Identifying High Throughput Paths in 802.11 Mesh Networks: a Model-based Approach

Theodoros Salonidis, Thomson Paris Research Lab
Michele Garetto, Università di Torino, Italy
Amit Saha, Tropos Networks, CA
Edward Knightly, Rice University, Houston, TX

***Abstract*— We address the problem of identifying high throughput paths in 802.11 wireless mesh networks. We introduce an analytical model that accurately captures the 802.11 MAC protocol operation and predicts both throughput and delay of multi-hop flows under changing traffic load or routing decisions. The main idea is to characterize each link by the packet loss probability and by the fraction of busy time sensed by the link transmitter, and to capture both intra-flow and inter-flow interference. Our model reveals that the busy time fraction experienced by a node, a locally measurable quantity, is essential in finding maximum throughput paths. Furthermore, metrics that do not take this quantity into account can yield low throughput by routing over congested paths or by filtering-out non-congested paths. Based on our analytical model, we propose a novel routing metric that can be used to discover high throughput path in a congested network. Using city-wide mesh network topologies we demonstrate that our model-based metric can achieve significant performance gains with respect to existing metrics.**

I. Introduction

Mesh networks offer inexpensive wireless coverage over large areas via use of wireless multi-hopping to wireline gateway nodes. Recently, cities are expanding use of mesh networks from public service and public safety to also include large-scale public broadband wireless access, potentially serving millions of users.[1] Such deployments will carry high amounts of traffic that will stress the 802.11 mesh backbone, thus causing unfairness and starvation, well known problems of the 802.11 CSMA protocol. Given this limitation, modeling and understanding 802.11 in conjunction with congestion control, traffic engineering and routing schemes is of paramount importance.

In this paper, we address the problem of identifying high throughput paths in an 802.11 mesh network by introducing a model that accurately predicts throughput and delay of multi-hop flows under fixed or changing traffic conditions. Existing models for 802.11 mesh networks focus on predicting throughput of a set of single-hop, single-receiver flows [6], [11], [13], [14]. A recent paper [10] proposes an analytical approach to estimate the end-to-end throughput over a single path, which is limited to the case of nodes having a single receiver and requires a rather complex, centralized computation that cannot be translated into a routing protocol. In contrast to [10], our model applies to arbitrary traffic matrices and yields a routing metric that can be easily incorporated to an efficient routing protocol to discover the optimal path. In [15] the authors propose an admission control schemes for flows in a single-channel, multi-hop network based on knowledge of both local resources at a node and the effect of admitting the new flow on neighboring nodes. In contrast to [15], our approach is based on a more precise mathematical model of the behavior of 802.11, and can efficiently discover the path providing the largest bandwidth to the new flow.

Our model expresses the throughput and delay of each link of a multi-hop flow as a function of (i) average input rate, (ii) the fraction of busy time carrier-sensed by the link transmitter and (iii) the packet loss probability experienced on the link – all locally measurable with zero or minimal communication overhead. To model changing traffic conditions, we introduce a two-step technique to estimate available path bandwidth, defined as *the maximum additional rate a flow can push before saturating its path*. The first step computes the capacity of each link in the path using the busy time fraction and packet loss probability as a summary of the interference caused by other links outside the path. In the second step, the link capacities are coupled with a clique-based computation that captures interference of links within the path. We use the topology of a city-wide mesh network deployed in Chaska, MN to demonstrate the effectiveness of our model in accurately predicting throughput and delay under existing traffic conditions and in estimating available path bandwidth.

Armed with the ability to estimate available bandwidth over a single path, we use our model to study the ability of routing protocols' link-cost metrics to discover high throughput paths. Several mesh routing metrics have been proposed that take into account packet loss probability [7], [8], data transmission rates [4], and multi-channel, multi-radio capabilities [9], [16]. These metrics have been demonstrated to find higher throughput paths than minimum-hop metrics. However, their performance has never been investigated under congested conditions that naturally arise in gateway-centric mesh networks.

To address this question, we use our model and a series of experiments that gradually evolve from single-link to full-scale city-wide topologies. We find that all existing routing metrics are highly sensitive to traffic load and detect congestion through the packet loss probability. However, we show that packet loss does not always provide accurate information and can result in low-throughput routing decisions, either by selecting congested paths or by filtering-out non-congested paths. Instead, the busy time fraction is an additional factor essential to discover high throughput paths, especially under congested conditions.

We introduce a new available bandwidth metric that can be combined with a source-route link-state routing protocol to directly compute the path providing the highest throughput. Our metric takes into account intra-flow and inter-flow inter-

[1] See Houston's RFP for example: www.houstontx.gov/it/wirelessrfp.html.

1-4244-1588-8/07/$25.00 ©2007 IEEE

ference using fraction of busy time as well as packet loss. We compare its performance with existing loss-based metrics in both the Chaska topology and a regular Manhattan network topology, under various traffic scenarios. Our experiments show that our proposed metric based on direct estimation of available bandwidth can yield high gains, being mostly effective in planned mesh networks of regular topology that provide several paths through spatial reuse.

The paper is organized as follows: We introduce the analytical model in Section II. In Section III, we evaluate the model's accuracy and introduce and evaluate the available bandwidth estimation technique. In Section IV, we investigate the ability of existing routing metrics to discover high throughput paths. In Section V, we compare model-based metrics for available bandwidth estimation to existing loss-based metrics. Section VI concludes this paper.

II. Analytical Model

In [11], [12] we introduced a general decoupling technique to analyze the behavior of each node in an 802.11 network with arbitrary topology. One important limitation of our previous work is that we limited ourselves to the case in which each source sends traffic to a single neighboring node. Since this case allows to analyze only particular traffic patterns, we now extend the analysis to the general case in which a node transmits to multiple neighbors. Notice that we use the term 'node' to refer to one network interface, i.e., one instance of the 802.11 MAC protocol which is shared by all flows passing through the node. We first review in Section II-A the modeling framework introduced in [11], [12]. The extension of the analysis to the case of multiple receivers is described in Section II-B.

A. Review of modeling framework

In 802.11, the behavior of a node is determined by what it senses on the channel, i.e., by the occupation of the 'air' around it in the frequency spectrum used. In a wireless mesh network, the state of the channel can be perceived differently by different nodes, because not all of them are in radio range of each other. As a consequence, existing techniques developed in the so called 'single cell' case (usually based on the classic analysis of [5]) are not applicable, and one has to consider the private channel view of each node.

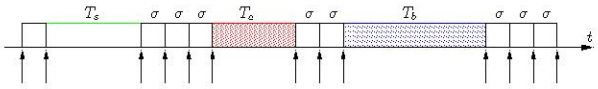


Fig. 1. Example of the evolution of the channel state perceived by a node

The evolution of the channel state experienced by a node can be described as a renewal process with four different states, as illustrated in the example of Fig. 1. The 4 states are : *(i)* idle channel; *(ii)* channel occupied by a successful transmission of the node; *(iii)* channel occupied by a collision of the node; *(iv)* busy channel due to activity of neighboring nodes, detected by means of either physical or virtual carrier sensing (the NAV). The time intervals during which the station remains in each of the four states above are denoted by σ, T_s, T_c, and T_b, respectively. While σ is constant, equal to one backoff slot, the duration of the other intervals can be variable (with general distribution), depending on the access mechanism (basic access or RTS/CTS), the frame size, and the sending rate of the transmitting station(s). Both T_s, T_c, and T_b include a deterministic idle slot at the end (see [12] for details). Let Π_σ, Π_s, Π_c, Π_b be, respectively, the occurrence probabilities of the four states described above. To compute these probabilities, we need to specify the events that can occur after an idle slot has elapsed. Let τ be the probability that the backoff counter of the node reaches zero after an idle slot; let e be the probability that when the backoff counter reaches zero, the transmission queue is empty; let p be the probability that a transmission of the station is not successful; at last, let b be the probability that if the station does not transmit after an idle slot, the channel becomes busy because of the activity of other nodes. Then we can express the occurrence probabilities of the four channel states as follows: $\Pi_s = \tau\,(1-p)(1-e)$, $\Pi_c = \tau\,p(1-e)$, $\Pi_\sigma = [(1-\tau)+\tau e]\,(1-b)$, $\Pi_b = [(1-\tau)+\tau e]\,b$.

