Measurement-Driven Modeling of Transmission Coordination for 802.11 Online Throughput Prediction

Eugenio Magistretti, *Student Member, IEEE*, Omer Gurewitz, *Member, IEEE*, and Edward W. Knightly, *Fellow, IEEE*

***Abstract*—In 802.11 managed wireless networks, the manager can address underserved 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 Management, Inference, and Diagnostics using Activity Share (MIDAS). 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 that, 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 underserved 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 a mean relative error as low as 4% in testbed experiments.**

***Index Terms*—802.11, coordination, inference, interference, WLANs.**

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 underserved link. Moreover, it is not immediately evident how much throttling any node will increase the throughput of the targeted underserved 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 that cause a target link to be underserved or to obtain poor performance. Moreover, MIDAS identifies effective management actions to increase the performance of the underserved 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: 1) *measurement collection*, which gathers reports from each node consisting of a small set of passive time-aggregate measurements; 2) *inference*, which infers the coordination among the transmissions of different sets of nodes using the reported measurements; 3) *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 that 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 how much conflicting nodes (i.e., neighbors of sender or receiver of a link) are destructive to the link requires understanding their coordination. In order to capture node coordination, we define network state as a set

Manuscript received January 25, 2011; revised October 26, 2011; accepted January 09, 2012; approved by IEEE/ACM TRANSACTIONS ON NETWORKING Editor C.-N. Chuah. This work was supported by the NSF under Grants CNS-0751173 and CNS-0721894 and by Cisco Systems under a grant. O. Gurewitz was supported in part by the Israeli MOITAL CORNET consortium. An earlier version of this paper appeared in the Proceedings of the ACM International Conference on Mobile Computing and Networking (MobiCom), Chicago, IL, September 20–24, 2010.

E. Magistretti and E. W. Knightly are with the Department of Electrical and Computer Engineering, Rice University, Houston, TX 77005 USA (e-mail: emagistretti@rice.edu; knightly@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).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNET.2012.2192482

[1]Network managers have a number of options for mitigation, including moving sets of access points (APs) or clients to alternate frequencies. Management, Inference, and Diagnostics using Activity Share (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.

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 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 a mean 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 this 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. MIDAS Framework

A link can be considered underserved 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: 1) 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; 2) link transmission rates: nodes that transmit few packets are less likely to interfere with a target link; and 3) link transmission coordination: the number of packets transmitted on a link and corrupted by a hidden terminal depends on how frequently the link sender and hidden terminal transmit simultaneously. Note that transmission rates and coordination strongly depend on the traffic load of each node.

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

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

measure the interference between nodes, it reflects node interactions. Thus, the Activity Share is affected by node topological 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 [11]) miss the important dynamic information about the coordination of the transmissions of multiple nodes.

A. Activity Share: Fundamental Element of Network Observation

As previously explained, our management framework aims to identify the originating causes of underserved 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: 1) reducing the time the medium is perceived as free by the sender, thereby forcing the sender to defer; 2) corrupting the packet reception at the receiver end, i.e., colliding. In multihop 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 following example.

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 underserved link. Let us consider the simple wireless network depicted in Fig. 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 underserved; 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. Fig. 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.

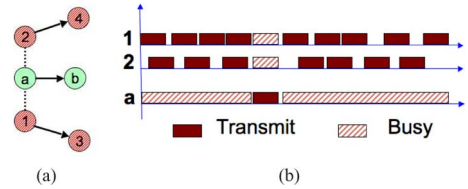

Fig. 1. Example of transmission alignment due to (lack of) carrier sensing. (a) Topology. (b) Typical timeline.

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 Fig. 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 that 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. $\vec{D} = (d_1, d_2, \dots d_N)$, $d_i \in \{0, 1\}$, 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 $\mathrm{AS}(L, \vec{D}_j)$, is the fraction of time during the interval $[0, L]$ for which the network was in state $\vec{D}_j$, i.e., $\mathrm{AS}(L, \vec{D}_j) = \frac{1}{L}\int_0^L \mathbf{1}_{[\vec{D}(t)=\vec{D}_j]}(t)dt$, 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 $\mathrm{AS}(L, \vec{D}_j)$ over all possible states adds to one:

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

We separately denote as *Activity Share*, $\vec{\mathrm{AS}}$, the distribution of time the network spent in each state during the time interval $[0, L]$, i.e., $\vec{\mathrm{AS}} = \{\mathrm{AS}(L, \vec{D}_j), \forall\, \vec{D}_j\}$. Note that if the network is stationary, $\lim_{L\to\infty} \mathrm{AS}(L, \vec{D}_j)$ 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 Fig. 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, but not be sufficiently greater than the background noise (plus interference) 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. Measurements

In MIDAS, each network node k continuously collects information and delivers a report R_k to the manager at every report interval.[3] In this paper, we suggest a new scheme that we will use to infer the Activity Share, given a set of measurements reported by the nodes $\vec{R} = \{R_1, R_2, \dots R_N\}$.

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 Fig. 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 [6]; therein, the authors show that the overhead is between 100 and 500 kb/s per node, without even considering the multiplicative effect of multihopping [4].

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, $T_k + B_k + I_k = 1, \forall\, k$; thus R_k needs to include only two out of the three time shares. 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. 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 $\vec{\mathrm{AS}}$ (Section III-B). Since the constraints do not generally identify a unique solution, we propose an optimization problem to choose a single representative $\vec{\mathrm{AS}}$ (Section III-C). The last element of the solution addresses the computational complexity of the proposed problem and reduces the dimension of the $\vec{\mathrm{AS}}$ 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 $V_{\mathrm{cs}}(i) = \{j | (i, j) \in E\}$. 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 [23] or passive online estimations [14]).

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 $\vec{\mathrm{AS}}$ 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:

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

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

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

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

where D_j^n denotes the nth 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 $V_{\mathrm{cs}}(k)$ is transmitting. Hence, the Activity Shares of states, in which any of the nodes in $V_{\mathrm{cs}}(k)$ are 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. Equation (4) relates to the idle time of the nodes and can be obtained with considerations analogous to the previous two.

