Implementing the Expected Transmission Time Metric for OLSR Wireless Mesh Networks

Pedro Miguel Esposito, Miguel Elias M. Campista, Igor M. Moraes,
Luís Henrique M. K. Costa, and Otto Carlos M. B. Duarte
Grupo de Teleinformática e Automação - PEE/COPPE - DEL/POLI
Universidade Federal do Rio de Janeiro
Rio de Janeiro, Brazil
Email:{pedro,miguel,igor,luish,otto}@gta.ufrj.br

Marcelo G. Rubinstein
PEL/DETEL/FEN
Universidade do Estado do Rio de Janeiro
Rio de Janeiro, Brazil
Email:rubi@uerj.br

***Abstract*— This paper presents the design of a plug-in for the Optimized Link State Routing (OLSR) protocol with the Expected Transmission Time (ETT) metric and experiments in an indoor testbed. The ETT metric is implemented as a plug-in, keeping portability and facilitating its deployment on operational networks. Our design identifies important implementation issues. Additionally, we run experiments in an indoor testbed to verify the performance of our ETT plug-in. Our results show that the ETT metric has the lowest packet loss rate and the lowest round trip time among the analyzed metrics, because it reproduces link quality conditions and also takes into account physical transmission rates.**

I. INTRODUCTION

The IEEE 802.11 standard is the most widespread wireless LAN technology that provides infrastructure and infrastructureless operating modes. In the infrastructureless mode, also called ad hoc mode, there are no centralizing elements, nodes are typically mobile, and nodes work in a collaborative way for routing packets across multiple hops to the destination. The absence of infrastructure and self-configuration capability make ad hoc networks well suited for low-cost applications. Nevertheless, node mobility, wireless medium instabilities, and the complete absence of infrastructure often result in low connectivity [1]. Hence, Wireless Mesh Networks (WMNs) [2] were proposed to increase ad hoc network connectivity because mesh networks extend the coverage area of access points by using multihop communications. The main difference between mesh and ad hoc networks is the use of a backbone of typically stationary wireless routers. These routers interconnect isolated LANs and may provide backhaul access to users. These backbone nodes usually do not have strict constraints on power consumption. As mobility and power savings are no longer the main problems, wireless mesh routing protocols can be optimized to consider link-quality metrics such as transmission capacity or error probability. This new paradigm is called quality-aware routing [3].

Different link-quality routing metrics have been proposed, but only a few have been implemented and evaluated in real networks [4]. Implementing new metrics for WMN routing protocols involve adaptation of protocols and/or metrics that is not straightforward. In this paper, we add the ETT metric to an implementation of the OLSR (Optimized Link State Routing) protocol [5], [6]. The OLSR protocol is a proactive link-state protocol that is being adopted in the upcoming IEEE 802.11s standard for WMNs. One of the most-known implementations of OLSR is the `olsrd` program (OLSR daemon) [7]. In its current version, `olsrd` can use hop count or ETX as metric. In this work, we build a plug-in that implements ETT for the OLSR daemon. The choice for a plug-in instead of modifying the OLSR source code provides portability and facilitates its deployment on operational wireless mesh networks. We analyze the performance of our implementation by running experiments in an indoor testbed. We compare the results obtained with the implemented ETT metric, hop count, and ETX metrics. Results show that, with ETT, the packet loss rate and the round trip time are lower, justifying the choice for ETT.

The remainder of this paper is organized as follows. Section II describes main implemented routing metrics for WMNs. The ETT metric implementation is described in Section III. Our experimental results are shown in Section IV. Section V presents our conclusions.

II. ROUTING METRICS

In ad hoc networks, the most used metric is hop count, which is convenient for ad hoc because frequent link breakages result from the mobility of users. On the other hand, as WMN routers are usually stationary, routing metrics that reflect link-quality variations are feasible.

One of the first metrics specifically proposed for wireless mesh networks is the Expected Transmission Count (ETX) [8]. The ETX metric estimates the expected number of tries needed to successfully transmit a frame on a link. Thus, the goal of using ETX is to find the route with the highest probability of packet delivery, instead of the shortest path. The ETX metric of a link is computed as $1/(d_f \times dr)$, where d_f and d_r are the forward and the reverse delivery ratios of the link. To estimate d_f and d_r, routers periodically broadcast small-size probes containing the fraction of probes received from each neighbor during the last w time window. Upon probe reception, routers become aware of the forward delivery ratio to each neighbor. To calculate the reverse delivery ratio, each node counts the number of probes received from each neighbor in the last w interval and computes the fraction received. The ETX computation considers both forward and reverse directions because of data- and ACK-frame transmission. The chosen route is the one with the lowest sum of ETX values along the route to the destination.