Using standard renewal-reward theory, the throughput of the node (expressed in packet/s) is given by

$$T_P = \frac{\tau(1-p)(1-e)}{\Pi_s\bar{T}_s + \Pi_c\bar{T}_c + \Pi_\sigma\sigma + \Pi_b\bar{T}_b} \tag{1}$$

Now, the probability τ is a deterministic function of p, which depends only on backoff parameters such as the window size, the number of backoff stages, etc. The complete expression of τ for 802.11 that takes into account the maximum retransmission limit jointly with the maximum window size, is given by

$$\tau = \frac{2q(1-p^{m+1})}{q(1-p^{m+1}) + W_0\big[1-p-p(2p)^{m'}\left(1+p^{m-m'}q\right)\big]} \tag{2}$$

where $q = 1-2p$, W_0 is the minimum window size, m is the *maximum retry limit*, and m' is the backoff stage at which the window size reaches its maximum value, $m' \leq m$.

The average durations $\bar{T}_s$ and $\bar{T}_c$ of a successful transmission or of a collision in which the station is involved can be computed *a priori* (see [5]), depending only on the distributions of packet sizes and data rates. It turns out that the only unknown variables in Equation (1) are: i) the occurrence probability b of a busy period, and its average duration $\bar{T}_b$; ii) p, the conditional packet loss probability; iii) e, the conditional probability of empty buffer.

The value of e depends on the traffic load of the node. The values of b, $\bar{T}_b$ and p, depend on the interaction of the node with the rest of the network. In [11] we described an iterative technique to compute these quantities, that allows to solve for the entire network and thus predict analytically the stationary behavior of each node, including its throughput. In particular, we introduced a methodology to evaluate the fraction of time f_B during which the channel is sensed busy, as well as the average duration $\bar{T}_b$ of a busy period. The value of f_B is related to our model through the following expression

$$f_B = \frac{\Pi_b\bar{T}_b}{\Pi_s\bar{T}_s + \Pi_c\bar{T}_c + \Pi_\sigma\sigma + \Pi_b\bar{T}_b} \tag{3}$$

Moreover, we proposed a technique to compute the packet loss probability p on the (single) link used by the node. In this paper we are not concerned with the *analytical* computation of f_B, $\bar{T}_b$ and p. The interested reader is referred to [11]. Indeed, in this work we assume that f_B, $\bar{T}_b$ and p are directly measured by each node through a combination of active and

passive measurements. For example p could be measured by a node either by sending broadcast probes or by keeping track of the number of retries associated with transmitted data packets.

B. Extension to multiple receivers

The extension of the model to the case of a node having multiple receivers has to take into account the fact that each outgoing link can have a different quality, which depends on the specific packet loss probability experienced by packets over that link. Notice that the MAC buffer is shared by all packets passing through the node, according to a FIFO queueing discipline. This can cause a head-of-line (HOL) blocking effect in case of heterogeneous outgoing links. In particular, a single link with poor quality can compromise the performance of many high quality links.

The main idea behind our modeling approach is to reconduct the analysis of a node having multiple receivers to the case of a single outgoing link, by defining a properly weighted node average transmission probability $\bar{\tau}$ and node packet loss probability $\bar{p}$, and basically reuse equations (1) and (3). At the same time, we model the dynamics of the shared MAC as a virtual server with given average service capacity.

Let n_L be the number of links used by the node to forward its traffic. Each link is associated to a packet loss probability p_i, $1 \leq i \leq n_L$, which maps into a per-link transmission probability τ_i by (2). Let λ_i be the arrival rate of packets destined to link i, and $\Lambda = \sum_{i=1}^{n_L} \lambda_i$ the total arrival rate of packets to the node. We introduce the weight $w_i = \lambda_i/\Lambda$, $1 \leq i \leq n_L$, which is equal to the probability that a generic packet to be served by the MAC has to go over link i. We also introduce the service rate μ_i of link i, and the aggregate service rate of the node $\mu = \sum_{i=1}^{n_L} \mu_i$.

We model the MAC buffer using a simple M/M/1/B model, where B denotes the buffer capacity expressed in packets. The traffic intensity at the queue is $\rho = \Lambda/\mu$. The probability that the queue is empty is $\pi_0 = (1-\rho)/(1-\rho^{B+1})$. We chose this queue model because it provides simple closed form expressions for the finite buffer case, and because we are mostly interested in the performance of links close to saturation, where the exact details on the arrival and departure processes at the queue are not important, and the exponential assumption provides accurate enough predictions.

To derive the node service rate μ, we put $e = 0$, i.e., we assume that the queue is not empty. We then compute the average number c_i of channel intervals (i.e., any of the four channel intervals described in Section II-A) required to serve a packet belonging to link i. We have

$$c_i = \sum_{j=0}^{m} \frac{W_j + 1}{2} p_i^j$$

where W_j is the window size at backoff stage j. Using quantities c_i, we can compute the probability s_i that a generic channel slot is part of the service time of a packet belonging to link i: $s_i = (w_i c_i)/\sum_{j=1}^{n_L}(w_j c_j)$. The average transmission probability of the node is $\bar{\tau} = \sum_{i=1}^{n_L} s_i \tau_i$, and the average packet loss probability of the node is $\bar{p} = \sum_{i=1}^{n_L} s_i \tau_i p_i/\bar{\tau}$. Let Δ be the average duration of a time slot when the queue is not empty. We have

$$\Delta = \bar{\tau}\ (1-\bar{p})\ \bar{T}_s + \bar{\tau}\ \bar{p}\ \bar{T}_c + (1-\bar{\tau})(1-b)\sigma + (1-\bar{\tau})\ b\ \bar{T}_b$$

The service rate μ_i over link i is the sum of two contributions, $\mu_i = \mu_{T_i} + \mu_{D_i}$. The first is the rate μ_{T_i} at which packets are transmitted successfully, given by

$$\mu_{T_i} = \frac{s_i \tau_i (1-p_i)}{\Delta}$$

The second is the rate μ_{D_i} at which packets are discarded by the MAC due to the maximum retry limit, given by

$$\mu_{D_i} = \frac{s_i \tau_i (1-p_i) p_i^{m+1}}{\Delta(1-p_i^{m+1})}$$

Let $\mu_T = \sum_{i=1}^{n_L} \mu_{T_i}$ and $\mu_D = \sum_{i=1}^{n_L} \mu_{D_i}$. The node aggregate service rate is finally given by $\mu = \mu_T + \mu_D$. Notice that μ, μ_T and μ_D can be expressed as functions of the only unknown variable b.

