Joint MAC-aware Routing and Load Balancing in Mesh Networks

Vivek Mhatre[*]
vivekmhatre@alcatel-lucent.com

Francois Baccelli[†]
francois.baccelli@ens.fr

Henrik Lundgren[‡]
henrik.lundgren@thomson.net

Christophe Diot[‡]
christophe.diot@thomson.net

[*]Bell Labs, Bangalore, India
[‡]Thomson Research Lab, Paris, France
[†]ENS-INRIA, Paris, France

ABSTRACT

Past approaches to routing in mesh networks either (i) do not account for the MAC-layer interactions between the links in a tractable manner, or (ii) are agnostic to load-balancing across gateways. Our answer to these problems is MaLB (MAC-aware and Load Balanced routing algorithm), a greedy, tractable, and distributed mesh routing algorithm. Since the underlying objective function has high combinatorial complexity, MaLB uses a greedy approach. MaLB finds an optimum routing forest (union of trees rooted at the gateways) by taking into account MAC-layer interaction between links, as well as optimum multi-hop association of mesh nodes to gateways. MaLB builds on top of ETP (Expected ThroughPut), a recently proposed MAC-aware routing metric. We also propose a low complexity variant of MaLB called LB (Load Balanced routing algorithm) which performs load balancing in a MAC-agnostic manner. MaLB performs especially well in networks with skewed topologies that result from unplanned mesh network deployment, as well as in the presence of gateway failures. Simulations with an enhanced version of ns-2 show that MaLB results in up to 60% higher throughput than a shortest path algorithm with ETX (Expected Transmission Count). Furthermore, MaLB results in up to 30% improvement over the LB algorithm, as well as a shortest path algorithm with ETT (Expected Transmission Time).

1. INTRODUCTION

Wireless mesh networks provide a cost-effective solution to expand the coverage of WiFi access points. In such networks, a subset of nodes, referred to as *gateways*, have direct connectivity to the Internet. The rest of the nodes, referred to as *mesh nodes*, use multi-hop communication to share the Internet connectivity of the gateways. Examples of such networks are community mesh networks such as MIT Roofnet [1] and Rice Mesh Network [2].

Routing algorithms for mesh network have to be robust to node failures, and link quality variations due to packet loss, and shadow-fading. It has been shown in [3] that in addition to the above factors, the MAC-layer interactions also play a significant role in determining the capacity of a link. In particular, depending on the MAC-layer contention, the goodput of a link can be substantially lower than its bit rate. Past work has primarily focused on defining routing metrics to determine high throughput routes. Examples of such routing metrics are MAC-agnostic routing metrics, such as ETX [4], ETT [5], and MAC-aware routing metrics, such as ETP[6], EDR [7], MIC [8].

When shortest path routing is used with any of the above routing metrics, the resulting routing algorithms do not support load balancing[1]. This is because the load-agnostic behavior of shortest path algorithms can result in scenarios where a few gateways have too many associated mesh nodes, while other gateways are underutilized. Further load imbalance in the network could be caused by gateway failures. 100% uptime for gateways is not possible, since the connection of the gateways to the Internet may fail occasionally, or the gateways may be unreachable [9]. Consequently, failure of one or more gateways can lead to overloading of a select few gateways. Load balancing and routing algorithms recently proposed in [10] to handle such load imbalance do not take the MAC-layer interaction into account. *The novelty of our approach is that we jointly study the problems of MAC-aware routing and load balancing.*

In order to incorporate the MAC-interactions between the links, we build on top of the ETP routing metric [6]. ETP is a MAC-aware routing metric which takes into account the capacity reduction of a link due to

[1] Load balancing in the context of this paper refers to balancing the number of mesh nodes communicating with different gateways. We do not consider traffic splitting via multipath routing.

*CoNEXT'*07, December 10-13, 2007, New York, NY, U.S.A.
Copyright 2007 ACM 978-1-59593-770-4/ 07/ 0012 ...$5.00.

its interaction with other links in its contention domain. However, such a cross-layer approach faces the following challenge. With routing metrics, such as, hop count, ETX, and ETT, routes depend solely on the link weights, i.e., the link qualities. On the other hand, in MAC-aware routing metrics, such as ETP, the link quality in turn depends on the choice of routes. This is because the choice of routes determines the set of active links in the network. This in turn determines the contention domain, and therefore the MAC-layer capacity reduction of each link. Thus, routing has an impact on the link weights. Due to the above inter-dependence of MAC and routing, the problem of finding optimal routes has high combinatorial complexity.

To this end, we make the following contributions in this paper. We propose a greedy distributed routing algorithm, MaLB (MAC-aware and Load Balanced routing) that works around the tractability problem arising from the interaction between MAC and routing, and jointly solves the problem of MAC-aware routing and load balancing. Instead of using shortest path routing, MaLB determines a delay optimal routing forest (union of disjoint trees rooted at the gateways). We also propose a low complexity variant of MaLB, referred to as LB (Load Balancing) which performs MAC-agnostic load balancing. MaLB trades off optimality for tractability, and hence can only guarantee convergence to a local optimum. Despite this, we demonstrate that MaLB results in significant throughput improvements over a shortest path routing algorithm with ETX and ETT as routing metric, as well as the LB routing and load balancing algorithm.

Simulations performed following the guidelines provided in [11], and using our enhanced version of ns-2, show that for balanced topologies, the performance of MaLB is similar to other routing algorithms considered. However, for skewed topologies which often result from gateway failures [9], as well as unplanned gateway deployment [1], MaLB results in up to 60% improvement in throughput over shortest path routing with ETX, and up to 30% improvement over LB, and over shortest path routing using ETT.

The rest of this paper is organized as follows. Related work is discussed in Section 2. In Section 3, the analytical framework used in the routing problem formulation is presented. The MaLB and LB routing algorithms are discussed in Section 4. A simulation based comparative study of the proposed algorithms is presented in Section 5. Finally, conclusions are future extensions are presented in Section 6.

2. RELATED WORK

Several routing metrics have been proposed in the literature to choose high throughput routes in wireless mesh networks. In [4], the authors propose Expected Transmission Count (ETX) routing metric. This metric is designed to avoid links with high packet losses. In [5], the authors build on top of the ETX routing metric, and define the Expected Transmission Time (ETT) routing metric. ETT estimates the quality of a link by taking into account the packet loss probability as well as the bit rate of the link. Since 802.11 supports multi-rate links, this metric tends to prefer high bit rate links over low bit rate links. Furthermore, if the network has nodes with multiple radios, then the WCETT path metric function defined using ETT chooses paths that have higher spatial reuse.

Expected Data Rate (EDR) defined in [7] aims to account for MAC layer contention between the links *of a given path*. Expected Throughput (ETP) defined in [6] also takes into account MAC layer contention between links of a given path. EDR and ETP focus on the intra-flow contention, and do not consider MAC contention between different flows. MIC, or Metric of Interference and Channel Switching [8] is a MAC-aware routing metric that accounts for intra-flow as well as inter-flow MAC contention. However, MIC assumes that the wireless channel is equally shared by all the nodes irrespective of the link bit rates, a assumption that does not hold in general in multi-rate networks [3]. Furthermore, since MIC is non-isotonic [12] it requires a complex routing protocol based on virtual nodes (referred to as LIBRA in [8]). All the above routing metrics are used in conjunction with some variant of shortest path routing. Since shortest path routing can lead to scenarios in which mesh nodes may be associated in a sub-optimal fashion to the gateways, the combination of these routing metrics with shortest path routing does not provide any load balancing guarantees.

