

On Accurate and Asymmetry-aware Measurement of Link Quality in Wireless Mesh Networks

Kyu-Han Kim, *Student Member, IEEE* and Kang G. Shin, *Fellow, IEEE*

Abstract—This paper presents a highly efficient and accurate link-quality measurement framework, called EAR (*Efficient and Accurate link-quality monitor*), for multi-hop wireless mesh networks, that has several salient features. First, it exploits three complementary measurement schemes: passive, cooperative, and active monitoring. By adopting one of these schemes dynamically and adaptively, EAR maximizes the measurement accuracy, and its opportunistic use of the *unicast* application traffic present in the network minimizes the measurement overhead. Second, EAR effectively identifies the existence of wireless link asymmetry by measuring the quality of each link in both *directions* of the link, thus improving the utilization of network capacity by up to 114%. Finally, its cross-layer architecture across both the network layer and the IEEE 802.11-based device driver makes EAR easily deployable in existing multi-hop wireless mesh networks without system recompilation or MAC firmware modification. EAR has been evaluated extensively via both *ns-2*-based simulation and experimentation on our Linux-based implementation in a real-life testbed. Both simulation and experimentation results have shown EAR to provide highly accurate link-quality measurements with minimum overhead.

Index Terms—Wireless mesh networks, wireless link-quality, link asymmetry, measurement, distributed systems

I. INTRODUCTION

Recently, wireless mesh networks (WMNs) have been drawing considerable attention due mainly to their potential for last-mile broadband services, instant surveillance systems, and back-haul service for large-scale wireless sensor networks [1]–[4]. However, due to their deployment in large and heterogeneous areas and their use of open wireless media, wireless links often experience significant quality fluctuations and performance degradation or weak connectivity [5], [6].

To deal with such wireless link characteristics, significant efforts have been made to improve the network performance by reducing the overheads associated with unexpected link-quality changes. For example, ExOR [7], [8] is a routing protocol that tries to reduce the number of retransmissions via cooperative diversity among neighboring nodes. MASA [9] is a MAC-layer approach that tries to minimize the overhead in recovering lost frames via nearby “salvaging” nodes. Finally, NADV [10] is a link metric that assists a geographic routing protocol to choose the relay node by optimizing the trade-off between proximity and link quality.

In addition to the above efforts, accurate measurement of wireless link quality is essential to dealing with link-quality fluctuations for the following reasons. First, the above-

mentioned three solutions rely heavily on the availability of accurate link-quality information to select the best relay nodes. Second, applications, such as video streaming and VoIP, also need the link-quality information to support QoS guarantees over WMNs. Third, diagnosing a network, especially a large-scale WMN, requires accurate long-term statistics of link-quality information to pinpoint the source of network failures, and reduce the management overhead [11]. Finally, WMNs commonly use multiple channels [12]–[14], and determining the best-quality channel among multiple available channels requires the information on the quality of each channel.

Unfortunately, there are several limitations in using existing techniques to measure the quality of links in WMNs. First, Broadcast-based Active Probing (BAP) has been widely used for link-quality-aware routing [7], [12], [15]. Even though it incurs a small overhead (e.g., 1 packet per second), broadcasting does not always generate the same quality measurements as actual data transmissions due to different PHY settings (e.g., modulation). Thus, BAP provides inaccurate link-quality measurements. Moreover, its use of an identical type of probing in *both* directions of a link generates *bi-directional* results, thus un-/under-exploiting link asymmetry. Second, unicast-based probing provides accurate and *uni-directional* results owing to its resemblance to the use of actual data transmissions, but it incurs significant overheads. Finally, passive monitoring [10] is the most efficient and accurate since it uses actual data traffic, but it also incurs the overhead of probing idle links.

To overcome the above limitations of existing measurement techniques, we propose a high-accuracy and low-overhead distributed measurement framework, called EAR, that has the following three salient features. First, EAR consists of three complementary measurement schemes—passive, cooperative, and active monitoring—that commonly use *unicast* for its accuracy and “opportunisticly” exploit the egress/cross traffic of each node for efficiency. Using unicast, all three schemes measure link-quality under the same setting as the actual data transmission, thus yielding accurate results. By exploiting data traffic in the network as probe packets, and dynamically and adaptively selecting the most effective of the three schemes, EAR not only reduces the probing overhead, but also decreases the measurement variations, thanks to the large number of “natural” probe (i.e., real traffic) packets.

Second, EAR’s link-quality measurement is made *direction-aware* to effectively capitalize on link asymmetry. Wireless link quality is often asymmetric due to such environmental factors as hidden or heterogeneous nodes, obstacles, and even weather conditions [1], [15], [16]. The better-quality direction of an asymmetric link might often be good enough to transmit

The authors are with the Real-Time Computing Laboratory in the Department of Electrical Engineering and Computer Science at The University of Michigan, Ann Arbor, MI 48109-2122, U.S.A. E-mail: {kyuhkim,kgshin}@eecs.umich.edu

data frames in that direction, instead of taking a longer detour path. By direction-aware measurement of link quality derived from actual data transmissions and ACK receptions, EAR can identify and exploit link asymmetry, thus improving the utilization of network capacity as well as routing performance.

Finally, EAR is designed to run in a fully-distributed fashion and to be easily deployable on existing IEEE 802.11x-based WMNs. It runs on each node and periodically measures the quality of link to each of its neighbors to maintain up-to-date link-quality information. On each node, EAR is implemented at the network layer and a device driver, and intelligently uses several features of the MAC layer, such as transmission results and data rate, by interacting with the MAC Management Information Base (MIB) [17]. Moreover, this design does not require any system change or MAC firmware modification, thus making its implementation and deployment easy.

We conduct an in-depth evaluation of EAR via both *ns-2*-based simulation and experimentation on a Linux-based implementation in our testbed. Our simulation results show that EAR's unicast-based techniques decrease the root mean-square error (MSE) in measurements by at least a factor of four over the broadcast-based approach, while reducing the overhead by an average of 50%, even in large-scale WMNs. Next, EAR is implemented as a routing component along with the extension to Orinoco 802.11b device driver, and then evaluated on our experimental testbed. Experimental results show that EAR effectively exploits existing application traffic in measurement (up to 13 times more probing packets than BAP's). Moreover, our measurement results show that there exist many asymmetric links, each lasting for a few to dozens of minutes, and that EAR's uni-directional measurement helps the routing protocol improve the end-to-end throughput by up to 114%. Finally, EAR's implementation is extended and evaluated to demonstrate the feasibility of supporting multi-radio WMNs [12], [18], [19].

The rest of this paper is organized as follows. Section II describes the motivation of this work. Section III presents the EAR architecture and algorithms. Section IV evaluates EAR using *ns-2*-based simulation, and Section V describes our implementation of EAR and experimental results on our testbed. Section VI discusses the remaining issues associated with EAR, and finally concludes the paper.

II. MOTIVATION

We first advocate the importance of accurate measurements of wireless link quality to WMNs. Then, we identify the limitations in applying existing measurement schemes to WMNs.

A. Why Accurate Link-Quality Measurement?

Wireless link quality varies with environmental factors, such as interference, multi-path effects and even weather conditions [5], [20], [21]. Especially, in multi-hop WMNs, due to their usual deployment in large and heterogeneous areas, wireless link quality fluctuates significantly, and thus, the various network protocols, such as the shortest-path and

geographic routing protocols, designed under the strong link-quality assumption¹ often suffer performance degradation or weak connectivity [5], [6], [20].

Accurate link-quality measurement is essential to solve the problem associated with varying link-quality in WMNs, as one can see from the following use-cases.

- *Selection of the best relay node*: Accurate link-quality information can reduce the recovery cost of lost frames caused by link-quality fluctuations. For example, ExOR [7], [8] and MASA [9] attempt to reduce the number of transmissions with the help of intermediate relay nodes in retransmitting lost frames. Both solutions are based on *capture effects* that allow in-range nodes to cooperatively relay "overheard" frames, but one key question is how to select the relay node that has the best link-quality.
- *Supporting Quality-of-Service (QoS)*: Wireless link-quality information enables applications and network protocols to effectively meet users' QoS requirements. For example, applications, such as VoIP and IPTV, can dynamically adjust their service level that can be sustained by varying link-quality in the network. On the other hand, link-quality-aware routing protocols [12], [15] can accurately locate a path that satisfies the QoS (e.g., throughput and delay) requirements based on the link-quality information.
- *Network failure diagnosis*: Link-quality statistics can be used to diagnose and isolate faulty nodes/links (or faulty areas) in WMNs, facilitating network management [11], [22]. WMNs covering shopping malls, a campus or a city, usually consist of a number of nodes, and each node must deal with site-specific link conditions. Thus, WMNs require accurate information on link conditions for troubleshooting.

Motivated by these and other use-cases, we would like to address how to measure link-quality and how beneficial accurate measurements can be in utilizing network capacity.

B. Limitations of Existing Techniques

There has been a significant volume of work on link-quality measurement. We discuss pros and cons of using existing techniques for WMNs.

1) **Accuracy and efficiency**: A measurement technique must yield accurate results at as low a cost as possible. First, Broadcast-based Active Probing (BAP) has been widely used for adopting link-quality-aware routing metrics such as Expected Transmission Count (ETX) [15] and Expected Transmission Time (ETT) [12]. BAP uses simple broadcasting of identical probe packets from each node and derives link-quality information by multiplying the percentage of successful transmissions in each direction.² Even though it is inexpensive, broadcasting uses a fixed and low data rate (e.g., 2Mbps), which is more tolerant of bit errors than other rates, and which may differ from the actual data-transmission rate (e.g., 11Mbps). Thus, as we will show later (in Figure

¹For example, *if I can hear you at all, I can hear you perfectly.*

²Even though the measurement technique (BAP) and the link-quality derivation (BAP-ETX) are orthogonal, in this paper we use the single term 'BAP' to mean both for simplicity.