Now, the aggregate node throughput can be evaluated either as $T_P = (1-\pi_0)\mu_T$, which is a function $h(b)$ of the only unknown b, or from equation (1), where τ and p are substituted by $\bar{\tau}$ and $\bar{p}$, respectively. This latter expression requires knowledge of both b and e. We can thus form a system of two equations in two unknowns, b and e:

$$\begin{cases} \frac{\bar{\tau}(1-\bar{p})(1-e)}{\Pi_s \bar{T}_s + \Pi_c \bar{T}_c + \Pi_\sigma \sigma + \Pi_b \bar{T}_b} & = \quad h(b) \\ \frac{\Pi_b \bar{T}_b}{\Pi_s \bar{T}_s + \Pi_c \bar{T}_c + \Pi_\sigma \sigma + \Pi_b \bar{T}_b} & = \quad f_B \end{cases}$$

which can be solved numerically. Notice that, in the above two equations, channel state probabilities Π_s, Π_c, Π_σ, Π_b are analogous to those introduced in Section II-A for the case of a single outgoing link, this time using node average probabilities $\bar{\tau}$ and $\bar{p}$.

Once b and e are known, we can evaluate T_P and individual link throughputs T_{P_i} as

$$T_{P_i} = \frac{T_P\ s_i\ \tau_i\ (1-p_i)}{\sum_{j=1}^{n_L} s_j\ \tau_j\ (1-p_j)} \qquad 1 \leq i \leq n_L \qquad (4)$$

Standard equations of the M/M/1/B queue model allow us to also compute the buffer overflow probability, the average number of packets in the queue, and the average queuing delay $\bar{Q}$. To compute the average queuing delay Q_i experienced by a packet destined to link i, we add the average time spent in the queue before service (this component is common to all packets, and equal to $\bar{Q} - 1/\mu$) to the specific service time of a packet sent over link i, obtaining $Q_i = \bar{Q} - 1/\mu + 1/\mu_i$.

III. Model Validation

To validate our analytical model and demonstrate how it can be exploited to predict the end-to-end throughput and delay of multi-hop flows, we proceed in two steps. First, we show that the model accurately predicts individual link behavior under given traffic conditions. Second, we show how the model can be used to predict the performance of a new flow to be inserted on a given multi-hop path, i.e., under a hypothetical change in traffic conditions. In both steps of the validation, we directly measure via simulation the fraction of time f_B sensed busy by each node, and the packet loss probability p_i on each link. These quantities are then fed into the model

outlined in Section II. We first describe how the measurement of f_B and p_i can be performed locally by the nodes at low communication overhead, then we validate the model under current traffic conditions, and finally we assess the accuracy of the performance prediction of multi-hop flows under changing traffic conditions.

In all experiments we have used the *ns-2* [2] simulator with wireless extensions. These extensions to *ns-2* model channel contention, packet collision, channel capture and backoff, based on the specifications of IEEE 802.11. The radio propagation model uses the two-ray ground reflection path loss model for large scale propagation and a Ricean fading model for small scale propagation. Apart from using the standard IEEE 802.11b parameters, we used 11 Mb/s data rate, 30-packet node buffer size, 150 m maximum transmission range, 212 m maximum carrier sense range, and turned off RTS/CTS in all simulations.

A. Estimation of f_B and p

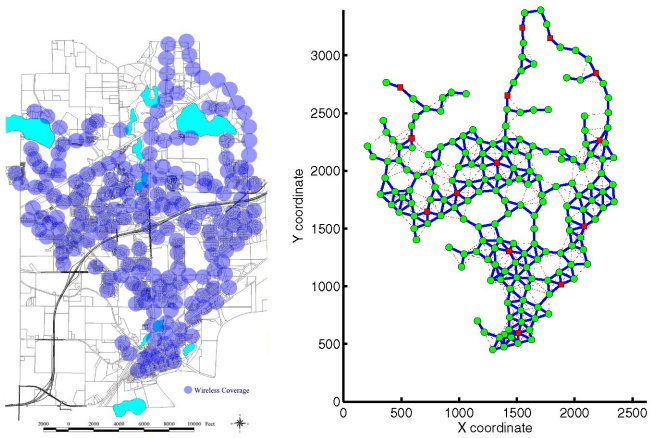
The fraction of busy time f_B provides a measure of the amount of air-time sensed busy by the node due to the activity of other nodes in its sensing range. Therefore, the estimation of f_B has to be done per-node, not per-link. In principle, the computation of f_B can be done through a passive measurement technique, thereby incurring zero communication overhead. The value of f_B can be very well approximated by the fraction of time that the activity of a node is suspended because the NAV timer is pending. Although this computation is simple to implement, current network-card drivers do not allow access to the NAV variable. Regardless, we assume that f_B can be computed locally with a more accessible device driver.

Estimation of the packet loss probability has to be done per-link, and requires active measurements in case a link is not currently used. Active measurement of link loss probability can be performed using either unicast or broadcast probes sent at low rate by the node, so as to minimize communication overhead. Unicast packets yield more accurate prediction with higher overhead in case a node has to monitor the quality of several links. Broadcast probe packets, originally proposed in [7], can be used to simultaneously measure the packet loss probability over all links, resulting in reduced communication overhead at the expense of decreased accuracy (broadcast probes have to be sent at the lowest data rate, thus can be subject to a different packet loss probability than unicast data packets). In our *ns-2* implementation, we use broadcast probes inserted at the head of the MAC queue to avoid undesired queuing delays. In practice, this can be accomplished by using the traffic classes provided by the 802.11e standard, as suggested in [9]. We also augment our measurement by counting the number of retries required for data packets being transmitted on each link. Existing device drivers such as MadWifi enable computation of this quantity.

B. Link estimation under current traffic conditions

We now show that our model can predict with high level of accuracy the performance of individual links under static traffic conditions, provided that the value of f_B and p are known. Notice that a node can estimate the performance of each of its outgoing links locally based on its measurement of f_B and p, i.e., there is no need to exchange information between neighboring nodes.

We consider the topology of the residential high-speed wireless mesh network commercialized by Chaska.net [1], which is reported in Figure 2. The network is comprised of 194 APs and we select 14 of these nodes as gateways.



(a) Chaska coverage map (b) Extracted Chaska topology

Fig. 2. Topology of the Chaska wireless mesh network. The coverage map (left) is taken from [1]. On the extracted topology (right), square nodes denotes gateways, while solid (dashed) edges connect nodes which are in transmission (sensing) range.

We first consider the case of downstream traffic alone. We assume that all gateways are saturated, and that the arrival rate of packets destined to each Access Point associated with a given gateway is the same. More specifically, the amount of application data arriving at each gateway is 5.6 Mb/s (large enough to saturate the queues of all gateways), equally divided among the APs. The payload of each data packet is constant, equal to 1000 bytes.

