time instant is in state $\vec{D}_j$. In the following, we consider L large enough to satisfy stationarity, and we drop L from our notation.

The estimation of the Activity Share is challenging because it cannot be locally measured by the nodes. Specifically, the nodes cannot identify the transmitters of all the packets they carrier-sense. In fact, some of the overlapping packets (e.g., sent by 1 and 2 in Figure 1) may collide at the intermediate nodes (e.g., node a), preventing the decoding of at least one of them. Another obstacle is the strength of the received signal, which may exceed the carrier sense threshold or generate collisions, but not be sufficiently greater than the background noise to permit the decoding of the packet. In order to overcome these challenges, it is necessary to analyze the combined measurements of different nodes.

B. The Measurements

In MIDAS, each network node k continuously collects information, and delivers a report R_k to the manager at every report interval. In this paper, we suggest a new scheme which we will use to infer Activity Share, given a set of measurements reported by the nodes $\vec{R} = \{R_1, R_2, ...R_N\}$.

A tradeoff emerges between the amount of information contained in R_k and the estimation accuracy of the Activity Share. If R_k contained complete traces of the exact times and durations of all transmissions of node k, the manager could use the reports to reconstruct a global trace of the transmissions in the network (such as in Figure 1(b)), and hence obtain the Activity Share by inspection. However, the amount of information that needs to be collected and the timely delivery of such traces would overwhelm the network resources. For example, a set of traces satisfying our requirements is collected in [5]; therein, the authors show that the overhead is between 100 kbps and 500 kbps per node, without even considering the multiplicative effect of multi-hopping [3].

We consider a highly simplified and easily measured set of inputs R_k consisting of information passively collected from the local network card and time averaged over the report interval. Each node observes the local channel in three states: T if the measuring node is transmitting; B if the node is not transmitting but the total received energy exceeds the carrier sensing threshold; I if neither the received energy exceeds the carrier sensing threshold, nor the node itself is transmitting. Notice that the state B reflects the activity of all carrier sensed nodes and does not distinguish between different transmitters. The report R_k includes the time shares T_k, B_k, I_k node k observed the channel in any of the three states during the report interval. Clearly, $T_k + B_k + I_k = 1$, $\forall k$. An implementation of the measurement collection tool is presented in Section V-A. In contrast to trace-based solutions, our reports only include two numerical values.

III. The Inference Tool

The reconstruction of the Activity Share from the reports is challenging because the time-average measurements in R_k are the result of the transmissions either of the individual node k (i.e., T_k) or of *all* its neighbors (i.e., B_k). In both cases, it is not possible to locally determine the overlapping intervals of subsets of neighbors, and of sets of nodes that do not share neighbors. In this section, we will show how to overcome this issue; our solution consists of three elements. First, in order to obtain accurate estimations, we use the $\vec{R}$ inputs to constrain the domain of the feasible $\vec{AS}$ (Section III-B). Since the constraints do not generally identify a unique solution, we propose an optimization problem to choose a single representative $\vec{AS}$ (Section III-C). The last element of the solution addresses the computational complexity of the proposed problem, and reduces the dimension of the $\vec{AS}$ solution space using protocol rules of 802.11 (Section III-D). In the experimental results in Section V we consider practical implementation issues, such as report losses and time-varying channel.

A. Network Model

We consider a single-radio, single-channel network, and we abstract it as a graph $G = \{V, E\}$, where the vertices V represent the N nodes, and the edges E represent the carrier sensing relationships among the nodes. The existence of a sensing edge $(i, j) \in E$ means that node i carrier senses transmissions from node j and vice versa. We define the set of the nodes that node i carrier senses as $V_{cs}(i) = \{j|(i, j) \in E\}$. We assume that the topology of the graph with respect to E is fixed during any observation interval and known to our inference tool (e.g., via offline link profiling [19], or passive online estimations [13]).

B. Report-based Constraints

In order to obtain an accurate estimation of the Activity Share, we use the reported measurements $\vec{R}$ to constrain the feasible domain. Since the local observations of the channel of any node provide information about the cumulative duration of sets of network states, the actual $\vec{AS}$ must satisfy the constraints imposed by all local observations, and hence lies in the feasible region the observations define. Accordingly, we can derive the following constraints:

$$\sum_{j:(D_j^k=1)} AS(\vec{D}_j) = T_k \tag{2}$$

$$\sum_{j:(D_j^k=0)\wedge(\exists s\in V_{cs}(k):D_j^s=1)} AS(\vec{D}_j) = B_k \tag{3}$$

$$\sum_{j:(D_j^k=0)\wedge(D_j^s=0,\forall s\in V_{cs}(k))} AS(\vec{D}_j) = I_k \tag{4}$$

$$\forall k \in [0..N]$$

where D_j^n denotes the n-th component of the $\vec{D}_j$ vector. Equation (2) constraints the time share each node is transmitting: the sum of the Activity Shares of states in which node k transmits should be equal to the fraction of time k transmitted. Equation (3) is related to the busy time of the nodes. In our network model, the state of a node k is busy if the node is not transmitting and any of the nodes in $V_{cs}(k)$ is transmitting. Hence, the Activity Shares of states, in which any of the nodes in $V_{cs}(k)$ is transmitting and node k is not, sum up to B_k. Notably, also the busy time of the nodes carries information about the Activity Share, by inducing constraints on the duration of the network states including transmissions from any neighboring node. The assumption that the links in E are fixed plays a crucial role in enforcing this constraint. Even though this is a simplifying assumption, related research shows that threshold-based carrier sensing relationships can be reasonably well approximated as binary [18]. Our experimental results, and a specific discussion in Section V-B, evaluate the effects of this assumption. Equation (4) relates to the idle time of the nodes, and can be obtained with considerations analogous to the previous two. Simple considerations show that any of the three equations associated with each node is redundant with respect to the remaining two and Equation (1). This fact can be easily verified by noticing that the state indexes used for the three constraints (2), (3), (4) are a partition of the whole set of indexes, thus they sum up to the left hand-side term of Equation (1). Thus, we consider Equation (4) redundant for all nodes.

C. Entropy-based Statistical Solution

In this subsection, we show how to determine a representative $\vec{AS}$ close to the actual $\vec{AS}$ occurred during the measured interval. The representative $\vec{AS}$ should satisfy the report constraints, since the actual $\vec{AS}$ determines the reported measurements. However, the constraints we defined do not identify in general a single $\vec{AS}$, but rather a feasible solution domain. Each Activity Share distribution $\vec{AS}$ in the domain defined by the reports would have generated the exact same observations obtained by the nodes; hence, the selection of any of these $\vec{AS}$ is admissible. However, a key observation is that not all feasible solutions are equally likely, e.g., 802.11 introduces a bias against states that include simultaneous transmissions of mutually carrier sensing nodes. We formalize this bias using the *a priori* distribution of the states, and we select our representative $\vec{AS}$ as the feasible solution closest to the *a priori* distribution.

Protocol-driven a priori information. As shown in Section II-A, we can give a statistical interpretation of the components of the Activity Share. Each $AS(\vec{D}_j)$ corresponds to the probability the network is in the state $\vec{D}_j$ at a random time instant. Because of the carrier sensing behavior of 802.11, not all network states have *a priori* identical probabilities of occurrence, i.e., $AS(\vec{D}_j)$ is *not a priori uniform* (i.e., equal to $\frac{1}{2^N}$) over all states $\vec{D}_j$. In fact, 802.11 carrier sense aims to prevent the occurrence of states where neighboring nodes transmit simultaneously, i.e., $\{\vec{D}_j \mid \exists k, l : l \in V_{cs}(k), D_j^k = 1, D_j^l = 1\}$. Practically, two neighbors can transmit simultaneously only if their backoffs expire within a slot interval, while non-adjcent nodes can initiate their transmissions independently. As a consequence, among the admissible $\vec{AS}$, our scheme should favor the $\vec{AS}$ that do not assign large probabilities to states including neighbor transmissions.

We model the protocol behavior of 802.11 by identifying an *a priori* distribution that assigns probabilities to the states $\vec{D}_j$ unequally. Since trying to capture the exact prior probability of each state according to 802.11 is very complicated, we use a coarse-grained approximation. Nonetheless, we will show in Section V that our technique attains high accuracy. Our fundamental idea is to assign to the network states *a priori* probabilities exponentially decreasing with the number of adjacent transmitters they contain, e.g., the states containing two pairs of adjacent transmitters have half the probability of those that contain only one pair. Notice that this assignment partitions the states $\vec{D}_j$ in classes, where all the states in the same class contain identical numbers of adjacent transmitters, and thus have equal probabilities. For instance, class 0 includes all states that do not contain adjacent transmitters and have probability p, class 1 includes all states that contain only one pair of adjacent transmitters and have probability $p/2$, etc.