The ETX implementation has two shortcomings: IEEE 802.11 broadcast frames are sent at the network basic physical rate and probes are usually smaller than data packets. Thus, ETX does not distinguish links with different capacities, and the loss probability of small probes differs from the loss probability of data packets. To cope with these problems, the Expected Transmission Time (ETT) [9], [10] metric was proposed. The ETT metric estimates the time a data packet needs to be successfully transmitted on a link.

The ETT metric computation is a matter of implementation choice. For Draves *et al.* [9], the ETT metric can be calculated by adjusting the ETX metric according to the packet size and the transmission capacity of the link. Thus, $ETT = ETX \times S/B$, where S is the packet size and B is the link capacity. To estimate B, nodes use the packet-pair technique [9]. In this technique, two back-to-back probes, one small followed by a large one, are sent to each neighbor. Each neighbor then measures the inter-arrival time between the two packets and reports it back to the sender of the probes. Upon receiving a predefined number of delay samples, the sender estimates the capacity of the link by dividing the size of the larger probe by the smallest delay sample obtained. Another approach to compute the ETT metric is proposed by Aguayo *et al.* [10]. The authors estimate the loss probability by considering that IEEE 802.11 uses data and ACK frames. The idea is to periodically compute the loss rate of data and ACK frames to each neighbor. The former is estimated by broadcasting a number of packets of the same size as data frames, one packet for each data rate as defined in IEEE 802.11. The latter is estimated by broadcasting small packets of the same size as ACK frames transmitted at the basic rate, which is the rate used for ACKs. Note that broadcasting packets at higher data rates may require firmware modifications. According to Aguayo *et al.*, the ETT metric considers the best throughput achievable (r_t) and the delivery probability of ACK packets in the reverse direction (p_{ACK}). Thus, $ETT = 1/(r_t \times p_{ACK})$.

In this work, we use the ETT metric definition according to Draves *et al.* [9]. Considering overhead, in a n-node network, where each node has v neighbors, the number of probes sent using ETT is $O(nv)$ whereas using ETX it is $O(n)$. This difference shows that the transmission time computation comes along with additional overhead because of unicast transmissions. Similarly to ETX, the chosen route is the one with the lowest sum of ETT values.

III. THE EXPECTED TRANSMISSION TIME IMPLEMENTATION

The Optimized Link State Routing (OLSR) protocol [5] is a proactive link-state protocol. In OLSR, each node periodically broadcasts `HELLO` messages in order to know the state of local links. Each node lists its neighbors in `HELLO` messages and consequently, a node is aware of its two-hop neighbors. The link-state advertisements are broadcasted in Topology Control (`TC`) messages. Moreover, the OLSR protocol limits the number of nodes in charge of disseminating `TC` messages to avoid redundancies. Each node selects its MultiPoint Relay (MPR) set, which is composed by nodes responsible for forwarding `TCs` from the selector node. A node fills its MPR set with the minimum number of one-hop neighbors needed to reach every two-hop neighbor. The use of MPRs reduces the number of `TC` messages. A smaller number of `TC` messages, however, may become a problem when the link-quality is low. As there are less redundant messages, their eventual losses make routing tables not correctly updated. As topology maps in different nodes may be not synchronized, routing loops may occur, leading to congestion and more message losses in an indefinitely repeated cycle.

The OLSR daemon (`olsrd`) [7] natively implements the hop count and ETX metrics. Nevertheless, instead of creating probes, `olsrd` uses OLSR `HELLO` messages to measure ETX, avoiding extra overhead. Our first design choice is to implement the ETT metric as a plug-in for `olsrd`[1], privileging portability and its use on routers with `olsrd` already installed. The main task of our ETT plug-in is to implement the packet-pair technique. Nevertheless, our design had to cope with issues related to OLSR characteristics and restrictions of the API of `olsrd`.

