A Unified Metric for Routing and Rate Adaptation in Multi-rate Wireless Mesh Networks

Tae-Seok Kim*, Gentian Jakllari‡, Srikanth V. Krishnamurthy*, Michalis Faloutsos*
*University of California, Riverside ‡BBN Technologies
{*tskim, krish, michalis*}*@cs.ucr.edu* *gentian@bbn.com*

Abstract—**In this paper, we propose a new metric that is applicable both to routing and rate adaptation in multi-rate wireless mesh networks. Unlike many previous efforts, our metric is comprehensive; it considers several factors that affect end-to-end performance such as the effect of the relative positions of the links on a path when choosing the rates of operation and the importance of avoiding congested areas. We call our metric ETM (for Expected Transmission cost in Multi-rate wireless networks). We analytically derive the ETM metric. We show that the ETM metric can be used (a) to determine the best end-to-end path with a greedy routing approach and (b) it can be used to dynamically select the best transmission rate for each link on the path via a dynamic programming approach. Based on ETM, we design and implement the ETM-framework on an indoor wireless mesh network and compare its performance with that of the popular ETT and the recently proposed ETOP metrics. Our experiments show that the ETM-framework yields throughput improvements of up to 253% and 368% as compared with the ETT and ETOP frameworks.**

I. Introduction

The goal of this work is to maximize the end-to-end throughput of flows in a multi-rate wireless mesh network. The throughput depends on a number of factors that should be considered *jointly* during the selection and management of a route. To our best knowledge, no previous work addresses together the issues of: (a) the selection of a path, (b) link rate adaptation on a per path-basis (and not per link in isolation), and (c) the effect of competing flows in the network. We discuss each of these issues below.

a. Path selection: The routing metric should *capture* the influence of all of the following interdependent factors that affect the end-to-end throughput: (a) the number of links on the path, (b) the achievable rates on these links and, (c) the relative positions of the links on the path. The last factor is a direct consequence of (i) the typical use of a finite number of retransmission attempts at the link layer (as with 802.11) and (ii) as many transmissions as needed to deliver the packet with a reliable transport layer protocol (such as TCP) [1]; packet drops closer to the destination induce end-to-end (e2e) retransmissions on links that were successfully traversed in previous e2e attempts and this increases the load and thereby congestion. Previously proposed routing metrics (such as [2], [1], [3]) account for a sub-set of the above factors; to the best of our knowledge, *there is no routing metric that captures the impact of all of these factors.*

b. Rate adaptation: Link rates need to adapt to the changing channel conditions during the life of a routing path. Most current link-rate adaptation schemes (e.g. [4], [5], [6], [7], [8], [9], [10]) operate on a *per link* basis. They do not take into account, the position of the link along the path. As pointed out in [1], link layer packet drops close to the destination are expensive in terms of retransmission costs: the packet has to be retransmitted from the source. This suggests that we want the reliability of the links to increase as we approach the destination and one way to achieve this is to use *less aggressive rate increases* on these links. At the same time, simply reducing the rates could have a negative impact on the throughput. There is no previous metric for rate adaptation that accounts for this above factor.

c. Interaction with other flows: Selection of higher-quality links has an indirect disadvantage. It could cause multiple flows to converge on common paths and can thus cause congestion. This congestion can decrease or even reverse the gains due to the reduced retransmission costs. Avoiding congested links has been considered previously [11], [12], but not jointly with path selection and rate adaptation.

In this paper, we propose a new comprehensive metric that can apply to both routing and rate adaptation in multi-rate wireless mesh networks. Our metric takes into account all of the above mentioned factors. We call our metric ETM (for Expected Transmission cost in Multi-rate wireless networks). ETM is arguably the most comprehensive routing metric to date. From an end-to-end performance point of view, it enables us to: (a) find reliable high-quality paths, (b) identify the appropriate rate for each link, and (c) avoid congested areas.

The contributions of our work are as follows:

Computing and using the ETM metric: We analytically derive the ETM cost of a path. We solve the problem of finding the route with the minimum ETM cost (we call this the Optimal Path Problem or OPP) using a greedy approach, which we show is optimal. We solve the problem of choosing the right rates (rate adaptation) in between route changes, using a dynamic programming approach (we call this the Optimal Rate Problem or ORP).

Building an ETM framework: We design and implement a framework based on ETM by using the Roofnet

This work was partially supported by the Multi-University Research Initiative (MURI) grant W911NF-07-1-0318 and the NSF NeTS grant 0626912.

module in the popular Click toolkit [13]. For comparisons, we also implement the ETT-and ETOP-frameworks [3], [1] and evaluate all the protocols on an indoor wireless mesh network consisting of 21 nodes. We consider 802.11 as the link layer protocol and TCP as the e2e protocol, given that these are readily available and are likely to be used in today's deployments.

Evaluating ETM: The ETM-framework yields higher TCP throughputs compared to the ETT-and the ETOP-frameworks. First, considering TCP flows in isolation, the median throughput with the ETM-framework improves by 131% and 30% over that achieved with the ETOP-and ETT-frameworks, respectively. Second, considering multiple flows, we observe that the ETM metric load-balances traffic successfully. The throughput improvements with ETM are 253% and 368%, as compared with ETOP and ETT, respectively.

The paper is organized as follows. Related work is described in Section II. In Section III, we analytically compute ETM. We formulate the Optimal Path and the Optimal Rate Problems and propose solutions for each in Section IV. Our evaluations are in Section V. We conclude the paper in Section VI.

II. Related Work

In this section, we first review the relevant related routing metrics. We then discuss related work on rate control.

Link quality based routing metrics: There are several previously proposed link quality based routing metrics.

The ETX metric: Proposed by De Couto *et al.* [2], ETX computes the expected number of transmissions (including retransmissions) needed to send a packet over a link, by measuring the forward and reverse packet delivery ratios (PDR) between a pair of neighboring nodes. However, when computing ETX an infinite number of retransmissions are implicitly assumed at the link layer. Furthermore, ETX does not consider multiple transmission rates. Measurements on wireless testbeds [14], [2] show that the use of ETX as a routing metric results in paths that yield higher throughputs than with minimum hop count based routing.

The ETT metric: The ETT metric was proposed in [3] for multi-rate wireless networks. The ETT on a given link is defined as the expected time to send a 1500-byte packet at the rate that yields the highest throughput on that link. ETT also accounts for the time taken for retransmissions at each rate. The ETT cost of a route is the sum of the ETTs of each link on the route. ETT implicitly assumes (as with ETX) an infinite number of retransmissions on each link.

The ETOP metric: The ETOP metric proposed in [1] accounts for the finite number of retransmission attempts at the link layer. The authors identify that due to this, packet drops closer to the destination can be costly. However, the ETOP metric has the following limitations. **First**, ETOP estimates the link quality by using probes broadcasted at the basic rate. This results in paths that may not support high rates. **Second**, ETOP uses *small* probe packets and this does not accurately reflect the loss rates for the larger data packets; this can lead to the selection of *lossy* links even close to the destination. **Third**, the framework with ETOP was not designed to respond to link quality fluctuations *in between* route changes. Thus, even though chosen routes account for link positions, unforeseen retransmissions could result with temporal variations in link quality due to the improper choice of transmission rates by an underlying rate adaptation mechanism (these mechanisms do not account for link positions). Such packet drops close to the destination will induce costly e2e retransmissions.

Other Related Efforts: In [3], Draves *et al.* propose a new routing metric, WCETT to take the intra-flow interference into account. Koksal *et al.* [15] propose mETX and ENT; these metrics extend ETX to account for highly variable link reliabilities. Other efforts that attempt to reduce energy consumption due to retransmission costs in a mesh network setting include [16], [17]. The problem of load balancing in mesh networks has been studied in [11], [12]. Unlike in our work, none of the above methods account for the finite number of retransmissions at the link layer, multi-rate capabilities, and the impact of queuing delay together.