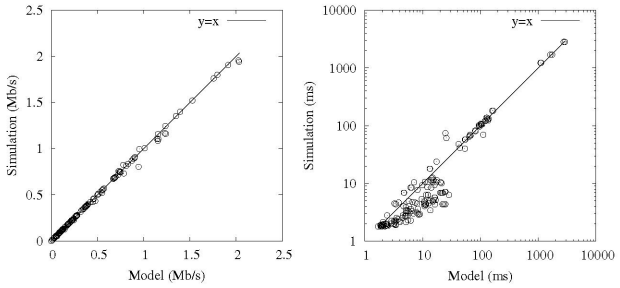


Fig. 3. Comparison between simulation and analysis for the throughput (left) and queuing delay (right) of each link, in case of saturated downstream traffic, with all flows active

Figure 3 depicts results for the case that all downstream flows are active (i.e., all APs receive traffic from the gateway they are connected to). We compare model and simulation results for the throughput and queuing delay on each traversed link in the network. The total amount of traffic successfully delivered to the APs is 31.46 Mb/s. We have also considered the case in which only half of the flows (randomly chosen) are active and found similar results.

We observe a good match between model and simulation results for the vast majority of links. Notice that, in Figure 3, in the case of downstream traffic, only a few links are congested (those directly attached to a gateway), while links that are

TABLE I

SUMMARY OF RESULTS FOR CURRENT NETWORK LOAD

scenario	throughput error	delay error
downstream, all flows	0.0098	0.674
downstream, half flows	0.0083	0.782
upstream, all flows	0.0305	0.215
upstream, half flows	0.0355	0.282

a few hops away from the gateway are lightly loaded and correspond to the clouds of dots with delay around 10 ms. Congested links exhibit, instead, a queuing delay in the order of hundreds of milliseconds. We argue that because of the use of a simple M/M/1/B queue model the delay prediction is not very accurate for lightly loaded links whereas the delay over the most critical, congested links is accurately estimated by the model.

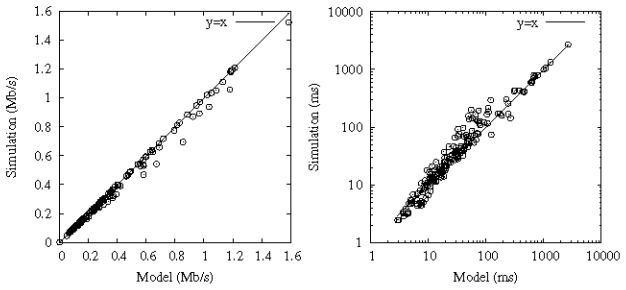


Fig. 4. Comparison between simulation and analysis for the throughput (left) and queueing delay (right) of each link, in case of aggregate upstream traffic per gateway of 2.4 Mb/s, with all flows active

Next, we consider the more critical case of upstream traffic. Here, several congested points may arise as traffic is pushed from the APs to the gateways. We assume that the aggregate amount of data sent to each gateway is 2.4 Mb/s, and that an equal amount of traffic is sent by each of the APs connected to a given gateway. Results of this experiment with all flows active are reported in Figures 4. We tried with only half the flows active and found similar results. The amount of traffic that arrives at the gateways is 25.16 Mb/s which is less than the aggregate amount of traffic sent by all Access Point, 33.6 Mb/s, indicating that the gateways are fully loaded.

We observe again a good match between model and simulation results for most of the links. The number of links experiencing delays in the order of 100 ms is much higher than in the case of downstream traffic, suggesting the presence of many congested points in the network. The delay prediction is more accurate than in the case of downstream traffic, because more links are congested.

Table I summarizes the accuracy of the model's predictions in the four scenario considered above. It contains, for each case, the mean relative error in the throughput prediction and the mean relative error in the delay prediction, averaged over all active links.

C. End-to-end estimation under changing traffic conditions

Here, we show how the model can be used to predict end-to-end throughput and delay of a multi-hop flow that is going to be added to the network. First, we describe a technique to compute the available throughput on a path based on the model in Section II. Then, we evaluate the accuracy of the proposed technique considering the Chaska topology.

The available bandwidth of a multi-hop flow over a path is the maximum throughput it can achieve subject to the condition that no queue along the path gets overloaded, i.e., the traffic intensity on each link is kept smaller than or equal to 1. Our estimation technique takes into account inter-flow and intra-flow interference separately and consists of two steps described below:

Inter-flow step: For each link l in the path, we first find the maximum additional input rate ϵ_l that can be added under the constraint that traffic intensity does not exceed 1. The maximum allowable value of ϵ_l is found by iteratively applying the model according to a binary search for the maximum value of λ_l. While doing so, we keep the measured values of f_B and p fixed relative to a node. This is indeed a crucial point of our technique because when we add traffic on a link, we perturb the network state. Our main approximation is thus to assume that f_B and p do not vary significantly when we add traffic on a single link without saturating the link.

Intra-flow step: The quantities ϵ_l computed in the first step provide an estimate of the throughput that each link could obtain if operated in isolation. However, the throughput available on a multi-hop path is smaller, because the links contend with each other. To account for self-interference, we proceed as follows. We construct an intra-flow link contention graph that captures interference relationships among the links of the path. Each vertex in the contention graph corresponds to a link. An edge exists between two vertices if they contend with each other. Each clique j of the contention graph provides a constraint on the maximum achievable throughput F over the path in the form $\sum_{l \in j} \frac{F}{\epsilon_l} \leq 1$, i.e., the sum of the normalized time shares required to send the amount of traffic F over each link of the clique cannot exceed 1. It follows that $F \leq F_j$, $\forall j$, where $F_j = \sum_{l \in j} (\frac{1}{\epsilon_l})^{-1}$. Finally, the path available throughput is computed taking the minimum among all F_j. Figure 5 illustrates an example of computation of the available path throughput in a simple case.

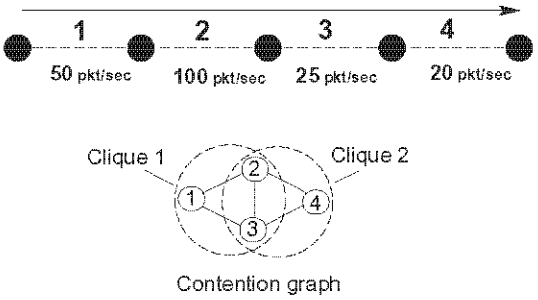
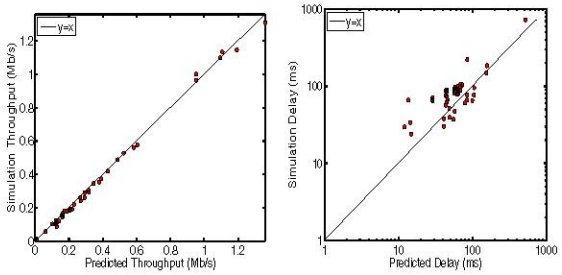


Fig. 5. A 5-hop flow and its intra-flow contention graph. Dotted lines denote interference range and dotted circles denote cliques. The path available throughput is $\min\{(\frac{1}{50}+\frac{1}{100}+\frac{1}{25})^{-1}, (\frac{1}{100}+\frac{1}{25}+\frac{1}{20}\})^{-1}\} = 10$ pkts/sec.

D. Accuracy of the throughput estimation technique

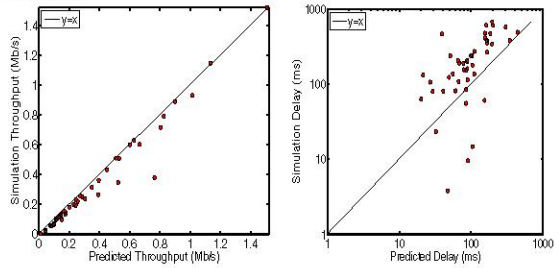
To validate our technique to estimating the available throughput over a multi-hop path, we consider the Chaska topology with the network already loaded with 100 flows. We separately consider upstream and downstream traffic and set the total amount of traffic sent to or from each gateway to 2 Mb/s, equally divided among the flows associated to it. Then we insert one additional flow, randomly chosen among those not already present in the network, and we repeat this

experiment 50 times. The rate of the new flow is limited to the value computed by our technique. Results for the case of downstream and upstream traffic are shown in Figures 6 and 7, respectively. Despite several approximations, our technique is able to obtain a good match with simulation, especially for the throughput prediction.