In [10], the authors study the problem of load balancing in mesh networks. However, the graph theoretic model considered in [10], and other works on flow control over trees [13] do not incorporate the effect of bandwidth sharing at the MAC layer.

Our work differs from the past work in that we *jointly* address the problems of MAC-aware routing and load balancing, and provide a distributed routing algorithm to solve these problems.

3. ANALYTICAL MODEL

In this section, we present the analytical framework for MAC-aware and load balanced routing. Within this framework, we formulate an optimization problem for determining a delay-optimal routing forest. In Section 4, we propose routing and load balancing algorithms to solve the proposed optimization problem.

Consider a mesh network in which $\mathcal{G}$ is the set of gateways that have direct connectivity to the Internet, and $\mathcal{M}$ is the set of mesh nodes. Each mesh node $i \in \mathcal{M}$ connects to a single gateway. Nodes are organized in disjoint[2] trees T_{g_i}, each tree rooted at a gateway node $g_i \in \mathcal{G}$. Let $\mathcal{T}$ be the union of all these trees. Thus, $\mathcal{T}$ corresponds to one possible choice of routing. The subtree rooted at mesh node $i \in \mathcal{M}$ is denoted by T_i. Note that T_i includes node i. All the nodes have a single radio for mesh communication, and operate over a common channel similar to the mesh network in [1]. The latter assumption is used because if a single radio is available for mesh communication, monitoring of link qualities over multiple channels is difficult. We assume that another radio is available at each mesh node for communication with its clients over an orthogonal channel.

Consider the example topology shown in Fig. 1, where the triangles represent gateway nodes g_1 and g_2, and the circles represent mesh nodes. The tree rooted at each gateway node is shown with solid lines. The dashed lines show those communication links between the node that are not used in the routing tree. Let l_i be the link used by mesh node i to associate (along with its subtree T_i) to its parent node in the routing forest. All the nodes in the subtree of node i are contained in the dotted ellipse. We use l_i as the state variable of node i. *Note that under the assumption of single-path routing, all the routes in the network are completely determined if for all mesh nodes i, their state variable l_i, i.e., their link of attachment to their respective parent is known.* Let P_{l_i} be the probability that a packet transmission on link l_i is successful. This probability includes the packet success probability in both directions to account for data as well as ACK frames [4]. Let R_{l_i} be the nominal bit rate of link l_i.

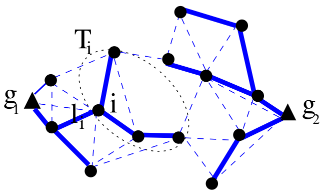


Figure 1: Routing forest with g_1 and g_2 as gateway nodes.

In the following subsections, we use an analytical model for bandwidth sharing between links at the MAC layer, and use this model to study impact on MAC-interactions on routing decisions.

3.1 Bandwidth sharing at MAC layer

Two links contend with each other at the MAC layer if the carrier sensing at the senders or the RTS-CTS mechanism preclude them from being active simultaneously. The indicator function $\mathbf{1}_{\{l_i \triangle l_j\}}$ is one if links l_i and l_j contend with each other, and zero otherwise. In [3], the authors have shown that in 802.11 DCF, the presence of low bit rate links considerably reduces the throughput of a nearby high bit rate link, since the slower links occupy the channel for longer duration. For example, consider k wireless links having nominal bit rates of $R_j, j = 1$ to k, and assume that all the links lie within the same contention domain, i.e., only one link from this set can be active at a time. For this scenario, 802.11 DCF allocates on average an equal number of channel access opportunities to all the links [14]. Based on this observation, the authors in [6] propose the following routing metric which is referred to as the Expected Throughput (ETP),

$$\frac{1}{\mathrm{ETP}_{l_i}} = \frac{1}{P_{l_i}} \cdot \sum_{j \in \mathcal{M}} \frac{\mathbf{1}_{\{l_i \triangle l_j\}}}{R_{l_j}} \tag{1}$$

In the model in Eq. (1), the sum of the inverse of the data rates models the equal time share received by all the contending links, while the success probability term, P_{l_i} is used to obtain the useful throughput of the link. As noted in [6], the above model assumes that *all* contending links can hear each other, i.e., they form a maximal clique. When the links do not form a maximal clique, more elaborate models are needed to predict the link throughput (see [15]). In these models, it is necessary to numerically solve complex fixed-point equations in a centralized manner. Furthermore, these models require knowledge of all the *active* links in the network. Since the set of active links in the network directly defines the routing policy in the context of mesh networks, this results in a chicken-and-egg problem. In order to keep our model tractable, we adopt the ETP model proposed in [6], and approximate the bandwidth received by l_i using Eq. (1). However, note in Eq. (1), we include *all* the active links, and not just the links in a given path. Consequently, unlike [6], we take intra, as well as inter-flow contention into account. Although an approximation, the above model captures at least the first order effects of the reduction of link capacities due to MAC-layer contention. Depending on routing, each mesh node j has a unique link l_j for attaching itself to its parent. By summing over $\{j \in \mathcal{M}\}$, we include only these *active links* in the bandwidth computation. Thus we explicitly take into account the *impact of routing on link quality.*

3.2 Objective function

Shortest path routing algorithms which rely on routing metrics, such as ETX and ETT do not include support for load balancing across multiple gateways. In this subsection, we build on top of ETP to incorporate load balancing. In routing forest $\mathcal{T}$, let $\mathcal{P}_j$ be the route of node j to its associated gateway, i.e., $\mathcal{P}_j$ is the set of links between node j and its associated gateway. A

[2] An extension to include multi-path routing is part of our future work.

mesh node i is connected to its parent node through link l_i. If link l_i belongs to path $\mathcal{P}_j$, i.e., $j \in T_i$, then a fraction of the bandwidth of link l_i is used for the traffic of node j. For simplicity, we assume that all the nodes in a subtree are treated equally, i.e., we assume fair bandwidth sharing within a subtree. Since the expected throughput of link l_i is ETP_{l_i}, the throughput received by traffic of node j over link l_i is $\frac{\mathrm{ETP}_{l_i}}{|T_i|}$, where $|T_i|$ is the number of nodes in subtree T_i (including node i). Hence the transmission delay of a bit of node j over link l_i is $\frac{|T_i|}{\mathrm{ETP}_{l_i}}$. Thus, the total transmission delay for node j in sending a bit to its associated gateway, and the total transmission delay over all of the mesh nodes for sending a bit to their respective gateway nodes are:

$$D_j(\mathcal{T}) = \sum_{k:l_k \in \mathcal{P}_j} \frac{|T_k|}{\mathrm{ETP}_{l_k}}, \quad \mathcal{D}(\mathcal{T}) = \sum_{j \in \mathcal{M}} D_j(\mathcal{T}) \qquad (2)$$

Note that the above cost function includes the effect of spatial multiplexing through the ETP metric. This is unlike shortest path routing with ETT [5] where the computed path metric *does not* account for the possible concurrent operation of links separated by more than three hops. As a result, ETT unfairly penalizes long paths.