Rate Control: Rate control mechanisms that adapt to link quality variations are proposed in [5], [6]. In [6] an approach which controls the sender's rate dynamically to improve responsiveness to channel variations is proposed. Choi *et al.* [5] consider collision effects on rate control. There are also proposals that utilize control message exchanges to determine the signal to noise ratio (SNR), which is then used to apply rate control (for e.g. [7], [8]). The Auto Rate Fallback (ARF) protocol [9] is the first commercial implementation that exploits the multi-rate capability of an IEEE 802.11 network. Implemented on MIT's Roofnet [4], SampleRate selects the data rate that has the smallest average per-packet transmission time in order to achieve high throughput. Onoe [10] is implemented in the 802.11 device driver for Atheros cards in Linux and Free BSD. However, unlike our approach, the above rate control approaches operate on a per-link basis and do not account for the impact of position of the link on the rate. Our framework meets this requirement.

III. Computing the ETM metric

In this section, we analytically compute an expression for the ETM cost of a path.

System Model: We assume that each link can support R transmission rates. Note that IEEE 802.11a supports a total of eight transmission rates. We interchangeably use R to denote both the set of possible rates and the cardinality of the set. For each rate r $(\in R)$ there is an associated probability p_r of a packet transmission success. At the link layer the number of transmission attempts is limited to K; if a packet fails in K successive attempts, the link layer drops the packet. We assume that such a packet drop induces a transport layer retransmission (with a protocol like TCP) from the source. Consider the traversal of a packet over a path $(v_0, \cdots, v_n)$ consisting of $n+1$ nodes (correspondingly, n links). We label the link between node v_{i-1} and node v_i as link i. Let the transmission rate used on the link between nodes v_{i-1} and v_i be r_i and the corresponding probability of successfully delivering the packet across the link in a single attempt be p_{r_i}. First, the source node v_0 initiates an end-to-end (e2e) connection. A transmission is then performed at the link layer by selecting a rate r_1 out of the R available rates for link 1. If the packet is received successfully within K transmission attempts by node v_1, then v_1 initiates a forwarding of the packet to node v_2, and so forth. However, if all K transmission attempts on say link i, fail, then the packet will be dropped by the link layer of node v_{i-1}. The packet drop causes the transport layer of the source node v_0 to initiate an e2e retransmission. In addition, each node is assumed to uses a FIFO queue for outbound packets. Our interest then, is in answering the following question: *given a path* $(v_0, \cdots, v_n)$*, the corresponding rate set* $(r_1, \cdots, r_n)$ *and the associated delivery probability set* $(p_{r_1}, \cdots, p_{r_n})$*, what is the expected transmission time required for a packet to be successfully delivered end to end (from* v_0 *to* v_n*)?*

Analytically computing ETM: Let z_n denote the random variable representing the number of e2e transmissions required in order for the packet to be delivered to the destination on an n-link path $(v_0, \cdots, v_n)$. Let h_i denote the number of consecutive hops that are successfully traversed along the path, in the i^{th} e2e transmission attempt. Thus, $h_i = 0$ if the packet fails to reach node v_1 from node v_0, and $h_i = n$ if the packet reaches the destination, v_n; $h_i < n$ indicates that the $(i+1)^{th}$ e2e retransmission is attempted. We assume that the random variables $h_1, h_2, \cdots$ are independent and identically distributed (**i.i.d**). Since the fading can be expected to be short-term [18], this assumption is reasonable. With this, we represent the variables by a single random variable h. Let l_{i,r_j} denote the number of link layer transmissions needed to deliver the packet over link j with data rate r_j in the i^{th} e2e transmission attempt. If the packet has successfully traversed link j, we have $l_{i,r_j} \leq K$. Otherwise, $l_{i,r_j} = K$ and a new e2e transmission attempt is started at node v_0. For each node v_j, we assume that $l_{1,r_j}, l_{2,r_j}, \cdots$ are *i.i.d* random variables, and the notation l_{r_j} is used to represent this common random variable.

In calculating the time taken for a packet to be delivered over a link, we account for the following three components: (i) the time for transmitting a packet, (ii) the time for which the node backs-off between transmission attempts, and (iii) the average queuing delay experienced by a packet at the transmitter node.

For calculating (i), we assume that the size of a data packet is fixed, and the transmission of a packet at rate r takes T_r seconds [19]. For the calculation of (ii), let $B(k)$ be the sum of the expected times spent in back-off over the k retransmission attempts, including the random period prior to the initial transmission attempt. The 802.11 MAC randomly selects the back-off window CW_i for the i^{th} retransmission attempt from the interval $(0, 2^{i-1}CW_{min})$, where CW_{min} is the minimum back-off window size. The interval increases to $(0, CW_{max})$ with the attempts, and after that, there is no longer an increase; CW_{max} is the maximum back-off window size. We assume that $CW_{max} = 2^6\ CW_{min}$ conforming to the IEEE 802.11 standard [20]. Then, as derived in [3], we can express $B(k)$ as:

$$\begin{aligned} B(k) &= T_{slot} \sum_{i=1}^{k} \mathbf{E}\left[CW_i\right] \\ &= \begin{cases} \frac{CW_{min}}{2}(2^k - 1)T_{slot}, & k \leq 7 \\ \frac{CW_{min}}{2}\{63 + 64(k-7)\}T_{slot}, & k \geq 8, \end{cases} \end{aligned} \tag{1}$$

where T_{slot} denotes the size of each slot in time units. Unfortunately, the calculation of (iii) is not straightforward; the variation in queuing delay depends on the network dynamics (e.g., the traffic load) and the underlying MAC protocol. Statistical models [21], [22] have been applied to analyze the queuing behavior in a wireless network using the 802.11 MAC protocol. In this work, we simply assume that measured values of the average queuing delays are available (We discuss how this is measured in our implementation later.).

Let the queuing delay experienced by a packet at node j be q_j. Then, if it is delivered over link j after l_{r_j} transmission attempts, the time spent on link j is $l_{r_j}T_{r_j} + B(l_{r_j}) + q_j$.

For a n-link path, the total cost $\mathbf{C}(\ {}_n)$ for successfully transmitting a packet with rates $\ {}_n = (r_1, r_2, \cdots, r_n)$ over the path, consists of the cost due to $z_n - 1$ unsuccessful transport layer transmission attempts and the cost incurred with the one last successful transmission, and is formally given by:

$$\begin{aligned} \mathbf{C}(\ {}_n) = \sum_{i=1}^{z_n - 1} \Bigg\{ \sum_{j=1}^{h_i} \Big(l_{r_j}T_{r_j} + B(l_{r_j}) + q_j \Big) + KT_{r_{h_i+1}} + B(K) \\ + q_{h_i+1} \Bigg\} + \sum_{j=1}^{h_{z_n}} \Big(l_{r_j}T_{r_j} + B(l_{r_j}) + q_j \Big). \end{aligned} \tag{2}$$

Note that with z_n set to 1, Eq. (2) reduces to the the ETT cost. From Eq. (2), we derive the ETM cost below.

Proposition 1. *The expected cost,* $\mathbf{E}\left[\mathbf{C}(\ {}_n)\right]$*, for delivering a packet over path* $(v_0, \cdots, v_n)$ *with rates* $\ {}_n = (r_1, \cdots, r_n)$ *is*

$$\begin{aligned} &\mathbf{E}\left[\mathbf{C}(\ {}_n)\right] \\ &= \mathbf{E}\left[z_n - 1\right] \Bigg\{ \sum_{j=1}^{n-1} \Big(L_{r_j}T_{r_j} + B(L_{r_j}) + Q_j \Big) \mathbf{P}\left[h > j-1 | h < n\right] \\ &+ \sum_{j=1}^{n} \Big(KT_{r_j} + B(K) + Q_j \Big) \mathbf{P}\left[h = j-1 | h < n\right] \Bigg\} \\ &+ \sum_{j=1}^{n} \Big(L_{r_j}T_{r_j} + B(L_{r_j}) + Q_j \Big), \end{aligned} \tag{3}$$

where, $L_{r_j} = \mathbf{E}\left[l_{r_j} | l_{r_j} \leq K\right]$ *and* $Q_j = \mathbf{E}\left[q_j\right]$.