In OLSR, all control messages are sent in broadcast. Therefore, the `send` function of the `olsrd` API does not have the destination address as an input. This is an obstacle to our plug-in implementation because the packet-pair technique requires nodes to send unicast back-to-back probes to each neighbor to estimate the link capacity [9]. Probes transmitted in broadcast would prevent the packet-pair technique to accurately estimate the current link capacity because broadcast uses the IEEE 802.11 basic physical rate. To solve this problem, we create a specific unicast socket to send and receive packet-pair probes. Other possible approaches would be to retrieve information from the network driver or to broadcast probes at different physical rates. The former is dependent on the driver implementation and may not be accurate [9]. The latter, on the other hand, requires firmware modification [11], which is not possible with our network devices. A second disadvantage of using `olsrd` to send the ETT probes is that the OLSR protocol aggregates control messages inside OLSR IP packets aiming at saving network resources. This aggregation hinders the correct operation of the packet-pair technique, which requires inter-arrival time samples. Instead of transmitting the small probe followed by the larger one, OLSR might transmit both in one packet. The use of a separate socket in our plug-in also solved this problem.

Initially, the ETT socket is passed to the `olsrd` process. Then, the plug-in is under control of `olsrd`, which uses a `select` system call to avoid blocking the OLSR processing

---

[1] The plug-in implementation as well as the source code can be found at `http://www.gta.ufrj.br/wmn`.

TABLE I

CONFIGURATION OF DEVICES USED ON THE TESTBED.

Device	Operating System	Processor	RAM	HD/ROM
Computer C_1	Linux Debian 3.1, kernel 2.4.27	Athlon XP 1.25 GHz	512 MB	40 GB
Computer C_2	Linux Debian 3.1, kernel 2.4.27	Pentium 4 2.80 GHz	1 GB	80 GB
Computer C_3	Linux Debian 3.1, kernel 2.4.27	Pentium 4 3.40 GHz	512 MB	80 GB
Computer C_4	Linux Debian 3.1, kernel 2.6.18	Pentium D 3.20 GHz	1 GB	80 GB
Routers ($R_1 - R_6$)	Linux OpenWrt RC4, kernel 2.4.30	BCM3302 V0.7 200 MHz	8 MB	2 MB

while waiting for probes. Nevertheless, this solution has a problem. When the ETT socket is passed to the `olsrd` process, the time measured between the two probes is the time between the deliveries of each probe from the daemon to the plug-in. Our design choice then is to use a specific thread for our plug-in, apart from the main `olsrd` thread. Opening a socket within another thread does not block the `olsrd` process while allowing the time measurement to happen as soon as probes arrive at the ETT socket.

Our last design choice deals with two problems related to probe forwarding, in order to guarantee correct packet-pair operation. First, neighbor nodes are learned through `HELLO` messages, which are transmitted at the network basic rate. On the other hand, packet-pair probes are unicast and sent at data rates. With IEEE 802.11, the lower the physical rate, the larger the reception range. Consequently, a neighbor according to `HELLO` messages may be out of range at higher rates. This problem leads to neighborhood and routing table mismatches. A second problem is related to the routing choice itself. Node A can choose a route to its neighbor B using another neighbor, say C, as next hop, if the route $A - C - B$ has lower ETT than $A-B$. In that case, unicast probes sent by A to B follow the 2-hop route, no longer giving an estimation of link $A-B$ ETT. Our solution uses source routing to send probes. Thus, we fix the route to each `HELLO`-message neighbor and we avoid possible multihop routes, precisely measuring each link capacity.

IV. EXPERIMENTAL RESULTS

We run experiments in an indoor testbed to validate our ETT plug-in. The performance of the OLSR protocol considering the ETT, the hop count, and the ETX metrics is evaluated.

A. Setup

Our testbed is deployed at the Federal University of Rio de Janeiro, Brazil. The mesh network is composed by ten nodes; four personal computers equipped with Netgear IEEE 802.11 PCI cards based on the Atheros AR5212 chipset and six Linksys WRT54G IEEE 802.11 wireless routers. Device configurations are shown in Table I.

All devices run `olsrd` [7] version 0.4.10. Personal Computers use the Madwifi driver version 0.9 [12] and routers the `Broadcom-drv` from Broadcom. None of the devices have additional antennas.

The nodes are placed on the third floor of the building (Figure 1). They are positioned to keep the network connected while maximizing the number of hops and the number of available routes. The distance between the two most distant nodes (R_3 and R_4) is approximately 45 m. Room A has a mezzanine and is divided into smaller rooms by wood walls. Nodes C_1, C_2, and R_1 are located at the bottom of the mezzanine. Node C_3 is at the top of the mezzanine. Rooms A to D are separated by simple masonry walls. The walls between all rooms and the hall are double and of masonry. It is worthy mentioning that there are other wireless networks in the area. In the worst case, we have found four networks, one of them using the same channel (number 6) as our testbed.