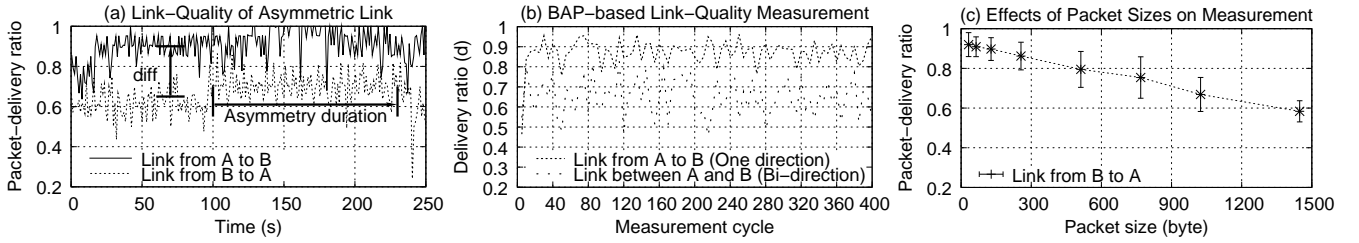


Fig. 1. Needs for Asymmetry-Awareness: (a) Wireless link-quality is often asymmetric. (b) BAP’s bi-directional measurement under-estimates quality of asymmetric links as shown in measurement results for 4000s. Finally, (c) measurement results are affected by the size of probing packets.

12), BAP yields less accurate link-quality information than a unicast-based approach (e.g., 10.2% error by broadcast vs. 1.6% error by unicast). Note that although recent device drivers (e.g., MADWiFi [23]) allow for using multiple data rates in broadcasting data, such multi-rate BAP will not only increase the probing overhead but also suffer from inaccurate measurements due to proprietary algorithms built into the firmware of NICs, as we will show in Section V-C4.

Next, the unicast-based approach to measuring link bandwidth [12], [24], [25] can yield accurate results as it uses the same data rate for probing a link as that for actual data transmissions over the link. However, frequent probing of link to each neighbor incurs a higher overhead than BAP. As the number of neighbors increases, probe packets might throttle the entire channel capacity.

Finally, without injecting probe packets, passive monitoring yields accurate link-quality measurements without incurring any overhead. Signal-to-Noise Ratio (SNR) monitoring may be the cheapest, but it is shown to be not strongly related to actual link-quality [5]. Self-monitoring [10] could be attractive due to its use of actual data-frame transmission results. However, it also incurs a large overhead in probing links when there are no data packets sent over them.

2) **Link-asymmetry-awareness:** Measurement schemes must be able to identify and exploit wireless link asymmetry that results from interference, obstacles, or weather conditions [1], [15], [16]. For example, if there is interference in the vicinity of node *A*, then signals from a remote node *B* to *A* might be disrupted, whereas signals from node *A* are normally strong enough to overcome the interference. While *B* might reach *A* via node *C* that has high-quality links to both *A* and *B*, node *A* can use the direct link to *B*, saving network resources.

First, BAP has limited asymmetry-awareness. It was originally designed to be aware of link asymmetry [12], [15]. BAP independently measures the quality³ of the link’s both directions, and then multiplies them. However, the results are bi-directional—giving the same link quality in both directions—due to the same type of probing used in both directions, and often under-estimate the quality of asymmetric links. Figure 1(a) shows the packet-delivery ratio of each direction of an asymmetric link in our testbed, and Figure 1(b) is a sample measurement result of BAP over the link for a long period (4000 s). As shown in these figures, even though one direction of link has good quality (upper curve in Figure

1(b)), the measurement result via bi-directionality often under-rates the quality of the link’s both directions (lower curve). Even though BAP may overcome this limitation using multiple types of probing (e.g., different-size probe packets), such an approach incurs additional overheads, and using broadcast may still under-/over-estimate link-quality as we will show in Section V-C2.

Next, unicast-based probing and passive monitoring are usually uni-directional in the sense that their measurement includes the delivery ratio of data and ACK frame transmissions.⁴ Thus, the measurement results accurately reflect the link-quality of actual data transmission. Again, in the first example, because ACK packets of a small size are usually successfully transmitted in the severe interference as shown in Figure 1(c), uni-directional measurements derived from the high-quality link direction of DATA transmission and the reverse-direction quality of ACK transmissions helps effectively identify asymmetric link and guide node *A* to directly transmit packets to node *B* without taking a detour path.

3) **Flexibility and feasibility:** Measurement techniques must be flexible enough to cope with time-varying link-quality. First, *aperiodic* measurements, which capture link-quality only for a certain period as in [26], [27], might be the simplest way to monitor link conditions. However, it yields poor measurement accuracy in wireless environments due to frequent link-quality fluctuations or requires significant efforts to determine the optimal measurement period.

On the other hand, the simple *on-demand* link-quality measurement used in MANETs [28], [29] might be cost-effective. However, it mainly focuses on link connectivity (i.e., a binary value) instead of actual wireless link quality. Even though several approaches (e.g., [30]) have been proposed to elaborately measure link-quality using SNR, their main purpose is to maintain stable connectivity, rather than adapting to the link dynamics in real time.

Finally, the measurement techniques have to be easily implementable and deployable in existing WMNs. BAP and unicast-based approaches can be implemented at any protocol layer without requiring any significant system change. Passive monitoring can be developed in the network and MAC layers. However, it needs to exploit the information from the MAC layer, which might not be available to the public [31].

³For the time-being we use the delivery ratio of data frames of link $A \rightarrow B$ as the link quality. We will elaborate on this in Section III-B.

⁴Although unicast-based probing includes transmissions in both directions of each link, we use the term “uni-directional” to emphasize the dominant effect of a data-packet transmission on link quality.

III. THE EAR ARCHITECTURE

This section details the architecture of EAR. First, the design rationale and main algorithm of EAR are outlined. Second, we define the link-quality parameters that EAR deals with, and then describe its three measurement schemes. Next, we explain EAR’s cross-layer interaction, and finally analyze the complexity of EAR.

A. Overview of EAR

EAR is a low-overhead and high-accuracy measurement framework that is aware of asymmetric wireless links and also easily deployable in 802.11-based WMNs. EAR has the following distinct characteristics.

- *Hybrid approach*: EAR adaptively selects one of three measurement schemes (passive, cooperative, and active) to opportunistically exploit existing application traffic as probe packets. If there is no application traffic over a link, EAR uses active probing on the link at a reasonable cost. Otherwise, EAR switches itself to passive or cooperative monitoring that gratuitously uses existing traffic for collecting the link-quality information.
- *Unicast-based uni-directional measurement*: EAR uses *unicast* (instead of broadcast) in each direction of a link for measuring its quality. Unicast, which uses the same settings as the actual data transmissions, allows different schemes to generate homogeneous measurements. Moreover, since the quality of each link’s direction is independently measured via unicast, the measurement results are uni-directional.
- *Distributed and periodic measurement*: EAR independently measures the quality of link from a node to its every neighbor in a fully-distributed way. This measurement is also taken periodically to cope with the varying link-quality, and its period is adapted based on a link-quality history.
- *Cross-layer interaction*: EAR is composed of “inner EAR” (*iEAR*) that periodically collects and derives link-quality information in the network layer and “outer EAR” (*oEAR*) that monitors egress/cross traffic at the device driver. These two components interact across the two layers to intelligently exploit MAC-layer information without any modification of MAC’s firmware.

EAR’s overall operation can be described in four sequential steps as shown in Algorithm 1. First, during a measurement period (M_x), every node monitors link quality using one of passive, cooperative, and active measurement schemes per neighbor. Then, at the end of M_x , a node records the measured link quality and exchanges the information with neighboring nodes, if necessary. Next, during an update period (U_x), nodes process link-quality reports from their neighbors, if any. Finally, after an ordered pair of M_x and U_x (called the *measurement cycle*, C_x ⁵), each node updates its local link-state table with directly and indirectly measured link-quality information, and then decides on its measurement scheme for the next cycle.

⁵We set C_x to 10 seconds ($=9s (M_x) + 1s (U_x)$) in our evaluation.

Algorithm 1 EAR at node i during C_x

- (1) During a Measurement-Period, $t \in (C_{x-1}, M_x)$
 - for every** neighbor node j **do**
 - $S_{ij} \leftarrow$ a monitoring scheme for the link from node i to j
 - if** $S_{ij} ==$ PASSIVE or ACTIVE **then**
 - monitor egress traffic to node j
 - else if** $S_{ij} ==$ COOPERATIVE **then**
 - monitor egress traffic from node i to k that node j overhears
 - end if**
 - if** node i received a cooperation request (ℓ) from node j **then**
 - overhear cross traffic from node j to node ℓ
 - end if**
 - end for**
- (2) At the end of a Measurement-Period, $t = M_x$
 - for every** neighbor j **do**
 - record measurement results from node i to node j
 - if** node i received a cooperation request (ℓ) from node j **then**
 - send node j a report of overhearing traffic
 - end if**
 - end for**
- (3) During an Update-Period, $t \in (M_x, M_x + U_x)$
 - process a measurement report(s) from other nodes, if any
- (4) End of an Update-Period, $t = M_x + U_x$ (or, $t = C_x$)
 - for every** neighbor j **do**
 - calculate the quality of link from node i to j using Eq. (1)
 - run the transition algorithm (in Figure 2) for node j
 - if** transition to COOPERATIVE **then**
 - choose node k that node j can overhear
 - send a cooperation request (k) to node j
 - else if** transition to ACTIVE **then**
 - schedule active probe packets
 - end if**
 - end for**

B. Link-Quality of Interest

EAR focuses on link cost and capacity as link-quality parameters, which are defined as follows. First, the link cost is defined as the inverse of the delivery ratio (d) of MAC frames. This definition reflects the expected transmission count of each data frame. The cost (\mathbb{C}) of link A \rightarrow B is calculated by

$$\mathbb{C} = \frac{1}{d_i} \quad \text{and} \quad d_i = (1 - \alpha) \times d_{i-1} + \alpha \times \frac{N_s}{N_t} \quad (1)$$

where d_i is the smoothed delivery ratio, α a smoothing constant,⁶ N_s the number of successful transmissions, and N_t the total number of transmissions and retransmissions during a measurement period of the i -th cycle.