(a) Throughput (b) Delay

Fig. 6. Download scenario: Comparison of simulation and analytical results in case of 2Mbps traffic, with 100 flows initially active and 50 additional multi-hop flows.



(a) Throughput (b) Delay

Fig. 7. Upload scenario: Comparison of simulation and analytical results in case of 2Mbps uplink traffic, with 100 flows initially active and 50 additional multi-hop flows.

IV. Evaluation of routing metrics

Several link-cost routing metrics have been proposed in the literature with the goal of finding high throughput paths in 802.11 mesh networks. Most are primarily based on the packet loss probability measured on the links. In this section, we use our model to evaluate the performance of existing routing metrics in congested networks. After introducing the metrics, we show through simple scenarios and graphs that they can fail to discover high-throughput paths, essentially because they rely on link quality measures that provide only partial, incomplete information about the throughput achievable on a given path. In contrast, we show that there is significant space for improvement if we directly estimate the end-to-end throughput over a path, which can be done using a model-based approach such as the one described in this paper.

In an effort to understand fundamental properties that govern the correlation of routing metrics and end-to-end throughput in wireless mesh networks, we focus on the baseline case of single channel, single data rate, and UDP traffic. Our model can also be used to study the case of different, fixed per-link data rate through the incorporation of appropriate packet transmission duration, or study the case of multi-radio systems where each radio executes a separate instance of 802.11 MAC protocol on a different channel. The baseline case is a necessary first step before considering more heterogeneous network scenarios, dynamic data rate adjustment mechanisms such as Autorate Fallback, as well as the role of transport protocols such as TCP.

A. Routing metrics

The Expected Transmission Count (ETX) metric, originally proposed in [7], computes for each link l the average number of transmission attempts ETX_l required to send successfully a packet over the link:

$$\text{ETX}_l = \frac{1}{1 - p_l} \tag{5}$$

where p_l is the packet loss probability at the MAC layer, that can be estimated locally using for example the broadcast probes technique.

The Expected Transmission Time (ETT) metric [9] is an improved version of ETX, that also takes into account the packet size S_l and the data rate B_l used on the link:

$$\text{ETT}_l = \text{ETX}_l \times \frac{S_l}{B_l} \tag{6}$$

The Weighted Cumulative ETT (WCETT) metric [9] improves ETT by introducing an additional (weighted) component related to the channel diversity of the path. We do not further consider this metric because we focus on single-channel networks.

The Interference-aware Resource Usage (IRU) metric [16] is defined as:

$$\text{IRU}_l = \text{ETT}_l \times N_l \tag{7}$$

where N_l is the set of neighbors that the transmission on link l interferes with. Notice that, in all metrics above the only dynamic variable is the packet loss probability while all other quantities depend only on network topology, hence not affected by network load. Therefore, we will refer to them as the class of *loss-based* metrics. Moreover, with a single-rate and constant packet-size, all metrics assign to the links a cost that is proportional to the ETX metric. Even the IRU metric, that includes the number of interfering nodes, is proportional to ETX when the node density is homogeneous.

Being based on packet loss probability, all the above metrics depend significantly on the network load. Indeed, in wireless mesh networks, packet loss probabilities can be very large (e.g., larger than 0.5) even under medium load, due to a variety of problems related to the MAC protocol itself (such as hidden terminals). Under many circumstances, losses induced by channel contention can be much higher than those due to channel corruption (fading, shadowing, etc) [11]. Therefore, *loss-based* metrics are significantly affected by network load, although they are expected to provide more stable link costs (e.g., subject to less random fluctuations) than other load-sensitive metrics such as those based on packet-pair techniques [8] or RTT measurements [3].

We conclude that the performance of *loss-based* metrics should be assessed within the larger family of load-sensitive metrics. The fundamental question then is the following: within the class of load-sensitive metrics, how effective are *loss-based* metrics at finding high-throughput paths? We address this question in the following sections, with the help of both our model and simulation experiments.

B. Performance of a single link

We first consider the performance of a single link. According to *loss-based* metrics, the cost of a link is proportional to

the expected number of retransmissions to successfully send a packet (ETX). The service time on the link is implicitly assumed proportional to the number of retransmission attempts. Hence, the maximum throughput on the link (the inverse of the service time) is inversely proportional to ETX and can be expressed as $T_P(p) = T_{max}(1-p)$, where T_{max} is the maximum achievable link throughput (when $p = 0$).

Instead, according to our model in Section II, the throughput on a link depends on three variables: the packet loss probability p, the fraction of busy time f_B and the average duration of a busy period $\bar{T}_b$.

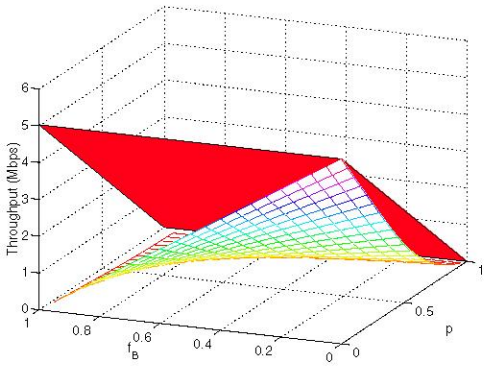


Fig. 8. The throughput of a single link as a function of the fraction of busy time f_B and of the packet loss probability p, according to the model (grid surface) and *loss-based* metrics (solid surface)

Figure 8 depicts the throughput of a single link as a function of f_B and p, according to our model[2] and according to *loss-based* metrics. We observe that they differ considerably. In particular, *loss-based* metrics neglect the impact of f_B and assume that the throughput decreases linearly with p. Instead, the throughput depends on p in a non-linear way due to the exponential backoff mechanism of 802.11. More importantly it decreases linearly with f_B. This can be explained by the fact that $1 - f_B$ is essentially the air-time available to the link transmitter.

Thus, for the simplest case of a single link we have a first clear indication that the throughput estimation according to *loss-based* metrics can deviate substantially from the actual behavior of 802.11.

C. Single path performance

We investigate the throughput performance of loss-based metrics for a single multi-hop flow in isolation, i.e., a single path without any interference from other nodes outside the path. It is well known that the throughput along a single path suffers from intra-flow interference. A typical behavior as a function of the number of hops is illustrated in Figure 9, where we have simulated a backlogged source, standard 802.11b parameters, no RTS/CTS, and nodes equally spaced apart so that any node is in range of just its predecessor and successor nodes. As the number of hops increases, the throughput exhibits a quick drop and then stabilizes to a value of approximately 1.2 Mb/s. A similar qualitative behavior is observed under different configurations such as node spacing, unequal sensing and transmission ranges, use of RTS/CTS, etc.

[2]It turns out that the throughput as predicted by our model is almost insensitive to $\bar{T}_b$, which plays a minor role with respect to f_B. Thus, in Figure 8 we set $\bar{T}_b = \bar{T}_s$, and focus only on the impact of f_B and p.