[3]The actual placement of the manager's node is an important design parameter; in fact, an optimal placement would permit to reduce MIDAS overhead by efficiently aggregating node reports and to shorten the report delivery delays. However, in case the manager's node is not optimally placed, MIDAS reports can still be delivered leveraging regular wireless network routing, with minimal overhead penalty due to the reports' small size.

Simple considerations show that any of the three equations associated with each node is redundant with respect to the remaining two and (1). This fact can be easily verified by noticing that the state indexes used for the three constraints (2)–(4) are a partition of the whole set of indexes, thus their Activity Shares sum up to the left-hand-side term of (1). In the following formulation of our inference problem, we consider (4) redundant for all nodes with respect to (1)–(3).

We conclude this section with two remarks. First, the report-based constraints, which are key to our inference methodology, are exclusively based on node cumulative temporal channel observations. This makes our methodology robust to heterogeneous packet lengths and physical transmission bit rates of the participating nodes. Second, the assumption that the links in E are fixed plays a crucial role in enforcing the constraints in (3) and (4). Even though this is a simplifying assumption, related research shows that threshold-based carrier-sensing relationships can be reasonably well approximated as binary [22]. Our experimental results (see Section V-B), and a specific discussion in [17], evaluate the effects of this assumption in a static indoor environment. It is part of our ongoing work in the generalization of the report-based constraints to encompass cases of very high channel variability. Specifically, we are considering to associate weights to the Activity Share elements in the left-hand-side summations of (3) and (4), representing the probability that the signal strength received by node k when the network is in state $\vec{D}_j$ overcomes the carrier-sense threshold.

C. Entropy-Based Statistical Solution

In this section, we show how to determine a representative $\vec{\mathrm{AS}}$ close to the actual $\vec{\mathrm{AS}}$ occurred during the measured interval. The representative $\vec{\mathrm{AS}}$ should satisfy the report constraints since the actual $\vec{\mathrm{AS}}$ determines the reported measurements. However, the constraints we defined do not identify in general a single $\vec{\mathrm{AS}}$, but rather a feasible solution domain. Each Activity Share distribution $\vec{\mathrm{AS}}$ 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 $\vec{\mathrm{AS}}$ 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 $\vec{\mathrm{AS}}$ 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 $\mathrm{AS}(\vec{D}_j)$ 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., $\mathrm{AS}(\vec{D}_j)$ 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., $\{\vec{D}_j | \exists\, k, l : l \in V_{\mathrm{cs}}(k), D_j^k = 1, D_j^l = 1\}$. Practically, two neighbors can transmit simultaneously only if their backoffs expire in the same slot, while nonadjacent nodes can initiate their transmissions independently. In general, the larger is the number of neighboring transmitting nodes in a state, the lower is the probability of occurrence of that state since such occurrence would require that a number of backoff counters expired exactly in the same slot. As a consequence, among the admissible $\vec{\mathrm{AS}}$, our scheme should favor the $\vec{\mathrm{AS}}$ 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. The computation of the exact *a priori* probability of each state is complicated because the probability of occurrence of states including multiple adjacent transmitters depends on the global network topology. In order to provide a simple solution, we use a coarse-grained approximation that assigns to each network state an *a priori* probability exponentially decreasing with the number of adjacent transmitters the state contains. For example, a state containing two pairs of adjacent transmitters has half the *a priori* probability of a state that contains 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 $\vec{\mathrm{AS}}$ Inference: In the previous paragraph, we formalized our knowledge of the protocol behavior by using an *a priori* distribution of $\vec{\mathrm{AS}}$. Our objective is to select the feasible $\vec{\mathrm{AS}}$ closest to the defined *a priori* distribution. We propose to use the concept of Kullback–Leibler distance [7] to quantify the distance between two distributions and select the representative $\vec{\mathrm{AS}}$ as the feasible solution that minimizes such distance from the *a priori* distribution. Accordingly, the problem is formulated following the Minimum Relative Entropy Principle.[4] 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.

$\vec{\mathrm{AS}}$ Inference Problem: We formulate the $\vec{\mathrm{AS}}$ inference problem as

$$\begin{aligned} \min_{\boldsymbol{x}} \quad & \sum_{j=0}^{\gamma-1} x_j \log \frac{x_j}{w_j} \\ \text{s.t.} \quad & \boldsymbol{\Phi} \cdot \boldsymbol{x} = \boldsymbol{T} \\ & \boldsymbol{\Psi} \cdot \boldsymbol{x} = \boldsymbol{B} \\ & \mathbf{1}' \cdot \boldsymbol{x} = 1 \\ & \boldsymbol{x} \geq \mathbf{0} \end{aligned} \tag{5}$$

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 jth entry, x_j, is $\mathrm{AS}(\vec{D}_j)$; $\boldsymbol{w}$ is a γ-dimensional vector, whose jth entry, w_j, is the *a priori* distribution of the network state $\vec{D}_j$; $\boldsymbol{\Phi}$ is an $N \times \gamma$ matrix, whose ijth entry is 1 if $D_j^i = 1$, 0 otherwise; $\boldsymbol{\Psi}$ is an $N \times \gamma$ matrix, whose ijth entry is 1 if $D_j^i = 0$ and $\exists\ s \in V_{\mathrm{cs}}(i) : D_j^s = 1$; $\boldsymbol{T}$ and $\boldsymbol{B}$ are N-dimensional vectors,

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

whose kth 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 $\boldsymbol{w}$; furthermore, the first and second constraints (each N-dimensional) correspond to (2) and (3), respectively, while the third constraint (one-dimensional) corresponds to (1).

D. Protocol-Based State-Space Reduction

The solution space of the $\vec{\mathrm{AS}}$ 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 $\vec{\mathrm{AS}}$ that assign large probabilities to states including neighboring transmissions is unlikely. We take advantage of this consideration by excluding from the solution space the $\vec{\mathrm{AS}}$ with $\mathrm{AS}(\vec{D}_j) > 0$, 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 [8]) but smaller than 2^N.