EAR also measures link capacity by using the data rate obtained from MAC frame transmissions. The data rate can be an upper bound of capacity that the link can achieve, and is used to derive a net capacity along with link cost via such metrics as ETX [15] and ETT [12]. In EAR, the rate is derived based on the recent statistics of dominantly-used rate at the MAC layer during the previous measurement cycle. This is done jointly with the collection of the link cost (N_s, N_t). Upon completion of data transmission to its neighbor, EAR updates the frequency of the data rate used. At the end of the measurement cycle, EAR uses the frequency to infer the MAC’s current data rate for the neighboring node. This simple

⁶We set α to 0.3, but also evaluate other values of α for link-quality awareness, as shown in Figure 13(b).

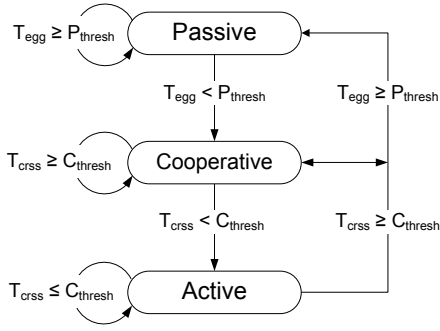


Fig. 2. Three measurement schemes and their inter-transitions: EAR consists of passive, cooperative, and active measurement phases. Based on the amount of egress/cross traffic (T_{egg}, T_{crss}), EAR adaptively switches from one measurement scheme to another. P_{thresh} and C_{thresh} are the thresholds for passive and cooperative schemes, respectively.

algorithm enables EAR to work with any rate-control scheme (e.g., *fixed*, *auto*) in MAC and yields accurate link-capacity information without incurring any communication overhead.

Note that even though EAR can be easily extended to measure other parameters, such as delay and jitter, as described in Section VI-A, we will focus on the link cost and capacity as main link-quality parameters in the remainder of this paper.

C. Hybrid Approach

As mentioned earlier, EAR consists of passive, cooperative and active measurement schemes, which are complimentary to each other. On the one hand, all of these schemes unicast probe packets through which any of the schemes provides consistent measurement results. On the other hand, although one scheme (i.e., active probing) provides accurate measurement results (e.g., 7% error in d as we will see in Section IV-B) compared to BAP (34% error), the other schemes can further improve the accuracy (1.5% error) by opportunistically exploiting a node’s egress/cross traffic, if any.

Figure 2 depicts the EAR’s hybrid measurement approach based on the three schemes. When a measuring node (m) has egress traffic, T_{egg} , to a neighbor node (n), m passively monitors the traffic. When T_{egg} decreases below a certain threshold, P_{thresh} ,⁷ m finds another neighbor node to which m has egress traffic and that n can overhear the traffic, and cooperatively (with node n) measures the quality of link $m \rightarrow n$. Finally, when the actual traffic over the link is low ($< C_{thresh}$), m actively measures link quality by unicasting probe packets over the link. Next, we give a detailed account of each measurement scheme with its rationale.

1) **Passive measurement via egress traffic:** When there is enough egress traffic, EAR favors passive monitoring over active monitoring for its accuracy and efficiency. The passive scheme (e.g., [10], [32]) can collect accurate and stable link-quality information from a large volume of existing data traffic without incurring any overhead. By contrast, many active schemes (using either broadcast or unicast probe packets as in [15], [24], [25]) must consume network resources for probing,

⁷We set both P_{thresh} and C_{thresh} to 10, which is the rate of active probing (e.g., 1 packet per second during a 10-second measurement cycle).

yet cannot provide as accurate results as the passive scheme (that uses the actual traffic).

In a WMN, there is usually enough egress and relay traffic through each node. EAR employs the passive scheme to accurately measure link quality by capitalizing on this real traffic while minimizing the measurement overhead. There are, however, several design issues to be resolved before using the scheme as follows.

- *Heterogeneous packet sizes:* The packet size greatly affects the delivery ratio [5] (also shown in Figure 1(c)), and thus, a measurement scheme has to derive the ratio by using packets of same or similar size in order to obtain accurate and consistent link cost. EAR’s passive scheme monitors packets within a 100-byte range of each of three popular sizes used in the Internet [33]—60, 512 and 1448 bytes—and derives the link cost corresponding to each size. EAR can also measure the link costs for other packet sizes similarly, or by using the estimation technique in [10].
- *Network-level vs. MAC-level:* Passive monitoring can be implemented at either the network layer or the MAC layer. The network layer solution is simple, but requires a neighboring node’s feedback on each successful packet delivery. This consumes network bandwidth, and its result is oblivious of the retransmission results at the MAC layer. EAR eliminates this overhead by placing itself at a device driver and monitoring transmission results based on MAC’s built-in ACK mechanism without additional cost or MAC modification (see Section V-A).
- *Use of MAC information:* EAR obtains (and uses) MAC information via a device driver’s interface to get around the difficulty of modifying MAC firmware. Proprietary MAC firmware makes it very difficult, if not impossible, for designers to modify MAC for direct use of channel information. Through a device driver’s interface, EAR can access MAC management variables—TxRetryLimit Exceeded, TxSingleRetryFrames, and TxMultiple RetryFrames⁸—to infer transmission results.

Suppose, as an example, that node A has (statistically) enough egress traffic to node B . Then, A requests its device driver to record the status of each of its packet transmissions. The device driver then keeps track of the three variables of MIB for the traffic, and derives the number of successful transmissions (N_s), the total number of transmissions (N_t), and the data rate. Next, at the end of a measurement period (M_i), EAR at the network layer obtains the measurement results from the device driver. Finally, at the end of an update period (U_i), it derives link quality using Eq. (1).

2) **Cooperative measurement using cross traffic:** EAR switches to cooperative monitoring when a measuring node (e.g., B in Figure 3) has no egress traffic to a neighbor node (C), but to others (A). We call the neighbor node with no traffic a “cooperative” node. Due to the broadcast nature of wireless

⁸Note that these variables are specified in IEEE 802.11 standard [17], and most of 802.11 chipsets, including Prism, Hermes and Atheros, provides interfaces to access these variables from a device driver or above [23], [34].

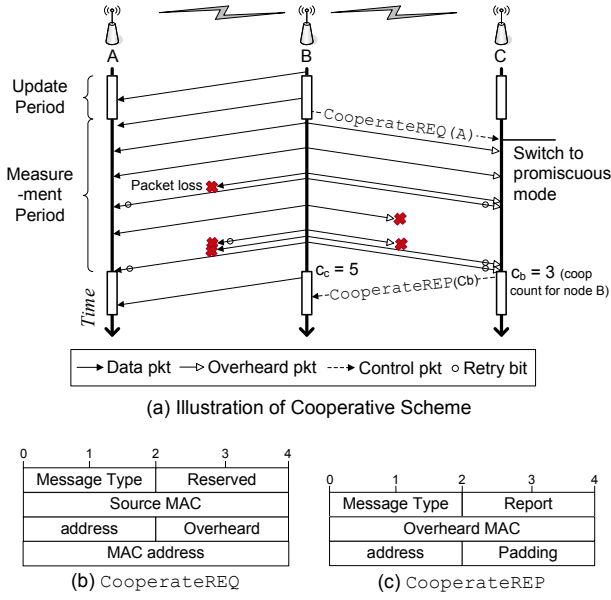


Fig. 3. Example of node B 's cooperative monitoring with node C . Once node B exchanges a cooperation message with C (CooperateREQ : source MAC address(B), overheard MAC address(A)), node C switches its NIC into a promiscuous mode and starts overhearing traffic from node B to node A . Then, it sends the overheard results back to node B (CooperateREP : report (C_b), overheard MAC address (A))

media, the cooperative node (C) can overhear the traffic from the measuring node (B) to the other neighbors (A)—we call the traffic *cross traffic*. The overhearing result is then used for the measuring node to derive the quality of link $B \rightarrow C$. This scheme not only helps the measuring node avoid the active probing, but also improves the measurement accuracy by using a large amount of cross traffic. Note that all nodes in WMNs are assumed to faithfully cooperate. Preventing malicious behaviors, such as DoS attacks, is beyond the scope of this paper.

To incorporate this scheme into EAR, we must resolve the following design issues.