Minimum Relative Entropy $\vec{AS}$ inference. In the previous paragraph, we formalized our knowledge of the protocol behavior by using an *a priori* distribution of $\vec{AS}$. Our objective is to select the feasible $\vec{AS}$ closest to the defined *a priori* distribution. We propose to use the concept of Kullback-Leibler distance [6] to quantify the distance between two distributions, and select the representative $\vec{AS}$ as the feasible solution that minimizes such distance from the *a priori* distribution. Accordingly, the problem is formulated following the Minimum Relative Entropy Principle.[3] Out of the feasible solutions that have equal Kullback-Leibler distance from the *a priori* distribution, the Minimum Relative Entropy Principle favors the solutions that spread the probability of the states in the same class as evenly as possible. In fact, in absence of any other information about the 802.11 protocol behavior, all states that the *a priori* distribution assigns to the same class have identical probability. Hence, any different probability assignment would introduce an unmotivated bias.

The $\vec{AS}$ Inference problem. We formulate the $\vec{AS}$ inference problem as:

$$\begin{aligned} \min_{\boldsymbol{x}} \quad & \sum_{j=0}^{\gamma-1} x_j \log \frac{x_j}{w_j} \\ \text{s.t.} \quad & \boldsymbol{\Phi} \cdot \boldsymbol{x} = \boldsymbol{T} \\ & \boldsymbol{\Psi} \cdot \boldsymbol{x} = \boldsymbol{B} \\ & \boldsymbol{1}' \cdot \boldsymbol{x} = 1 \\ & \boldsymbol{x} \geq \boldsymbol{0} \end{aligned}$$

where γ is the cardinality of the set of admissible network states (2^N in this case); $\boldsymbol{x}$ is a γ-dimensional vector, whose j-th entry is $AS(\vec{D}_j)$; $\boldsymbol{w}$ is the prior distribution of the network states; $\boldsymbol{\Phi}$ is an $N \times \gamma$ matrix, whose ij-th entry is 1 if $D_j^i = 1$, 0 otherwise; $\boldsymbol{\Psi}$ is an $N \times \gamma$ matrix, whose ij-

[3] Note the that minimizing the relative entropy is equivalent to maximizing the expected value of the log-likelihood.

th entry is 1 if $D^i_j = 0$ and $\exists s \in V_{cs}(i) : D^s_j = 1$; $\boldsymbol{T}$ and $\boldsymbol{B}$ are N-dimensional vectors, whose k-th entries are the measurement results T_k, and B_k respectively. Notice that the objective function is the relative entropy between the solution $\boldsymbol{x}$ and the prior distribution $\boldsymbol{w}$; further, the first and second constraints (each N-dimensional) correspond to Equations (2) and (3) respectively, while the third constraint (1-dimensional) corresponds to Equation (1).

D. Protocol-based State Space Reduction

The solution space of the $\vec{AS}$ inference problem is generated by 2^N variables, i.e., the Activity Share components that correspond to all possible network states; as the number of network nodes N increases, the exploration of such a large space to find the best candidate solution becomes computationally complex. In order to reduce space and complexity, we again leverage the protocol properties of 802.11 which permit to discover unlikely states.

As we observed, due to carrier sensing, the occurrence of $\vec{AS}$ that assign large probabilities to states including neighboring transmissions is unlikely. We take advantage of this consideration by excluding from the solution space the $\vec{AS}$ with $AS(\vec{D}_j) > 0$, for any $\vec{D}_j$ including neighboring transmitters. Practically, this is equivalent to reducing the number of Activity Share components, by eliminating those corresponding to the unlikely $\vec{D}_j$. In terms of graph theory, the set of transmitters in any allowed state is an independent set of the graph G. Thus, the number of network states, and of Activity Share components to be estimated, reduces to the cardinality of the set of the independent sets, which is generally still exponential (in graphs with bounded node degree [7]) but smaller than 2^N.

By using this simplification, the resulting inference problem can be obtained from (III-C), by equating γ to the cardinality of the set of the independent sets of the network and by replacing w_j with $\frac{1}{\gamma}$, $\forall j$. The latter substitution reduces the Minimum Relative Entropy objective to Maximum Entropy: the probability of all the states in the $\vec{AS}$ solution will be spread as evenly as possible according to the constraints.

In our experiments, we verified that the enhancement described above permits to double the network size that we can solve with similar time budget. While simplifying the computation, the illustrated state space reduction is only an approximation of the reality and may penalize the accuracy of the obtained solution. We investigate the performance of the state space reduction in Section V-D, while we adopt the full state space representation in the testbed results in Section V-C.

IV. Mitigation of Hindering Transmissions

In this section, we address our goal of improving the throughput of under-served links. Specifically, we show how MIDAS uses the Activity Share to predict how limiting the transmission rate of any hindering node will benefit the throughput of the problematic link. Our solution is comprised of two procedures: *i)* we address the main challenge of estimating the Activity Share after the management operation; *ii)* based on the new Activity Share, we estimate the potential throughput gain that any single link can obtain, in particular the target link. With regard to the first procedure, the key technique we devise follows a differential approach in which we consider that small deviations from the current network conditions have limited effect on the nodes other than the rate-limited and the under-served. The second procedure uses a simple model that identifies how the Activity Share affects the busy time and collision probability of the under-served link. In this section, we discuss each step separately.

A. Evolution of the Activity Share after Rate-limiting

In order to obtain the potential throughput gain of the under-served link by rate-limiting a specific node (Section IV-B), we first compute the Activity Share after rate-limiting. Our methodology follows a differential approach that assumes that small changes on the transmission rate of a node do not affect the relative durations of the states in which that node transmits. In particular, we assume that the Activity Share of the states in which the rate-limited node transmits will reduce in proportion to their values before rate-limiting. Note that based on the differential approach, the total time the nodes transmit, other than the under-served and rate-limited nodes, is not affected by the change. In practice, this can be realized, e.g., by having the transmission rates of neighboring links fixed to the value before the management operation. In the following, we illustrate the analytical aspects of the differential approach, while its accuracy is implicitly evaluated by the experimental results in Section V (see in particular, Figures 8 and 17-19).

Denote AS^o (Activity Share *old*) and AS^n (Activity Share *new*) the Activity Share before and after the rate-limiting action, respectively. Let us consider the case of rate-limiting the packet transmission rate (i.e., at the MAC layer) of a single conflicting node k of a quantity RL_k. We define $\{\vec{D}^{k0}_l\}$ the states in which k does not transmit (i.e., $D^k_l = 0$), and $\{\vec{D}^{k1}_l\}$ the states in which k does (i.e., $D^k_l = 1$), and we establish that the j-th states, i.e., $\vec{D}^{k0}_j$ and $\vec{D}^{k1}_j$, differ only for the k-th entry, i.e., $\vec{D}^{k0}_j = \{d_{j1} \dots d_{jk-1}\, 0\, d_{jk+1} \dots d_{jN}\}$ and $\vec{D}^{k1}_j = \{d_{j1} \dots d_{jk-1}\, 1\, d_{jk+1} \dots d_{jN}\}$. Using the differential approach, the Activity Share of the network states (in $\{\vec{D}^{k1}_l\}$) in which k transmits decreases proportionally to the duration of those states in AS^o, and the state $\vec{D}^{k0}_j$ benefits from the decrease of the state $\vec{D}^{k1}_j$, for all j. Formally,

$$AS^n(\vec{D}^{k1}_j) \approx AS^o(\vec{D}^{k1}_j) - \frac{AS^o(\vec{D}^{k1}_j)}{\sum\limits_{l:D^k_l=1} AS^o(\vec{D}_l)} \cdot h \cdot RL_k \quad (5)$$

$$AS^n(\vec{D}^{k0}_j) \approx AS^o(\vec{D}^{k0}_j) + \frac{AS^o(\vec{D}^{k1}_j)}{\sum\limits_{l:D^k_l=1} AS^o(\vec{D}_l)} \cdot h \cdot RL_k \quad (6)$$

where h is the duration of the packets sent by k, and RL_k is the rate-limiting amount of node k in terms of packets per second. For ease of exposition, we assume fixed duration of the data packets transmitted over all links. Next, we will use the AS^n to obtain the new collision probability of the under-served link.