Note that in Eq. (2), all terms are of the form $\frac{|T_i|}{\mathrm{ETP}_{l_i}}$ for some i. Furthermore, a term such as $\frac{|T_i|}{\mathrm{ETP}_{l_i}}$ corresponds to link l_i, and appears exactly $|T_i|$ times, once for each node in the subtree T_i. Thus, we can rewrite (2) as follows:

$$\mathcal{D}(\mathcal{T}) = \sum_i \frac{|T_i|^2}{\mathrm{ETP}_{l_i}}, \qquad (3)$$

where ETP_{l_i} is given by (1). Our objective is to determine a routing forest $\mathcal{T}$ that minimizes the above cost function.

4. ROUTING ALGORITHMS

In this section, we propose two distributed load balancing and routing algorithms. Unlike the shortest path routing algorithms that work with hop count, ETX and ETT metrics, the proposed algorithms aim to balance the traffic load across the gateways in the network.

4.1 MaLB: A MAC-aware Load Balanced Routing Algorithm

In this subsection, we solve the problem of determining routes that are optimal with respect to the cost function defined in Eq. (3). Minimizing the objective function in Eq. (3) requires us to evaluate the objective function over all the possible routing forests (exponential complexity). Traditional shortest path algorithms used in conjunction with ETX and ETT are not applicable here, since the link weights in our problem formulation are not fixed, but depend on the routing configuration itself. *We are not aware of any algorithms that compute optimum trees when link weights depend on the choice of tree.* Hence we propose an algorithm, referred to as MaLB (MAC-aware and Load Balanced routing) that is guaranteed to converge at least to a local minimum of Eq. (3). MaLB is a greedy, and distributed routing algorithm. In the following, we first briefly describe the idea behind the proposed algorithm, and then present the exact algorithmic details.

We assume that as an initial configuration, the network is organized in a forest structure (not necessarily optimum). This could be generated through hop count metric and any generic shortest path routing protocol. We assume that a separate protocol for tree initialization is used during network bootstrap. Starting with this initial topology, the proposed algorithm progressively reconfigures the network topology. Each node maintains a periodic timer. When the timer at node i expires, it finds the best point of attachment to the routing forest, or equivalently, the best parent node, and then migrates to the new parent node along with its entire subtree. For example, in Fig. 2, node i migrates along with its subtree from parent j to parent k. A potential migration has an impact on two types of terms in the cost function Eq. (3); the tree size $|T_m|$ of an arbitrary node m, and ETP_{l_n} of an arbitrary node n. This is depicted in Fig. 2 (c) and Fig. 2 (d).

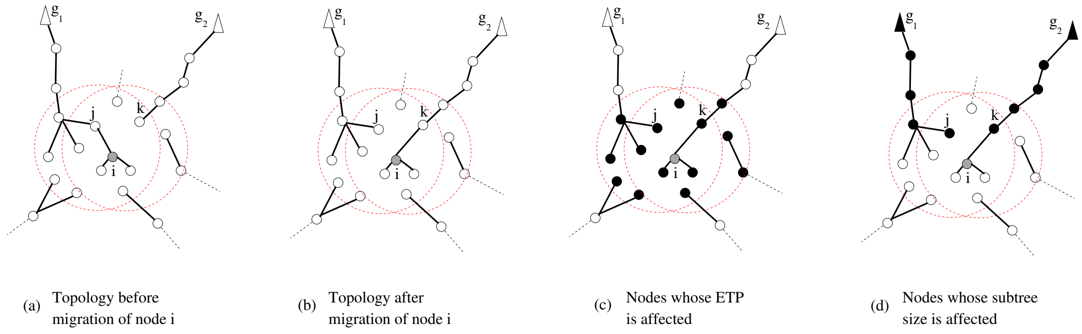
Since the link between node i and j, denoted by l_i^j is no longer used after migration, this also has an impact on the ETP of the active links in the contention domain of nodes i and j. Likewise, after the migration, the link between i and k, denoted by l_i^k which was initially inactive, becomes active. This has an impact on the ETP of other active links in the contention domain of nodes i and k. Fig. 2 shows the nodes of the links whose ETPs are affected by this migration.

We denote F_i as the current parent of node i. Set $\mathcal{A}_k$ denotes the set of ancestors of node k including itself. The set of candidate parent nodes of node i is denoted by $\mathcal{S}_i$. This set includes those neighbors of node i that are not in its subtree (to avoid routing loops). The set of nodes that belong to the contention domain of link l_j is denoted by CD_{l_j}. A node belongs to the contention domain of a link if it is in the carrier sensing range of either endpoints of the link. The choice of new parent node is made as follows.

Algorithm 1: MaLB
When the periodic timer of node i expires, it performs the following steps:

1. For each $k \in \mathcal{S}_i$, form $\mathcal{H}_{ik}$, the set of nodes that are affected if node i migrates to node k.

$$\mathcal{H}_{ik} = \mathrm{CD}_{l_i^k} \cup \mathrm{CD}_{F_i} \cup \mathcal{A}_{F_i} \cup \mathcal{A}_k. \qquad (4)$$



(a) Topology before migration of node i

(b) Topology after migration of node i

(c) Nodes whose ETP is affected

(d) Nodes whose subtree size is affected

Figure 2: Nodes affected by migration of node i from node j to node k are depicted as black circles.

2. For each $k \in \mathcal{S}_i$, compute the following quantity which determines the change in the global objective function resulting from the migration of node i to node k.

$$\Delta_{ik} = \sum_{j\in\mathcal{H}_{ik}} \left\{ \frac{|T'_j|^2}{\text{ETP'}_{l_j}} - \frac{|T_j|^2}{\text{ETP}_{l_j}} \right\}, \qquad (5)$$

   where $|T_j|$ and $|T'_j|$ are the respective subtree sizes, while ETP_{l_j} and ETP'_{l_j} are the respective ETPs of node j before and after the migration of node i to node k. The quantity Δ_{ik} can be computed locally at node i as shown in Subsection 4.3.

3. Find the potential parent, F'_i that results in the highest reduction of the global objective function, i.e., $F'_i = \text{argmin}_{k\in\mathcal{S}_i}\Delta_{ik}$.

4. Migrate to F'_i if $F'_i \neq F_i$.

Typically, estimation of link quality (bit rate and packet success ratio) requires sending periodic probes, and averaging the observed results over time durations of a few minutes. This is required, because the random time-varying shadow-fading results in substantial fluctuations in link qualities, and hence responding to such such short term fluctuations can lead to route flaps as shown in [16]. We assume that the period of the timers running on mesh nodes for making migration decisions are also of the order of few minutes, since reliable link quality estimates are not available at a smaller time scale.

We also assume that the message exchange between a node, its children, and its ancestors during a migration can be completed within a few milliseconds, since this involves exchanging short association/disassociation messages. Under this model, it is reasonable to assume that with high probability, at any given instant, only one node is involved in migration in the entire network. For example, in a network of 100 nodes, with a migration timer period of 2 minutes, and assuming that the multi-hop association/dissociation message of 40 bytes travels 10 hops at 6Mbps, the probability that at least two nodes are simultaneously in the migrating phase is less than 10^{-3}. Thus, the routing tree information at each node is consistent at every time instant with high probability. With the timer values tuned in accordance with the above observation, we have the following result on the convergence of MaLB.