Proof: By taking the expectation on both sides of Eq. (2), and replacing the term within the first summation on the right-hand side of Eq. (2) with Λ_i, we have

$$\mathbf{E}\left[\mathbf{C}(\ {}_n)\right] = \mathbf{E}\left[\sum_{i=1}^{z_n - 1} \Lambda_i\right] + \mathbf{E}\left[\sum_{j=1}^{h_{z_n}} \Big(l_{r_j}T_{r_j} + B(l_{r_j}) + q_j \Big)\right]. \tag{4}$$

We derive the two terms of Eq. (4) separately. Since z_n is independent of Λ_i (e2e attempts experience i.i.d losses), $\mathbf{E}\left[\sum_{i=1}^{z_n-1}\Lambda_i\right]$ reduces to $\mathbf{E}[z_n-1]\,\mathbf{E}[\Lambda]$.

Omitting the index relating to the e2e transmission attempt i (since these attempts are i.i.d):

$$\begin{aligned}&\mathbf{E}[\Lambda]\\ &=\mathbf{E}\left[\sum_{j=1}^{h}\left(l_{r_j}T_{r_j}+B(l_{r_j})+q_j\right)+KT_{r_{h+1}}+B(K)+q_{h+1}\right].\end{aligned} \tag{5}$$

Conditioning on h (the number of links traversed during an unsuccessful attempt) in Eq. (5), we have

$$\begin{aligned}\mathbf{E}[\Lambda]=\sum_{i=0}^{n-1}&\left\{\sum_{j=1}^{i}\left(\mathbf{E}\left[l_{r_j}|l_{r_j}\le K\right]T_{r_j}\right.\right.\\ &\left.\left.+\mathbf{E}\left[B\left(\mathbf{E}\left[l_{r_j}|l_{r_j}\le K\right]\right)\right]+\mathbf{E}[q_j]\right)\right\}\mathbf{P}[h=i|h<n]\\ &+\sum_{i=0}^{n-1}\left(KT_{r_{i+1}}+B(K)+\mathbf{E}[q_{i+1}]\right)\mathbf{P}[h=i|h<n],\end{aligned} \tag{6}$$

By expanding the first term on the right-hand side of Eq. (6), and using the relation $\sum_{j=i}^{n-1}\mathbf{P}[h=j|h<n]=\mathbf{P}[h>i-1|h<n]$, we have

$$\begin{aligned}\mathbf{E}[\Lambda]=\sum_{i=1}^{n-1}&\left\{\mathbf{E}[l_{r_i}|l_{r_i}\le K]\,T_{r_i}+\mathbf{E}[B(\mathbf{E}[l_{r_i}|l_{r_i}\le K])]\right.\\ &\left.+\mathbf{E}[q_i]\right\}\mathbf{P}[h>i-1|h<n]+\sum_{i=0}^{n-1}\left(KT_{r_{i+1}}+B(K)\right.\\ &\left.+\mathbf{E}[q_{i+1}]\right)\mathbf{P}[h=i|h<n].\end{aligned} \tag{7}$$

Note that in the above expression, $l_{r_j}\le K$ if a link j is successfully traversed with rate r_j. Furthermore, in simplifying Eq. (6) to Eq. (7) $\sum_{j=1}^{0}=0$ is used.

Since $h_{z_n}=n$, the second term in Eq. (4) reduces to:

$$\begin{aligned}\mathbf{E}\left[\sum_{j=1}^{h_{z_n}}\left(l_{r_j}T_{r_j}+B(l_{r_j})+q_j\right)\right]=\sum_{j=1}^{n}&\left(\mathbf{E}\left[l_{r_j}|l_{r_j}\le K\right]T_{r_j}\right.\\ +\mathbf{E}\left[B\left(\mathbf{E}\left[l_{r_j}|l_{r_j}\le K\right]\right)\right]&\left.+\mathbf{E}[q_j]\right).\end{aligned} \tag{8}$$

Inserting Eq. (7) and (8) into Eq. (4) and using the definition of L_{r_j} and Q_j, Eq. (3) is obtained. ■

Now we estimate L_{r_j}, $\mathbf{P}[h>j-1|h<n]$ and $\mathbf{P}[h=j-1|h<n]$ (in Eq. (3)) in order to express $\mathbf{C}(\ _n)$ in terms of the link success probabilities for each rate. From its definition, L_{r_j} can be further expressed as:

$$L_{r_j}=\sum_{i=1}^{K}\frac{i\cdot\mathbf{P}\left(l_{r_j}=i\right)}{1-\mathbf{P}\left(l_{r_j}>K\right)}=\sum_{i=1}^{K}\frac{i\cdot\left(1-p_{r_j}\right)^{i-1}p_{r_j}}{1-\left(1-p_{r_j}\right)^{K+1}}. \tag{9}$$

We define π_{r_j} to be the probability that a packet transmitted with data rate r_j is not dropped on link j, i.e., $\pi_{r_j}=1-(1-p_{r_j})^K$. With this, the probability that a packet transmitted with rates $(r_1,\cdots,r_i)$ is delivered over i successive hops is given by $\rho_i=\pi_{r_1}\times\cdots\times\pi_{r_i}$. We then obtain $\mathbf{P}[h>j]=\pi_{r_1}\times\cdots\times\pi_{r_{j+1}}=\rho_{j+1}$. In particular, from $\mathbf{P}[h\ge j]=\rho_j$, $\rho_0=1$ is induced. Furthermore, since $z_n=l$ indicates that there are $l-1$ transport layer transmission failures before the first success, z_n has a geometric distribution with parameter $\mathbf{P}[h\ge n]=\rho_n$. With the new variables defined above, $\mathbf{P}[h>j-1|h<n]$ and $\mathbf{P}[h=j-1|h<n]$ in Eq. (3) can be expressed as follows:

$$\mathbf{P}(h>j-1|h<n)=\frac{\mathbf{P}(h>j-1)-\mathbf{P}(h\ge n)}{1-\mathbf{P}(h\ge n)}=\frac{\rho_j-\rho_n}{1-\rho_n}. \tag{10}$$

and

$$\begin{aligned}\mathbf{P}(h=j-1|h<n)&=\frac{\mathbf{P}(h\ge j-1)-\mathbf{P}(h>j-1)}{1-\mathbf{P}(h\ge n)}\\ &=\frac{\rho_{j-1}-\rho_j}{1-\rho_n}.\end{aligned} \tag{11}$$

With these new simplifications, we now re-express $\mathbf{C}(\ _n)$ in Theorem 1, in terms of the link success probabilities.

Theorem 1. *The expected cost,* $\mathbf{E}[\mathbf{C}(\ _n)]$*, for delivering a packet with rates* $\ _n=(r_1,\cdots,r_n)$ *over path* $(v_0,\cdots,v_n)$ *is:*

$$\begin{aligned}\mathbf{E}[\mathbf{C}(\ _n)]=\sum_{j=1}^{n}&\left\{\frac{\rho_j}{\rho_n}\left(L_{r_j}T_{r_j}+B\left(L_{r_j}\right)+Q_j\right)\right.\\ &\left.+\frac{\rho_{j-1}-\rho_j}{\rho_n}\left(KT_{r_j}+B(K)+Q_j\right)\right\}.\end{aligned} \tag{12}$$

Proof: We substitute Eq. (10) and Eq. (11) into Eq. (3). Since z_n follows a geometric distribution, Eq. (3) reduces to:

$$\begin{aligned}\mathbf{E}[\mathbf{C}(\ _n)]=\left(\frac{1}{\rho_n}-1\right)&\left\{\sum_{j=1}^{n-1}\frac{\rho_j-\rho_n}{1-\rho_n}\left(L_{r_j}T_{r_j}+B(L_{r_j})+Q_j\right)\right.\\ &\left.+\sum_{j=1}^{n}\frac{\rho_{j-1}-\rho_j}{1-\rho_n}\left(KT_{r_j}+B(K)+Q_j\right)\right\}\\ &+\sum_{j=1}^{n}\left(L_{r_j}T_{r_j}+B(L_{r_j})+Q_j\right)\\ =&\sum_{j=1}^{n}\frac{\rho_j}{\rho_n}\left(L_{r_j}T_{r_j}+B(L_{r_j})+Q_j\right)\\ &+\sum_{j=1}^{n}\frac{\rho_{j-1}-\rho_j}{\rho_n}\left(KT_{r_j}+B(K)+Q_j\right).\end{aligned} \tag{13}$$

Putting both the terms on the right-hand side in the last equation together within a single summation yields Eq. (12). ■

Theorem 2. *ETM satisfies the following recursive equation:*

$$\begin{aligned}\mathbf{E}[\mathbf{C}(\ _{n+1})]=&\frac{\mathbf{E}[\mathbf{C}(\ _n)]}{\pi_{r_{n+1}}}+L_{r_{n+1}}T_{r_{n+1}}+B(L_{r_{n+1}})+Q_{n+1}\\ &+\frac{1-\pi_{r_{n+1}}}{\pi_{r_{n+1}}}\left(KT_{r_{n+1}}+B(K)+Q_{n+1}\right).\end{aligned} \tag{14}$$

where $\mathbf{E}[\mathbf{C}(\ _0)]=0$.