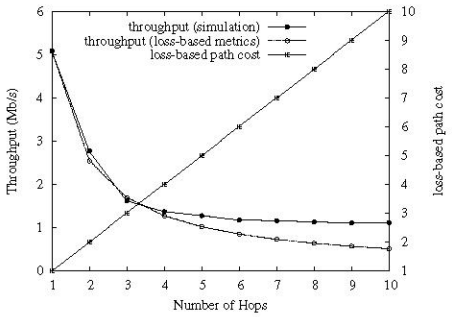


Fig. 9. Throughput over a single path as a function of the number of hops, according to simulation and *loss-based* metrics

The right-side y-axis of Figure 9 reports the path cost of *loss-based* metrics when only broadcast probes exist without any data traffic injected over the path. In this case, the packet loss probability experienced by the probe packets is close to zero on each link. Hence, all loss-based metrics yield a path cost proportional to the number of hops.

Analogous to the single link case, we also report the throughput prediction according to *loss-based* metrics, i.e., the inverse of the path cost. We observe that this quantity represents well the throughput available on the chain for number of hops less than 4. However, longer paths that would in reality provide a stable throughput of 1.2 Mbps, are increasingly penalized by *loss-based* metrics as the number of hops increases.

D. Path selection performance

We now investigate the performance of *loss-based* metrics in a congested network in which multi-hop flows contend with each other. We focus on the simplest case where there exist only two alternative independent paths and show that *loss-based* metrics can indeed lead to a sub-optimal decision: select a path providing significantly lower throughput than the alternate path.

A typical situation encountered in the large mesh network of Figure 2 is the generic scenario depicted in Figure 10. Here, mesh AP $\mathbf{A}$ must select a path either toward gateway $\mathbf{G_1}$ or an independent path toward gateway $\mathbf{G_2}$. The number of hops between $\mathbf{A}$ and $\mathbf{G_1}$ or $\mathbf{G_2}$ is variable and possibly different. In addition, both nodes $\mathbf{G_1}$ and $\mathbf{G_2}$ can be loaded with upstream and/or downstream flows. We investigate the throughput loss

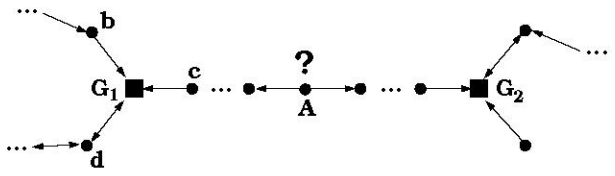


Fig. 10. Generic path selection problem: mesh AP $\mathbf{A}$ must send traffic to either gateway $\mathbf{G_1}$ or $\mathbf{G_2}$.

due to sub-optimal routing decisions of *loss-based* metrics by evaluating the individual impact of two critical parameters: high packet loss and high busy time fraction.

Impact of high packet loss. Figure 11 depicts an instance of the generic path selection problem of Figure 10. With respect to gateway $\mathbf{G_1}$, AP $\mathbf{A}$ has a two-hop path through

relay node $\mathbf{B}$. The path load is induced by AP $\mathbf{C}$ which has established a single-hop UDP upload flow to gateway $\mathbf{G1}$ and can create high packet loss to link $\mathbf{A}-\mathbf{B}$ because it is hidden from $\mathbf{A}$. With respect to gateway $\mathbf{G}_2$, $\mathbf{A}$ has a path with a variable number of hops and no interference is induced by other flows. In this case, the available bandwidth and path cost are only a function of the number of hops as in Figure 9.

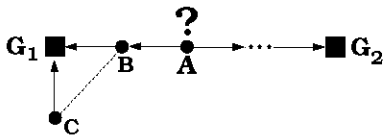


Fig. 11. Example topology used to derive results in Figure 12

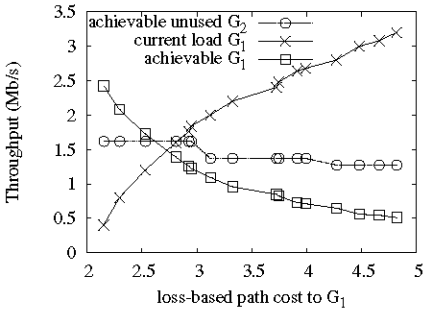


Fig. 12. Load on gateway $\mathbf{G}_1$ and achievable throughput on gateways $\mathbf{G}_1$ and $\mathbf{G}_2$ as a function of the *loss-based* path cost to $\mathbf{G}_1$, for the example topology of Figure 11

We now perform a simulation experiment focusing on path $\mathbf{A}-\mathbf{G}_1$. We gradually increase the load on link $\mathbf{C}-\mathbf{G}_1$, and for each load value we measure: i) the *loss-based* path cost from $\mathbf{A}$ to $\mathbf{G}_1$; ii) the maximum throughput $\mathbf{A}$ would achieve if it selected the path to gateway $\mathbf{G}_1$.

Figure 12 depicts three quantities as a function of the measured path cost $\mathbf{A}-\mathbf{G}_1$. The first two quantities are the (input) load on link $\mathbf{C}-\mathbf{G}_1$ and the maximum achievable throughput of path $\mathbf{A}-\mathbf{G}_1$. The third quantity is the maximum achievable throughput on the alternate path $\mathbf{A}-\mathbf{G}_2$, given that this path would not be selected due to higher cost; more specifically, for each value c_1 of the measured path cost $\mathbf{A}-\mathbf{G}_1$, we report the throughput of Figure 9 that corresponds to a path $\mathbf{A}-\mathbf{G}_2$ of cost $\lceil c_1 \rceil$ hops.

As expected, the path cost from $\mathbf{A}$ to $\mathbf{G}_1$ increases as the load on link $\mathbf{C}-\mathbf{G}_1$ increases. However, the selection of gateway $\mathbf{G}_1$ resulting from *loss-based* metrics can lead to significantly less throughput than the achievable throughput on the path to gateway $\mathbf{G}_2$. Indeed, paths toward $\mathbf{G}_2$ with less than 4 hops can provide from 0 to 100% more throughput than that obtained toward $\mathbf{G}_1$. Longer paths can provide even higher gains but could be more difficult to find.

We also observe that, no matter how high the load on $\mathbf{G}_1$ and the corresponding path cost are, there can always be a long enough path toward $\mathbf{G}_2$ not selected by *loss-based* metrics while providing higher throughput, at least equal to 1.2 Mb/s, the stable path throughput in Figure 9.

Impact of high busy time fraction. Figure 13 depicts an instance of the generic path selection problem of Figure 10 where the *loss-based* path cost and throughput $\mathbf{A}-\mathbf{G}_1$ are mainly affected by the high fraction of busy time sensed at intermediate node $\mathbf{B}$ due to the activity of two independent links $\mathbf{C}-\mathbf{G}_1$ and $\mathbf{D}-\mathbf{G}_3$ within its sensing range.[3]

Focusing again on path $\mathbf{A}-\mathbf{G}_1$, we repeat the experiment on the scenario of Figure 13. We equally increase the loads of links $\mathbf{C}-\mathbf{G}_1$ and $\mathbf{D}-\mathbf{G}_3$ and measure the *loss-based* path cost and the maximum achievable throughput to $\mathbf{G}_1$. The results are reported in Figure 14, which also depicts the achieved throughput for a higher path cost to gateway $\mathbf{G}_2$.