- **Overhearing cross traffic:** The promiscuous mode in IEEE 802.11 NIC allows each node to overhear data frames destined for nodes other than itself. Due to the broadcast nature of wireless media, packets with the same network ID (or ESSID) can be captured by MAC and sent up to the upper layer. EAR at a device driver can choose this mode upon making/accepting a cooperation request, and monitor the cross traffic immediately. Even though this cross traffic consists of data-frame transmissions in one direction of a link, the cooperative scheme yields measurement accuracy comparable to the passive scheme's. While the passive and active schemes use data/ACK frame transmission results in measurement, the direction of ACK transmissions shows mostly a good delivery ratio even over a highly lossy link (see the case of 80 bytes in Figure 1(c)), due to its small packet size (64 Bytes) and low/reliable transmission rate (2Mbps). Therefore, link quality is mostly governed by the direction of data-frame transmissions, and the use of cross traffic makes a marginal impact on its measurement

accuracy, as we experimentally show in Section IV-B.

- **Selective overhearing:** A cooperative node has to selectively overhear cross traffic whose data rate is the same as the rate from a measuring node to itself as if it were the destination of the traffic. Because the data rate affects greatly the delivery ratio as we will show in Section V-C2, overhearing all cross traffic with different rates yields inaccurate and noisy results. In EAR, the measuring node (B) selects, based on its local information, neighbor nodes (A) that the cooperative node (C) has to monitor, and then includes the selection in its cooperation request message (i.e., $\text{CooperateREQ}(A)$) sent to the cooperative node.
- **Ambiguity of retransmissions:** The cooperative scheme must exclude packet retransmissions in measurements, since retransmissions are not caused by the link to the cooperative node. In addition, retransmissions cause both the measuring node and the cooperative node ambiguity in counting overheard packets. In Figure 3, because the cooperative node (C) cannot receive duplicate frames from its MAC layer even in the promiscuous mode, the measuring node B cannot use the retransmitted packets for measurements (e.g., the fourth overheard packet). Also, if there are multiple retransmissions, the cooperative node cannot count the total number of packets that are successfully delivered to node C , due to a single retry bit in the frame and the ignorance of duplicate frames at MAC (e.g., the last overheard packet delivered to node C).

Let's consider the example in Figure 3(a). In the first update period, node B decides to use the cooperative scheme, based on the algorithm in Figure 2, to measure the quality of link $B \rightarrow C$ by using traffic $B \rightarrow A$. Next, $\text{CooperateREQ}(A)$ is sent to node C . On receiving the request, node C switches its NIC mode to the promiscuous mode, and starts to overhear the traffic from B to A . At the same time, node B also begins counting, within the cross traffic, first-time successful transmissions (C_c)—the number of total transmissions for measurements. In the second update period, a report of overheard results ($\text{CooperateREP}(C_b)$)—the number of successful transmissions—as in Figure 3(c) from C is sent to B , and then a new delivery ratio (i.e., $\frac{C_b}{C_c} = \frac{3}{5}$) is calculated. Note that all messages are reliably delivered to the destination nodes through timer-based message/ACK handshakes.

3) **Active measurement using shared unicast:** When there is no egress/cross traffic, EAR switches to active monitoring and opportunistically sends unicast probe packets to neighbor nodes. Since it uses unicast-based probing, EAR can collect more accurate results than broadcast-based probing. On the other hand, by employing "cooperative" monitoring, EAR can reduce the active probing overhead to as low as BAP's overhead (e.g., 1 packet per second). Also, it can further reduce the probing overhead by adaptively adjusting the probe frequency based on the history of the link's quality.

To incorporate this scheme into EAR, one must address the following design issues.

- **Minimize the interference caused by probing traffic:** There are cases when a node needs to do active probing of link to one of its neighbors even though a channel is heavily

used by others. For example, in Figure 4, (a) a channel is used by A , B and D , but C needs active probing of links to the other three, and (b) D has enough ingress traffic (e.g., video streaming), but it needs to probe links to B and C . EAR reduces the probing overhead by sharing the probe packets via cooperative monitoring. In Figure 4 (a), C probes only the link to A and also measures the quality of links to B , D through cooperation with B and D , which overhear the probing traffic from C to A . Note that this is different from BAP in the sense that probe packets are transmitted at the same rate as that of data transmissions (as opposed to a broadcasting rate).

- *Reduce the probing overhead on stable/idle links:* If a link has a small quality-variance and experiences low activities, EAR need not trigger active probes often. Thus, it uses an activity-based backoff timer that (i) is exponentially increased upon its expiration, with an upper bound (*window*), if the variance/activity has been below a minimum threshold, and (ii) linearly decreases every measurement cycle. On the other hand, if there has been either the minimum activity or quality-fluctuation (*var*), EAR resets the timer to 0 and triggers the active probing, as described in Algorithm 2.(1).
- *Need to probe at different rates:* A measuring node that uses several data rates to its neighbors cannot ‘share’ probe packets with all neighbors. Instead, the measuring node needs the same number of sets of probes as the number of data rates the node uses for its neighbors, which might, in turn, generate lots of probe packets during one measurement cycle. To reduce this possibility, EAR distributes a set of probing packets over several cycles during which it is not scheduled to probe links due to its backoff timer. Because links are idle under the active scheme and the backoff timer increases exponentially, there are usually enough unused cycles to accommodate all sets. If not, EAR schedules probes for all data rates in a round robin fashion over available cycles so that every rate has an equal chance to be probed (see Algorithm 2.(2)).

Let’s consider an illustrative example. Suppose node C in Figure 4(a) switches to active monitoring. Based on its link-quality variance and data-rate history, C classifies A and B

Algorithm 2 Active-probing scheduler (every C_x in a node)

- ```

(1) Update an exponential back-off timer (t)
 for every link i do
 if $t_i == 0$ then
 $w_i \leftarrow (w_i \times 2 > window) ? window : w_i \times 2$
 $t_i \leftarrow \text{rand}(0, w_i)$
 end if
 $t_i \leftarrow (variance_i > var) ? 0 : t_i - 1$
 end for
(2) Select an active-probing group for the next measure-cycle
 for every active-coop group g do
 if g has links with expired back-off timer t then
 $G \leftarrow G \cup \{g\}$
 end if
 $g_{next} \leftarrow$ group $g \in G$ that has waited for probing most
 schedule active-probing to group g_{next} if g_{next} exists
 end for

```
- 

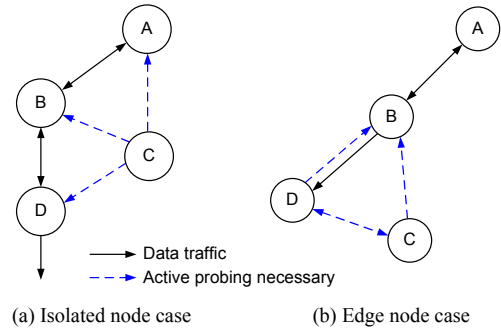


Fig. 4. Need for active monitoring: In Figure 4(a),  $C$  does not have any egress/cross traffic, and thus needs active probing while other nodes do not. In Figure 4(b),  $D$  has ingress traffic, but the opposite direction does not.

to be in the 11 Mbps-group and  $D$  the 2 Mbps-group, respectively. Also, based on the backoff timer,  $C$  schedules the active probing to the 11 Mbps-group first. During the first update period,  $C$  broadcasts a cooperation request (`ActiveCooperateReq(B)`) indicating  $B$ ’s cooperation. Then, for the following measurement period,  $C$  triggers the active probing to  $A$  and measures the quality of link  $C \rightarrow A$  and  $C \rightarrow B$  through passive and cooperative monitoring, respectively. In the second measurement period,  $C$  schedules the active probing to the 2 Mbps-group (i.e.,  $D$ ) based on the above scheduling rule. In the third update period, if the links  $C \rightarrow A$  and  $C \rightarrow B$  show stable quality and had no activity, EAR skips its probing for the 11 Mbps-group and schedules the probing for the next group (i.e.,  $D$ ) if its backoff timer has been expired.

#### D. Cross-Layer Interaction

As shown in Figure 5, EAR’s two-tier architecture (*i*EAR and *o*EAR) allows for cross-layer interactions as well as easy extension of EAR to support multi-radio WMNs.

- *Access to information on underlying layers:* As explained in the hybrid approach, *i*EAR at the network layer uses information on the MAC layer through *o*EAR in a device driver. Recently, a couple of open-source device drivers provides various interfaces to access those information. For example, the MADWiFi [23] device driver includes Hardware Abstraction Layer (HAL) for Atheros [35] chipsets to access or configure parameters in the MAC/PHY layers. EAR’s design allows to use this capability for accurate and flexible link-quality measurements by decoupling link-quality monitoring from link-quality management.
- *Architecture for multi-radio extension of EAR:* Today’s WMNs are usually equipped with multiple radios [12], [18], [19] to increase network capacity. EAR can be easily extended to support link-quality measurements on multi-radio WMNs. As shown in Figure 5(b), EAR’s architecture can accommodate as many interfaces as possible in each mesh router. Basically, EAR simply requires an additional monitoring module (*o*EAR) per NIC and interacts with one management module *i*EAR, making EAR scalable.
- *Pull-based cross-layer interaction:* Cross-layer interactions in EAR rely on pull-based message exchanges. For each interface (NIC <sub>$i$</sub> ), individual *o*EAR <sub>$i$</sub>  in its device driver (*DD* <sub>$i$</sub> )

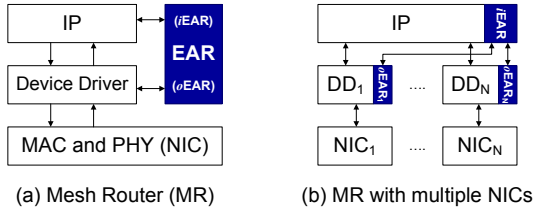


Fig. 5. Cross-Layer Approach in EAR: EAR’s two components (i.e.,  $i$ EAR and  $o$ EAR) interacts with each other to derive accurate link-quality information (Figure 5(a)). In addition, this approach enables EAR to support link-quality measurements for multi-radio WMNs as shown in Figure 5(b).

monitors network traffic that is passed through the interface and extracts the monitoring results. Then,  $i$ EAR in the network layer periodically pulls the monitoring information from each  $o$ EAR and aggregates the information. Based on the aggregated results,  $i$ EAR updates routing protocols and coordinates the monitoring schemes across multiple NICs.