B. Relationship between the Collision Probability of the Under-Served Link and the Activity Share

According to [9], we can express the maximal throughput of any link after the rate-limiting action by estimating its busy time and collision probability. The busy time of a link can be obtained from the new Activity Share using Equation (3). In this section, we show how to use the new Activity Share to determine the collision probability of any link, and in particular of the under-served. Given the Activity Share, the main challenge in computing the collision probability is in the transformation of the cumulative time the colliding nodes have transmitted simultaneously into the number of collided packets. For instance, let τ be the sum of the Activity Share of the states where colliding nodes a and b transmit simultaneously; since (assuming a fixed packet duration h) a packet can collide at most with two different packets, the total number of collided packets between these two nodes can be any integer in the range $[\frac{\tau}{h}, 2\min\{transmitted\ packets\ by\ a\ or\ b\}]$. In the following, we use a binary channel assumption; accordingly, a packet on (i,j) is corrupted if it overlaps for any arbitrary small duration of time with any other packet reception at j.

In order to compute the collision probability $p^{i,j}$ of a problematic link (i,j), we determine the success probability, i.e., the probability that the transmission of a packet from i to j entirely fits within a time interval during which its hidden terminals are not transmitting. To estimate this probability, we model the transmission attempts of i as the sampling of an ON/OFF process representing the aggregate transmissions of all the hidden terminals of i [9], [19]. The ON period is the interval during which at least an hidden terminal is transmitting, the OFF period is the gap in the activity of all the hidden terminals that node i has to discover randomly.

In the analysis of this process, we make the following assumptions. *1)* In general, the transmissions of the hidden terminals are not coordinated and may overlap; thus, the durations of the ON and OFF periods are variable. In this case, it is a common assumption to model them distributed exponentially. *2)* The duration of an ON period can range from very short, e.g., an individual ACK transmission, to much longer than the duration of a data packet h, in case of consecutive overlapping transmissions of different hidden terminals. We balance these cases, by approximating the average duration of an ON period, $\bar{T}_{ON}$, with h. *3)* Conditioned on the fact that i can transmit, i.e., that the nodes in $V_{cs}(i)$ are not transmitting, we assume that the transmissions of i occur at random points in time.

In order to succeed, a packet transmitted on (i,j) needs to start during an OFF period, and be entirely received during the OFF period. Thus, using assumptions 1) and 3), we can write the collision probability as: $p^{i,j} = 1 - \frac{\bar{T}_{OFF}}{\bar{T}_{ON}+\bar{T}_{OFF}} e^{-\frac{h}{\bar{T}_{OFF}}}$ [9]; assumption 2) permits to obtain $p^{i,j}$ as a function of $\frac{\bar{T}_{ON}}{\bar{T}_{ON}+\bar{T}_{OFF}}$. In the remainder, we show how to express $\frac{\bar{T}_{ON}}{\bar{T}_{ON}+\bar{T}_{OFF}}$ (and thus $p^{i,j}$) as a function of the Activity Share.

In order to do this, we compute the total duration the process is in ON and $\{ON\ or\ OFF\}$ states during a measurement interval ΔT: the ratio between these two quantities is equal to the ratio of their averages $\frac{\bar{T}_{ON}}{\bar{T}_{ON}+\bar{T}_{OFF}}$. Recall that the ON and OFF states model the sampling of node i of the channel at the receiver, and that node i cannot sample the ON/OFF process (i.e., transmit) during the transmissions of nodes in $V_{cs}(i)$. Hence, we prune all time intervals in which at least one of i's neighbors is transmitting, i.e., we consider only time intervals in which no node in $V_{cs}(i)$ is transmitting. Thus, the whole duration of the ON-OFF process in ΔT is $(1-B_i)\Delta T$. Let us denote $V_{ht}(i,j)$ the set of hidden terminals of (i,j). Then, the whole duration of the ON period in ΔT is the time at least one hidden terminal is transmitting and no node in $V_{cs}(i)$ is transmitting. By using the Activity Share, we denote the latter interval as $AS^{HT*}\Delta T$, where

$$AS^{HT*} = \sum_{l:(\exists m \in V_{ht}(i,j): D_l^m = 1) \wedge (D_l^n = 0, \forall n \in V_{cs}(i))} AS(D_l) \tag{7}$$

Finally, the identity between $\frac{\bar{T}_{ON}}{\bar{T}_{ON}+\bar{T}_{OFF}}$ and the ratio of their total durations in ΔT discussed above leads to

$$\frac{\bar{T}_{ON}}{\bar{T}_{ON}+\bar{T}_{OFF}} = \frac{AS^{HT*}\Delta T}{(1-B_i)\Delta T} \equiv AS^{normHT*} \tag{8}$$

By replacing (8) into $p^{i,j}$, we can write:

$$p^{i,j} = 1 - (1 - AS^{normHT*}) e^{-\frac{AS^{normHT*}}{1-AS^{normHT*}}} \tag{9}$$

which expresses the collision probability of a link using exclusively the Activity Share. Using Equation (9) we can compute the throughput according to [9].

V. Performance Evaluation

In this section, we validate MIDAS through an extensive set of testbed and simulation experiments. After introducing our experimental platform and implementation, we investigate the performance of MIDAS in a real testbed deployment. Finally, we extend the evaluation by simulating a broader set of topologies with larger numbers of nodes, in order to determine the sensitivity of the tool to node density and traffic load, and show its robustness to missing reports and real traffic distribution.

A. Experimental Testbed

WARP. To validate MIDAS, we used the Wireless Open-Access Research Platform (WARP) developed at Rice University [1]. The platform, built around a Xilinx Virtex processor, includes the MAX2829 radio chipset that provides RSSI readings. Moreover, WARP implements an OFDM layer similar to 802.11a. In our configuration, the boards operate at 6 Mbps using BPSK modulation, and are equipped with a 3 dBi antenna; all boards are controlled by a laptop via Ethernet connections.

Inference Tool Implementation. The implementation of the inference tool consists of two basic components. *i)* The *transmission duration counter* measures the time duration the radio is in transmission state by timing the functions that control the transmission operations. *ii)* The *sub-packet RSSI time sampler* measures the time duration the received signal strength, including noise and interference, exceeds a given threshold. In contrast to existing off-the-shelf drivers, such as MadWifi for Atheros chipsets,[4] which only provide an RSSI sample per packet, our implementation samples the RSSI values at regular time intervals shorter than the packet duration, and compares them to the carrier sense threshold.

Validation Tool. Two additional components were implemented only for validation purposes. *i)* The *fast RSSI sampler* behaves identically to the sub-packet RSSI time sampler described above, but supports higher sampling rates via a digital design, thus improving the precision of the busy time estimation. *ii)* The *trace collection logic* provides the ground truth of our experiments by collecting and storing on the board's memory the timestamps and durations of all radio-transmitted packets and sends batch traces to a control station. The individual node traces are **not** used by the inference tool, but permit to reconstruct offline a network-wide global trace of the transmitting activity of all nodes and to extrapolate the actual Activity Share. In order to synchronize the individual traces from different nodes, the control station issues an Ethernet broadcast to the boards at the beginning of each experiment, which is used to reset their clock. We verified that our technique achieves clock offsets below a few microseconds.

Testbed Setup. We conduct our experiments on a five-node indoor testbed. In order to verify the robustness of MIDAS to different node densities, we alternately deployed our nodes in different topological configurations. We list the locations used in our topologies in decreasing order of density, with reference to Figure 2: in the single-hop topology $S1$ all nodes are next to each other close to position b; in the multi-hop topology $M1$ the nodes are located in the positions $\{a, b, c, d, e1\}$; in the multi-hop $M2$ the nodes are in positions $\{a, b, c, d, e2\}$. Each board transmits 1000-byte data packets, with constant inter-packet time whose value depends on the experiment. Each experiment run lasts 10 seconds and, where not differently specified, the reported results are cumulative over 10 runs.

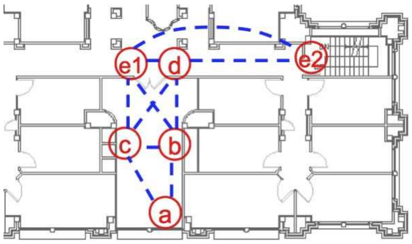


Fig. 2. Layout of our testbed deployment.

B. RSSI-based Busy Time Discovery

The challenge in the experiment setup is to devise a technique to consistently measure the node busy time, i.e., the total duration of *all and only* the transmissions of a limited set of neighboring nodes. In Section II-A, we argued that we cannot use packet reception statistics to measure node busy time with the needed accuracy because of packet losses. In this section, we show how to use the received signal strength to discover neighbors' transmissions. The advantage of our technique is that it does not rely on packet decoding, thus being resilient to losses.