By using this simplification, the resulting inference problem can be obtained from Problem (5) 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 $\vec{\mathrm{AS}}$ 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-C, while we adopt the full state space representation in the testbed results in Section V-B.

IV. Mitigation of Hindering Transmissions

In this section, we address our goal of improving the throughput of underserved 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. The manager uses our prediction tool to anticipate the outcome of alternative corrective actions (a corrective action is a pair of conflicting node and rate-limiting amount) and to choose the most profitable according to the management policy. The rate-limiting amount of any conflicting node may be determined, e.g., considering the current throughput surplus of the links that node forms, with respect to a previously agreed minimum. In case rate-limiting a single conflicting node reveals insufficient to reach the target throughput on the underserved link, the manager iterates the evaluation after collecting new reports.

Our prediction tool is comprised of two procedures: 1) we address the main challenge of estimating the Activity Share after a potential corrective action; 2) 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 underserved. The second procedure uses a simple model that identifies how the Activity Share affects the busy time and collision probability of the underserved 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 underserved 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 underserved 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, Figs. 7 and 13–15).

Denote AS^{o} (Activity Share *old*) and AS^{n} (Activity Share *new*) as 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 $\{\vec{D}_l^{k0}\}$ as the states in which k does not transmit (i.e., $D_l^k = 0$), and $\{\vec{D}_l^{k1}\}$ as the states in which k does (i.e., $D_l^k = 1$), and we establish that the jth states, i.e., $\vec{D}_j^{k0}$ and $\vec{D}_j^{k1}$, differ only for the kth entry, i.e., $\vec{D}_j^{k0} = \{d_{j1} \cdots d_{jk-1} 0\ d_{jk+1} \cdots d_{jN}\}$ and $\vec{D}_j^{k1} = \{d_{j1} \cdots d_{jk-1} 1\ d_{jk+1} \cdots d_{jN}\}$. Using the differential approach, the Activity Share of the network states (in $\{\vec{D}_l^{k1}\}$) in which k transmits decreases proportionally to the duration of those states in AS^o, and the state $\vec{D}_j^{k0}$ benefits from the decrease of the state $\vec{D}_j^{k1}$, for all j. Formally

$$\mathrm{AS}^{\mathrm{n}}(\vec{D}_j^{k1}) \approx \mathrm{AS}^{o}(\vec{D}_j^{k1}) - \frac{\mathrm{AS}^{\mathrm{o}}(\vec{D}_j^{k1})}{\sum_{l:D_l^k=1} \mathrm{AS}^{\mathrm{o}}(\vec{D}_l)} \cdot h \cdot \mathrm{RL}_k \quad (6)$$

$$\mathrm{AS}^{\mathrm{n}}(\vec{D}_j^{k0}) \approx \mathrm{AS}^{\mathrm{o}}(\vec{D}_j^{k0}) + \frac{\mathrm{AS}^{\mathrm{o}}(\vec{D}_j^{k1})}{\sum_{l:D_l^k=1} \mathrm{AS}^{\mathrm{o}}(\vec{D}_l)} \cdot h \cdot \mathrm{RL}_k \quad (7)$$

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; the use of different bit rates

and thus packet durations on different links may be accommodated extending h to a vector form. Next, we will use the AS^{n} to obtain the new collision probability of the underserved link.

B. Relationship Between the Collision Probability of the Underserved Link and the Activity Share

According to [10], we can express the maximal throughput of any link after the rate-limiting action by estimating the new busy time of its sender and the new collision probability. The new busy time of the sender can be obtained from the new Activity Share using (3). In this section, we show how to use the new Activity Share to determine the new collision probability of any link, and in particular of the underserved. For the sake of simplicity, our derivation only considers the packet collisions with hidden terminals, which are typically the overwhelming majority.

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. We illustrate this issue via a simple example. Consider an underserved link (a, b) affected by a hidden terminal c, and suppose that we aim to determine the collision probability on (a, b) using the Activity Share distribution. Let τ be the sum of the Activity Share of the states where nodes a and c transmit simultaneously. Since a packet on (a, b) can collide at most with two different packets sent by c (assuming a fixed and identical duration h of the packets sent by a and c), the total number of packet collisions that c may have caused on (a, b) can be any integer in the range $[\frac{\tau}{h}, 2\min\{\mathrm{Pkt}_a, \mathrm{Pkt}_c\}]$, where Pkt_a (resp. Pkt_c) denotes the number of packets transmitted during the observation interval by node a (resp. c). This shows that a large range of collision probabilities is consistent with the same Activity Share distribution. We remark that the example above is only provided for illustrative purposes and is not intended to limit the applicability of our solution. 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 [10], [23]. The ON period is the interval during which at least a 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, $\bar{T}_{\mathrm{ON}}$, with h.[5]
3) Conditioned on the fact that i can transmit, i.e., that the nodes in $V_{\mathrm{cs}}(i)$ 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: $p^{i,j} = 1 - \frac{\bar{T}_{\mathrm{OFF}}}{\bar{T}_{\mathrm{ON}}+\bar{T}_{\mathrm{OFF}}} e^{-\frac{h}{\bar{T}_{\mathrm{OFF}}}}$ [10]; assumption 2) permits to obtain $p^{i,j}$ as a function of $\frac{\bar{T}_{\mathrm{ON}}}{\bar{T}_{\mathrm{ON}}+\bar{T}_{\mathrm{OFF}}}$. In the remainder, we show how to express $\frac{\bar{T}_{\mathrm{ON}}}{\bar{T}_{\mathrm{ON}}+\bar{T}_{\mathrm{OFF}}}$ (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 $\{\mathrm{ON}$ or $\mathrm{OFF}\}$ states during a measurement interval ΔT: The ratio between these two quantities is equal to the ratio of their averages $\frac{\bar{T}_{\mathrm{ON}}}{\bar{T}_{\mathrm{ON}}+\bar{T}_{\mathrm{OFF}}}$. 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 $V_{\mathrm{cs}}(i)$. 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 $V_{\mathrm{cs}}(i)$ is transmitting. Thus, the whole duration of the ON/OFF process in ΔT is $(1 - B_i)\Delta T$. Let us denote $V_{\mathrm{ht}}(i, j)$ as 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 $V_{\mathrm{cs}}(i)$ is transmitting. By using the Activity Share, we denote the latter interval as $\mathrm{AS}^{\mathrm{HT}*}\Delta T$, where

$$\mathrm{AS}^{\mathrm{HT}*} = \sum_{l:(\exists\ m\in V_{\mathrm{ht}}(i,j):D_l^m=1)\wedge(D_l^n=0,\forall\ n\in V_{\mathrm{cs}}(i))} \mathrm{AS}(D_l). \tag{8}$$