Theorem 1 *Starting from a given forest structure $\mathcal{T}_0$, MaLB converges in a finite number of steps to $\mathcal{T}^*$, which is a local minimum of Eq.* (3).

PROOF. Consider the migration of node i from its current parent j to new parent k (with $k \neq j$). Using Step 3 of Algorithm MaLB, we have

$$\Delta_{ik} < \Delta_{it} \;\; \forall \text{ potential parents } t \text{ of } i.$$
$$\Rightarrow \mathcal{D}(\mathcal{T}_{n+1}) - \mathcal{D}(\mathcal{T}_n) = \Delta_{ik} < \Delta_{ij} = 0 \quad \text{using Step 4.}$$

Thus, any time a node migrates, the value of the objective function *strictly* decreases. Note that the total number of distinct routing forests is finite. Hence the algorithm converges to a local optimum in finite number of steps. □

Simulation results in Section 5 show that although MaLB may produce a locally optimum configuration, it results in substantial throughput improvements over other routing algorithms.

4.2 LB: A MAC-agnostic Load Balanced Routing Algorithm

The two distinguishing features of MaLB approach are (i) Inclusion of MAC-layer interactions through ETP, and (ii) Load balancing through efficient multi-hop association of mesh nodes to gateways. As discussed in Section 3, the MAC interactions between the links lead to substantial increase in the complexity of ETP. Hence, we define a new routing algorithm that performs load balancing, but uses a simple model for estimating the capacity of a link. In this model, the MAC interactions between the links are ignored, and the product of the bit rate and the packet success probability of the link is used as a measure of its capacity. Using the approach in Section 3, the routing problem is equivalent to finding optimum forest structures which minimize the following objective function corresponding to Eq. (3).

$$\tilde{\mathcal{D}}(\mathcal{T}) = \sum_i \frac{|T_i|^2}{P_{l_i} R_{l_i}}, \tag{6}$$

Corresponding to Algorithm 1, we define Algorithm 2 for finding the optimum routes. We refer to this algorithm as LB (Load Balancing). LB forms the middle ground, since it is an intermediate low complexity solution between load-agnostic routing algorithms, such as shortest path with ETX, ETT, and a MAC-aware load balancing routing algorithm such as MaLB.

Algorithm 2: LB
This algorithm is identical to Algorithm 1, except that in the first step, the set of affected nodes $\mathcal{H}_{ik}$ does not account for MAC-layer impact, i.e., $\mathcal{H}_{ik}$ is defined as follows:

$$\mathcal{H}_{ik} = \mathcal{A}_{F_i} \cup \mathcal{A}_k. \tag{7}$$

Observe that the reciprocal of the term in the denominator of Eq. (6) is the ETT of the parent link of node i. Therefore, comparing Eq. (3) and Eq. (6), we note that *LB corresponds to the case of joint routing and load balancing when the reciprocal of ETT is used as the estimate of the link bandwidth.*

In Section 5, we show that the throughput increases progressively as we go from ETX to MaLB. LB has higher throughput than ETX and ETT due to its load balancing attribute. However, further throughput increase can be obtained using MaLB when the MAC-layer interaction is also taken into account.

4.3 MaLB, LB: Distributed Implementation

The appealing feature of shortest path algorithms such as OLSR and DSDV which is exploited by ETX and ETT, is that they can be implemented in a distributed manner. In this subsection, we show that MaLB is also amenable to a *distributed implementation.* Distributed implementation of LB is along the lines of MaLB, and is considerably simplified, since it does not rely on the ETP metric. Referring to the description of MaLB, we note that for distributed implementation, it is necessary that a node be able to compute Δ_{ik} for each of its potential parents using Eq. (5) using *local* information. We show that this can indeed be achieved through limited message exchange between the mesh nodes.

Recall that $\mathcal{H}_{ik}$ is the set of nodes that are affected by the migration of node i to node k. The effect of migration is either at the MAC layer (through ETP) or at the network layer (through changes in tree size). We can therefore partition set $\mathcal{H}_{ik}$ into two disjoint sets; (i) $\mathcal{U}_{ik}$, the set of nodes whose ETP is affected by the migration of node i (e.g., the black nodes in Fig. 2 (c)), and (ii) $\mathcal{V}_{ik}$, the set of remaining nodes whose link ETP is unaffected by the migration of node i (e.g., the black nodes in Fig. 2 (d)). Thus, $\mathcal{H}_{ik} = \mathcal{V}_{ik} \cup \mathcal{U}_{ik}$, and $\mathcal{U}_{ik} \cap \mathcal{V}_{ik} = \phi$. Thus, Eq. (5) can be rewritten as follows.

$$\Delta_{ik} = \sum_{j \in \mathcal{U}_{ik}} \left\{ \frac{|T'_j|^2}{\text{ETP'}_{l_j}} - \frac{|T_j|^2}{\text{ETP}_{l_j}} \right\} + \sum_{j \in \mathcal{V}_{ik}} \left\{ \frac{|T'_j|^2 - |T_j|^2}{\text{ETP}_{l_j}} \right\} \tag{8}$$

For every active link that a node is part of, the node includes information about the bit rate and success probability of that link in its periodic beacon messages. Using this information from the beacon messages of all the nodes in its contention domain, a node can determine the ETP of all its active links using Eq. (1). Each node then embeds the ETP value of all its active links in its beacon messages. When node i wishes to determine the impact of its migration on the ETP of a contending link, say l_j, it first determines if l_j belongs to the contention domain of its old link $l_i^{F_i}$ and new link l_i^k. The new ETP of l_j is then computed as follows:

$$\text{ETP'}_{l_j} = \frac{P_{l_j}}{\left(\frac{P_{l_j}}{\text{ETP}_{l_j}} - \mathbf{1}_{\left\{ l_j \Delta l_i^{F_i} \right\}} \frac{1}{R_{l_i^{F_i}}} + \mathbf{1}_{\left\{ l_j \Delta l_i^k \right\}} \frac{1}{R_{l_i^k}} \right)} \tag{9}$$

We also assume that each node j embeds its tree size, and the ID of its parent node in its beacon message. Using this information, node i can determine if node j is its current ancestor or new ancestor (or both), and can thus determine the new tree size of node j. Thus, the new ETPs as well as the new tree sizes of all the nodes in $\mathcal{U}_{ik}$ (the MAC-neighborhood of node i), and hence the first summation on the right hand side of Eq. (8) can be locally computed.