We identify the neighbor set of a node as the set of nodes whose transmissions consistently exceed an RSSI threshold. Because wireless links are heterogeneous, the received signal strength varies from link to link; because of fading, it fluctuates in time. The first issue requires each node to autonomously calibrate its own threshold value, depending on the local topology, so that *all and only* neighbors' transmissions consistently exceed it. The choice of an RSSI threshold sufficiently lower (resp. higher) than the average signal strength of every neighbor node (resp. non neighbor node) permits to cope with the temporal variations.

In order to demonstrate our calibration technique, we design an experiment where we evaluate many possible RSSI thresholds for each node in the topology *M2*. In the experiment, each node in the network takes turns of 10 seconds transmitting broadcast packets at maximum rate, while the others measure the fraction of time the RSSI exceeds a threshold. For each transmitter, we perform the experiment 8 times, increasing the threshold of the potential receivers (i.e., all other nodes) from -90 dBm to -66.2 dBm in steps of 3.4 dBm; for each threshold value, we perform 10 iterations. In Figure 3, we show the results we obtained for node a; for the other nodes we obtained similar results. The X-axis represents the threshold value, while the Y-axis is the ratio between the total time the received RSSI exceeds the threshold and the total transmission time of the source node. The figure shows that, for certain values of the threshold, node a can discover almost all transmissions from a set of nodes and almost none from the others. For instance, for a threshold of -80 dBm node a can discover more than 99% of the transmissions of nodes b and c, and less than 1% of those of d and $e2$. This threshold defines a neighbor set of node a including b and c. The experiment shows that it is possible to identify suitable RSSI thresholds consistently exceeded by neighbor nodes, and rarely by non-neighbors. In our experiment, the calibration procedure just described was repeated once for each topology at the beginning of each session, and the thresholds were left unchanged during hour-long repetitions. To address unusual situations of severe fading where fixed thresholds cannot be identified, we are investigating a technique based on the evaluation of the threshold crossing probability of each neighbor.

C. Testbed Results

Experimental Methodology. We evaluate the accuracy of the inference tool, by assessing its predictions in different testbed and simulation settings. At the *end of each experiment* performed, we collect *a single* report from each node including its transmission time and busy time, which represent the parameters $\boldsymbol{T}$ and $\boldsymbol{B}$ in Problem (III-C). We compute the

[4] Multiband Atheros Driver for Wifi. Available at http://madwifi.org/

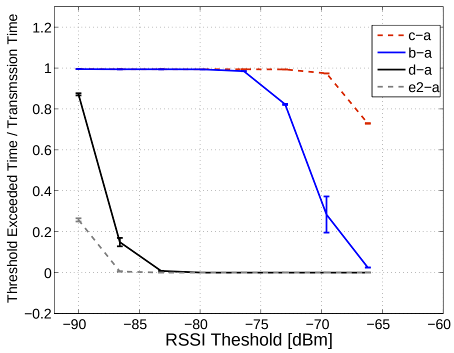


Fig. 3. Effect of the RSSI threshold on the busy time of node a

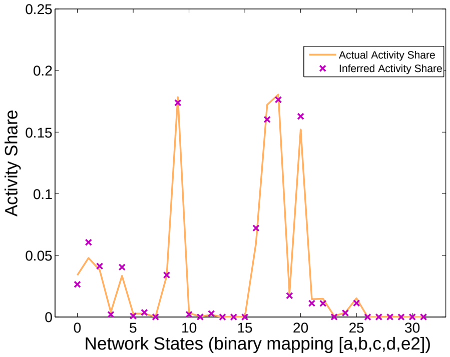


Fig. 4. Activity Share inference (testbed).

optimal solution of Problem (III-C) corresponding to the collected values using the Matlab solver *fmincon*. We establish the accuracy of the Activity Share inference by comparing our estimations with the ground truth provided by an *omniscent centralized approach* based on the collection of detailed traces (see the Validation Tool above).

An example of the results obtained from a single run on topology $M2$ is shown in Figure 4. In the Figure, we present the scatterplot of the predicted and actual (ground truth) Activity Share obtained in the single run. Each value k on the X-axis denotes a *network state* $\vec{D}$ corresponding to the binary representation of k (once mapped the bit indices 0 through 4 to the nodes positioned in a, b, c, d, and $e2$, respectively, e.g., $k = 20$ maps to the network state $\{10100\}$, i.e., where only nodes $e2$ and c transmit). The graph shows an excellent agreement between the inferred Activity Share and the actual Activity Share obtained from the traces. Further, we can observe that a number of states have very short durations: these typically include simultaneous transmissions of nodes in carrier sensing range, which occur less frequently than the others.

Sensitivity to Network Density. Network density can highly affect the accuracy of the Activity Share estimation. In order to infer the Activity Share, it is challenging to estimate the duration of overlapping transmissions of non-neighboring nodes, by combining their transmission reports with the busy timeshare reports of their common neighbors. On the one hand, the lower is the density, the more tightly the busy timeshare reports constrain the overlapping transmissions of non-neighboring nodes, but also the fewer reports reflect such events. For example, if a node z has only two transmitting neighbors x and y (where x and y are not neighbors), z's busy timeshare report permits to exactly recover the share of time that transmissions of x and y overlapped, as $T_x + T_y - B_z$. However, if a node has three transmitting neighbors (which are non-neighbors to one another), based on the busy timeshare of the node it is not possible to determine the amount of overlapping transmissions of any two or all three of them. In fact, the higher the density, the more combinations of overlapping transmissions of a node's neighbors are consistent with the node's busy timeshare report. On the other hand, the higher is the density, the larger is the number of nodes that observe the transmitting activity of a given set of transmitters. Accordingly, more constraints can be imposed on the Activity Share estimation based on the diversity of the reports of different neighbors. In order to investigate the influence of network density on the Activity Share accuracy, we run experiments on all three different topologies of our testbed: topology $S1$ which is densest, as all nodes are connected to one another, topology $M1$ which is less dense, and topology $M2$ which is the sparsest.

Figure 5 shows the CDF of the normalized relative error of the Activity Share estimation, where the relative error committed in a state is weighted by the Activity Share of that state, i.e., proportionally to the duration. The X-axis indicates the normalized relative error committed, while the Y-axis is in (non-dimensional) time ratio units. For instance, the point in $(0.1, 0.7)$ indicates that the network spends 70% of the time in states where our inference tool commits an error of 10% or less. All plots show that our inference technique is remarkably accurate under all density conditions; further, $S1$ is the most accurate solution, while the $M1$ plot mostly dominates $M2$. The respective average normalized relative errors, i.e., the relative error committed in a randomly sampled instant, are 4.6% for $S1$, 9.9% for $M1$, and 11.5% for $M2$. These results are obtained for broadcast packets; however, similar values have been obtained using one-hop unicast flows, i.e., 4.8% for $S1$, 6.1% for $M1$, and 7.7% for $M2$. *We conclude that the Activity Share inference tool is accurate under all density conditions: in low density, a small number of reports reflects overlapping transmission events, but they impose tight constraints; in high density, the large report diversity compensates for the looser constraints imposed by the reports.*

The influence of network density on the Activity Share is revisited by simulating larger topologies and the results can be found in Section V-D.

Sensitivity to Traffic Load. Similarly to network density, traffic load can also affect the accuracy of the Activity Share estimation, since it influences the overlapping transmissions sensed by a node. The higher the traffic load, the larger is the amount of overlapping transmissions, which challenge the estimation of the Activity Share by enlarging the feasible state space. In fact, as noted earlier, in case of overlapping transmissions, several combinations of Activity Shares may generate identical observations (i.e., node busy and transmis-

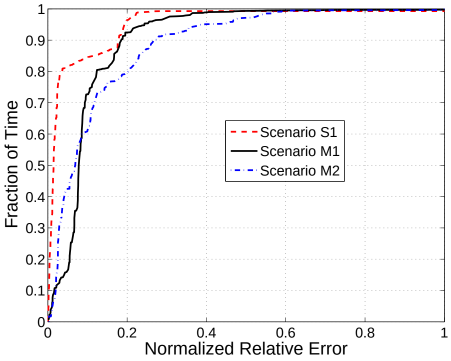


Fig. 5. Inference sensitivity to network density (testbed).