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

$$\frac{\bar{T}_{\mathrm{ON}}}{\bar{T}_{\mathrm{ON}}+\bar{T}_{\mathrm{OFF}}} = \frac{\mathrm{AS}^{\mathrm{HT}*}\Delta T}{(1-B_i)\Delta T} \equiv \mathrm{AS}^{\mathrm{normHT}*}. \tag{9}$$

By replacing (9) into $p^{i,j}$, we can write

$$p^{i,j} = 1 - (1 - \mathrm{AS}^{\mathrm{normHT}*})e^{-\frac{\mathrm{AS}^{\mathrm{normHT}*}}{1-\mathrm{AS}^{\mathrm{normHT}*}}} \tag{10}$$

which expresses the collision probability of a link using exclusively the Activity Share. Using (10), we can compute the throughput according to [10].

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. Additional results can be found in [16] and [17].

[5]Notice that in case different links use different bit rates and thus packet durations, the average duration of an ON period may be approximated as the average of the durations; the evaluation of this enhancement is beyond the scope of this paper.

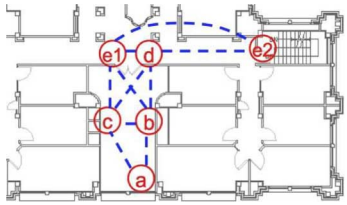

Fig. 2. Layout of our testbed deployment.

A. Experimental Testbed

WARP: To validate MIDAS, we used the Wireless Open-Access Research Platform (WARP) developed at Rice University, Houston, TX[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 Mb/s 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. 1) The *transmission duration counter* measures the time duration the radio is in transmission state by timing the functions that control the transmission operations. 2) The *subpacket 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,[6] 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-sensing threshold.

Validation Tool: Two additional components were implemented only for validation purposes. 1) The *fast RSSI sampler* behaves identically to the subpacket RSSI time sampler described above, but supports higher sampling rates via a digital design, thus improving the precision of the busy time estimation. 2) 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 Fig. 2. In the single-hop topology $S1$, all nodes are next to each other close to position b. In the multihop topology $M1$, the nodes are located in the positions $\{a, b, c, d, e1\}$. In the multihop topology $M2$, the nodes are in positions $\{a, b, c, d, e2\}$. Each board transmits 1000-B data packets, with constant interpacket time whose value depends on the experiment. Each experiment run lasts 10 s and, where not differently specified, the reported results are cumulative over 10 runs.

[6]Multiband Atheros Driver for Wifi available at http://madwifi.org/

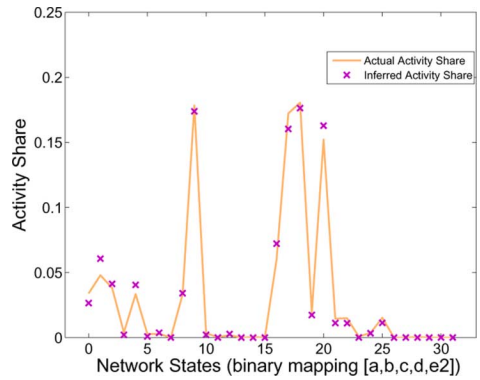

Fig. 3. Activity share inference (testbed).

B. 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 (5). We compute the optimal solution of (5) corresponding to the collected values using the MATLAB solver *fmincon*. We establish the accuracy of the Activity Share inference by comparing our estimations to the ground truth provided by an *omniscient 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 Fig. 3. 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 0–4 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. Furthermore, 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 nonneighboring nodes by combining their transmission reports with the busy time share reports of their common neighbors. On the one hand, the lower the density, the more tightly the busy time share reports constrain the overlapping transmissions of nonneighboring nodes, but also the fewer reports that 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 time share 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 nonneighbors to one another), based on the busy time share of the node, it is not possible to determine the amount of overlapping transmissions of any

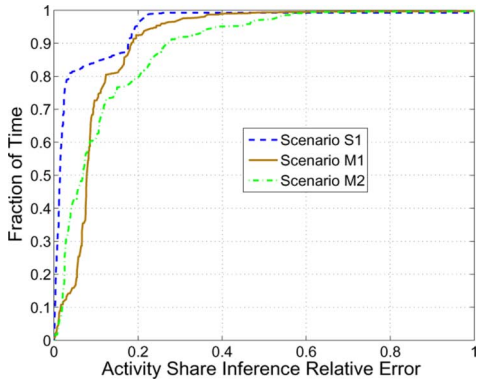

Fig. 4. Inference sensitivity to network density (testbed).

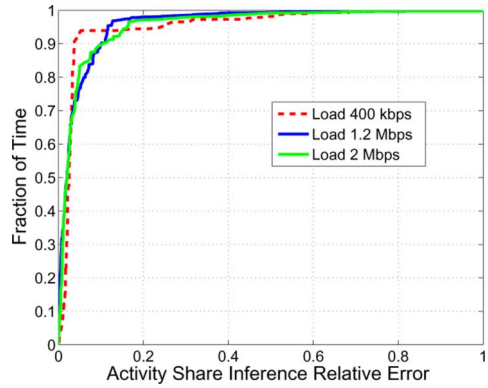

Fig. 5. Inference sensitivity to traffic load (testbed).

two or all three of the neighbors. In fact, the higher the density, the more combinations of overlapping transmissions of a node's neighbors are consistent with the node's busy time share report. On the other hand, the higher 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.