We observe that the throughput loss due to the sub-optimal routing decision is higher than in Figure 12. The reason is that the throughput reduction due to high fraction of busy time sensed by node $\mathbf{B}$ is not captured by *loss-based* metrics. This additional penalty on the path to $\mathbf{G}_1$ is not included in the *loss-based* path cost. As a result, an alternate non-chosen path to $\mathbf{G}_2$ of 4 hops could provide 200 % higher throughput, i.e., the throughput loss due to *loss-based* metrics is approximately twice the throughput loss observed in Figure 12.

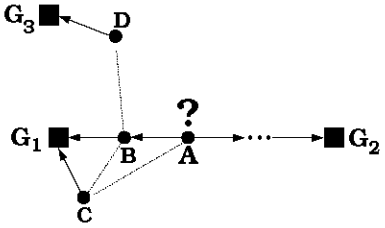


Fig. 13. Example topology used to derive results in Figure 14

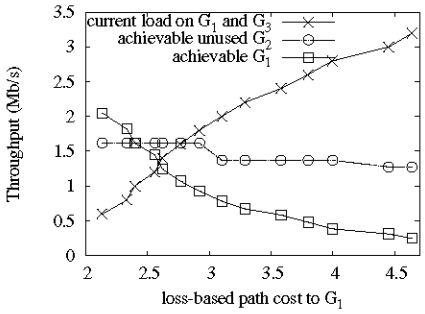


Fig. 14. Load on gateways $\mathbf{G}_1$ and $\mathbf{G}_3$ and achievable throughput on gateways $\mathbf{G}_1$ and $\mathbf{G}_2$ as a function of the corresponding path cost to gateway $\mathbf{G}_1$, for the example topology of Figure 13

We conclude that *loss-based* metrics can fail to discover high-throughput paths, possibly leading to severe throughput losses, in the order of 100% or more. Indeed, the packet loss probability on the links forming a path provides very partial information about the achievable throughput on the path. Thus, a routing strategy based on this information can select highly suboptimal paths.

V. Performance Evaluation

In this section we compare the performance of our model-based metric for available bandwidth estimation with the performance of existing loss-based metrics. As in Section III, we used the *ns-2* simulator for this evaluation. We first outline in Section V-A the design of a routing protocol based on our metric. In Section V-B, we compare the performance of the different metrics in various network scenarios.

[3]The reader is referred to [11] for an explanation of why node $\mathbf{B}$ will sense a large fraction of busy time.

A. The Routing Protocol

Our available bandwidth metric is meant to be used in combination with a proactive routing protocol similar to LQSR [9]. We emulated the behavior of a link state protocol using a centralized routing database that is instantaneously updated whenever nodes provide new measurements of f_B and p, link ETX, link IRU, etc. In particular, measurements are updated using the average of the samples collected during the last 20 seconds. When a source node **S** needs to send a packet to a destination node **D**, **S** retrieves a route from the centralized database and puts the entire path in the packet header. Intermediate nodes only need to relay packets based on the source-routing information carried in the packet.

We use a modified version of Dijkstra's shortest path algorithm. For Dijkstra's shortest path algorithm to be applicable, the cost of a link should be strictly positive so that the cost of a path (route), which is defined to be the sum of the cost of the constituent links, increases monotonically with the increase in the number of constituent links. For any route, since the loss-based metrics, i.e., IRU and ETX, increase monotonically with the number of constituent links, the routing protocol can use the standard version of Dijkstra's shortest path to select routes.

However, when our proposed model-based metric is used, the routing protocol uses a modified version of Dijkstra's shortest path algorithm for route selection. In this modified version, the cost of a link is the available bandwidth from the source to the destination of the link, and the cost of any path is defined as the maximum available bandwidth between the source and destination. The available bandwidth over a path, computed according to the technique described in Section III-C, is guaranteed to monotonically decrease with the hop count. Hence, we can use a modified Dijkstra's algorithm, shown in Algorithm 1, to efficiently compute the available bandwidth from a source node to all other nodes in the network.

Algorithm 1 Modified Dijkstra's shortest path algorithm to search for path with maximum available bandwidth.

Require: N: Total nodes.
Require: s: Source node.
Require: Adjacent[u]: Set of all nodes adjacent to u.
  for $i = 1$ to N **do**
    $available[i] = 0$;
    $previous[i] = null$;
  end for
  $available[s] = \infty$; /* available bandwidth from s to s */
  $\mathcal{S} \leftarrow \emptyset$; /* initialize set of visited nodes */
  $\mathcal{Q} = 1, 2, ..., N$; /* set of all unvisited nodes */
  while $\mathcal{Q}$ is not empty **do**
    $u \leftarrow$ Extract_Max($\mathcal{Q}$) ;
    $\mathcal{S} \leftarrow \mathcal{S} \cup u$;
    for all $v \in$ Adjacent[u] **do**
      $B =$ **CliqueBandwidth(u,v)** ;
      if $available[v] < B$ **then**
        $available[v] = B$;
        $parent[v] \leftarrow u$;
      end if
    end for
  end while

The function "CliqueBandwidth(u, v)" computes the available bandwidth from the source node s to node v using the newly added node u as the previous hop. The available bandwidth computation is done using the intra-flow clique-based procedure of Section III-C over a single path $s \rightarrow v$, formed by link (u, v) and the path defined by the parent[] entries of u and its parent nodes back to source s.

B. Results

For our first set of experiments, we consider the Chaska topology in Figure 2. We randomly pick 100 nodes and start an upstream flow to its nearest gateway according to minimum-hop count. The traffic load sent by each source node is chosen as a uniform random value between 0 and $2/m$ Mb/s, where m is the number of flows sending traffic to the same gateway. By doing so, we randomize the load on the 14 gateways present in the network. After allowing these 100 flows to run for 30 seconds, we randomly choose another node, different from the 100 existing sources, and start an upstream flow from this node with the freedom to choose the destination gateway. For the new flow we compute the best route to a gateway according to ETX, IRU, and our metric, that we call AVAIL; once computed, the flow uses the same route for the rest of the simulation.

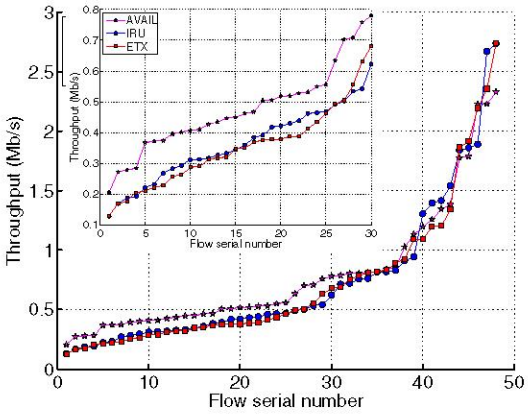
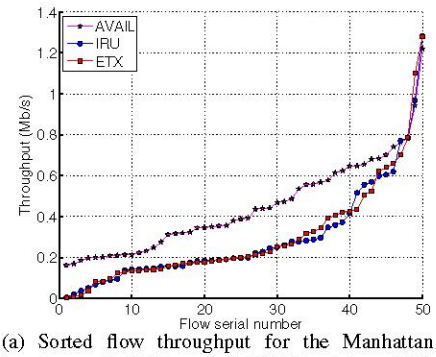


Fig. 15. Sorted flow throughput for the Chaska topology. Since the topology does not allow many alternate independent paths from AP's to gateways, all the metrics perform similarly.