Proof: Let $\mathbf{E}[\mathbf{C}(\ _{n+1})]$ be the cost of path $(v_0,\cdots,v_{n+1})$. From Eq. (12), $\mathbf{E}[\mathbf{C}(\ _{n+1})]$ is given by:

$$\mathbf{E}\left[\mathbf{C}(\ {}_{n+1})\right] = \frac{1}{\pi_{r_{n+1}}} \sum_{j=1}^{n} \left\{ \frac{\rho_j}{\rho_n} \left(L_{r_j} T_{r_j} + B\left(L_{r_j}\right) + Q_j \right) + \frac{\rho_{j-1} - \rho_j}{\rho_n} \left(KT_{r_j} + B(K) + Q_j \right) \right\} + L_{r_{n+1}} T_{r_{n+1}} + B\left(L_{r_{n+1}}\right) + Q_{n+1} + \frac{\rho_n - \rho_{n+1}}{\rho_{n+1}} \left(KT_{r_{n+1}} + B(K) + Q_{n+1} \right) = \frac{\mathbf{E}\left[\mathbf{C}(\ {}_{n})\right]}{\pi_{r_{n+1}}} + L_{r_{n+1}} T_{r_{n+1}} + B(L_{r_{n+1}}) + Q_{n+1} + \frac{1 - \pi_{r_{n+1}}}{\pi_{r_{n+1}}} \cdot \left(KT_{r_{n+1}} + B(K) + Q_{n+1} \right). \tag{15}$$

■

IV. Routing and Rate Adaptation with ETM

In this section, we formulate the Optimal Path Problem (**OPP**) and the Optimal Rate Problem (**ORP**), alluded to earlier in Section I. The solution to OPP would find the best path with the least ETM cost. The solution to ORP should select the best rates for each link on a given path. Although the metric is non-commutative, a greedy algorithm can solve OPP; the solution to ORP is based on a dynamic-programming approach.

A. Our Route Selection Algorithm

Formally OPP may be posed as follows: *Given a node pair, find the path between the pair with the minimum ETM cost.*

Lemma 1. *OPP satisfies the greedy-choice property.*

Proof: Consider a path $(v_0, \cdots, v_{n+1})$ with its corresponding rate set $\ {}_{n+1} = (r_1, \cdots, r_{n+1})$. The cost $\mathbf{E}\left[\mathbf{C}(\ {}_{l+1})\right]$ is given by Eq. (14). We see that the cost for the path $(v_0, \cdots, v_{n+1})$ is expressed in terms of (i) the cost for the sub-path $(v_0, \cdots, v_n)$, $\mathbf{C}(\ {}_{n})$, (ii) the weight $p_{r_{n+1}}$ of the link (v_n, v_{n+1}) with rate r_{n+1}, and (iii) the expected queuing delay Q_{n+1} at the sender of the link (v_n, v_{n+1}). Thus, the solution to finding the path $(v_0, \cdots, v_{n+1})$ is possible by greedy selection over the link to be newly added viz., (v_n, v_{n+1}). This implies that a global optimal solution can be arrived at by making a locally optimal choice; in other words the greedy-choice property is satisfied. ■

Lemma 2. *OPP satisfies the optimal sub-structure property i.e., the sub-path of an optimal path is itself optimal.*

Proof: We prove this lemma by contradiction. Let us assume that $\ {}^{*}_{l+1} = (v_0, \cdots, v_l^*, v_{l+1})$ is the optimal path from the source v_0 to the destination v_{l+1} i.e., its path cost, $\mathbf{E}\left[\mathbf{C}(\ {}^{*}_{l+1})\right]$ is the minimum. We assume that for the path upto node v_l^* in $\ {}^{*}_{l+1}$, the optimal substructure property does not hold. In other words, the cost $\mathbf{E}\left[\mathbf{C}(\ {}^{*}_{l})\right]$ for the path $\ {}^{*}_{l} = (v_0, \cdots, v_l^*)$ is not optimal. Thus, there must exist a path $\ {}'_{l}$ from v_0 to v_l^* with rate set $\ '$ and with cost $\mathbf{E}\left[\mathbf{C}(\ {}'_{l})\right]$ satisfying $\mathbf{E}\left[\mathbf{C}(\ {}'_{l})\right] < \mathbf{E}\left[\mathbf{C}(\ {}^{*}_{l})\right]$.

We can then construct a new path $\ {}'_{l+1}$ from v_0 to v_{l+1} by replacing $\ {}^{*}_{l}$ in the previously considered $\ {}^{*}_{l+1}$ with $\ {}'_{l}$; its path cost $\mathbf{E}\left[\mathbf{C}(\ {}'_{l+1})\right]$ then satisfies the following inequality:

$$\mathbf{E}\left[\mathbf{C}(\ {}'_{l+1})\right] = \frac{\mathbf{E}\left[\mathbf{C}(\ {}'_{l})\right]}{\pi_{r^*_{l+1}}} + L_{r^*_{l+1}} T_{r^*_{l+1}} + B(L_{r^*_{l+1}}) + Q^*_{l+1} + \frac{1 - \pi_{r^*_{l+1}}}{\pi_{r^*_{l+1}}} \left(KT_{r^*_{l+1}} + B(K) + Q^*_{l+1} \right) < \mathbf{E}\left[\mathbf{C}(\ {}^{*}_{l+1})\right]. \tag{16}$$

However, this contradicts the assumption that $\mathbf{E}\left[\mathbf{C}(\ {}^{*}_{l+1})\right]$ is the minimum cost from v_0 to v_{l+1}. ■

Theorem 3. *OPP is optimally solved by a greedy algorithm.*

Proof: An optimization problem can be solved by a greedy algorithm if the greedy choice property and the optimal substructure property hold [23]. The proof thus follows from Lemma 1 and Lemma 2. ■

Based on the above, we design an optimal route selection algorithm. We modify the well known Dijkstra's algorithm for finding the shortest path between any two nodes. In the original Dijkstra's algorithm, the path cost $dist[v]$ from the source node to node v via node u is given by $dist[v] = dist[u] + cost(u, v)$, where $cost(u, v)$ is the link cost between u and v. In our case, we use the following equation for computing $dist[v]$:

$$dist[v] = \min_{r_{u,v} \in R} \left\{ \frac{dist[u]}{\pi_{r_{u,v}}} + L_{r_{u,v}} T_{r_{u,v}} + B(L_{r_{u,v}}) + Q_{u,v} + \frac{1 - \pi_{r_{u,v}}}{\pi_{r_{u,v}}} \left(KT_{r_{u,v}} + B(K) + Q_{u,v} \right) \right\}, \tag{17}$$

where $r_{u,v}$ and $\pi_{r_{u,v}}$ are the rate from u to v and the probability that the packet is not dropped within K attempts from u to v when transmitted at rate $r_{u,v}$, respectively. The minimization in Eq. (17) ensures the selection of the best rate between u and v (at the time of path computation), and thereby the minimum ETM cost from the source to v. The proposed path selection algorithm is represented in Algorithm 1 below.