Fig. 4 shows the cumulative distribution function (cdf) of the relative error of the Activity Share inference (notice that the probability of a state used to compute the cdf is the Activity Share of that state, i.e., its duration). The x-axis indicates the relative error committed, while the y-axis is in (nondimensional) 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. Furthermore, $S1$ is the most accurate solution, while the $M1$ plot mostly dominates $M2$. The respective mean relative errors, i.e., the relative error committed in the state occupied 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 [16].

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 transmission time shares). 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 traffic loads of all nodes to 400 kb/s, 1.2 Mb/s, and 2 Mb/s (also in this case, each experiment is repeated 10 times).

Fig. 5 depicts the cdf of the relative error of the Activity Share estimation. As can be seen in the figure, the Activity Share inference tool attains a very low relative error. Furthermore, the variations among the three plots are minimal and are comparable to the results attained for the fully backlogged case. In particular, the mean relative error is 4.6%, 4.0%, 4.5% for 400 kb/s, 1.2 Mb/s, and 2 Mb/s, 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 nonuniform traffic patterns over larger topologies to the simulation section (see the paragraph "Robustness to Real Traffic Distribution" in Section V-C).

Sensitivity to Report Interval Length: In the previous experiments, we used report intervals of 10 s, i.e., each node k sent one report every 10 s including the busy and transmission time shares 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). In order to clarify the last issue, we notice that our entropy-based inference tool is most accurate if the node reports reflect steady-state observations of the underlying process, i.e., formed by the transmissions of all network nodes (see Section III). As the report intervals shorten, the actual realizations of the process during each interval may largely depart from the steady state, thus degrading the inference accuracy. 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 traffic.

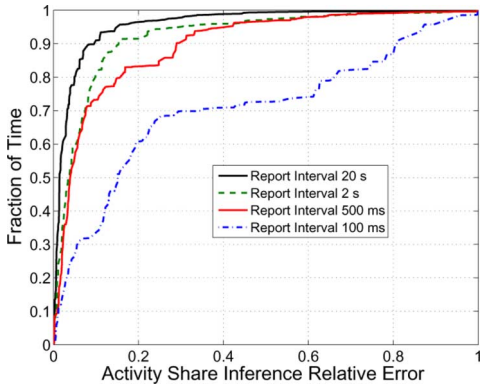

Fig. 6. Inference sensitivity to short report intervals (testbed).

The experiments show that the inference tool is accurate also for short report intervals (Fig. 6). 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 mean 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 B, 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 $\{a \rightarrow c; b \rightarrow a; c \rightarrow a; d \rightarrow b; e1 \rightarrow c\}$. For each set of experiments, we consider a target underserved link whose traffic is fully backlogged, and we perform a baseline run, measuring the throughput of the target link when all others transmit at a rate randomly chosen in the [400 kb/s, 900 kb/s] 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 kb/s). Then, we perform four additional runs on the testbed (one per conflicting node), alternately rate-limiting a different conflicting node for the same 400-kb/s 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.

Fig. 7 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 kilobits per second); 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 73 kb/s different from the actual throughput gain (i.e., about 18% of the rate-limiting value of 400 kb/s, or 26% of the average actual throughput gain of approximately 280 kb/s), with the per-link error being $\{84, 67, 80, 58, 74\}$ kb/s (sorted in the alphabetical order of the sender). *We conclude that, despite rate-limiting, different conflicting nodes can have largely different impacts on the throughput of an underserved link, and our prediction tool adequately captures the heterogeneous effects.*

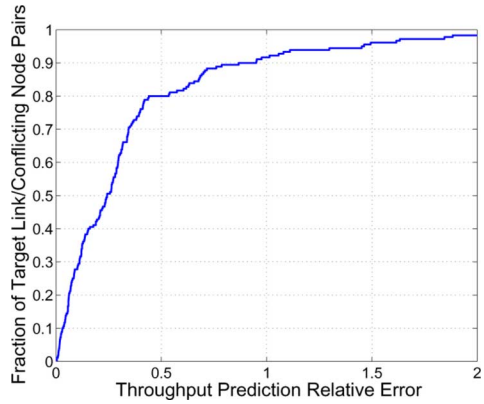

Fig. 7. Throughput increase estimation for concurrent nodes with loads in [400 kb/s, 900 kb/s] (testbed).

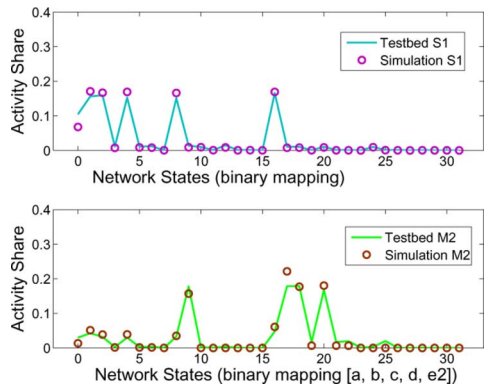

Fig. 8. Activity share: simulation versus testbed.

C. 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 this paper implement such enhancement. We also investigate the robustness of the inference technique to realistic traffic conditions and report losses.

Simulation Settings: We consider scenarios where each node generates 1000-B UDP packets directed toward a single neighbor, with constant interpacket 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 Fig. 8, which is obtained using 802.11a at 6 Mb/s, all results in this section are obtained using 802.11b at 11 Mb/s data rate in order to experiment with different conditions. We refer to the analogous simulation results obtained for 802.11a at 6 Mb/s as needed.

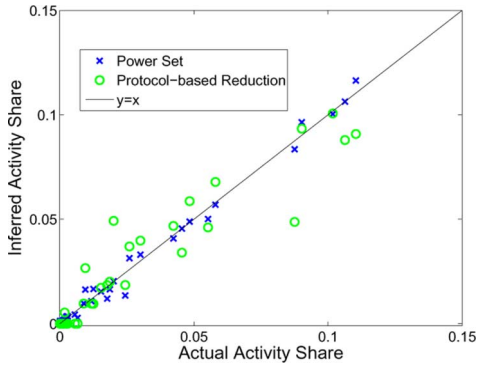

Fig. 9. Inference with protocol-based reduction.