The OLSR parameters used in our experiments are the default of `olsrd`, as follows. `HELLO` messages are sent every 2 s. Each `HELLO` has validity time of 20 s. Loss rate is computed using 10 samples. `TC` messages are sent every 5 s and are valid for 30 s. The MPR mechanism is used by default on *olsrd*. Regarding the packet-pair technique, we use 6 inter-arrival time samples to estimate the bandwidth (B) of each link. Two back-to-back probes are sent every 10s. Each sample is valid for 60 s. Main parameters are summarized in Table II.

TABLE II

MAIN OLSR AND ETT PARAMETERS.

Parameter	Value
HelloInterval	2.0 s
HelloValidityTime	20.0 s
LinkQualityWinSize	10
TcInterval	5.0 s
TcValidityTime	30.0 s
ett_window	6
emission_interval	10.0 s
ett_expiration_time	60.0 s

We choose two representative pairs of nodes in our topology to measure packet loss rate, round trip time, and route length. The experiment consists of `ping` packet transmissions from C_3 to R_3, and from C_4 to R_4. We vary the packet size from 100 to 1472 bytes. Each experiment consists of transmitting

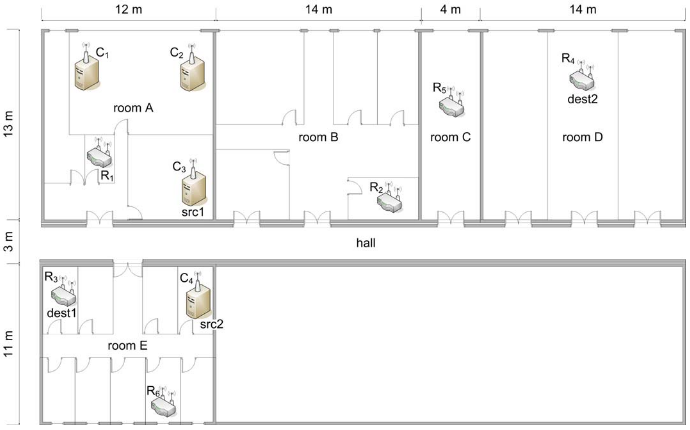
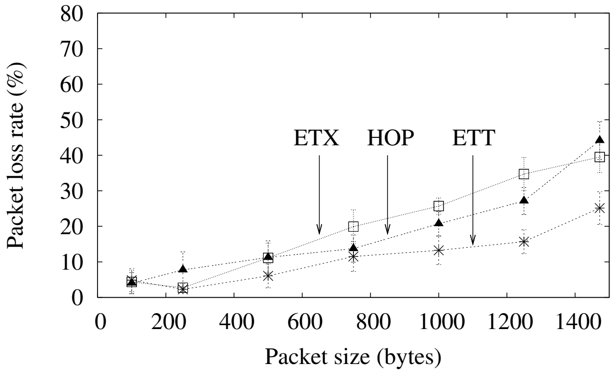


Fig. 1. Experimental scenario.

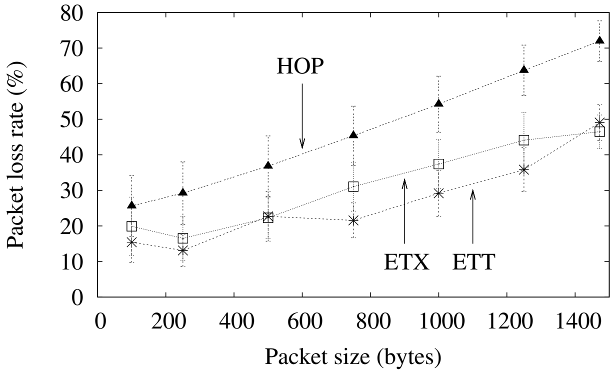
100 `pings` during 5s, where `pings` are transmitted at a constant rate of 20 `pings` per second. Consecutive experiments are separated by idle periods of 5s. In our experiments, we do not inject additional background traffic in the network.

B. Results

We compare the performance of ETT, ETX, and hop count metrics implemented in OLSR. All measurements are presented with 95% confidence intervals represented in the figures by vertical bars.



(a) From node C_3 to node R_3.