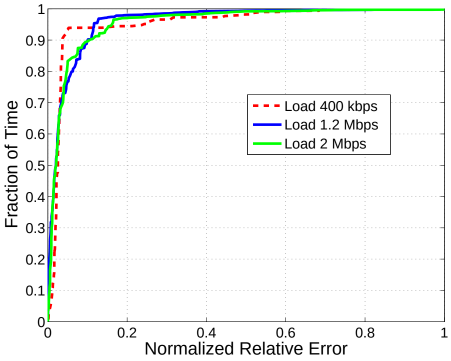


Fig. 6. Inference sensitivity to traffic load (testbed).

sion timeshares). However, light traffic conditions increase the free airtime observed by a node, which in turn weakens the coordination attained by carrier sensing, by decoupling the transmitting patterns of the nodes and leaving larger room to randomness. We study the impact of traffic load on the Activity Share inference tool, by running the experiment on a fixed topology with various traffic loads. Specifically, we iterate scenario $M1$ three times, fixing the unicast traffic loads of all nodes to 400 kbps, 1.2 Mbps, and 2 Mbps (also in this case, each experiment is repeated 10 times).

Figure 6 depicts the CDF of the normalized relative error of the Activity Share estimation. As can be seen in the figure, the Activity Share inference tool attains a very low normalized relative error. Furthermore, the variations among the three plots are minimal, and are comparable with the results attained for the fully backlogged case. In particular, the average normalized relative error is 4.6%, 4.0%, 4.5% for 400 kbps, 1.2 Mbps, and 2 Mbps, respectively. *We conclude that, even though heavier traffic challenges the Activity Share inference by increasing the amount of overlapping transmissions, while lighter traffic increases randomness, the accuracy of our solution is largely independent of the traffic load of the nodes.*

We defer the investigation of the sensitivity of the inference tool to large topologies and non-uniform traffic patterns, to the simulation section.

Sensitivity to Report Interval Length. In the previous experiments, we used report intervals of 10 seconds, i.e., each node k sent one report every 10 seconds including the busy and transmission timeshares B_k and T_k that k measured during the same interval. The report interval introduces tradeoffs of reporting overhead (favoring long intervals), responsiveness to network changes (favoring short intervals), and obtaining statistically significant data (favoring long intervals). We assess how short report intervals affect the performance of the inference tool, by measuring the accuracy in the scenario $M1$, for various report interval lengths, varying from 20 s to as low as 100 ms, for fully backlogged unicast traffic.

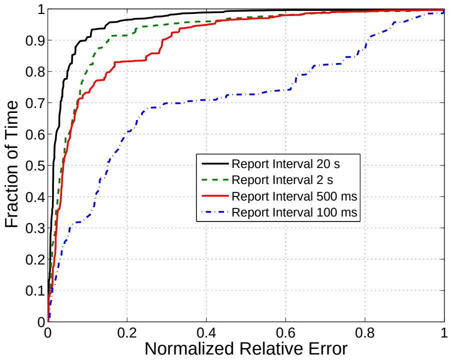


Fig. 7. Inference sensitivity to short report intervals (testbed).

The experiments show that the inference tool is accurate also for short report intervals (Figure 7). In particular, as the report interval decreases from 20 s to 500 ms, the accuracy decrease is minimal. When the report interval is further reduced, and is set up to (as small as) 100 ms, i.e., the reported values are based on approximately 20 packets sent by each node, the accuracy declines. The average normalized errors are 4.1%, 7.6%, 10.2%, and 29% for the cases of 20 s, 2 s, 500 ms, and 100 ms, respectively. *We conclude that, in order to better capture the network dynamics, the network manager can adapt the duration of the report intervals, with a small penalty on inference accuracy.* Note that since each report includes only two entries, the overhead is minimal. For example, in our implementation, the reports R_k include only two floating point values for a total of 16 bytes, i.e., they easily fit within a single packet, and can be aggregated or even piggybacked in regular traffic.

Throughput Prediction Accuracy With Heterogeneous Concurrent Load. We evaluate the accuracy of the model in Section IV, by comparing its predictions with testbed experiments in the topology $M1$ with single-hop flows $\{a \to c; b \to a; c \to a; d \to b; e1 \to c\}$. For each set of experiments, we consider a target under-served link whose traffic is fully backlogged, and we perform a baseline run, measuring the throughput of the target link when all others transmit at rate randomly chosen in the [400 kbps, 900 kbps] interval. At the end of the baseline run, we collect the node reports, infer the Activity Share, and predict the throughput increase of the target link obtained by rate-limiting any of the four conflicting nodes of a fixed quantity (400 kbps). Then, we perform four additional runs on the testbed (one per conflicting node), alternately rate-limiting a different conflicting node for the same

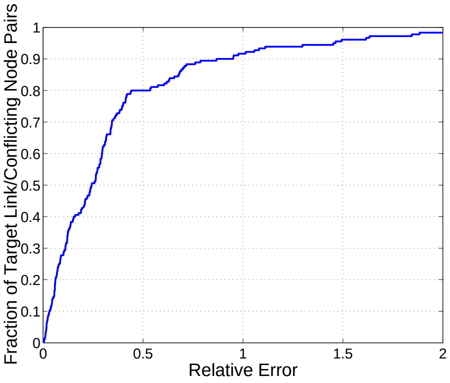


Fig. 8. Throughput increase estimation for concurrent nodes with loads in [400 kbps, 900 kbps] (testbed).

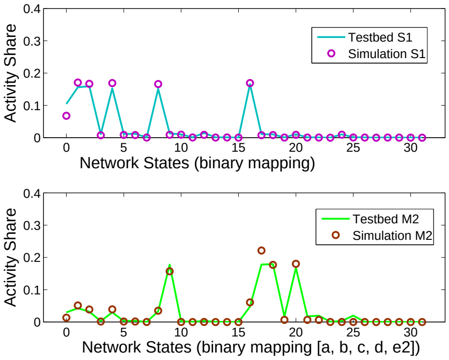


Fig. 9. Activity Share: simulation vs testbed.

400 kbps quantity, and we record the actual throughput gain of the target link. Finally, we compare the actual throughput gain obtained in the testbed with the throughput gain predicted by our model.

Figure 8 shows the CDF of the relative error for all possible target link/conflicting node pairs for 10 repetitions of our scenario (200 predictions in total). The long tail of the distribution is due to few combinations for which the actual gain is very small (on the order of a few kbps); in those cases, even an error of few packets is decisive in relative terms. In terms of the absolute error, the predicted throughput gain is on average less than 72 kbps different from the actual throughput gain (i.e., 18% of the rate-limiting value of 400 kbps, or around 26% of the average actual throughput gain of approximately 280 kbps). *We conclude that, despite rate-limiting different conflicting nodes can have largely different impacts on the throughput of an underserved link, our prediction tool adequately captures the heterogeneous effects.*

D. Simulation Results - Inference Tool

In order to evaluate the inference tool on various topologies including a larger number of nodes, we performed an extensive set of ns-2 simulations following the inference experimental methodology adopted in the previous section. In this section, we first compare testbed and simulation. Then, we evaluate the accuracy loss due to the state space reduction discussed in Section III-D; all the results in the remainder of the paper implement such enhancement. We also investigate the sensitivity of the inference technique to network density, unsaturated and real traffic conditions, and its robustness to report losses, and short report interval lengths.

Simulation Settings. We consider scenarios where each node generates 1000-byte UDP packets directed toward a single neighbor, with constant inter-packet time. The traffic is generated for 100 s at a fixed rate. We use the FreeSpace propagation model, with node transmission and interference ranges equal to 210 m. We generate scenarios with a certain network density (i.e., where each node has on average a predetermined number of neighbors), by deploying the nodes in random positions, and scaling the size of the deployment area. Except for the experiment in Figure 9, which is obtained using 802.11a at 6 Mbps, all results in this section are obtained using 802.11b at 11 Mbps data rate in order to experiment with different conditions. We refer to the analogous simulation results obtained for 802.11a at 6 Mbps as needed.

Comparison between Testbed and Simulations. The simulations introduce simplifications about actual channel propagation and abstract operational details, such as the WARP board's packet processing time. For this reasons, our first experiment compares the simulations and testbed results. We consider the topologies $S1$ and $M2$ used in the testbed section and fully backlogged nodes. Using the *omniscent centralized approach*, we extract the Activity Share from the traces of simulation and testbed, and we compare them. Figure 9 shows the actual Activity Share (Y-axis) for all 32 possible states (X-axis) sorted similarly to Figure 4.[5] *The plots show an excellent agreement between the two environments; the small discrepancies are due to non-ideal packet processing times and carrier sensing relationships in the testbed.*

Effect of the Protocol-based State Space Reduction. The next experiment evaluates the effect of the protocol-based reduction discussed in Section III-D. We generate a random topology of 10 nodes, with an average number of 7 neighbors per node, and we compare the Activity Share obtained using the reduced (labeled "Protocol-based Reduction") and the entire 2^N state spaces (labeled "Power Set").