Implementation details for the cross-layer approach will be provided in Section V-A, and a prototype of EAR’s multi-radio extension will be discussed in Section V-C5.

#### E. Complexity of EAR

The operation of EAR consumes less network resources than the broadcast-based approach due mainly to its use of hybrid monitoring. As the egress/cross traffic increases, EAR in each node passively monitors its traffic at the sender side, eliminating the need for active probing. With cooperative monitoring, EAR only requires a periodic report message per cycle from a cooperating node to the measuring node. Since the cooperating node shares a *hello* message to send a report every cycle, its overhead is negligible. Finally, even in case of active monitoring, EAR’s resource consumption is less than that of the broadcast approach due to its exponential active-timer, triggering active probing less frequently.

### IV. PERFORMANCE EVALUATION

We conducted simulation to evaluate EAR under controlled environments. We first describe our simulation model and then present the evaluation results of EAR.

#### A. The Simulation Model & Method

The *ns-2* [36] is used to evaluate the advantages of EAR under the approximated link-quality models. The simulation was run on the topologies of Figure 6 for evaluating the accuracy of both EAR and BAP as well as random topologies for evaluating EAR’s scalability. Note that nodes in all topologies do not move (as in mesh networks), and two adjacent nodes are separated by 150–200 m.

In all simulation runs, we used the shadowing radio propagation model in the *ns-2* to simulate time-varying wireless link quality as suggested in [20] and adjusted the standard deviation of the model as a link-quality parameter. The standard deviation is based on the values in [36], and a wireless channel is modified so that each direction of the channel can be set to

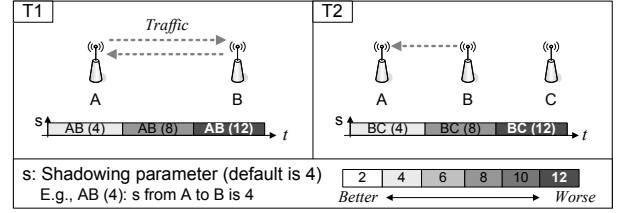


Fig. 6. Simulation topologies: We used the above topologies with different traffic and shadowing model values to evaluate the accuracy of EAR. The bottom of each topology shows the change of link quality in a time domain.

the different values to simulate asymmetric and varying link-quality. CMU 802.11 wireless MAC extension in *ns-2* is used.

Throughout the simulation, the following parameter settings were used. First, RTS/CTS handshake was disabled to study the effects of link-quality fluctuations and co-channel interferences. Second, UDP flows were mainly used to emulate users’ traffic with an exponential distribution and a packet size of 1000 bytes. Third, a default MAC data rate was set to 11 Mbps. Finally, all experiments were run for 1000 seconds, and the results of 10 runs were averaged unless specified otherwise. Note that we intentionally did not use a rate control algorithm because of the lack of relation between a radio propagation model and different modulation schemes. The SNR-based radio propagation model relies on a unit disc model, which has limitations in simulating realistic channel characteristics [5]. However, we enable the control algorithm and evaluate EAR over real wireless links in our test-bed, as we will show in Section V-B.

#### B. Accuracy

We show the accuracy of EAR with fluctuating and asymmetric link-quality and compare it with the accuracy of BAP.

1) *EAR*: We first evaluated the accuracy of each of EAR’s measurement schemes. To simulate time-variant asymmetric link-quality, we set the quality of a link’s one direction (D1) to the default value, 4, of the shadowing model, while the opposite direction (D2)’s quality is set to 4 during [0s, 200s), to 8 during [200s, 600s), and to 12 during [600s, 1000s]. Given this scenario, we used T1 of Figure 6 to evaluate the passive scheme by measuring the delivery ratio, while running one UDP flow in both directions at 1.0 Mbps. We also used the above settings without UDP traffic for the active scheme. Finally, we used the topology T2 and one UDP flow from *B* to *A* for the cooperative scheme; while changing the quality of link *B*→*C* to the same as D1 and D2 each, we measured the delivery ratio over the link between *B* and *C*.

Figure 7 shows the progression of the delivery ratio measured by EAR and BAP for stable (D1) and unstable (D2) directions of a link. First, EAR’s passive and cooperative schemes show almost the same results as the ideal case as shown in two upper figures of Figures 7 (a) and 7 (b). Specifically, the root mean-square errors of the passive scheme’s delivery ratio ( $rmse_d$ ) are 0.012 for D1 and 0.015 for D2, and those for the cooperative scheme are 0.017 and 0.021. It is worth mentioning that the use of cross traffic



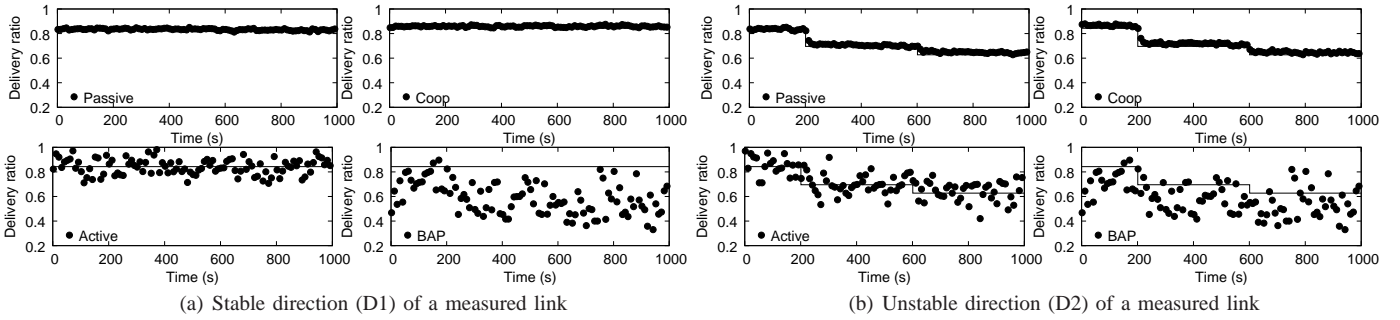


Fig. 7. Accuracies of EAR and BAP: Figure 7(a) shows that EAR yields accurate results close to the ideal values (solid lines), while BAP generates fluctuating and skewed results, affected by unstable direction. Figure 7(b) shows that EAR accurately measures the link quality even with unstable link states, whereas BAP shows inaccuracies and large variances.

in the cooperative scheme makes a marginal impact on the measurement accuracy (e.g., 0.005 difference in  $rmse$  over the passive scheme), conforming its design rationale. On the other hand, the both schemes quickly adapt themselves to the change of link quality— $rmse$  of the ratio’s standard deviation ( $rmse_s$ ) is 0.002 and 0.003 for the passive scheme, and 0.002 and 0.006 for the cooperative scheme, respectively—thanks to the use of a large portion of existing traffic as probe packets.

On the other hand, the accuracy of the active scheme lies between the previous two schemes’ and BAP’s accuracies. For example, for stable direction (D1), the active scheme increases  $rmse_d$  (0.064) by a factor of 4 over the passive scheme, whereas BAP increases  $rmse_d$  (0.287) by a factor of 26. Even though the active scheme increases the error rate due to the small number of probe packets, the error rate (7%) is much lower than BAP’s (34%). Moreover, the active scheme successfully captures link-quality asymmetry (in contrast to BAP), as shown in two lower figures of Figures 7(a) and 7(b). Note that the error in the active scheme is slightly large due to the probabilistic nature of the shadowing channel model in *ns-2*. However, as we will show in Section V-C2, the active scheme evaluated in our testbed incurs only 1.6% error, whereas BAP incurs 10.4%, demonstrating EAR’s ability of accurately capturing link quality even under complex and realistic channel conditions.

2) *BAP*: We also evaluated the accuracy of BAP for the purpose of comparison with EAR. We used topology T1 in Figure 6 with no traffic, and measured bi-directional link quality based on the link cost in Eq. (1). As shown in two BAP figures in Figure 7, BAP yields poor measurement accuracy (i.e.,  $rmse_d$  is 0.287 for D1 and 0.158 for D2), due mainly to the bi-directional nature of BAP. BAP’s accuracy is 4 times worse than the active scheme’s and 26 times worse than the passive scheme’s. On the other hand, although BAP is sensitive to varying link-quality (i.e., D2) and yields measurements relatively close to the ideal case, its variance is still (around twice) larger than the active scheme’s (Figure 7(b)).