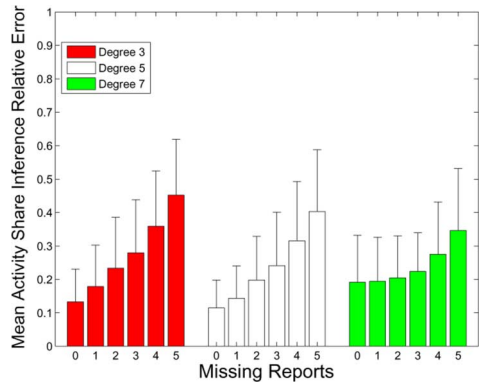

Fig. 10. Inference robustness to missing reports.

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 reason, 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 *omniscient centralized approach*, we extract the Activity Share from the traces of simulation and testbed, and we compare them. Fig. 8 shows the actual Activity Share (y-axis) for all 32 possible states (x-axis) sorted similarly to Fig. 3.[7] *The plots show an excellent agreement between the two environments; the small discrepancies are due to nonideal 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. As remarked therein, the reduction of the state space improves the computational performance of our inference tool by decreasing the size of the feasible solution domain. However, the exclusion from the domain of the states including neighboring transmissions introduces an inconsistency between the domain and the measurement reports. In fact, the excluded states likely occurred, even if for relatively short durations, during each report interval and thus influence the reports input to the inference tool. However, the protocol-based state-space reduction ignores those states in the construction of the feasible solution domain. In order to assess how this approximation affects the accuracy of the inference tool, we generate a random topology of 10 nodes, with an average number of seven 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").

Fig. 9 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, which 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 is higher when considering the entire state space since the solution domain of the protocol-based state reduction ignores states that contributed to the reported measurements.*

Robustness to Incomplete Information: In the case of severe network congestion, some of the reports could be lost. As the number of report losses increases, the accuracy of the inference tool is expected to decrease because of the less constrained, and thus larger, feasible solution domain including the actual Activity Share. A larger solution domain entails a higher uncertainty in the search of the actual Activity Share since all solutions within the domain may have occurred (see Section III-C). We evaluate how report losses affect the accuracy of the inference tool by simulating the loss of up to five out of the 10 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). Fig. 10 shows the mean relative error of the Activity Share inference 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 85th 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.*

Robustness to Real Traffic Distribution: In our previous experiments, all traffic sources generate packets according to predefined interpacket 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 [6] in our simulation environment. In particular, we randomly select 10 nodes from the UCSD traces, and we play 10 s of their traffic on a network topology obtained as follows. Two nodes are considered disconnected if the replayed 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 an overlapping valid only if its duration exceeds 100 μs); otherwise, the two nodes are considered connected. Each simulated node generates packets according to the transmission times of the UCSD node it represents in the trace. Fig. 11 shows a

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

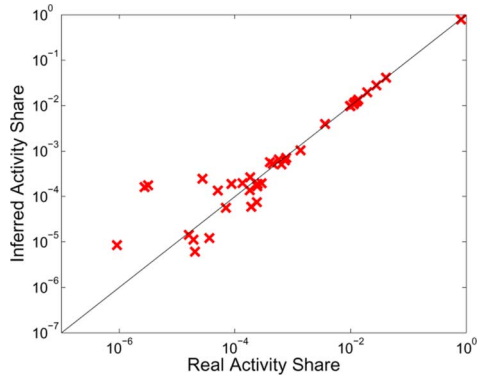

Fig. 11. Inference robustness to realistic traffic.

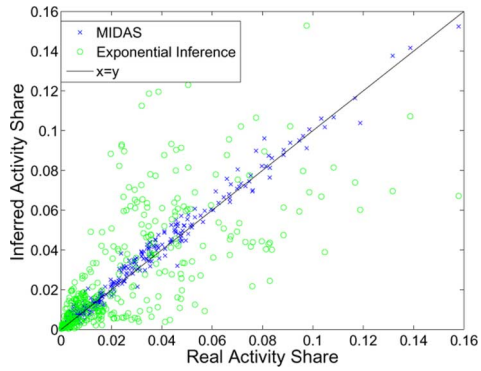

Fig. 12. Comparison to an exponential inference method.

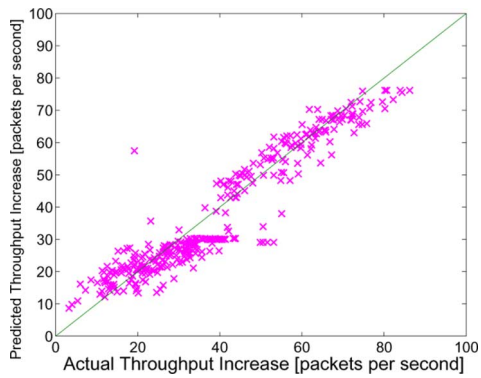

Fig. 13. Throughput increase estimation for scenarios with density 3.

scatterplot of the measured Activity Share versus 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 mean 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.*

Comparison to an Exponential Inference Technique: In order to provide a comparison of our inference technique to an alternative baseline method, we design an Exponential Inference Technique based on the assumption that the time duration the system remains in any *network state* is exponentially distributed. This permits us to model the temporal evolution of the system, by using a continuous-time Markov chain [9]; notice that this modeling technique is common to several related works [3], [10], [23]. From any state D, the system transitions to the state $D \cup i$ with rate λ_i, where λ_i is the transmission rate of i in state D; the system transitions to the state $D - i$ with rate μ_i, where μ_i is the termination rate of i in state D. Specifically, for the states D for which $D \cup i$ is an independent set, λ_i *is equal to* the transmission rate of i, once normalized over the duration of the states in which i can transmit because it senses the medium idle (i.e., $1 - B_i$), $\lambda_i = 0$ for the other states; for the states D for which $i \in D$, μ_i is the reciprocal of the duration of a data packet transmission, $\mu_i = 0$ for the other states. For fairness of comparison, differently from all related works, we directly measure the input parameters from the operational network.[8] We compare MIDAS to the exponential inference solution for topologies of 10 nodes, where each node has on average five neighbors and transmits fully backlogged traffic. Fig. 12 shows the scatterplot of all the predictions obtained for 10 scenario repetitions, where X's denote MIDAS predictions, while O's denote the exponential modeling solution. The figure shows that MIDAS largely outperforms the alternative, and the mean relative error is 9% versus 36%. *We conclude that a simplifying exponential assumption does not adequately characterize the real system behavior and does not permit to design a technique to accurately infer the Activity Share.*

[8]Of course, we do not use such inputs in MIDAS (see Section II-B).