Figure 10 shows the scatterplot of the Activity Share. The X-axis is the actual value of the Activity Share, while the Y-axis is the estimated value; each mark represents a single state. As expected, the solution including the power set is more accurate (crosses are closer to the line than circles). The concentration of circles on the X-axis close to the origin are due to the states including adjacent nodes transmitting, that the protocol-based reduction excludes. Note that the actual Activity Share values of those states are not significantly larger than 0, as the simultaneous transmissions of neighboring nodes are relatively unlikely. The power set solution benefits from accounting for the unlikely states, not only in the prediction of the Activity Share of those states, but also of states including only independent sets of transmitters. *We conclude that the accuracy of the inference tool increases by re-introducing the*

[5] Note that for scenario S1 the actual node mapping is immaterial.

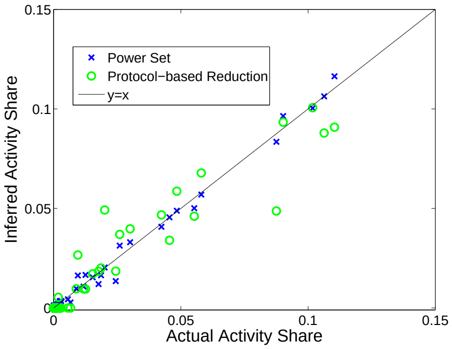


Fig. 10. Inference with protocol-based reduction.

states excluded by the protocol-based reduction, since those states contribute to the reported measurements.

Sensitivity to Network Density. This subsection revisits the issue of network density on large topologies. In contrast to Section V-C, we run our scheme on the reduced state space. We evaluate the normalized relative error between inferred and actual Activity Share, for 30 topologies of 10 and 15 nodes, with average node neighbors from 3 to 13 and fully backlogged traffic. Recall that the normalized relative error is the relative error committed in a state weighted by the Activity Share (i.e., proportionally to the duration) of the state. Figures 11 and 12 show that our inference tool is accurate under different densities, e.g., the network spends more than 70% of time in states whose relative error is below 20% for all tested densities in 10-node topologies. Figure 11 shows that for 10 nodes a density increase from 3 to 5 improves the accuracy of the inference tool, while for density 7 the performance decreases. The average normalized relative errors are 12.2%, 10.2%, 17% for densities 3, 5, and 7 respectively. We ran this experiment also using 802.11a at 6 Mbps, and we obtained normalized relative errors of 13.7%, 12.5%, 15.2%, respectively. Figure 12 shows a similar trend for 15-node networks; the accuracy grows for density increase from 3 to 7, but it reduces for density 13. The average normalized relative errors are 18%, 14%, 26%, for densities 3, 7, and 13 respectively. Both figures clearly depict the existence of an accuracy tradeoff related to the network density. As explained in Section V-C, the denser the network the more constraints can be imposed on the Activity Share estimation. However, as the network approaches a clique, the probability of simultaneous transmissions of neighboring nodes increases, thus generating network states that are excluded by the protocol-based state space reduction. For example, the accuracy degrades for 10-node networks with density 7, and for 15-node with density 13. In contrast, in 15-node networks with density 7, nodes in close proximity likely observe different channel busy intervals, due to the diverse sets of carrier sensed nodes; thus, their simultaneous transmissions are less frequent. Notice that, as shown later in more detail, traffic intensity has a similar effect on the validity of the protocol-based approximation, and that fully backlogged traffic is a worst case due to higher occurrence of adjacent node transmissions. *We conclude that, although the protocol-based reduction is accurate in all evaluated settings, high network densities challenge the validity of its approximation.*

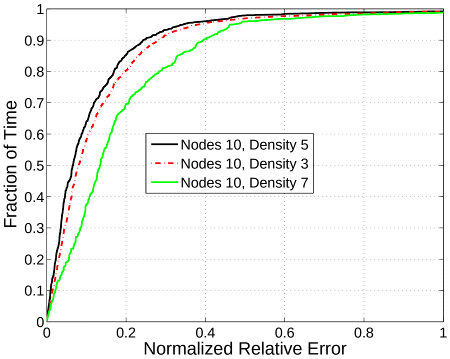


Fig. 11. Inference sensitivity to density (10 Nodes).

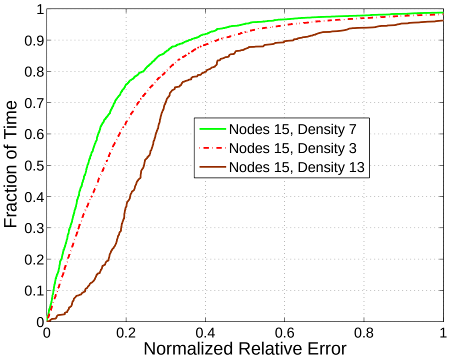


Fig. 12. Inference sensitivity to density (15 Nodes).

Sensitivity to Traffic Load. Not only the network density affects the occurrence of states excluded by the protocol-based approximation, but also the traffic intensity. In fact, as the traffic decreases, the probability that carrier sensing nodes transmit simultaneously decreases. Figure 13 shows the bar graph of the average normalized relative error between inferred and actual Activity Share, for 10 nodes, density from 3 to 7, and target transmission rate from 300 kbps to fully backlogged for each node (in a single scenario, all nodes are subject to identical target transmission rates). The graph shows that the error increases with the target transmission rate for each density value. Furthermore, the error increases with the network density, because of the reasons explained above. In particular, as the density increases the network becomes fully backlogged for lower transmission rates (the effect is evident in the case of 900 kbps for density 7). *We conclude that as the network traffic decreases the performance of our methodology improves, because of the increased accuracy of the protocol-based approximation.*

Robustness to Real Traffic Distribution. In our previous experiments, all traffic sources generate packets according to predefined inter-packet distributions. In this experiment, we investigate how critical this assumption may be for our inference technique to operate in real traffic scenarios. In order to reproduce real traffic, we replay actual traffic traces collected within UCSD Jigsaw project [5] in our simulation

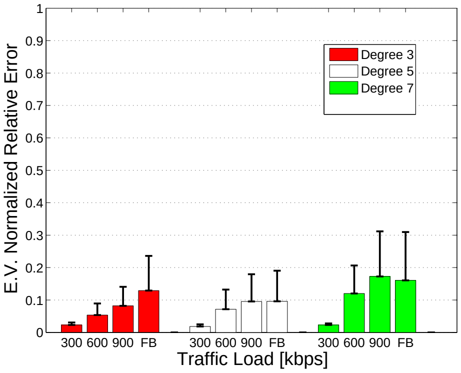


Fig. 13. Sensitivity of the Activity Share estimation to traffic load (10 Nodes).

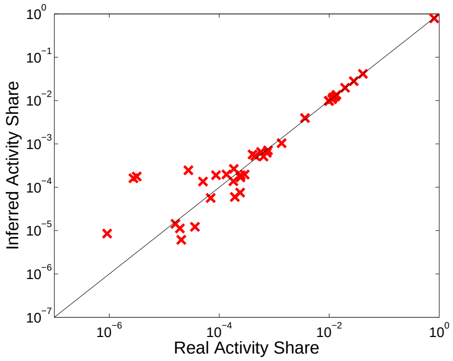


Fig. 14. Inference robustness to realistic traffic.

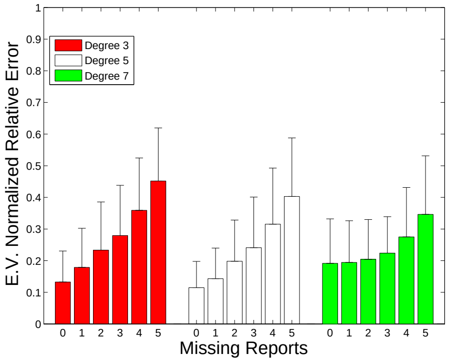


Fig. 15. Inference robustness to missing reports.

environment. In particular, we randomly select 10 nodes from the UCSD traces, and we play 10 seconds of their traffic on a network topology obtained as follows. Two nodes are considered disconnected if the considered traces include five overlapping transmissions of data packets from the two nodes (in order to safely consider synchronization errors and simultaneous neighbor transmission events, we consider the overlapping valid only if its duration exceeds 100 μs); otherwise, the two nodes are considered disconnected. Each simulated node generates packets according to the transmission times of the UCSD node it represents in the trace. Figure 14 shows a scatterplot of the measured Activity Share vs. the inferred Activity Share for 10 repetitions of the experiment, with traffic from 10 different time intervals. Note that, in order to visually capture a large range of Activity Share values, we plot both axes in logarithmic scale; for this reason, for small values of the Activity Share the error is visually magnified, although it is only a few percent. The plot shows that also in these real traffic conditions, our inference technique achieves very accurate results. Quantitatively, the average normalized relative error is about 3%. *We conclude that our inference technique is robust to real traffic conditions, i.e., it is accurate also in case the traffic is not generated according to a predefined distribution.*