B. Our Rate Selection Algorithm

Once a route is determined between a node pair, the route is used for a while (since frequent route changes can be overhead intense). However, the rates determined by the routing algorithm may not be optimal for the lifetime of the route. To cope with this, we propose an ETM-based rate adaptation algorithm that tunes the rate dynamically; our algorithm implicitly accounts for the link positions on the path to ensure that packet drops closer to the destination are less likely. In particular, here we address the ***Optimal Rate Problem (ORP)****: Given a path $(v_0, \cdots, v_n)$, determine the rate set $\ {}_{n}$ that minimizes the expected transmission cost for the path,* $\mathbf{E}\left[\mathbf{C}(\ {}_{n})\right]$.

Lemma 3. *ORP satisfies the overlapping property, i.e., the problem can be broken down into smaller subproblems that retain the same structure.*

Proof: Formally, ORP can be expressed as:

Algorithm 1: Path Algorithm

```
Input: Graph $G(V, E)$ and Source $s$
Initialization: $dist[u] \leftarrow \infty$ and $prev[u] \leftarrow \emptyset$ for $\forall\ u \in V/\{s\}$,
$dist[s] \leftarrow 0$, $rate[u, v] \leftarrow 0$, and $\Phi \leftarrow V$;
/* $prev[u]$ is the node that immediately follows $u$,
and $v$ is the neighbor of $u$                                  */;
while $\Phi \neq \emptyset$ do
    $u \leftarrow MIN(\Phi)$;
    /* $MIN(\Phi)$ is the element in $\Phi$ that has the
    least path cost                                             */;
    $\Phi \leftarrow \Phi - \{u\}$;
    for each neighbor $v$ of $u$ do
        $tmp \leftarrow \min_{r_{u,v} \in R} \Big\{ \frac{dist[u]}{\pi_{r_{u,v}}} + L_{r_{u,v}} T_{r_{u,v}} + B(L_{r_{u,v}}) +$
        $+ Q_{u,v} \frac{1-\pi_{r_{u,v}}}{\pi_{r_{u,v}}} \Big( KT_{r_{u,v}} + B(K) + Q_{u,v} \Big) \Big\}$;
        if $tmp < dist[v]$ then
            $dist[v] \leftarrow tmp$;
            $prev[v] \leftarrow u$;
            $rate[u, v] \leftarrow$
            $\text{argmin}_{r_{u,v} \in R} \Big\{ \frac{dist[u]}{\pi_{r_{u,v}}} + L_{r_{u,v}} T_{r_{u,v}} + B(L_{r_{u,v}}) +$
            $Q_{u,v} + \frac{1-\pi_{r_{u,v}}}{\pi_{r_{u,v}}} \Big( KT_{r_{u,v}} + B(K) + Q_{u,v} \Big) \Big\}$;
        end
    end
end
```

$$\min_{{}_{n} \in R^{n}} \mathbf{E}\left[\mathbf{C}\left(\ \ {}_{n}\right)\right] = \min_{r_n \in R} \left\{ \frac{\min_{\ \ {}_{n-1} \in R^{n-1}} \mathbf{E}\left[\mathbf{C}_{n-1}(\ \ {}_{n-1})\right]}{\pi_{r_n}} \right.$$
$$\left. + L_{r_n} T_{r_n} + B(L_{r_n}) + Q_n + \frac{1 - \pi_{r_n}}{\pi_{r_n}} \left(KT_{r_n} + B(K) + Q_n\right) \right\}, \tag{18}$$

where R^n is an n-dimensional vector space over the set of rates R. From Eq. (18), computing $\min_{\ \ {}_{n} \in R^n} \mathbf{E}[\mathbf{C}(\ \ {}_{n})]$ requires the calculation of $\min_{\ \ {}_{n-1} \in R^{n-1}} \mathbf{E}[\mathbf{C}(\ \ {}_{n-1})]$. In other words, the ORP for the path $(v_0, \cdots, v_n)$ can be solved if the sub-problem (again an ORP) for the path $(v_0, \cdots, v_{n-1})$ is solved. ■

Lemma 4. *ORP satisfies the optimal substructure property: if we have the optimal rate set for a problem, then the associated rate set for each sub-problem is also optimal.*

Proof: The proof is similar to that of Lemma 2; we omit the details due to space limitations. ■

Theorem 4. *ORP is solvable by dynamic programming.*

Proof: An optimization problem can be solved by dynamic programming if the problem satisfies both the overlapping and the optimal substructure properties [23]. The proof is immediate from Lemmas 3 and 4. ■

Based on the above properties of ORP, we propose an optimal rate selection algorithm using a dynamic programming technique. Our rate selection algorithm chooses a rate for link j (for $j = 1, \cdots, n$) so as to satisfy the following equation:

$$\mathcal{V}_j = \min_{r_j \in R} \left\{ \frac{\mathcal{V}_{j-1}}{\pi_{r_j}} + L_{r_j} T_{r_j} + B(L_{r_j}) + Q_j \right.$$
$$\left. + \frac{1 - \pi_{r_j}}{\pi_{r_j}} \Big(KT_{r_j} + B(K) + Q_j\Big) \right\}, \tag{19}$$

where $\mathcal{V}_0 = 0$ for $j = 0$ and $\mathcal{V}_j$ is the expected cost for delivering a packet from node v_0 to node v_j. The rate selection algorithm is provided in Algorithm 2 below.

Note that given the graph $G(V, E)$, it can be easily shown that the complexities of Algorithm 1 and Algorithm 2 are $O(|V|^2)$ and $O(|V|)$, respectively [23].

Algorithm 2: Rate Algorithm

```
Input: $\mathcal{V}_j$
Output: $\mathcal{V}_{j+1}$
foreach $v_j$ in the path $(v_0, \cdots, v_n)$ do
    $\mathcal{V}_{j+1} \leftarrow \min_{r_{j+1} \in R} \Big\{ \frac{\mathcal{V}_j}{\pi_{r_{j+1}}} + L_{r_{j+1}} T_{r_{j+1}} + B(L_{r_{j+1}}) +$
    $Q_{j+1} + \frac{1-\pi_{r_{j+1}}}{\pi_{r_{j+1}}} \Big( KT_{r_{j+1}} + B(K) + Q_{j+1} \Big) \Big\}$;
    $r_{j+1} \leftarrow \text{argmin}_{r_{j+1} \in R} \Big\{ \frac{\mathcal{V}_j}{\pi_{r_{j+1}}} + L_{r_{j+1}} T_{r_{j+1}} +$
    $B(L_{r_{j+1}}) + Q_{j+1} + \frac{1-\pi_{r_{j+1}}}{\pi_{r_{j+1}}} \Big( KT_{r_{j+1}} + B(K) + Q_{j+1} \Big) \Big\}$;
end
```

V. Implementation and Evaluations

In this section, we first describe the implementation of a framework based on routing and rate adaptation with ETM, the ETM-framework. Then, we compare its performance with that of the ETOP-and ETT-frameworks. The term ETT-framework (ETOP-framework) refers to the combination of ETT-routing (ETOP-routing) and the original SampleRate algorithm. To begin with, we set the queueing delay $Q_j = 0$. Our goal is to exclusively capture the retransmission costs. The impact of the queuing delay is subsequently evaluated.

Testbed Description: Our wireless testbed consists of 21 Soekris net5501 nodes, which run a Debian Linux distribution with kernel v2.6.16.19. Each node is equipped with a CM9 802.11 a/b/g miniPCI card [24], which embeds the Atheros AR5213 chipset and an omnidirectional antenna of 5 dBi gain. We use the Roofnet module in the popular Click toolkit [13]. We modify the MadWifi-0.9.3 driver [10] so that it can recognize RAW packets generated by Click. Experiments are performed with the 802.11a mode to avoid interference from co-located 802.11b/g networks. We disable the RTS/CTS messages (as is commonly done [2]), and the default long retry limit of 7 is used by the wireless cards.