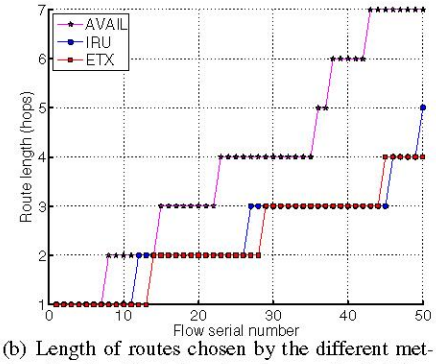
Having selected the route, irrespective of the routing metric used, we use our model to predict the available bandwidth along the chosen route and rate-limit the newly added upstream flow to the predicted bandwidth value. This is done for the sake of a fair comparison with ETX and IRU, since neither of them can be used to compute the sustainable sending rate of the new UDP stream. For each metric, we repeat the experiment 50 times, each experiment having the same initial 100 flows but different source nodes from which to start the new flow to be added in the network. We measure the throughput achieved by the new flow in each run, and sort the 50 values in increasing order.

Figure 15 indicates that the AVAIL metric provides a gain with respect to the other metrics, especially for lower throughput values, as shown in the insert of Figure 15. However, in all cases the gain is small. The reason is that the simulated Chaska topology is composed of almost disconnected, dense clusters of nodes, hence it provides low spatial reuse and a limited number of independent paths towards distinct gateways.

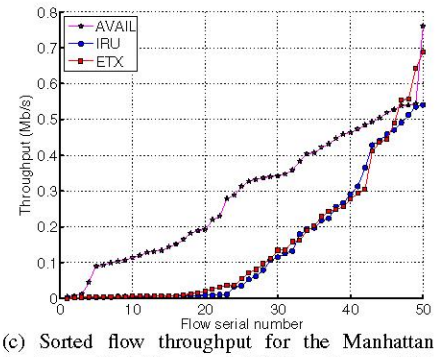
We next consider a richer and structured topology providing more spatial reuse and degrees of freedom in the selection of routes. More specifically, we consider a manhattan network of



(a) Sorted flow throughput for the Manhattan topology with 3 Mbps load. AVAIL achieves 49% (51%) higher throughput than ETX (IRU) on average.



(b) Length of routes chosen by the different metrics for the Manhattan topology with 3 Mbps maximum load.



(c) Sorted flow throughput for the Manhattan topology with 4 Mbps load. AVAIL achieves 108% (127%) higher throughput than ETX (IRU) on average. Also, under ETX and IRU half of the flows starve.

Fig. 16. Manhattan topology

196 nodes arranged in a 14 x 14 grid. Each node is within sensing and transmission range of only its North, South, East, and West neighbors. We randomly select 10 of these nodes to act as gateways[4]. The load on each gateway is uniformly distributed between 30% and 100% of a preset maximum and this load is equally distributed among the flows ending at the gateway. We run two sets of experiments for two values of this preset maximum, namely, 3 Mb/s, shown in Figure 16(a) and 4 Mb/s, shown in Figure 16(c). Figure 16(b) reports the length of the routes selected by the different metrics in the 3 Mb/s case.

Since the topology provides a rich selection of independent paths the AVAIL metric is able to find routes with considerably higher throughput than the routes chosen by ETX or IRU. In Figure 16(a), AVAIL achieves 49% (51%) higher throughput with respect to ETX (IRU) on the average. As shown in Figure 16(b), AVAIL typically selects longer routes than those chosen by IRU or ETX. This is because AVAIL does a better job at finding longer routes with higher available bandwidth than shorter routes. The throughput gain of AVAIL increases when the offered load in the network is increased. In Figure 16(c), AVAIL achieves 127% (108%) higher throughput with respect to ETX (IRU) on the average. More importantly, under *loss-based* metrics almost half the flows receive close to zero throughput while AVAIL is able to identify non-starving paths.

VI. Conclusions

We proposed a novel technique to discover high throughput paths in a congested 802.11 mesh network. The novelty of our approach lies in the direct computation of the end-to-end available bandwidth through the use of an accurate analytical model of 802.11, that we have extended to the case in which nodes send to multiple receivers. We have also shown how to leverage our technique within a distributed routing protocol which exploits local measurements performed by the nodes to effectively route newly added flows, achieving significant gains with respect to existing routing metrics. We plan to extend our approach to study the coupling of mesh routing protocols with dynamic data rate adjustment mechanisms such as AutoRate Fallback and congestion control protocols such as TCP.

[4]In a real deployment the flexibility of placing gateways might be limited since gateway nodes typically require a dedicated link to the wired backbone.

VII. Acknowledgements

This research was supported by NSF Grant CNS-0325971 and by a grant from Cisco.

References

[1] Chaska.net, Residential High Speed Wireless Internet Access. `http://www.chaska.net`.

[2] The Network Simulator, ns-2. `http://www.isi.edu/nsnam/ns/`.

[3] A. Adya, V. Bahl, J. Padhye, A. Wolman, and L. Zhou. A Multi-Radio Unification Protocol for IEEE 802.11 Wireless Networks. In *IEEE Broadnets*, San Jose, CA, USA, October 2004.

[4] B. Awerbuch, D. Holmer, and H. Rubens. The Medium Time Metric: High Throughput Route Selection in Multirate Ad Hoc Wireless Networks. *Kluwer Mobile Networks and Applications (MONET) Journal, Special Issue on Internet Wireless Access: 802.11 and Beyond*.

[5] G. Bianchi. Performance analysis of the IEEE 802.11 distributed coordination function. *IEEE Journal on Selected Areas in Communications*, 18(3):535–547, Mar. 2000.

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

[7] D. S. J. D. Couto, D. Aguayo, J. Bicket, and R. Morris. A high-throughput path metric for multi-hop wireless routing. In *Proc. ACM MobiCom*, San Diego, CA, USA, September 2003.

[8] R. Draves, J. Padhye, , and B. Zill. Comparison of routing metrics for static multi-hop wireless networks. In *Proc. ACM SIGCOMM*, Philadelphia, PA, USA, Sept. 2004.

[9] R. Draves, J. Padhye, and B. Zill. Routing in Multi-Radio, Multi-Hop Wireless Mesh Networks. In *Proc. ACM MobiCom*, Philadelphia, PA, USA, Sept. 2004.

[10] Y. Gao, D. Chiu, and J. Lui. Determining the end-to-end throughput capacity in multi-hop networks: methodology and applications. In *Proc. ACM SIGMETRICS*, Saint-Malo, France, 2006.

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

[12] M. Garetto, J. Shi, and E. Knightly. Modeling media access in embedded two-flow topologies of multi-hop wireless networks. In *Proceedings of ACM MobiCom '05*, Cologne, Germany, Sept. 2005.

[13] 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*, Barcelona, Spain, April 2006.

[14] X. Wang and K. Kar. Throughput modelling and fairness issues in CSMA/CA based ad-hoc networks. In *Proc. IEEE INFOCOM*, Miami,FL,USA, March 2005.

[15] Y. Yang and R. Kravets. Contention-Aware Admission Control for Ad Hoc Networks. *IEEE Trans. on Mobile Computing*, 4(4):363–377, 2005.

[16] Y. Yang, J. Wang, and R. Kravets. Designing routing metrics for mesh networks. In *Proceedings of WiMesh*, Santa Clara, CA, Sept. 2005.