Robustness to Incomplete Information. In the case of severe network congestion, some of the reports could be lost. We evaluate how report losses affect the accuracy of the inference tool, by simulating the loss of up to five out of the ten reports transmitted in 10-node networks, with densities of 3, 5 and 7 (i.e., where each node has on average 3, 5, and 7 neighbors, respectively). Figure 15 shows the average normalized relative error of the Activity Share computed out of all possible states obtained from 30 random topologies, where we evaluate the lack of all possible combinations of missing reports (bars indicate 85-th percentiles). We observe that the performance gracefully degrades as the number of missing reports increases. This is because the reports of neighboring nodes are related: for instance, part of the busy time of neighboring nodes is generated by transmissions of common neighbors. *We conclude that our inference technique is robust to report losses, due to inherent redundancy of node reports.*

Report Interval Length. In this section, we extend the result obtained in the testbed experiments to larger topologies. Furthermore, differently from the testbed experiment, we use the state space reduction which, as shown above, may affect the accuracy of our solution. Specifically, in the previous simulations, we used report intervals of 100 s, i.e., each node k sent one report every 100 s including the busy and transmission timeshares B_k and T_k that k measured during the same interval. This result assesses the accuracy of our inference technique in a 10-node network, with density 5, for report interval lengths as low as 50 ms. Figure 16 shows that the inference tool is accurate also for short report intervals. In particular, as the report interval is decreased from 100 s to 2 s, the accuracy decrease is minimal. When the report interval is small and set to 50 ms, i.e., the reported values are based on approximately 10 packets sent by each node, the accuracy decreases. The average normalized errors are 10%, 11%, 14%, and 25% for the cases of 100 s, 2 s, 500 ms, and 50 ms, respectively. *Our simulations confirm the conclusions we obtained in our testbed results; the slight performance decrease is mainly due to the state space reduction and to the more complex interactions of larger topologies.*

E. Simulation Results - Throughput Prediction Tool

We investigate the performance of the prediction tool with ns-2 simulations with the same experimental methodology used to evaluate the throughput prediction accuracy in Section V-C. We start by running MIDAS on a ten node random topology with density 3. Node transmission rates are set to

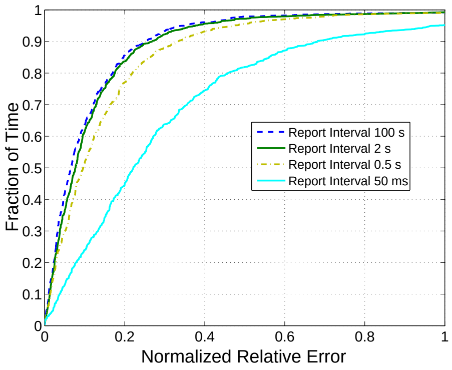


Fig. 16. Inference robustness to short report intervals.

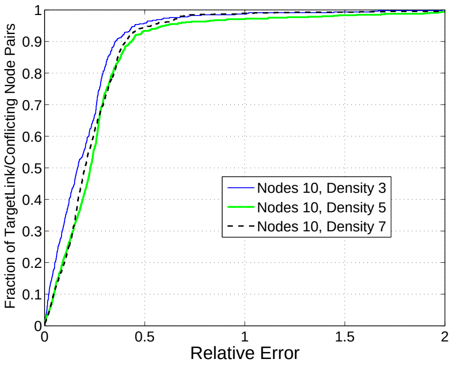


Fig. 18. Throughput prediction sensitivity to density (600 kbps).

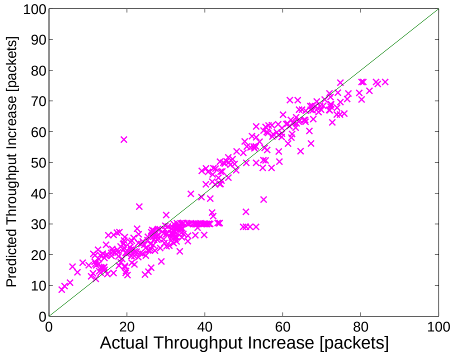


Fig. 17. Throughput increase estimation for scenarios with density 3.

600 kbps. As in the experimental case, we pick one target underserved link, we increase its load until it is fully backlogged, and successively repeat the experiment, rate limiting each time a different conflicting flow by 400 kbps; we iterate this procedure for all links in the network. In Figure 17, we compare the scatterplot of the predicted throughput gain with the actual throughput increase collected for 10 different random topologies. The X-axis index identifies the actual throughput increase for a saturated link by rate-limiting one of its conflicting nodes, while the Y-axis represents the predicted value for the same rate-limiting action, e.g., a point on the diagonal represents a perfect match between the actual and the predicted throughput gain of the tagged link; points above and below the diagonal represent an overestimate and an underestimate of the predicted over the actual throughput gain of the tagged link, respectively. The graph shows an excellent agreement between the prediction and the simulation. It is a notable finding that by rate-limiting different conflicting nodes of the same fixed amount, the throughput increase of the target link can range from 7% to 172% of the rate-limited quantity 400 kbps, i.e., from 28 kbps to 688 kbps. In the remainder of this section, we evaluate how the prediction accuracy depends on network density and traffic load.

Sensitivity to Network Density. As previously shown, network density influences the accuracy of the Activity Share inference, which is the basis of throughput prediction (see Section V-C). In addition, network density determines the number of neighbors and neighbor's neighbors (potentially hidden terminals) that respectively affect the link busy time and collision probability, which in turn are key to our prediction tool. We investigate these effects by evaluating our predictions for all possible target link/conflicting node pairs in 10 topologies with 10 nodes, and densities of 3, 5, and 7, with node transmission rates of 600 kbps. Figure 18 shows the empirical CDF of the relative error between the predicted and actual throughput increase. The plot for density 3 (i.e., for topologies with 3 neighbors per node) is the most accurate, while the case for density 5 is the least; the average relative errors are 17%, 26%, 22% for densities 3, 5, and 7, respectively. Surprisingly, the accuracy in throughput prediction does not exactly reflect the accuracy in the inference of the Activity Share (we checked that the trends in Figure 11 were respected also in this set of scenarios). The main reason is that our model is more accurate in the computation of the fraction of busy time than of the collision probability, since the former imposes less stringent assumptions (see Section IV-B). Thus, the case of density 3, where the number of hidden terminals is restricted by the degree of the receiver, is most accurate. In terms of the absolute error, i.e., the difference between the actual throughput gain and the predicted gain, the predicted throughput gain is within 80 kbps (i.e., 20% of the rate-limiting value of 400 kbps) from the actual throughput gain in 83% to 92% of the cases. *We conclude that the accuracy of the prediction model increases as the number of hidden terminals decreases, because of the less stringent assumptions we impose on the computation of the fraction of busy time of the under-served link.*

Sensitivity to Traffic Load. In this experiment, we investigate the effect of traffic load on the accuracy of our predictions, by repeating the simulations above for node transmission rates of 900 kbps. Figure 19 shows the same ranking among the curves relative to different densities as for the case of 600 kbps. However, the accuracy obtained for 600 kbps is higher than for 900 kbps. This is due to two reasons: first, the Activity Share inference technique based on the protocol state-space reduction is more accurate for lower traffic loads (see Figure 13); second, in terms of the relative error the prediction of small throughput gains is more challenging than the prediction of large gains. As the neighbor load increases, rate-limiting actions produce on average a lower benefit for the under-served link, thus increasing the influence of the less

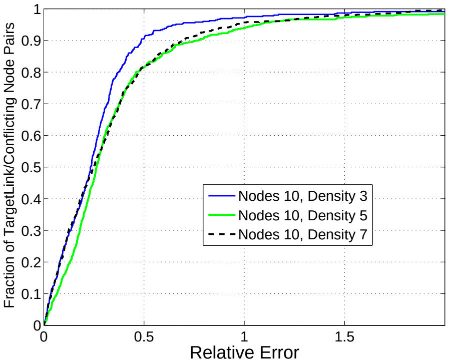


Fig. 19. Throughput prediction sensitivity to density (900 kbps).

accurate results for lower gains on the CDF. For example, for density 5 and 600 kbps, on average the under-served link gains 0.6 of the rate-limiting amount (i.e., 240 kbps out of 400 kbps in this experiment), while for the case of density 5 and 900 kbps the under-served link gains 0.4 (i.e., 160 kbps out of 400 kbps). This explains why the relative error is larger for 900 kbps than for 600 kbps.