We now show that even the second term in Eq. (8), consisting of contributions from links in $\mathcal{V}_{ik}$, can be locally computed at node i. The ETP of the links of the nodes in $\mathcal{V}_{ik}$ are unaffected by the transition of node i. However, their tree size could still be affected by node i's migration. Recall that $\mathcal{A}_k$ is the set of ancestors of node k including itself. Hence, $\mathcal{V}_{ik} \subseteq \mathcal{A}_{F_i} \cup \mathcal{A}_k$. Thus, the second term in Eq. (8) can be re-written as follows.

$$\begin{aligned} &\sum_{j \in \mathcal{V}_{ik}} \left\{ \frac{|T'_j|^2 - |T_j|^2}{\text{ETP}_{l_j}} \right\} \\ =& 2|T_i| \left\{ \sum_{j \in \mathcal{V}_{ik} \cap \mathcal{A}_{F_i}} \frac{|T_j|}{\text{ETP}_{l_j}} \right\} - |T_i|^2 \left\{ \sum_{j \in \mathcal{V}_{ik} \cap \mathcal{A}_{F_i}} \frac{1}{\text{ETP}_{l_j}} \right\} \\ &+ |T_i|^2 \left\{ \sum_{j \in \mathcal{V}_{ik} \cap \mathcal{A}_k} \frac{1}{\text{ETP}_{l_j}} \right\} - 2|T_i| \left\{ \sum_{j \in \mathcal{V}_{ik} \cap \mathcal{A}_k} \frac{|T_j|}{\text{ETP}_{l_j}} \right\} \end{aligned} \tag{10}$$

Although the expression in Eq. (10) appears to be complex, it can be evaluated relatively easily by observing that each of the summations can be readily interpreted as a cumulative sum along the tree branches. We assume that starting from the one-hop nodes attached to the gateways, and moving towards the leaf nodes, each node propagates the cumulative sum (up to the parent, but not including the node's own terms) corresponding to two quantities; $1/\mathrm{ETP}_{l_j}$, $|T_j|/\mathrm{ETP}_{l_j}$. Thus, at each node, a cumulative sum of the above two quantities along all hops from the gateway node to its parent are available. The terms corresponding the node are not added, since they are included in separate fields ($|T_i|, \mathrm{ETP}_{l_i}$). Let f_i be the ancestor of node i that is within the contention domain of l_i, and farthest from node i (in terms of number of hops). Node i can determine $1/\mathrm{ETP}_{l_{f_i}}$, as well as the cumulative sum $\sum 1/\mathrm{ETP}_{l_j}$ up to node f_i, using simple message exchange with its parent node, since f_i belongs to contention domain of either i or its parent node or both. Using this information, node i can determine all the terms in Eq. (10). We use the beacon-stuffing approach in [17], whereby an information element (253 bytes) in IEEE 802.11 beacon message is used to embed information about all the active links of a node (see Fig. 3).

$\lvert T_i \rvert$ (1 byte)	$\sum \frac{1}{ETP_{l_j}}$ (6 bytes)	$\sum \frac{\lvert T_j \rvert}{ETP_{l_j}}$ (6 bytes)

Parent	MAC addr (6 bytes)	$R_{l_i^k}$ (1 byte)	$P_{l_i^k}$ (1 byte)	$ETP_{l_i^k}$ (2 bytes)
Child 1				
Child 2				
⋮ Child r				

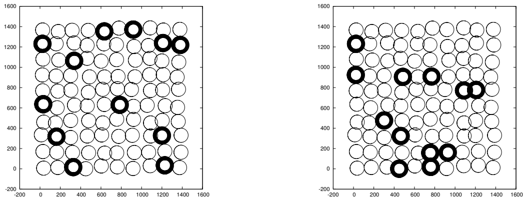
Figure 3: Fields in the information element of a beacon message of a mesh node in MaLB.

4.3.1 Computing Tree size

We assume that when a node i joins or leaves its parent node, it informs its parent node about its migration, and also about the size of its subtree, T_i. Besides, when the size of the subtree of a node changes (due to nodes joining or leaving the subtree), this information is propagated to all the ancestors of the node. Note that the above message exchange is only limited to the paths joining the affected node to its old and new gateways. Thus, each node i has the latest information about its tree size $|T_i|$.

4.3.2 Computing ETP

We assume that each node has a mechanism for monitoring the "health" of all links originating from that node. Thus, the bit rate $R_{l_i(j)}$, and the packet success probability of link $P_{l_i(j)}$, to all the potential parent nodes j of node i is available at node i. Periodic broadcast and/or unicast probes could be used for this purpose [4, 5]. A link is active if it is part of the forest structure, and is therefore being used for routing traffic. Each node j embeds in its beacon message the following information about all the active links that originate at that node: R_{l_j}, the bit rate of the link, and P_{l_j} the packet success probability of that link. We assume that a node can decode the beacon messages of all nodes that are within its contention domain[3]. When the information about the bit rates and the success probabilities of the contending links gathered at node j is combined with the corresponding information of node i, the contention domain of link $l_i(j)$ is completely known to node i. Using this combined information, node i can compute $ETP_{l_i(j)}$ for all of its potential parent links $l_i(j)$ from (1). In case of LB, the ETP is replaced by the product of the bit rate and the packet success probability, and both these quantities are locally available at each node.



(a) Topology 1

(b) Topology 2

Figure 4: Perturbed grid topologies used for simulations. Bold circles represent gateway nodes.

5. PERFORMANCE EVALUATION

It is of paramount importance to incorporating a realistic wireless environment when performing wireless simulations [11]. We have therefore followed the guidelines provided in [11], and added several new features to the ns-2 simulator [18] to make the simulation scenarios as close as possible to reality. In this section, we use ns-2 simulations to compare the throughput performance of MaLB and LB routing algorithms with a shortest path routing algorithm using ETX and ETT as routing metrics. We first describe our simulation methodology, and then discuss the detailed set-up of the simulator.

5.1 Methodology

Since performing dynamic link-layer probing, and updating routes during the run-time of a simulation is difficult in ns-2, and most other simulators, we take the following approach. Our objective is to first simulate probe agents, gather link quality statistics, compute routes based on these statistics, and then run traffic with the same topology. Note that the shadowing chan-

[3] The beacon messages are short, and are sent at the lowest and the most resilient bit rate. Hence the assumption that all the nodes within the contention domain of a given node receive its beacon messages is realistic.

nel gains between the nodes are not time-varying, but there are small scale fading variations due to movement of objects in the neighborhood of the nodes. It is necessary to ensure that the link probes and the traffic are simulated using the exact same topology, and the exact same shadowing gains.

Hence, we first fix a shadowing seed, and run one set of simulations during which ETX and ETT probes are generated to determine the link qualities. The results from these measurements (packet success ratio, and bit rate) are then fed into a Dijkstra shortest path computation engine to determine the routes corresponding to ETX and ETT. The measurements are also fed into the MaLB/LB computation engine which generates routes corresponding to MaLB and LB. We use the routing generated by ETT as the initial non-optimized routing forest structure for MaLB. All the required quantities for shortest path routing with ETT, such as bit rates and packet success probabilities of the links, are also required by MaLB, and therefore assumed to be readily available at all the nodes. These quantities can therefore be used to initialize the ETT-based routing forest. We then run TCP traffic (first downlink, then uplink) with these routes on the same topology, using the same shadowing seed as the one used for collecting probe results. The entire process is then repeated for the next shadowing seed. Eight independent shadowing seeds are used, and throughputs are averaged over these eight runs.

For simulating ETX probes, we use periodic broadcast probes (sent once every second) at the lowest bit rate (6Mbps) [4]. For ETT, we use a different probing mechanism from the one in [5]. In [5], the authors propose using unicast packet-pairs once every minute to estimate the current bit rate of the link. We simulate short periodic unicast `ping-no-reply` packets between each node sent once every second, and recover the current bit rate of the link directly from the auto-rate algorithm. We adopt the above approach to obtain more accurate bit rate estimates.