D. 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-B. We start by running MIDAS on a 10-node random topology with density 3. Node transmission rates are set to 600 kb/s. As in the experimental case, we pick one target underserved link, increase its load until it is fully backlogged, and successively repeat the experiment, rate-limiting each time a different conflicting flow by 400 kb/s; we iterate this procedure for all links in the network. The scatterplot in Fig. 13 compares 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 kb/s, i.e., from 28 to 688 kb/s. 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

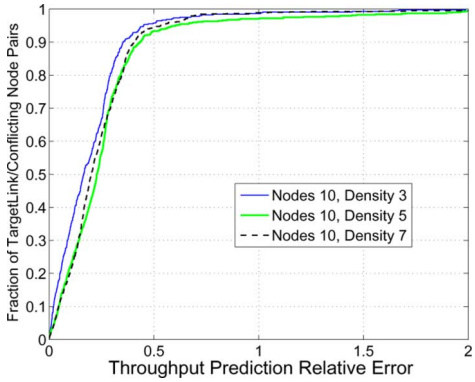

Fig. 14. Throughput prediction sensitivity to density (600 kb/s).

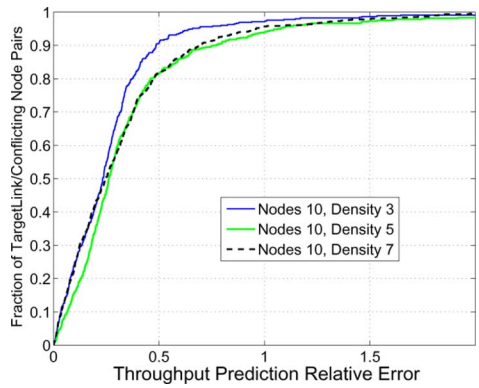

Fig. 15. Throughput prediction sensitivity to density (900 kb/s).

Section V-B, [16], and [17]). 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 kb/s. Fig. 14 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 three 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 [16, Fig. 8] 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), e.g., our model ignores collisions with terminals in carrier-sensing range, whose incidence grows with the density of the scenarios. 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 kb/s (i.e., 20% of the rate-limiting value of 400 kb/s) from the actual throughput gain in 83%–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 underserved 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 kb/s. Fig. 15 shows the same ranking among the curves relative to different densities as for the case of 600 kb/s. However, the accuracy obtained for 600 kb/s is higher than for 900 kb/s. 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 [17]). 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 underserved link, thus increasing the influence of the less accurate results for lower gains on the cdf. For example, for density 5 and 600 kb/s, on average the underserved link gains 60% of the rate-limiting amount (i.e., 240 out of 400 kb/s in this experiment), while for the case of density 5 and 900 kb/s, the underserved link gains 40% (i.e., 160 out of 400 kb/s). This explains why the relative error is larger for 900 kb/s than for 600 kb/s.

VI. Related Work

Wireless Network Monitoring: Performance monitoring of single-hop WLANs has recently attracted research interest [6], [18]. 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 multihop wireless networks. In our work, we show that we can attain very accurate results with the use of small time-averaged reports. Furthermore, [6] and [18] 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 [3], [5], [10], [12], [13], [19], [23], [25]. 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 underserved links. Our scheme relies on active offline link profiling, such as [13] and [23], 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, [14] and [24] propose methods to avoid active offline profiling: [24] uses low overhead online probes; [14] uses passive online estimations using traces collected by deployed sniffers. While [14] and [24] only consider pairwise link interference and do not characterize the coordination between conflicting nodes, we can leverage the results therein for online link profiling.

Alternative Mitigation Techniques: Network managers have a number of options to optimize the performance of underserved links other than rate-limiting conflicting links.

Alternative mitigation techniques have been suggested in literature, e.g., channel selection [15], association control [2], transmit power control [20], and channel width adaptation [21]. We remark that they are orthogonal to our inference technique and can benefit from it.

VII. Conclusion

In this paper, we present a management framework for wireless networks called MIDAS. MIDAS addresses the problem of identifying the conflicting nodes that cause underperformance 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 a mean relative error as low as 4% and predicts the throughput gain of an underserved link corresponding to alternative rate-limiting actions with an error lower than 20% of the rate-limited quantity.

Acknowledgment

The authors are grateful to J. Kruys and S. Pandey for insightful comments and feedback on the project.

References

[1] Rice University, Houston, TX, "Rice University WARP project," [Online]. Available: http://warp.rice.edu

[2] Y. Bejerano, S. J. Han, and L. Li, "Fairness and load balancing in wireless LANs using association control," in *Proc. ACM MobiCom*, Sep. 2004, pp. 315–329.

[3] R. R. Boorstyn, A. Kershenbaum, B. Maglaris, and V. Sahin, "Throughput analysis in multihop CSMA packet networks," *IEEE Trans. Commun.*, vol. 35, no. 3, pp. 267–274, Mar. 1987.

[4] J. Camp, V. Mancuso, O. Gurewitz, and E. Knightly, "A measurement study of multiplicative overhead effects in wireless networks," in *Proc. IEEE INFOCOM*, Apr. 2008, pp. 1633–1641.

[5] M. Carvalho and J. Garcia-Luna-Aceves, "A scalable model for channel access protocols in multihop ad hoc networks," in *Proc. ACM MobiCom*, Sep. 2004, pp. 330–344.

[6] 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 *Proc. ACM SIGCOMM*, Sep. 2006, pp. 39–50.

[7] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. New York: Wiley, 1991.