F. Simulation Results - Real Network Topology

In this experiment we consider the topology of a real mesh network, TFA [3], deployed in south-east Houston. The topology consists of 18 backhaul nodes whose actual connections, as reported by WIANA [21], are depicted in Figure 20; the coverage area is $3km^2$, and the average node degree is 6.58. All nodes are equipped with 802.11b cards; the node denoted as GW, the gateway, is also wiredly connected to Internet via a 100 Mbps fiber. We considered a scenario of upload activity of the nodes, and the destination of each link is chosen according to the shortest path toward the gateway.

Inference Tool. Figure 21 compares the values of the Activity Share entries inferred by our solution with the actual values. All network nodes send fully backlogged traffic at 11 Mbps (as shown above, this represents a worst case for the inference tool). The X-axis index identifies the network states, while the Y-axis represents the Activity Share, i.e., the ratio of time the network spent in a state. The X-axis is sorted according to increasing values of the actual Activity Share of the states and, for the sake of readability, represent only the 237 states with highest Activity Share. The graph shows again an excellent agreement among inference and simulation.

Throughput Prediction. Similarly to the case in Figures 18-19, we evaluate the accuracy of our methodology by predicting the throughput increase on any target link, achievable by limiting the transmission rate of any conflicting node; we consider all possible target link / conflicting node pairs. Figure 22 shows the empirical CDF of the absolute error between the throughput increase predicted by the methodology and the actual throughput increase. The X-axis represents the absolute error in terms of packets per second, while the Y-axis represents the fraction of predictions. In this experiment, the target transmission rate is 600 kbps for all links excluding the target link, which is fully backlogged; the rate limitation is 50 packets per second (where the size of each packet is 1000 bytes). This permits to obtain appreciable throughput increases for several of the examined target link / conflicting node pairs. The figure shows that in 81% of the predictions the absolute error is less than 10 packets per second.

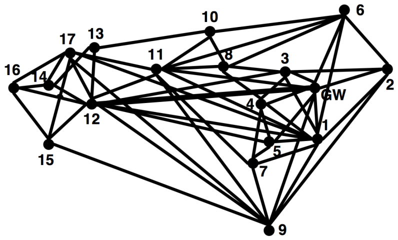


Fig. 20. Topology of the TFA network deployed in Houston, Texas.

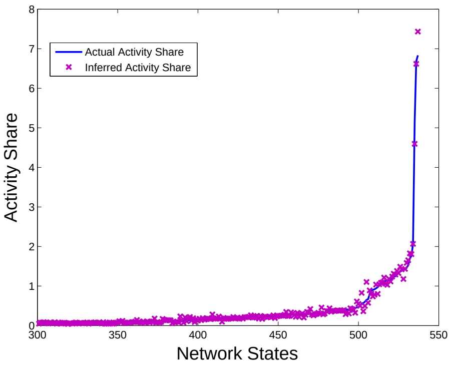


Fig. 21. Activity Share Inference for the TFA network (fully backlogged nodes).

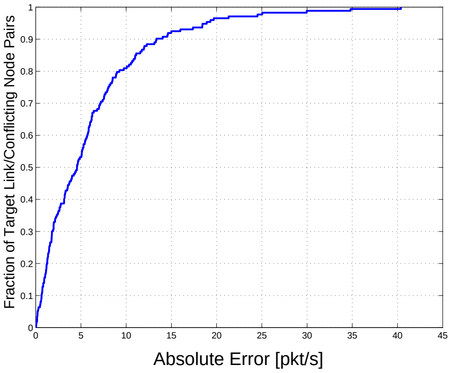


Fig. 22. Throughput Increase Prediction for the TFA network (transmission rate 600 kbps).

VI. Related Work

Wireless Network Monitoring. Performance monitoring of single-hop WLANs has recently attracted research interest [5], [16]. The proposed approaches reconstruct a global trace of all network packet transmissions by combining offline detailed traces reported by sniffers spread throughout the network. These solutions can provide a comprehensive survey of the network activity. However, they require the delivery of *detailed traces* from all (or at least most of) the nodes, which severely hinders the normal operations of multi-hop wireless networks. In our work, we show that we can attain very accurate results with the use of small time-averaged reports. Furthermore, [5], [16] do not address the problem of identifying the origins of poor link performance and rate-limiting the most hindering nodes.

802.11 Throughput Models. Several 802.11 throughput prediction models have been proposed in the literature [2], [4], [9], [11], [12], [17], [19], [20]. Their goal is either to compute the throughput of the network links given their traffic demands, or to compute the feasible region of the network. In contrast, we use measurements to infer the network behavior, particularly the coordination between node transmissions and the causes of poorly performing links, and use this understanding to improve the throughput of under-served links. Our scheme relies on active offline link profiling, such as [12], [19], to identify the carrier sensing and interference relationships between the nodes. In addition, we introduce passive online measurements during normal network operations, to capture the complex node interactions determined by the actual transmission patterns. Recently, [13] proposes a method to replace active offline profiling with passive online estimations using traces collected by deployed sniffers. While [13] does not characterize the coordination between conflicting nodes, nor predicts the effects of rate-limiting actions, we can leverage the result therein for passive link profiling.

VII. Conclusions

In this paper, we present a management framework for wireless networks called MIDAS. MIDAS addresses the problem of identifying the conflicting nodes that cause under-performance of a target link. We introduce the key concept of Activity Share that captures the coordination among the conflicting nodes. Since the Activity Share cannot be locally measured by the nodes, we show how MIDAS infers it using time-aggregate, passively collected measurements reported by the nodes. Finally, we design a throughput model based on the Activity Share that MIDAS utilizes to predict the benefit of rate-limiting conflicting transmissions. Our results show that MIDAS infers the Activity Share with an average normalized relative error as low as 4%, and predicts the throughput gain of an under-served link corresponding to alternative rate-limiting actions with an error lower than 20% of the rate-limited quantity.

References

[1] Rice University WARP project. Available at: http://warp.rice.edu.

[2] R. R. Boorstyn, A. Kershenbaum, B. Maglaris, and V. Sahin. Throughput analysis in multihop CSMA packet networks. *IEEE Transactions on Communications*, 35(3):267–274, Mar. 1987.

[3] J. Camp, V. Mancuso, O. Gurewitz, and E. Knightly. A measurement study of multiplicative overhead effects in wireless networks. In *IEEE INFOCOM*, 2008.

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

[5] Y.-C. Cheng, J. Bellardo, P. Benkö, A. C. Snoeren, G. M. Voelker, and S. Savage. Jigsaw: Solving the puzzle of enterprise 802.11 analysis. In *ACM SIGCOMM*, 2006.

[6] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. John Wiley, 1991.

[7] R. Diestel. *Graph Theory (Graduate Texts in Mathematics)*. Springer, 2005.

[8] R. Gallager. *Discrete Stochastic Processes*. Kluwer, 1990.

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

[11] A. Jindal and K. Psounis. Characterizing the achievable rate region of wireless multi-hop networks with 802.11 scheduling. *ACM/IEEE Transactions on Networking*, 18(3):257–281, Mar. 2009.

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

[13] A. Kashyap, U. Paul, and S. R. Das. Deconstructing interference relations in WiFi networks. In *IEEE Secon*, 2010.

[14] E. Magistretti, O. Gurewitz, and E. W. Knightly. Inferring and mitigating hindering transmissions in managed 802.11 wireless networks. In *ACM MobiCom*, 2010.

[15] E. Magistretti, O. Gurewitz, and E. W. Knightly. Inferring and mitigating hindering transmissions in managed 802.11 wireless networks. In *Rice University Technical Report*, 2010. TREE3610. Available at: http://www.ece.rice.edu/∼knightly/Infer.pdf.

[16] R. Mahajan, M. Rodrig, D. Wetherall, and J. Zahorjan. Analyzing the MAC-level behavior of wireless networks in the wild. In *ACM SIGCOMM*, 2006.

[17] K. Medepalli and F. Tobagi. Towards performance modeling of IEEE 802.11 based wireless networks: A unified framework and its applications. In *IEEE INFOCOM*, 2006.

[18] D. Niculescu. Interference map for 802.11 networks. In *ACM IMC*, 2007.

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

[20] W. Wang, and K. Kar. Throughput Modelling and Fairness Issues in CSMA/CA Based Ad-Hoc Networks. In *IEEE Infocom*, 2005.

[21] Wireless and Internet and Assigned and Numbers and Authority. http://www.wiana.org