We understand that the above approach of first simulating the probes, and then simulating the actual traffic, does not take into account the fact that the measured quantities can themselves be affected by the ongoing traffic. This can lead to issues related to routing instabilities as pointed out in [16]. A detailed comparative analysis of routing stability of ETX, ETT, LB, and MaLB is an important and interesting research topic on its own, and is part of our future work.

5.2 Simulation Set-up

5.2.1 *The simulator*

We use the ns-2 simulator with several mesh-specific enhancements [18] added to overcome some of the known limitations of wireless capabilities of ns-2 [11]. We have added the following new modules to ns-2.

- *Cumulative interference along with noise power for an accurate SINR-based reception model:* The current ns-2 implementation compares two signal powers to determine successful packet reception.
- *Combined shadow-fading with an accurate shadowing model:* The current shadowing model of ns-2 is inaccurate, since it recomputes a new independent random variable as the shadowing gain for every new packet reception.
- *Multi-SINR and multi-rate 802.11 a/g links, Autorate fallback (ARF) algorithm for rate adaptation:* Currently, ns-2 only supports links with fixed rates.
- *A framework for link quality estimation:* We have added support for using unicast and broadcast probes as required by most mesh routing protocols such as, ETX [4], ETT [5], and ETP [6].

5.2.2 *Topology*

We simulate a topology consisting of 100 nodes (mesh and gateway nodes). The nodes are deployed in a 10×10 grid with a random perturbation in x and y direction around the grid points. The grid points are separated by 150m, and a random perturbation of 20m is used around these points. This models the scenario of regular deployment of APs along streets on lamp-posts with small perturbations. We do not use uniformly random node deployment, since it results in disconnected nodes, as well as nodes that are arbitrarily close to each other, both of which are unlikely even in an unplanned mesh deployment. Two typical randomly generated topologies for which detailed results are presented in this paper are shown in Fig. 4. Similar results were obtained for some of the other randomly generated simulation topologies, and hence have been omitted.

The grid consists of four quadrants, and each quadrant has 3 randomly deployed gateways (indicated by dark circles). This models uniform deployment of gateways across the entire region, with a certain degree of randomness in the location of the gateways. We use the following model for gateway failure. A gateway is said to have failed if its connection to the Internet goes down. When this happens, the routes are reorganized so that the mesh nodes associated with the failed gateway find a route to a different gateway. Furthermore, we assume that in spite of the failure of gateway's Internet connection, the client nodes associated with this gateway (over a different radio) continue to generate traffic. Hence the failed gateway behaves like an ordinary mesh node, and participates in multi-hopping. We consider the following four scenarios. Scenario 0 is the default scenario, and in this scenario, all the gateways in each quadrant are functional. In scenario 1, one randomly picked gateway in the first quadrant (bottom left quadrant) has failed. In scenario 2, one randomly chosen gateway in the first quadrant, and one randomly

Parameter	Value
Transmit Power	23dBm
Noise Power	-96dBm
Receive Sensitivity	-93dBm
Carrier Sensing Threshold	-93dBm
Antenna gain	13dBi
Path loss constant (c in $c \cdot d^{-n}$)	-53dB
Path loss exponent (n in $c \cdot d^{-n}$)	3.3
Shadowing std dev	4dB
Shadowing correlation between forward and reverse directions of a link	0.5
K factor (Ricean fading)	10
Ambient object speed (Ricean fading)	0.2m/s
Number of independent shadowing runs	8

Table 1: Simulation Settings

chosen gateway in the third quadrant (top right) has failed. Scenario 3 has two failed gateways in first quadrant. Thus, scenario 3 has the highest imbalance in terms of the number of mesh nodes per gateway.

5.2.3 MAC and PHY-layer settings

We use 802.11g (link rates of 6 to 54 Mbps). All nodes have a single radio for mesh communication, and operate over a common channel. We assume that each mesh node communicates with its clients using a separate radio over an orthogonal channel (not simulated). The values of c and n in $c{\cdot}d^{-n}$ in the path loss model in Table 1 are taken from the measurement studies reported in [2] in a suburban region for 802.11g. Unlike the testbed nodes in [2] where 15dBi antennas were used, we use antenna gains of 13dBi in our simulations to keep the EIRP (Effective Isotropically Radiated Power) below the allowable limit of 36dBm. We use a shadowing standard deviation of 4dB instead of 6dB as reported in [2], since several rounds of simulations with 6dB shadowing resulted in many unusable links due to excessive shadowing, as well as many extremely long range links (over 1200m). The correlation between random shadowing gains in the forward and the reverse directions is taken to be 0.5 [19]. This enables us to simulate asymmetric links which are often observed in real testbeds [20, 11]. Important simulation parameters are summarized in Table 1.

5.2.4 Traffic model

Since the majority of traffic observed in the Internet is carried over TCP, we use TCP as the transport protocol for generating traffic. For the downlink scenario, a single FTP download is carried out from each gateway to each of its associated mesh nodes. In another independent set of runs, we simulate the uplink scenario in which a single FTP upload is carried out from each mesh node to its associated gateway. The simulation duration is 5 minutes (sufficient for TCP to reach equilibrium [20]). In contrast to the past work [4, 5], we carry out concurrent TCP sessions. We believe that this is a realistic scenario to simulate, since multiple users are simultaneously active in the network. We do not use UDP traffic, since depending on the network topology, the maximum sustainable rate of a path could vary substantially, and hence setting the input rate of each connection is a non-trivial task. Instead, we let TCP determine the highest sustainable rate.

5.2.5 Comparison Metrics

In a mesh network, depending on the location of the mesh nodes and the gateways, some of the mesh nodes may get substantially lower throughput than others. This aspect is inherent to the many-to-few communication architecture. When a data set has very high dispersion, instead of the mean, the median is a better metric for comparing performance. Furthermore, instead of using the standard deviation, the Semi-Inter-Quartile Range (SIQR) is a better metric to measure the spread of data (a measure of fairness), as it ignores extremely high and extremely low values in the data set. SIQR is half the difference between the 75th percentile and the 25th percentile of the data set. Hence, as in [20], we use the median and the SIQR for comparing the considered routing metrics.

5.3 Simulation Results

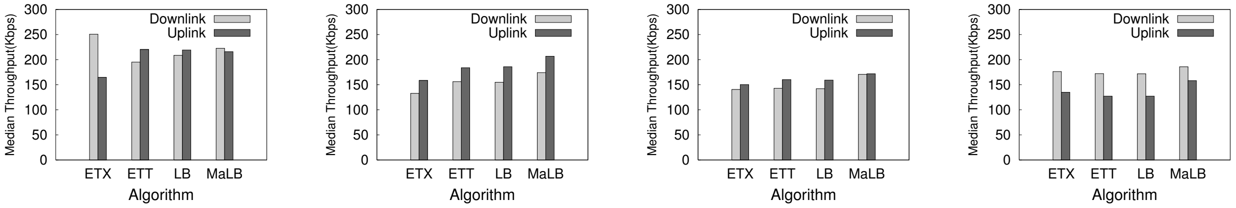
In this subsection, we present simulation results for the shortest path routing algorithm with ETX, ETT, and the MaLB and LB routing algorithms with different topologies, and under several gateway failure scenarios.

5.3.1 Throughput Dependence on Gateway Location

In Fig. 5 and Fig. 6, we plot the median throughput of all the routing algorithms[4] for topologies 1 and 2. These simulations were carried out with RTS/CTS on. With RTS/CTS off, we observed high packet losses and low throughputs due to hidden node problem (more details on the impact of RTS/CTS in Subsection 5.3.4).