Implementation of ETM-routing: We implement the ETM-routing as a modified version of Srcr [4], proposed for Roofnet; it is similar to the DSR protocol [25]. Each node periodically broadcasts a 1500-byte probe packet at each possible data rate and a 60-byte packet at the basic rate. The larger probes reflect data transmissions and the smaller probes characterize ACK transmissions. Each probe contains queuing delay information at the broadcasting node in its header. Each node maintains a cache of all the known routes, the average queuing delay at the nodes on the routes, and the delivery rates of the probes over their corresponding links. Whenever the source node needs to send a packet, it first checks if the destination is in the cache. If so, it runs Algorithm 1, proposed in Section IV, on the graph constructed with the topology in the cache to find the route with the minimum ETM cost. When each node on the route forwards a packet, it includes its latest queuing delay and the current delivery rates of the probe packets between itself and the previous node in the packet header; thus, at these times all the nodes on the route update the rate vectors and the queuing delay information in their caches. To avoid route flapping, average queuing delays are only updated if the change is higher than a preset threshold.

If the source cannot find the destination in its cache, it invokes a network-wide query. Every node which receives the query, inserts its own address, its queuing delay, and probe delivery statistics in the packet header and re-broadcasts it. When the destination receives the query, it responds on the reverse path corresponding to the route via which the packet was received and contains the information that was obtained in the query. The source updates its information and runs Algorithm 1 to compute a new route to the destination.

Estimating the mean queuing delays: Each packet is time-stamped when it is enqueued and when it is to be transmitted. The difference between the two time-stamps represents the queuing delay of the packet. Let $x_{j,t}$ be the measured queuing delay of the t^{th} enqueued packet at the transmitter of a link j. The average queuing delay Q_j for node j is estimated by using the following exponential smoothing formula:

$$Q_j = \alpha x_{j,t} + (1 - \alpha) s_{j,t-1}, \tag{20}$$

where α is the smoothing factor ($0 < \alpha < 1$). $s_{j,t-1}$ is the the weighted mean of the queuing delays of the last k enqueued packets; it is computed as $s_{j,t-1} = \sum_{n=1}^{k} \omega_n x_{j,t-n}$, where the weighted factors ω_n satisfy the property $\sum_{n=1}^{k} \omega_n = 1$. If $\alpha \approx 1$ there is a lesser extent of smoothing and recent changes are considered more important. In our experiments, we choose $\alpha = 0.5$ to balance the two factors.

Implementation of ETM-rate adaptation: We implement the ETM-rate adaptation module on top of SampleRate [4]. SampleRate periodically invokes transmissions of 1500-byte data packets at a randomly chosen bit-rate; it computes the packet delivery ratio (PDR) and uses this to compute the rate that yields the best throughput. The PDRs are also updated based on actual data transmissions (much faster time scale) and can thus, respond to the time-varying channel quality more quicky and accurately. We modify the SampleRate algorithm as follows. Whenever a node v_j wants to transmit a packet to its neighbor v_{j+1}, it runs Algorithm 2, proposed in Section IV, using the statistics obtained with the native SampleRate algorithm. It obtains r_{j+1} and $\mathcal{V}_j$ and includes this information in the packet header and sends the packet at rate r_{j+1}. The rate adapts to cope with temporal changes in link quality in the short term; as discussed, our approach (Algorithm 2) takes into account the position of the link on the path.

Implementation Complexity: The use of ETM requires the embedding of additional information (rate vector and the queuing delay for ETM-routing and the information on r_{j+1} and $\mathcal{V}_j$ for ETM-rate adaptation) in the packet header. As compared to ETT or ETOP, this adds an overhead of $2 \sim 5\%$ depending on path length. In particular, the information embedded with ETT (or ETOP) is of the order of 14 bytes while it is 20 bytes with ETM; compared to a packet size of 1500 bytes, we believe that this is sufficiently small. Additional processing delays due to ETM are not noticable in our implementation.

TABLE I
MEDIAN TCP THROUGHPUT VERSUS THE WEIGHTED AVERAGE PATH LENGTH FOR ALL 70 PAIRS

Path	# of	Median TCP Throughput (Kbps) / WPL		
Length	Pairs	ETOP	ETT	ETM
All length	70	1275 / 2.03	2255 / 2.18	2940 / 2.22
1-hop	26	18000 / 1.01	16250 / 1.13	17600 / 1.09
2-hops	22	1125 / 2.00	1955 / 2.19	2195 / 2.27
3-hops	18	586 / 3.10	979 / 3.20	1250 / 3.27
4-hops	4	379 / 4.00	575 / 4.31	783 / 4.39

Performance of TCP Flows: We first consider the impact of the three frameworks on the performance of TCP connections.

Setup: In our first set of experiments, 70 source-destination pairs are randomly chosen out of the 420 (= 21×20) possible pairs; 26 pairs are separated by a single-hop and 44 pairs are separated by multiple hops. At the start of an experiment, Roofnet runs for 20 sec in order to reach stable operations. Then, a source pings its destination for 20 sec at a rate of one packet per second. With this, Roofnet discovers the paths to the destination. The source then initiates a TCP connection with the destination, and sends data for 3 minutes. The maximum achieved TCP throughput is measured using Iperf [26]. The results with each framework are obtained within minutes of each other; we expect the slower time-scale channel conditions to have changed little during this time, which is something that we experimentally observe during our experiments. We repeat the experiment seven times and compute an average to reduce the impact of temporal variations.

TCP performance of each framework: In Table I we tabulate the median throughputs[1] and the weighted path length (**WPL**) for the 70 considered flows, with each framework. *WPL* is defined as the length of the path weighted by the number of packets sent over that path. A larger *WPL* value indicates that longer paths are found (possibly supporting higher rates on the component links). Path lengths can change during the course of the session. Node pairs are grouped into one of four types according to minimum path length between them. The results show that over all pairs the ETM-framework outperforms the ETOP-and ETT-frameworks in terms of the median TCP throughput by 131% and 30%, respectively.

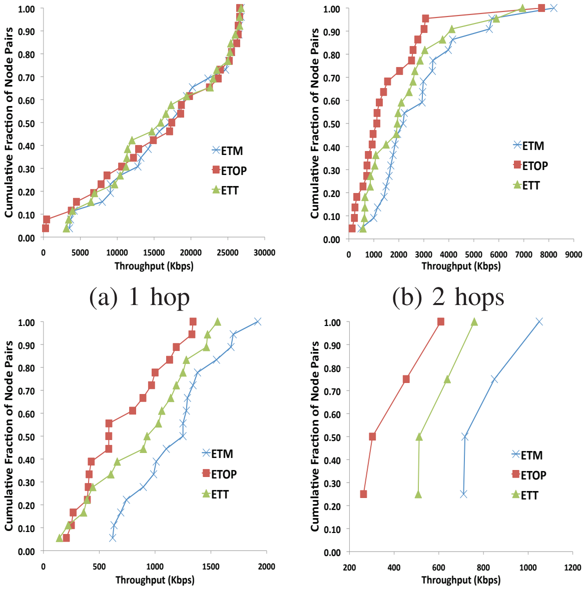
In Figure 1, we plot the CDFs of the TCP throughputs. The throughputs for the pairs separated by one hop, are statistically identical with the different frameworks (Figure 1(a)). This is because with all of the frameworks, the direct link between the considered pair is chosen rather than an alternate multi-hop path; the channel quality on the direct links is sufficiently good and the multi-hop forwarding penalty incurred on longer paths hurts the throughput. In this case the position of the link has little or no bearing and ETM offers little or no improvements

[1] When the distribution of the data is skewed (as it is in our case), the median is more representative of the observed behaviors than the mean [27].

TABLE II

TCP FLOW RESULTS FOR THE FOUR NODE PAIRS WITH LONG PATHS

	20 → 25			20 → 24			20 → 40			24 → 20		
	ETOP	ETT	ETM	ETOP	ETT	ETM	ETOP	ETT	ETM	ETOP	ETT	ETM
Throughput (Kbps)	303	512	717	263	508	711	454	759	849	609	638	1050
EnR	13.2	11.3	6.9	17.2	10.9	5.7	12.8	10.1	8.9	10.3	11.9	3.3
EnD (%)	7.5	8.1	2.6	11.3	8.9	4.1	10.2	8.8	7.1	6.9	7.4	4.1
WPR	0.14	0.21	0.39	0.19	0.27	0.48	0.26	0.31	0.39	0.38	0.31	0.61
WPL	4.0	4.9	4.4	4.3	4.9	5.2	4.0	4.3	4.7	4.2	4.7	5.0



(a) 1 hop

(b) 2 hops

(c) 3 hops

(d) 4 hops

Fig. 1. The CDF of the throughputs for pairs separated by different path lengths with each of the frameworks

over ETOP or ETT. ETM offers significant throughput improvements compared to what is achieved with ETT and ETOP for node-pairs that are separated by two or more hops. In particular, the ETM-framework achieves median throughput improvements up to 113% (for pairs separated by 4 hops) as compared with ETOP. The improvements over ETT amounts to 36% for pairs separated by 4 hops.