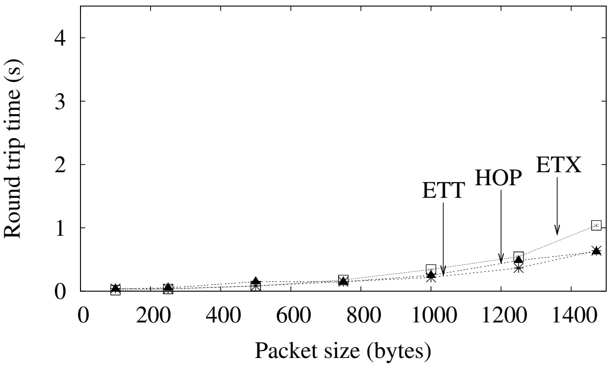
(b) From node C_4 to node R_4.

Fig. 2. Packet loss rate.

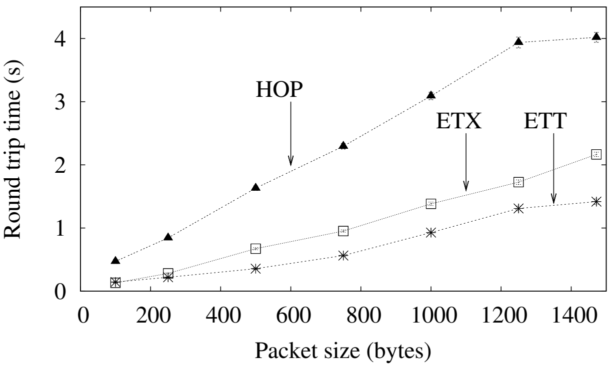
Figures 2(a) and 2(b) plot the average packet loss rate for each metric experienced by nodes C_3 and C_4 when pinging nodes R_3 and R_4, respectively. As the packet size increases, the packet loss rate also increases. Figures 2(a) and 2(b) show that the ETT metric has the lowest packet loss rate. This metric is designed to select the best route according to the current link quality, and reproduce physical conditions better than ETX. The performance of hop count is the worst because it is not aware of link-quality variation. Note that the communication between nodes C_4 and R_4 shows a higher loss rate than the communication between C_3 and R_3. This behavior occurs because the path between these nodes has more obstacles and uses more hops (Figure 1). Consequently, the difference in performance among the three metrics is larger, and hop count shows again the worst results.

Figures 3(a) and 3(b) plot the average round trip time for each metric. Similarly to the packet loss rate, the round trip time produced with the ETT metric is the lowest among the three metrics. The lower the packet loss rate, the lower the number of retransmissions, which reduces the round trip time. The round trip time is greater for the communication between nodes C_4 and R_4 when compared with the communication between nodes C_3 and R_3. Moreover, as the ETT metric is the only one that estimates the transmission time, this metric performs better then the ETX and the hop count metrics in terms of round trip time.

ETX and ETT metrics choose routes according to link quality, whereas hop count chooses the shortest route. This leads to a tradeoff between route length, which translates into the number of medium accesses, and link quality. Figures 4(a) and 4(b) plot the average route length for each metric. These figures show that ETX and ETT metrics produce longer routes than hop count. Although such fact results in a higher number of medium accesses, the route links have better quality and send packets using higher physical rates. This explains the lower packet loss rate and the lower round trip time for ETX and ETT metrics. According to Figures 4(a) and 4(b) the average route length is independent of the packet size. As expected, the average route length is greater for nodes C_4 and

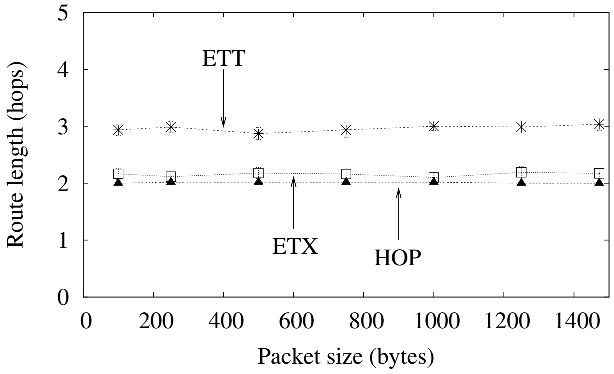


(a) From node C_3 to node R_3.