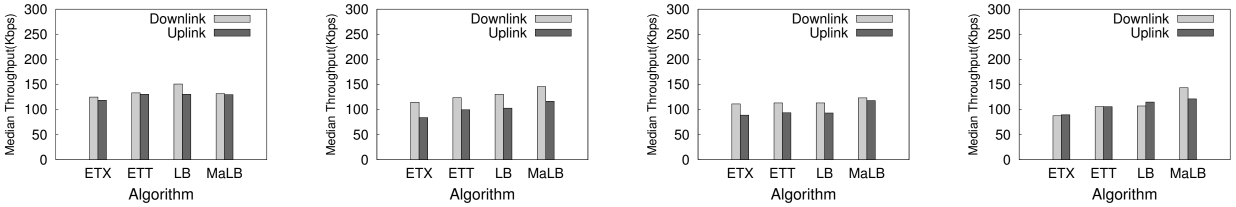
We note that the median throughput for topology 2 is lower than the median throughput for topology 1 (for all the scenarios). This is because topology 2 has some gateways very close to each other. This leads to inefficient usage of gateway bandwidth due to contention between the gateways. Furthermore, this also results in some clients being far from any of the gateways, since most of the gateways are concentrated in one region. Thus, the gateways should be spaced uniformly to reduce the number of hops between the mesh nodes and the gateways. However, note that depending on the location of a node, it may not be possible to provide Internet connectivity at that node. Or equivalently, it may not be possible to select an optimal gateway loca-

[4] In Fig. 5 to Fig. 7, the label "ETX" (respectively "ETT") along the x-axis stands for the shortest path algorithm with ETX (respectively ETT) as the routing metric.



(a) Scenario 0 (b) Scenario 1 (c) Scenario 2 (d) Scenario 3

Figure 5: Median throughput for topology 1 in Fig. 4(a). MaLB performs better load balancing, and results in higher throughput for skewed topologies of scenarios 1, 2, and 3.



(a) Scenario 0 (b) Scenario 1 (c) Scenario 2 (d) Scenario 3

Figure 6: Median throughput for topology 2 in Fig. 4(b). MaLB performs better load balancing, and results in higher throughput for skewed topologies of scenarios 1, 2, and 3.

tion. Hence we use a model in which the location of gateway within the network is randomized.

5.3.2 *Algorithm Performance*

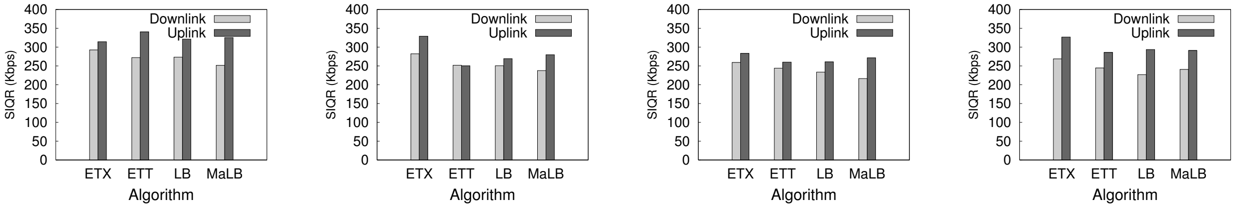
From Fig. 5 and Fig. 6, we note that for the case of a balanced load (scenario 0) in which all quadrants have 3 active gateways, all algorithms perform more or less identically. For scenarios 1, 2 and 3 which correspond to unbalanced networks, we note that the median throughput improves progressively as we go from ETX to MaLB. We note that LB performs better than ETX and ETT with shortest path. This demonstrates the benefit of load balancing. Furthermore, MaLB performs better than LB, thereby showing that it is beneficial to take MAC-layer interactions into account when determining routes.

For scenario 1 which corresponds to a single gateway failure in the first quadrant, MaLB performs up to 38% better than ETX, up to 17% better than ETT, and up to 13% better than LB. Scenario 2 corresponds to a single gateway failure in the first, as well as the third quadrant. For this scenario, MaLB performs up to 32% better than ETX, up to 25% better than ETT, and up to 26% better than LB. Thus, for all the scenarios in which one or more quadrant has a failed gateway, MaLB results in significant throughput improvement.

However, the highest gains from MaLB are attained for highly unbalanced topologies. To study the best case improvement of MaLB over other algorithms, we consider scenario 3 of topology 2 in which there are two failed gateways in the first quadrant. In Fig. 8, we plot the percentage improvement of MaLB over other algorithms for this scenario. For this scenario, MaLB results in more than 60% throughput improvement over ETX, more than 30% improvement over LB for the downlink. Although the throughput improvement over the uplink is not as high as the downlink, MaLB still outperforms other algorithms.

Thus, MaLB adapts to gateway failures by balancing load across gateways, and avoids MAC-congested paths to provide improved throughput. Load-agnostic routing protocols such as shortest path routing with ETX and ETT continue to send most of the traffic of mesh nodes in the first quadrant to the only surviving gateway, thereby overloading it. LB results in improvement over ETX and ETT through load balancing, although not as significant as MaLB (especially for the downlink). Hence we conclude that a combination of (i) MAC-awareness, and (ii) load balancing is required for optimizing throughput in mesh networks. Purely load-aware algorithms such as LB which have simple and intuitive graph-theoretic interpretations do not perform as well as MaLB, since they are MAC-agnostic.

In Fig. 7, we plot the SIQR of the node throughputs for uplink as well as downlink for topology 1. The smaller the SIQR, the lower the dispersion of the data. We note that the SIQR progressively decreases as we go from ETX to MaLB. Combining all our observations, we conclude that MaLB has the best fairness among all the studied algorithms.



(a) Scenario 0 (b) Scenario 1 (c) Scenario 2 (d) Scenario 3

Figure 7: SIQR plots for topology 1 in Fig. 4(a) showing that MaLB is comparable to ETX and ETT in terms of fairness.

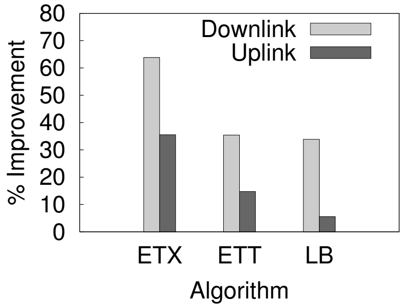


Figure 8: Performance of MaLB for scenario 3 of topology 2.

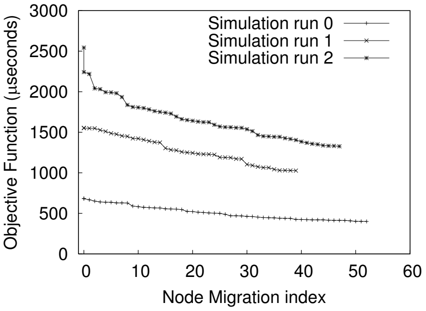
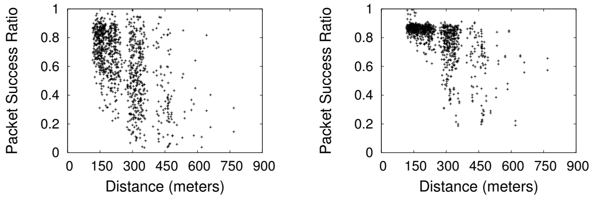


Figure 9: Evolution of the objective function with node migration in MaLB.