Performance analysis: To understand the performance differences between the frameworks, we present the detailed results with respect to four node pairs ($20 \rightarrow 24$, $20 \rightarrow 25$, $20 \rightarrow 40$, and $24 \rightarrow 20$) (see Figure. 2); the paths between these pairs are the longest in our experiments.

At each node the following statistics are gathered at the MAC layer using the Click-handler [28]: (i) the number of transmissions (including retransmissions), and (ii) the number of packets that are dropped (when the maximum limit on the number of possible retransmissions is exceeded).

We define two new metrics: the Effective number of transmissions and Retransmissions (**EnR**) and the Effective number of Dropped packets (**EnD**). For a TCP flow, *EnR* is the ratio of the packets received at the destination to the total number of transmissions and retransmissions attempted at the MAC layer for that flow. It reflects the cost of delivering a TCP packet successfully from the source to the destination. The *EnD* of a TCP flow is the ratio of the total number of MAC layer packet drops to the total number of transport layer transmission attempts performed for a given flow. With this we estimate the fraction of packets that were dropped en route the destination.

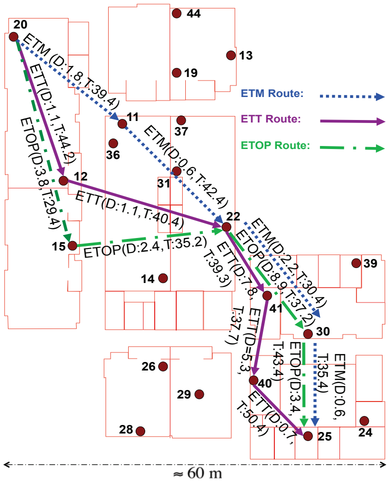


Fig. 2. The primary paths for node pair $20 \rightarrow 25$ selected by each framework; the notation is: D: the packet drop rate on the link (%), T: average transmission rate on the link (Mbps)

We also record all the paths traversed by a TCP flow and the number of packets sent over each of the paths. Based on the path and packet records, we introduce the metric Weighted Pair Reliability (**WPR**). *WPR* is defined as the path delivery ratio weighted by the number of packets sent over that path during a TCP flow; the path delivery ratio is computed as the product of the delivery ratios of the links on the path. A higher *WPR* value is representative of increased reliability.

Table II shows the performance for the four node pairs in terms of the newly defined measures. We see that the ETM-framework (in typical cases) reduces the *EnR* and *EnD* significantly as compared to the ETOP- and ETT-frameworks; this in turn, leads to higher TCP throughputs. We also observe from the *WPR* and *WPL* that ETM computes more reliable, albeit longer paths than both ETT and ETOP in typical cases.

An closer look at the paths with each framework: Next, we examine the routes computed with ETT, ETOP, and ETM for one of the four pairs, $20 \rightarrow 25$. From Table II, for this pair the ETM-framework achieves throughput improvements of about 40% and 137% as compared with the

ETT-framework and the ETOP-framework, respectively. The primary route established with each framework is depicted in Figure 2.

We make the following observations with regards to the considered flow from the figure:

(i) ETOP-routing finds a route that does not exploit multiple rates. Since small probe packets attain high packet delivery rates over long links, the route is short but can only support low rates. Thus the throughput is the poorest.

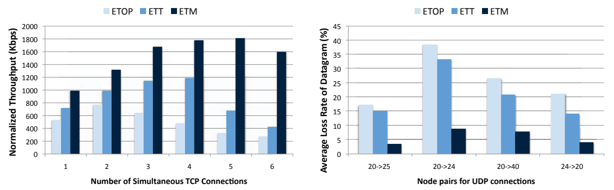
(ii) ETT-routing chooses longer paths as compared to ETOP-routing (much longer) and ETM-routing (comparable); in particular, the segment between the nodes 22 and 25 is longer. However, the use of SampleRate results in the use of high rates and thus, more drops on the links composing this segment. Packet drop rates of 7.8% and 5.3% are observed on the links $22 \rightarrow 41$ and $41 \rightarrow 40$; the average transmission rates on these links are 39.3Mbps and 37.7Mbps.

(iii) ETM-routing chooses lower rates closer to the destination. Links $22 \rightarrow 30$ and $30 \rightarrow 25$ experience packet drop rates of just 2.2% and 0.66%; the corresponding average transmission rates on these links are lower than that with ETT (with ETM-rate adaptation) and equal to 30.4Mbps and 35.4Mbps. Note that the results with other multi-hop pairs exhibit behaviors consistent with the findings above.

Effect of Interference: So far we only considered one active TCP flow. Next, we evaluate the ETM-framework with multiple simultaneous, interfering TCP connections.

Setup: Five nodes that are at the periphery of our network (nodes 14, 20, 24, 39, and 44) are chosen and 10 temporally staggered TCP flows are established between distinct node pairs. There are 5×4 possible pairs of nodes and 20×10 = 200 distinct TCP flows; each flow lasts for 3 mins. We control the number of simultaneously active connections by varying the times of initiations of the flows. We also randomize the order in which the TCP flows are established.

Results: We use the Multiplied Median Throughput (**MMT**) metric which was proposed in [14], as an estimate of the achieved network-wide TCP throughput. *MMT* is the product of the number of concurrent flows and the median throughput achieved by the flows. Figure 3 (a) shows the *MMT* values with ETOP, ETT, and the ETM-frameworks versus the number of simultaneously active connections. In all cases, the *MMT* value increases as we initially increase the number of concurrent flows; it then starts to drop. The peak *MMT* values with the ETOP, ETT, and ETM-frameworks appear with two, four, and five concurrent flows, respectively. As shown in Table II and Figure 2, the inherent selection of *lossy inefficient* routes causes the capacity with ETOP to saturate with few flows. Beyond four concurrent flows, the *MMT* with the ETT-framework significantly drops. The reduction in link layer retransmissions with ETM leads to lower overhead and thus, reduced inter-flow interference. Thus, ***the ETM-framework achieves the best performance even in interference dominated settings***; as we see, the *MMT* value peaks with five flows with ETM.



(a) Capturing the impact of multiple simultaneous TCP connections with the *MMT* metric

(b) Performance of the different frameworks with UDP in terms of the percentage datagram loss rate

Fig. 3. Performance with (a) interference and (b) UDP

Performance with UDP: Next we ask "How does ETM work with UDP" ? Recall that the ETM cost metric was derived considering e2e retransmissions; UDP does not perform e2e retransmissions. Nevertheless, the ETM-framework reduces costly packet drops and one might expect it to provide performance benefits with UDP as well. We demonstrate that this is the case in our next set of experiments.

Setup: For these experiments, we consider the paths with the same source-destination nodes pairs ($20 \rightarrow 24$, $20 \rightarrow 25$, $20 \rightarrow 40$, and $24 \rightarrow 20$) as listed in Table II; these paths correspond to the longest paths observed in our experiments. We establish and run a UDP flow on each pair for 3 minutes. For each flow, we set the sending rate to be 1 Mbps. Our performance metric is the percentage average loss rate of datagrams (the ratio of the number of lost datagrams to the total number of datagrams sent, expressed as a percentage). It is easy to compute the throughput from this percentage (the throughput in percentage is simply 100 % - percentage loss rate).

Results: Figure 3(b) shows the percentage average loss rate of datagrams for each source-destination pair. The ETM-framework achieves higher reliability than the other frameworks; the loss rates are 77% and 70% lower on average, as compared to the ETOP and ETT-frameworks, respectively. This reduction in loss rate is profound when compared to the 50% and 46% reduction in *EnD* with ETM (recall Table II). This is because of the TCP congestion control mechanism; TCP decreases the sending rate when packet drops occur and this decreases the achievable gains as compared to UDP. The reduction in packet losses increases UDP throughput, reduces congestion and thus, benefits the entire network overall. The behaviors reported here are consistent with what was observed with other node pairs in the network.