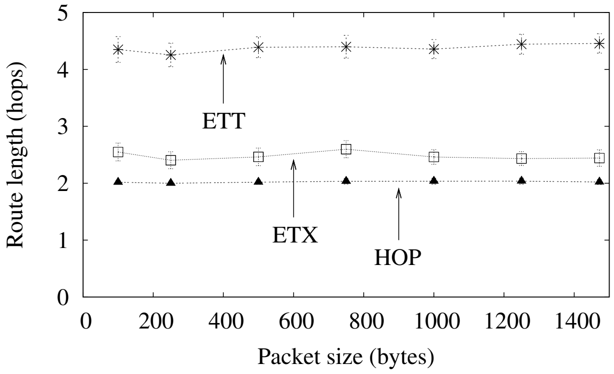


(b) From node C_4 to node R_4.

Fig. 3. Round trip time.



(a) From node C_3 to node R_3.



(b) From node C_4 to node R_4.

Fig. 4. Route length.

R_4 when compared with the average route length for nodes C_3 and R_3, especially for ETX and ETT metrics.

V. CONCLUSION

Today, there is a *consensus* that routing metrics that do not consider physical variations are not suitable for Wireless Mesh Networks (WMNs). Metrics unaware of link quality cannot guarantee reasonable stability and acceptable loss rates. Therefore, routing in wireless mesh networks has evolved by designing algorithms that take wireless medium conditions into account. Thus, the recently proposed metrics reflect various physical-layer characteristics, such as loss probability and transmission rate.

In this paper, we have implemented a plug-in of the ETT metric for the OLSR protocol. The plug-in maintains portability and can be easily implemented in operational networks. Our experiments show that the ETT metric performs better than both hop count and ETX metrics. Besides building a portable implementation of the ETT metric, we have identified important design choices that must be considered in the implementation of WMN routing protocols.

ACKNOWLEDGEMENT

This work was supported by CNPq, CAPES, COFECUB, FAPERJ, and FUJB. Also, we would like to thank Felipe Schiller for his contribution in the ETT implementation.

REFERENCES

[1] M. E. M. Campista, I. M. Moraes, P. M. Esposito, A. Amodei Jr., D. O. Cunha, L. H. M. K. Costa, and O. C. M. B. Duarte, “The ad hoc return channel: A low-cost solution for Brazilian interactive digital TV,” *IEEE Communications Magazine*, vol. 45, no. 1, pp. 136–143, Jan. 2007.

[2] I. F. Akyildiz, X. Wang, and W. Wang, “Wireless mesh networks: A survey,” *Computer Networks*, vol. 47, no. 4, pp. 445–487, Mar. 2005.

[3] C. E. Koksal and H. Balakrishnan, “Quality-aware routing metrics for time-varying wireless mesh networks,” *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 11, pp. 1984–1994, Nov. 2006.

[4] M. E. M. Campista, D. G. Passos, P. M. Esposito, I. M. Moraes, C. V. N. de Albuquerque, D. C. M. Saade, M. G. Rubinstein, L. H. M. K. Costa, and O. C. M. B. Duarte, “Routing metrics and protocols for wireless mesh networks,” *IEEE Network*, vol. 22, no. 1, pp. 6–12, Jan.-Feb. 2008.

[5] T. Clausen, P. Jacquet, A. Laouiti, P. Muhlethaler, A. Qayyum, and L. Viennot, “Optimized link state routing protocol,” in *IEEE International Multi Topic Conference (INMIC)*, Dec. 2001, pp. 62–68.

[6] T. Clausen and P. Jacquet, “Optimized link state routing protocol (OLSR),” IETF Network Working Group RFC 3626, Oct. 2003.

[7] “olsrd,” Available at http://www.olsr.org, 2008.

[8] D. S. J. D. Couto, D. Aguayo, J. Bicket, and R. Morris, “A high-throughput path metric for multi-hop wireless routing,” in *ACM MobiCom*, Sept. 2003, pp. 134–146.

[9] R. Draves, J. Padhye, and B. Zill, “Routing in multi-radio, multi-hop wireless mesh networks,” in *ACM MobiCom*, Sept. 2004, pp. 114–128.

[10] D. Aguayo, J. Bicket, and R. Morris, “SrcRR: A high throughput routing protocol for 802.11 mesh networks (DRAFT),” MIT, Tech. Rep., 2005.

[11] J. Bicket, D. Aguayo, S. Biswas, and R. Morris, “Architecture and evaluation of an unplanned 802.11b mesh network,” in *ACM MobiCom*, Aug. 2005, pp. 31–42.

[12] “Madwifi.org,” Available at http://www.madwifi.org, 2008.