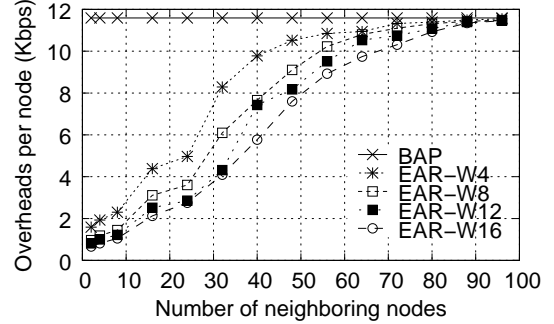


Fig. 8. Effects of the number of neighboring nodes on overheads

### C. Scalability

We evaluated the efficiency and scalability of EAR with a large number of neighboring nodes.<sup>9</sup> To simulate a large and dense WMN, we varied the number of nodes from 2 to 96 in an area of 200 m  $\times$  200 m. We measured a node’s average message rate during a measurement cycle, including the number of control and active probe packets. In this simulation, we did not transport any traffic to evaluate the EAR’s worst-case overhead (i.e., the active scheme) and did compare it with BAP through which each node injects one probe packet (of 1448 bytes) per second.

Even in the worst case, EAR measurement overheads are, on average, only one half of BAP’s, thanks to its activity/variance-based backoff timer. While maintaining a given measurement variance, EAR effectively avoids unnecessary probing of idle links. As shown by Figure 8, in case of low node density (1–10 nodes), EAR’s overhead is a one-sixth (e.g., *window*=4, or “EAR-W4”) of BAP’s. Even though the timer expires easily (thus increasing the overhead) as the number of nodes increases (10–70 nodes), EAR also reduces the overheads by an average of 50% (e.g., *window*=16, or “EAR-W16”) of BAP’s, by adjusting the maximum window size of the timer. Even in a highly dense environment (> 70 nodes), EAR’s overhead does not surpass BAP’s. Even though BAP can adopt this timer to reduce overhead, BAP triggers the timer

<sup>9</sup>The neighboring abstraction in EAR is not deterministic but probabilistic. That is, EAR actively collects *hello* messages from one-hop-away nodes and maintains links whose quality (packet-delivery ratio) is greater than a minimum threshold (0.2).

every measure-cycle because of its sharing of probing packets with all neighboring nodes and its error (large variance) in measurement (see Figure 7).

## V. SYSTEM IMPLEMENTATION AND EXPERIMENTATION

We also implemented EAR in Linux-based systems and evaluated it on our testbed. We first give the architectural details of this implementation, and then describe our experimentation setup. Finally, we present the experimental results.

### A. Implementation Details

We implemented EAR in Linux-based systems with both Pentium-based devices (e.g., laptops) and StrongARM-based devices (e.g., Stargates) and Lucent IEEE 802.11b NIC.

1) **iEAR at the network layer:** As shown in Figure 9, *iEAR* is implemented in the network layer as a loadable module of netfilter [37] and is composed of the following six components. First, *task queue with timers* is responsible for releasing periodic EAR messages, such as cooperation request/reports, and triggering measurement/update events. Next, *message and task processor* processes the EAR messages and dispatches them to the corresponding task functions in *iEAR*. If necessary, it sends/receives periodic reports and requests to/from neighboring nodes.

When measurement timers expire, *measurement components* in the middle of Figure 9 take measurements and derive link states as follows. First, the measurement scheme selected by EAR records the measurement results obtained from the *oEAR* (stamper), and then exchanges the results with neighboring nodes during the update-period, if necessary (exchanger). Finally, it updates link states and determines which measurement scheme to use for the next measurement period (transitioner).

*Link-state table and disseminator* updates the local link-state table at the end of measurement cycle. Then, the updated information is periodically disseminated to every other node (e.g., once every 30s) through a sequenced flooding message and is reflected to other nodes' link-state table. Based on the update information, the *routing-table manager* locally calculates new routing paths with link-quality-aware routing metrics, including ETX [15] and ETT [12], and invokes a kernel function that updates the kernel routing table, if

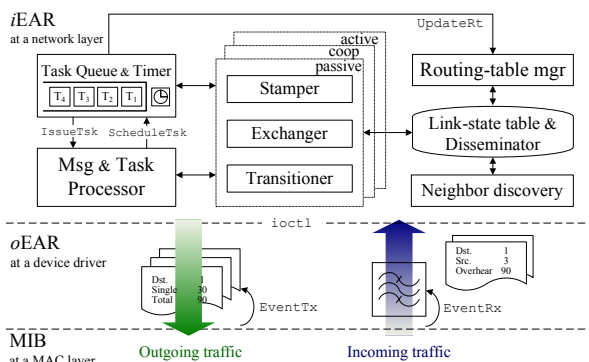


Fig. 9. EAR's software architecture: EAR is composed of an *iEAR* at the network layer and an *oEAR* at a device driver.

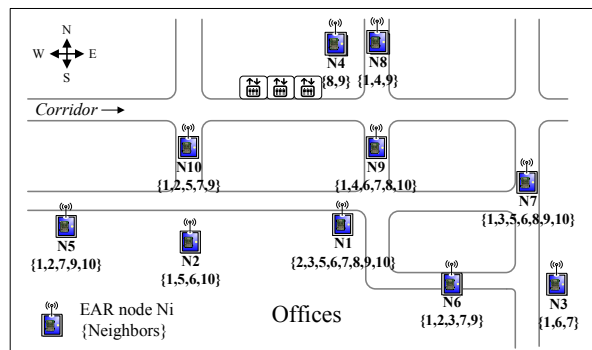


Fig. 10. EAR testbed: 10 EAR nodes are placed on either ceiling panels or high-level shelves to send/receive strong signals in the same floor of our Department building (70 m  $\times$  50 m).

there are route changes. Finally, *neighbor discovery* maintains neighbors by exchanging periodic *hello* messages.

2) **oEAR at a device driver:** *oEAR* is implemented as sub-functions in an Orinoco 802.11 Linux device driver, and is composed of two monitoring functions (i.e., outgoing and incoming traffic monitoring) and several interfaces with *iEAR* and MIB, as shown in Figure 9. First, *outgoing traffic monitoring* observes the egress traffic to each neighboring node and collects transmission statistics such as  $N_s$ ,  $N_t$  and a data rate, based on MAC MIB information. Next, *incoming traffic monitoring* overhears cross traffic. When there is a cooperation request from *iEAR*, *oEAR* switches the mode of NIC into a promiscuous mode and begins overhearing the cross traffic between two neighbors. Finally, *oEAR* has several *interfaces* through which it requests transmission/reception results from the MAC layer (i.e., *EventTx*, *EventRx*) and periodically delivers collected statistics to *iEAR* (i.e., *ioctl*).

### B. Experimental Setup

To evaluate our implementation, we constructed a testbed in the Electrical Engineering and Computer Science (EECS) Building at the University of Michigan. This building has rooms with floor-to-ceiling walls and solid wooden doors, and has relatively straight corridors. This environment provides enough multi-path effects from obstacles and interference from public wireless services.

In this environment, we deployed 10 nodes in the topology of Figure 10. We placed 5 laptops ( $N1-N5$ ) in different offices and 5 stargates ( $N6-N10$ ) along the corridors. All nodes were deliberately placed on either ceiling panels or high-level shelves to send/receive strong signals to/from neighbors.

All nodes were equipped with the same Lucent IEEE 802.11b PCMCIA card and were equipped with EAR. Each card operated at channel 11 (2.462 Ghz), less crowded channel in the building, and was set to use a built-in automatic rate control algorithm (i.e., *auto*) for its data rate. Next, each node dynamically loaded EAR into both the device driver (*oEAR*) and the network layer (*iEAR*). Finally, BAP was implemented and tested for the purpose of comparison.

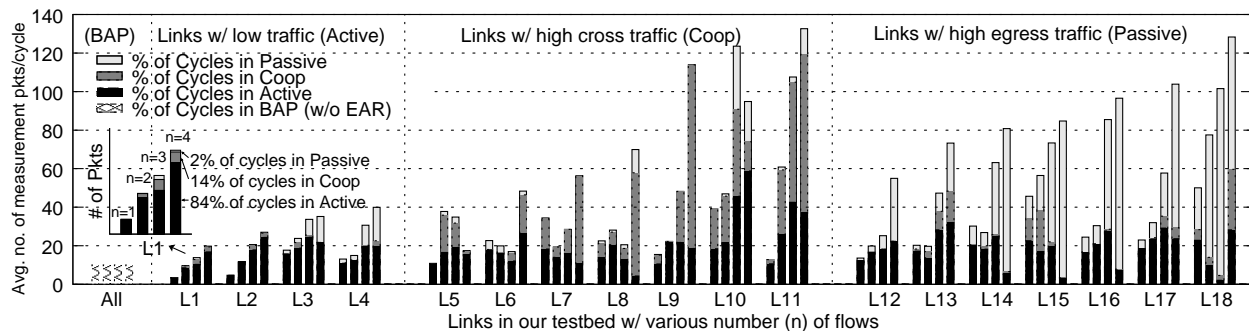


Fig. 11. Benefits of EAR’s hybrid approach: EAR effectively exploits existing traffic for their measurements through the hybrid approach. While increasing the number ( $n_f$ ) of UDP flows in our testbed, we measured the average number of packets that are used for measuring the quality of each link (i.e., the sum of both active probing packets and existing data packets), and the number of cycles (in percentage) used by each measurement scheme.