Effect of Congestion: Accounting for Queuing: All of the experiments so far do not consider the queuing delay in the ETM metric i.e., Q_j was set to zero. We now perform experiments with the queuing delay included in the metric. This has an impact primarily in settings with multiple-flows in the network; by incorporating Q_j in the metric, the ETM-framework balances the load across the network.

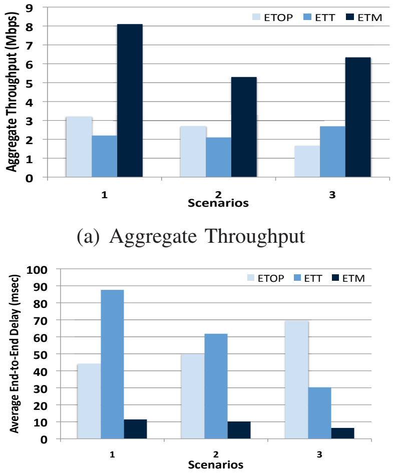
Setup: Three distinct scenarios are considered (please see Figure 2). In scenario 1, four pairs $11 \rightarrow 19$, $12 \rightarrow 13$, $15 \rightarrow 13$, and $20 \rightarrow 44$ are chosen. Pairs $11 \rightarrow 28$, $14 \rightarrow 26$, $22 \rightarrow 29$, and $31 \rightarrow 29$ and pairs $24 \rightarrow 41$, $25 \rightarrow 41$, $30 \rightarrow 40$ and $39 \rightarrow 40$ are considered in scenario 2 and 3, respectively. In each scenario, using each framework we simultaneously establish TCP connections between each chosen pairs and send traffic between each pair for 3 minutes. Again the results with each framework are obtained within minutes of each other. We set the smoothing factor α to 0.5 while estimating the average queuing delay with the ETM-framework.

Results: We observe that both ETOP- and ETT-routing produce routes with large *overlaps* (we do not pictorially show them due to space limitations). As an example, the paths chosen by the connections with the two frameworks in scenario 1 are (a) $< 11, 19 >$, $< 12, 11, 19, 13 >$, $< 15, 11, 19, 13 >$ and $< 20, 11, 19, 44 >$, and, (b) $< 11, 19 >$, $< 12, 11, 19, 13 >$, $< 15, 12, 11, 19, 13 >$, and $< 20, 12, 11, 19, 44 >$, respectively. Note that routes chosen by all of these flows include link $11 \rightarrow 19$. This causes large queuing delays at node 11 which in turn, increases the end-to-end delay and reduces the throughput of each flow. A careful examination reveals that the overlap is higher with the ETT-framework (link $12 \rightarrow 11$ is used by 3 of the flows) and thus, the impact is higher.

In contrast, we observe that ETM-routing results in load-balanced paths for the considered pairs. With respect to scenario 1, the paths used are $< 11, 19 >$, $< 12, 37, 13 >$, $< 15, 22, 13 >$ and $< 20, 11, 19, 44 >$. The link $11 \rightarrow 19$ is now used by only two of the flows. Load-balanced transmissions reduce the average end-to-end delay per flow as shown in Figure 4(b) and this leads to throughput improvements of about 253% and 368% as compared with the ETOP- and ETT-framework, respectively (see Figure 4(a)). Due to similar factors, the superior performance of the ETM-framework is also observed in scenarios 2 and 3, as shown in Figure 4.

VI. Conclusions

In this paper, we propose an comprehensive path metric ETM, that can be applied to routing and rate adaptation in multi-rate wireless mesh networks towards addressing the following goals: (a) transmit packets with increased reliability as they traverse closer to the destination, (b) achieve the best rates while adhering to the first goal, and (c) load-balance the traffic. We analytically compute the ETM metric. We design algorithms to determine the optimal path and the rates to be used on the path, in terms of ETM. Next, we design and implement a framework that integrates routing and rate adaptation based on ETM. Our evaluations on an indoor wireless mesh network show that the use of the ETM-metric provides significant performance gains over previously proposed metrics.



(a) Aggregate Throughput

(b) Average End-to-End delay

Fig. 4. The benefits from ETM in congested scenarios

References

[1] G. Jakllari, S. Eidenbenz, N. Hengartner, S. Krishnamurthy, and M. Faloutsos. Link positions matter: A noncommutative routing metric for wireless mesh network. In *IEEE Infocom*, 2008.

[2] D. Couto, D. Aguayo, J. Bicket, and R. Morris. A high-throughput path metric for multi-hop wireless routing. In *ACM Mobicom*, 2003.

[3] R. Draves, J. Padhye, and B. Zill. Routing in multi-radio, multi-hop wireless mesh networks. In *ACM Mobicom*, 2004.

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

[5] J. Choi, J. Na, K. Park, and C. Kim. Adaptive Optimization of Rate Adaptation Algorithms in Multi-Rate WLANS. In *IEEE ICNP*, 2007.

[6] D. Qiao and S. Choi. Fast-responsive link adaptation for IEEE 802.11 WLANs. In *IEEE ICC*, 2005.

[7] G. Holland, N. H. Vaidya, and P. Bahl. A rate-adaptive MAC protocol for multi-hop wireless networks. In *ACM Mobicom*, 2001.

[8] H. Zhu and G. Cao. rDCF: A Relay-enabled Medium Access Control Protocol for Wireless Ad Hoc Networks. In *IEEE Infocom*, 2005.

[9] A. Kamerman and L. Monteban. WaveLAN II: A high-performance wireless LAN for the unlicensed band. In *Bell Labs Technical Journal*, 1997.

[10] MadWifi, Multiband Atheros Driver for WiFi, http://www.madwifi.org/.

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

[12] V. Mhatre, F. Baccelli, H. Lundgren, and C. Diot. Joint MAC-aware routing and load balancing in mesh networks. In *ACM CoNEXT*, 2007.

[13] E. Kohler, R. Morris, B. Chen, and J. Jannotti. The click modular router. *ACM Transactions on Computer Systems (TOCS)*, 18(3):263–297, 2000.

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

[15] C. Koksal and H. Balakrishnan. Quality-aware routing metrics for time-varying wireless mesh networks. *IEEE Journal on Selected Areas in Communications*, 24(11):1984–1994, 2006.

[16] S. Banerjee and A. Misra. Minimum energy paths for reliable communication in multi-hop wireless networks. In *ACM Mobihoc*, 2002.

[17] Q. Dong, S. Banerjee, M. Adler, and A. Misra. Minimum energy reliable paths using unreliable wireless links. In *ACM Mobihoc*, 2005.

[18] J. Proakis. *Digital Communications*. McGrawHill, 2000.

[19] D. Qiao and S. Choi. Goodput enhancement of ieee 802.11a wireless lan via link adaptation. In *IEEE ICC*, 2001.

[20] IEEE Standard for Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specications. ISO/IEC 8802-11:2005/Amd.4:2006(E).

[21] O. Tickoo and B. Sikdar. Queuing analysis and delay mitigation in IEEE 802.11 random access MAC based Wireless networks. In *IEEE Infocom*, 2004.

[22] P. Engelstad and N. Østerbø. Queuing delay analysis of IEEE 802.11e EDCA. In *3rd Annual Conference on Wireless On-demand Network Systems and Services*, 2006.

[23] T. Cormen, C. Leiserson, and R. Rivest. *Introduction to Algorithms*. McGrawHill, 2000.

[24] CM9 802.11 a/b/g wireless card, http://www.wneweb.com/.

[25] D. Johnson, D. Maltz, and J. Broch. Dsr: The dynamic source routing protocol for multi-hop wireless ad hoc networks. *Ad Hoc Networking*. chapter 5, 139-172, 2001.

[26] Iperf-tool, http://dast.nlanr.net/projects/iperf/.

[27] R. Jain. *The Art of Computer Systems Performance Analysis*. John Wiley and Sons, 1991.

[28] The Click Modular Router Project. http://www.read.cs.ucla.edu/click/.