(a) RTS/CTS off (b) RTS/CTS on

Figure 10: Impact of RTS/CTS on packet success.

5.3.3 *Speed of Convergence of MaLB*

In Fig. 9, we plot the evolution of the global objective function against the migration step for three randomly chosen simulation runs. We note that after convergence, MaLB results respectively in 41%, 34% and 48% reductions in the values of the objective function. Over all the simulation runs, MaLB resulted in an average of 37% percent reduction in the objective function starting with the ETT generated routing topology. In the best case, there was up to 57% reduction in the value of objective function after MaLB terminated. Furthermore, over all the considered simulation scenarios, we observe that MaLB takes an average of 40 node migrations to converge. The maximum number of node migrations required was 57.

5.3.4 *RTS/CTS helps*

To study the impact of RTS/CTS, in Fig. 10, we plot the packet success ratio for the unicast ETT probe packets as a function of distance (for all the links in the network). These plots correspond to one of the shadowing seeds used in subsequent simulations. We note that RTS/CTS significantly improves the packet delivery ratio by alleviating the hidden node problem. For example, referring to Fig. 10, the links that are approximately 150m long, have more than 80% packet success with RTS/CTS. However, without RTS/CTS, the same links may have packet success ratio as low as 20%. Hence, in the rest of the paper, we only present results for scenarios with RTS/CTS turned on.

We also note that turning off RTS/CTS has much more impact on ETT, MaLB, and LB, as compared to ETX. This is because, the absence of RTS/CTS during the unicast probing phase of ETT results in high packet losses due to hidden node problem. Since the ARF auto-rate algorithm treats any form of packet loss as an indicator of poor channel conditions, it unnecessarily falls back to lower bit rates for robustness. Consequently, the bit rates estimated during the probing phase are lower than what the links can support. These inaccurate rate estimates lead to poor route selection by ETT, MaLB, and LB, all of which rely on the bit rate estimates. ETX, on the other hand, does not require rate estimates.

6. CONCLUSION

In this paper, we propose MaLB, a distributed MAC-aware load balancing and routing protocol, and LB, a distributed low complexity MAC-agnostic load balancing algorithm. MaLB is particularly beneficial in scenarios of load imbalance which often result from gateway failures. MaLB performs optimum load balancing while taking into account the MAC-layer interaction between the links, and thereby avoiding MAC-congested paths. We show through simulations with an enhanced version of ns-2 simulator that as compared to shortest path routing algorithms with routing metrics such as ETX, the MaLB algorithm results in more than 60% improvement in throughput. MaLB also results in up to 30% improvement over LB, as well as shortest path routing with ETT as the routing metric.

In the future, we plan to test the performance of the proposed algorithms over an experimental testbed. Another future direction that we will pursue is extension of MaLB to multi-radio mesh networks. Studying the stability, and robustness of ETX, ETT, LB and MaLB via a sensitivity analysis is part of our future work.

Acknowledgments

We would like to thank Roch Guerin, George Varghese, Don Towsley, and the anonymous reviewers for their valuable comments, and Ming Li and Marianna Carrera for helping us with scripts.

7. REFERENCES

[1] J Bicket, D. Aguayo, S. Biswas, and R. Morris. Architecture and evaluation of an unplanned 802.11b mesh network. In *ACM Mobicom 2005*, August 2005. Cologne, Germany.

[2] J. Camp, J. Robinson, C. Steger, and E. Knightly. Measurement driven deployment of a two-tier urban mesh access network. In *ACM MobiSys 2006*, June 2006. Uppsala, Sweden.

[3] M. Heusse, F. Rousseau, G. Berger-Sabbatel, and A. Duda. Performance anomaly of 802.11b. In *IEEE Infocom 2003*, March 2003. San Francisco, California.

[4] D S. J. De Couto, D. Aguayo, J. Bicket, and R. Morris. A high-throughput path metric formulti-hop wireless routing. In *ACM Mobicom 2003*, September 2003. San Diego, California, USA.

[5] R. Draves J Padhye and B. Zill. Routing in multi-radio, multi-hop wireless mesh networks. In *ACM MobiCom 2004*, September 2004. Philadelphia, Pennsylvania, USA.

[6] V Mhatre, H. Lundgren, and C. Diot. Mac-aware routing in wireless mesh networks. In *IEEE/IFIP WONS 2007*, January 2007. Obergurgl, Austria.

[7] J C. Park and S. Kasera. Expected data rate: An accurate high-throughput path metric for multi-hop wireless routing. In *IEEE Secon 2005*, 2005. Santa Clara, California, USA.

[8] Y Yang, J. Wang, and R. Kravets. Interference-aware load balancing for multihop wireless networks. Technical Report UIUCDCS-R-2005-2526, Department of Computer Science, University of Illinois at Urbana-Champaign, 2005.

[9] `http://www.netequality.net`.

[10] Y Bejerano, S. Han, and A. Kumar. Efficient load-balancing routing for wireless mesh networks. *Computer Networks*, 51:2450–2466, 2007.

[11] C Newport, D. Kotz, R. S., Gray J. Liu, Y. Yuan, and C. Elliott. Experimental evaluation of wireless simulation assumptions. *SIMULATION: Transactions of The Society for Modeling and Simulation International*, April 2007. To appear.

[12] J L. Sobrinho. Algebra and algorithms for qos path computation and hop-by-hop routing in the internet. In *IEEE Infocom 2001*, April 2001. Anchorage, Alaska, USA.

[13] J He, J. Chen, and S. H. Gary Chan. Extending wlan coverage using infrastructureless access points. In *HPSR 2005*, May 2005.

[14] B Kauffmann, F. Baccelli, A. Chaintreau, V. Mhatre, D. Papagiannaki, and C. Diot. Measurement-based self organization of interfering 802.11 wireless access networks anchorage. In *IEEE Infocom 2007*, May 2007. Alaska, USA.

[15] Y Gao, J. Lui, and D. M. Chiu. Determining the end-to-end throughput capacity in multi-hop networks: Methodology and applications. In *ACM Sigmetrics 2006*, June 2006. Saint Malo, France.

[16] K. Ramachandran, I. Sheriff, E. Belding-Royer, and K. Almeroth. Routing stability in static wireless mesh networks. In *Passive and Active Measurement Conference*, April 2007. Louvain-la-neuve, Belgium.

[17] R Chandra, J. Padhye, L. Ravindranath, and A. Wolman. Beacon-stuffing: Wi-fi without associations. In *IEEE Hotmobile 2007*, February 2007. Tucson, AZ, USA.

[18] V Mhatre. Enhanced wireless mesh networking for ns-2 simulator. *ACM SIGCOMM Computer Communications Review*, July 2007. (editorial).

[19] E Perahia and D. Cox. Shadow fading correlation between uplink and downlink. In *IEEE VTC 2001*, May 2001. Rhodes, Greece.

[20] R Draves, J. Padhye, and B. Zill. Comparison of routing metrics for static multi-hop wireless networks. In *ACM Sigcomm 2004*, August 2004. Portland, Oregon, USA.