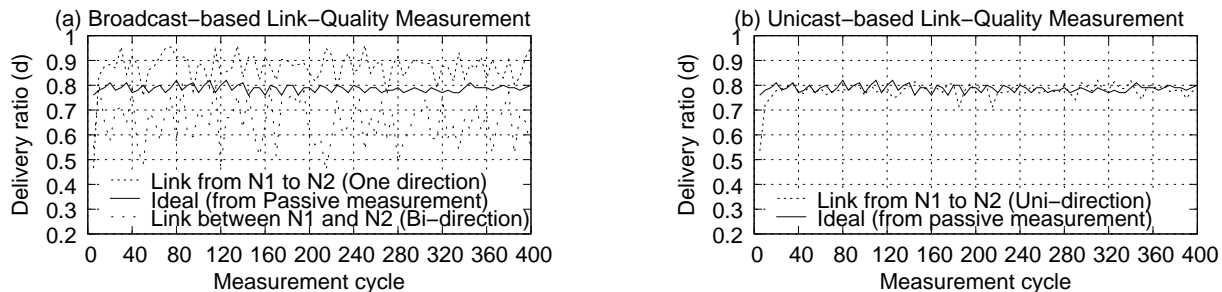


Fig. 12. Broadcast vs. unicast measurement accuracy: Broadcast usually yields high link quality because it uses lower (thus reliable) data rate than that for actual data transmission (top curve in Fig.12(a)). By contrast, EAR’s active probing provides almost the same results as ideal passive monitoring (Fig.12(b)).

### C. Experimental Results

Using the above setup, we first show how effectively EAR uses real data traffic with its hybrid approach for measuring link quality. Then, we show that by using the data traffic, EAR’s unicast-based approach measures link quality more accurately than the broadcast-based approach. Finally, we show that EAR’s uni-directional link quality effectively identifies link asymmetry, and improves the efficiency of utilizing the channel capacity over BAP’s bi-directional link quality.

1) *Effective exploitation of data traffic*: We evaluated the effects of EAR’s hybrid approach by measuring the number of probe packets per link. We ran several different numbers ( $n_f$ ) of UDP flows at 100 Kbps for 40 minutes, each pair of which were randomly chosen once every 4 minutes. While increasing  $n_f$ , we measured the average number of packets ( $n_p$ ) used for measurement of each link’s quality per cycle and derived the percentage of cycles during which each measurement scheme is used. Figure 11 plots representative links with different amounts of measurement packets.

While the broadcast-based approach uses a fixed number of probe packets (i.e., 10) per cycle, the hybrid approach in EAR indeed increases  $n_p$  as the number of flows increases. As shown in Figure 11,  $n_p$  of links with high egress traffic approaches 130, and  $n_p$  of links with high cross traffic grows up to 135 packets. On the other hand,  $n_p$  of links with low traffic is even smaller than BAP’s since EAR reduces active probing based on an exponential backoff timer.

Next, the percentage of each measurement scheme per link depends on the link’s geographical location and traffic pattern. In Figure 11, links with low traffic are located in edge nodes

in our testbed such as N3, N4 and N5, whereas links with high egress traffic are located at center nodes such as N1, N2 and N9, where large flows are often relayed. On the other hand, links with high cross traffic can be placed at center nodes that have lots of relay traffic, but might not use some links to transmit the traffic often.

2) *Improved accuracy with unicast packets*: We also evaluated the accuracy improvement of unicast-based measurement in EAR over the broadcast-based measurement. We used two adjacent nodes (N1, N2) and measured the delivery ratio of link N1→N2<sup>10</sup> with both BAP and EAR’s active probing for 400 cycles (i.e., 4000s). As a reference (called ‘Ideal’), we separately ran one UDP flow at 1 Mbps from N1 to N2 and measured the delivery ratio by EAR’s passive scheme. Note that the passive scheme provides accurate results as it derives link-quality information from the transmission of a large number of actual data packets. Due to its low, fixed data rate, BAP yields less accurate results than the unicast-based approach. The top line in Figure 12(a) shows the progression of one direction quality of link N1→N2 with broadcast probing. Since actual data transmission uses 11 Mbps, BAP generates a higher delivery ratio than the ideal does due to its low data rate (i.e., 2 Mbps), which is more tolerant of bit errors. By contrast, owing to the use of unicast packets, EAR’s measurement results (average is 0.778 (1.6% error), standard deviation 0.032) are closer to the ideal results (0.791, 0.014) than those (0.872 (10.2% error), 0.064) of BAP,

<sup>10</sup>This link is randomly chosen to show its link-quality among measurements of more than 24 wireless links in our testbed. Measurement results of all other wireless links are discussed in the next section.

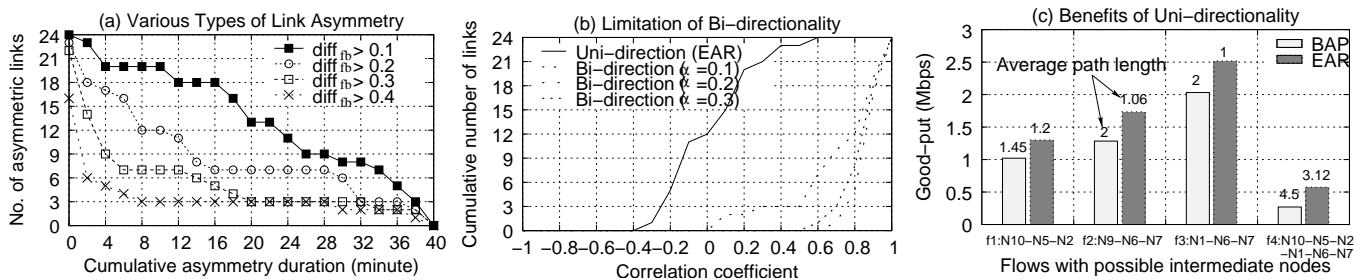


Fig. 13. Benefits of uni-directionality's on link-asymmetry awareness: Wireless link quality is often asymmetric as shown in (a), and the good-quality direction of an asymmetric link is under-estimated since the bi-directional result is affected mainly by the poor-quality direction as shown in (b). By contrast, EAR's uni-directional link quality improves capacity efficiency as shown in (c).

as shown in Figure 12(b).

On the other hand, the bi-directional link-quality information derived from BAP (the bottom line in Figure 12(a)) provides worse results than the ideal case. This is due to the poor quality of link  $N2 \rightarrow N1$  and yields under-estimated quality of link  $N1 \rightarrow N2$ . This bi-directionality is evaluated in the following experiment.

3) *Gains of uni-directionality on link asymmetry*: Before showing the uni-directionality benefits on asymmetry, we first measured the asymmetry of wireless links in our testbed and evaluated the limitation of BAP's bi-directionality on the asymmetry. To this end, we repeated the experiment in Section V-C1. This time, we fixed  $n_f$  to three, and measured the delivery ratio of all links in each direction as well as bi-direction with BAP.

From extensive measurements, we found that wireless links often have significant link asymmetry and show various interesting characteristics, in terms of lifetime and degree of asymmetry. Figure 13(a) shows the number of links in our testbed that have different asymmetry lifetimes with different link quality in each direction. For the case of  $\text{diff}_{fb} > 0.1$  (i.e.,  $|d_{forward} - d_{backward}| > 0.1$ ), a small degree of asymmetry occurs very often for short (4 minutes) to long periods (40 minutes). On the other hand, some links experience a high degree of asymmetry (e.g.,  $\text{diff}_{fb} > 0.4$ ) for more than 25 minutes of a 40-minute runtime.

Observing the various link-quality asymmetry, we found that bi-directional link quality measured by BAP is often affected by the worse-quality direction of an asymmetric link, thus yielding under-estimated results. To illustrate this, we first derived the correlation coefficient ( $\rho$ ) between bidirectional quality ( $d_i$ ) of each asymmetric link measured by BAP and the delivery ratio ( $s_i$ ) of the worse-quality direction of the asymmetric link over 40 minutes. As shown in the bi-direction cases of Figure 13(b), BAP generates skewed measurement results. More than 75% of links are closely correlated with the worse-quality direction of asymmetric links (i.e.,  $\rho > 0.8$ ). Next, we derived  $\rho$  between forward (e.g., link AB) and backward (link BA) link-quality of each asymmetric link measured by EAR. As shown in the solid line in Figure 13(b), EAR's unidirectional link-quality measurement shows independence between forward and backward link-quality. More than 75% of links show weak correlation ( $-0.2 < \rho < 0.2$ ).

Finally, we evaluated the improvement of EAR's uni-

directional link quality on utilization of asymmetric links. We began with a simple case using three nodes (N2, N5 and N10) and one UDP flow from N10 to N2. Since the quality of link  $N2 \rightarrow N10$ 's one direction is worse than the opposite direction, BAP's under-estimated bi-directional measurement makes the flow detour through N5. By contrast, EAR's unidirectional link quality enables the flow to directly route to N2, improving the good-put by 27.45% as shown in Figure 13(c). Similarly,  $N9 \rightarrow N6 \rightarrow N7$  and  $N1 \rightarrow N6 \rightarrow N7$  have 35.2% and 12.87% good-put improvements, respectively.

We further evaluated how much uni-directionality improves the overall network performance. This evaluation is done with six nodes (N1, N2, N5, N6, N7, and N10), two asymmetric links ( $N2 \rightarrow N10$ , and  $N1 \rightarrow N7$ ), and one UDP flow from N10 to N7. As shown in the  $f_4$ 's result of Figure 13(c), EAR's asymmetry awareness improves the network efficiency over BAP by up to 114%, mainly by finding shorter paths (e.g.,  $N10 \rightarrow N2 \rightarrow N1 \rightarrow N7$ ) with asymmetric links than detouring paths (e.g.,  $N10 \rightarrow N5 \rightarrow N2 \rightarrow N1 \rightarrow N6 \rightarrow N7$ ).