[8] R. Diestel, *Graph Theory (Graduate Texts in Mathematics)*. New York: Springer, 2005.

[9] R. Gallager, *Discrete Stochastic Processes*. Norwell, MA: Kluwer, 1990.

[10] M. Garetto, T. Salonidis, and E. W. Knightly, "Modeling per-flow throughput and capturing starvation in CSMA multi-hop wireless networks," in *Proc. IEEE INFOCOM*, Apr. 2006, pp. 1–13.

[11] K. Jain, J. Padhye, V. Padmanabhan, and L. Qiu, "Impact of interference on multi-hop wireless network performance," in *Proc. ACM MobiCom*, Sep. 2003, pp. 66–80.

[12] A. Jindal and K. Psounis, "Characterizing the achievable rate region of wireless multi-hop networks with 802.11 scheduling," *IEEE/ACM Trans. Netw.*, vol. 18, no. 3, pp. 257–281, Mar. 2009.

[13] A. Kashyap, S. Ganguly, and S. R. Das, "A measurement-based approach to modeling link capacity in 802.11-based wireless networks," in *Proc. ACM MobiCom*, Sep. 2007, pp. 242–253.

[14] A. Kashyap, U. Paul, and S. R. Das, "Deconstructing interference relations in WiFi networks," in *Proc. IEEE SECON*, Jun. 2010, pp. 1–9.

[15] B. Kauffman, F. Baccelli, A. Chaintreau, V. Mhatre, K. Papagiannaki, and C. Diot, "Measurement-based self organization of interfering 802.11 wireless access networks," in *Proc. IEEE INFOCOM*, May 2007, pp. 1451–1459.

[16] E. Magistretti, O. Gurewitz, and E. W. Knightly, "Inferring and mitigating hindering transmissions in managed 802.11 wireless networks," in *Proc. ACM MobiCom*, Sep. 2010, pp. 305–316.

[17] E. Magistretti, O. Gurewitz, and E. W. Knightly, "Inferring and mitigating a link's hindering transmissions in managed 802.11 wireless networks," Rice Univ., Houston, TX, Tech. Rep., 2010 [Online]. Available: http://networks.rice.edu/papers/inference-paper.pdf

[18] R. Mahajan, M. Rodrig, D. Wetherall, and J. Zahorjan, "Analyzing the MAC-level behavior of wireless networks in the wild," in *Proc. ACM SIGCOMM*, Sep. 2006, pp. 75–86.

[19] K. Medepalli and F. Tobagi, "Towards performance modeling of IEEE 802.11 based wireless networks: A unified framework and its applications," in *Proc. IEEE INFOCOM*, Apr. 2006, pp. 1–12.

[20] V. Mhatre, K. Papagiannaki, and F. Baccelli, "Interference mitigation through power control in high density 802.11 WLANs," in *Proc. IEEE INFOCOM*, May 2007, pp. 535–543.

[21] T. Moscibroda, R. Chandra, Y. Wu, S. Sengupta, P. Bahl, and Y. Yuan, "Load-aware spectrum distribution in wireless LANs," in *Proc. IEEE ICNP*, Oct. 2008, pp. 137–146.

[22] D. Niculescu, "Interference map for 802.11 networks," in *Proc. ACM IMC*, Oct. 2007, pp. 339–350.

[23] L. Qiu, Y. Zhang, F. Wang, M. K. Han, and R. Mahajan, "A general model of wireless interference," in *Proc. ACM MobiCom*, Sep. 2007, pp. 171–182.

[24] T. Salonidis, G. Sotiropoulos, R. Guerin, and R. Govindan, "Online optimization of 802.11 mesh networks," in *Proc. ACM CoNext*, Dec. 2009, pp. 61–72.

[25] X. Wang and K. Kar, "Throughput modelling and fairness issues in CSMA/CA based ad-hoc networks," in *Proc. IEEE INFOCOM*, Mar. 2005, pp. 23–34.

Eugenio Magistretti (S'04) received the Laurea and Doctorate degrees in computer engineering from the University of Bologna, Bologna, Italy, in 2003 and 2007, respectively, and is currently pursuing the Ph.D. degree in electrical and computer engineering at Rice University, Houston, TX.

His main research interests are in the area of wireless MAC protocols, with a focus on design, performance modeling, and evaluation.

Omer Gurewitz (S'00–M'05) received the B.Sc. degree in physics from Ben Gurion University, Beer Sheva, Israel, in 1991, and the M.Sc. and Ph.D. degrees in electrical engineering from the Technion—Israel Institute of Technology, Haifa, in 2000 and 2005, respectively.

He is an Assistant Professor with the Department of Communication Systems Engineering, Ben Gurion University. Between 2005 and 2007, he was a Post-doctoral Researcher with the Electrical and Computer Engineering Department, Rice University, Houston, TX. His research interests are in the field of performance evaluation of wired and wireless communication networks. His current projects include cross-layer design and implementation of medium access protocols for 802.11 as well as 802.16 (WiMAX) standards.

Edward W. Knightly (S'91–M'96–SM'04–F'09) received the B.S. degree from Auburn University, Auburn, AL, in 1991, and the M.S. and Ph.D. degrees from the University of California, Berkeley, in 1992 and 1996, respectively, all in electrical engineering.

He is a Professor of electrical and computer engineering with Rice University, Houston, TX. His research interests are in the areas of mobile and wireless networks and high-performance and denial-of-service resilient protocol design.

Prof. Knightly is a Sloan Fellow. He served as Associate Editor of numerous journals and special issues including the IEEE/ACM TRANSACTIONS ON NETWORKING and IEEE JOURNAL ON SELECTED AREAS OF COMMUNICATIONS Special Issue on Multi-Hop Wireless Mesh Networks. He served as Technical Co-Chair of IEEE INFOCOM 2005 and General Chair of ACM MobiHoc 2009 and ACM MobiSys 2007, and served on the program committee for numerous networking conferences including ICNP, INFOCOM, MobiCom, and SIGMETRICS. He is a recipient of the National Science Foundation CAREER Award. He received the Best Paper Award from ACM MobiCom 2008.