4) *Limitation of multi-rate BAP*: For fair evaluation, we have also implemented the multi-rate BAP (mr-BAP) and evaluated its accuracy. Recent advances in device drivers (e.g., MADWiFi [23]) for Atheros-based IEEE 802.11 chipsets enable BAP to use different data rates for broadcast transmissions, through which BAP may be able to overcome its limitation in measurement accuracy. To evaluate the effect of data rates, we have implemented mr-BAP in the MADWiFi device driver (version 0.93) on top of Atheros-based LinkSys 802.11a/b/g NICs. mr-BAP periodically injects a set of broadcast probing packets with a target data rate and derives the packet-delivery ratio in one direction of link. Note that we intentionally use one direction to exclude the effects from the bidirectionality of BAP.

Even though mr-BAP improves its accuracy over the use of a basic data rate, it still suffers from inaccuracy due to proprietary algorithms embedded into NICs. We ran mr-BAP with different data rates over a wireless link between two nodes equipped with mr-BAP in our testbed and measured packet-delivery ratios over 1000 seconds. Figure 14 shows the link-quality results measured by mr-BAP with different data rates as well as different amounts of data traffic. First, as expected, using low data rates (e.g., 1 Mbps, 18 Mbps) overestimates link-quality, as shown in Figure 14 (a) and (b) (RMSE is 0.19 and 0.23, respectively). However, mr-

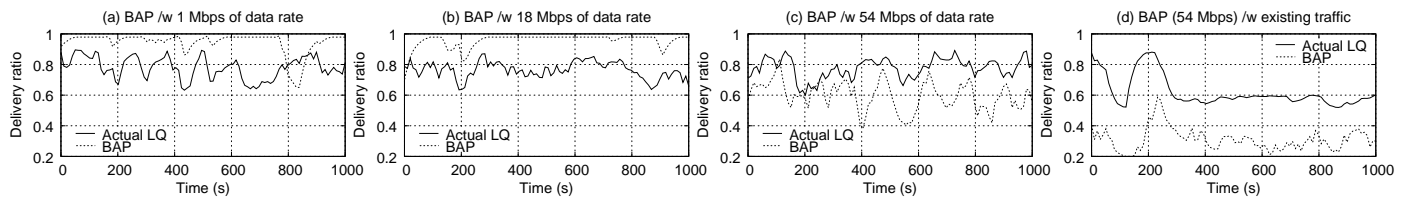


Fig. 14. Accuracy of mr-BAP: Multi-rate BAP is feasible, but inaccurate in measurement. As expected, using low data rates causes overestimation of link-quality—(a) and (b). However, using the same data rate as the actual data transmission for BAP also causes inaccuracy due to proprietary algorithms built in NICs (e.g., (c)) as well as the existence of data traffic (d). Resultant RMSE of each BAP is 0.195, 0.202, 0.203, and 0.3233, respectively.

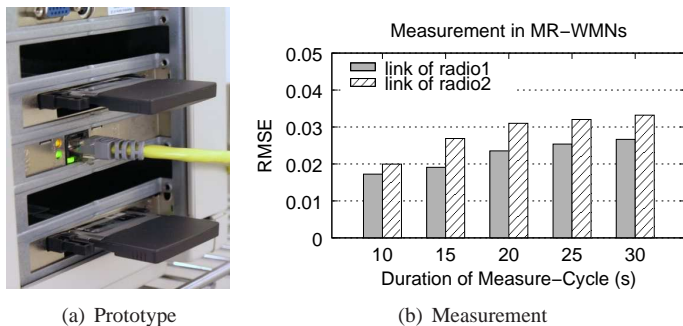


Fig. 15. Prototype and measurement of a multi-radio mesh node: EAR is implemented to support a multi-radio WMN and effectively measures the quality of links across multiple radios.

BAP with the same data rate as that for data transmissions (i.e., 54 Mbps) still experiences inaccuracy in measurement as shown in Figures 14(c) and (d). One reason would be manufacturers’ proprietary algorithms built into the NIC for broadcast transmission [38]. NIC manufacturers include their own algorithms for improving resource utilization, such as power consumption [39], which potentially causes fluctuation and inaccuracy in measurements.

5) *Support for multi-radio WMNs*: We have prototyped a multi-radio WMN (MR-WMN) and evaluated the EAR with multi-radio extension. As shown in Figure 15(a), we have built multi-radio mesh nodes, each with a low-power desktop (Pentium-III CPU, 256 MB memory) and two PCMCIA-type NICs. Multi-radio extension of EAR is implemented based on the architecture shown in Figure 5(b). Then, each radio in a node is tuned to a different orthogonal channel, and the qualities of links between two nodes are measured via EAR with multi-radio extension.

Figure 15(b) shows the progression of link-quality measured by EAR. As shown in the figure, EAR maintains accuracy, while supporting measurements for two radios (*RMSEs* in the packet-delivery ratio for radio 1 and radio 2 are 0.017 and 0.019, respectively). Thanks to EAR’s two-tier architecture, the link-quality of each radio can be measured independently. Link-quality in radio 1 is stable, and thus, more tolerant of a long measure-cycle than one in radio 2. EAR can adaptively change the measure-cycle *per radio*, depending on QoS or error-tolerance requirements.

## VI. CONCLUSION

### A. Remaining Issues

There are some remaining issues associated with EAR.

*Disseminating link-quality information*: Although this paper focused on how to measure link quality in WMNs, dissemination of the measured link-quality information is an equally important problem. Broadcast-based sequenced flooding [40] is one popular solution to this problem in small networks. There are also a couple of well-known approaches to the dissemination problem in MANETs [41], [42]. However, the information dissemination in WMNs has several challenges to overcome, including scalability and fault-tolerance. We will address these issues in a separate forthcoming paper.

*Measuring other link-quality parameters*: In this paper, the packet-delivery ratio and data rate—suitable for high-throughput metrics—are considered as the link-quality parameters. However, QoS parameters, such as delay and jitter, should be measured to support real-time applications. These parameters can be accurately measured by EAR, based on MIB [17] and NIC buffer clearing time [43]. Thus, along with the high-throughput parameters, EAR can support such applications as VoIP, IPTV that use the time-related parameters.

### B. Concluding Remarks

This paper has presented a novel link-quality measurement framework, called EAR, for wireless mesh networks. EAR is composed of three complementary measurement techniques—passive, cooperative, and active monitoring—which minimize the probing overhead and provide highly accurate link-quality information by exploiting each node’s egress and cross traffic. Moreover, based on accurate and direction-aware link-quality measurements, EAR identifies and exploits under-utilized asymmetric links, thus improving the utilization of network capacity by up to 114%. Finally, EAR is designed to be easily deployable in existing IEEE 802.11-based wireless mesh networks without any change of MAC firmware or system kernel compilation. EAR has been evaluated extensively via both *ns-2*-based simulation, and experimentation on a Linux-based implementation, demonstrating its superior accuracy and efficiency over existing measurement techniques.

## ACKNOWLEDGMENT

The work reported in this paper was supported in part by NSF under Grants CNS-0435023 and CNS-0721529. This is a superset of the paper [44] presented at ACM MobiCom 2006.

## REFERENCES

- [1] “MIT roofnet,” <http://www.pdos.lcs.mit.edu/roofnet>.

- [2] "Mesh networking summit," <http://research.microsoft.com/meshsummit>.
- [3] "Seattle wireless," <http://www.seattlewireless.net>.
- [4] "Mesh dynamics inc.," <http://www.meshdynamics.com>.
- [5] D. Aguayo, J. Bicket, S. Biswas, G. Judd, and R. Morris, "Link-level measurements from an 802.11b mesh network," in *Proceedings of ACM SigComm*, Portland, OR, Aug. 2004.
- [6] D. S. D. Couto, D. Aguayo, B. A. Chambers, and R. Morris, "Performance of multi-hop wireless networks: Shortest path is not enough," in *Proceedings of the First Workshop on Hot Topics in Networks (HotNets-I)*, Princeton, New Jersey, Oct. 2002.
- [7] S. Biswas and R. Morris, "ExOR: Opportunistic multi-hop routing for wireless networks," in *Proceedings of ACM SigComm*, Philadelphia, PA, Aug. 2005.
- [8] —, "Opportunistic routing in multi-hop wireless networks," in *Proceedings of the Second Workshop on Hot Topics in Networks (HotNets-II)*, Cambridge, MA, Nov. 2003.
- [9] C. Yu, K. G. Shin, and L. Song, "Link-layer salvaging for making routing progress in mobile ad hoc networks," in *Proceedings of ACM MobiHoc*, Urbana-Champaign, IL, May 2005.
- [10] S. Lee, B. Bhattacharjee, and S. Banerjee, "Efficient geographic routing in multihop wireless networks," in *Proceedings of ACM MobiHoc*, Urbana-Champaign, IL, May 2005.
- [11] L. Qiu, P. Bahl, A. Rao, and L. Zhou, "Troubleshooting multi-hop wireless networks," in *Proceedings of ACM SigMetrics (extended abstract)*, Alberta, Canada, Jun. 2005.
- [12] R. Draves, J. Padhye, and B. Zill, "Routing in multi-radio, multi-hop wireless mesh networks," in *Proceedings of ACM MobiCom*, Philadelphia, PA, Sep. 2004.
- [13] P. Bahl, R. Chandra, and J. Dunagan, "SSCH: Slotted seeded channel hopping for capacity improvement in IEEE 802.11 ad-hoc wireless networks," in *Proceedings of ACM MobiCom*, Philadelphia, PA, Sep. 2004.
- [14] A. Raniwala and T. Chiueh, "Architecture and algorithms for an IEEE 802.11-based multi-channel wireless mesh network," in *Proceedings of IEEE InfoCom*, Miami, FL, Mar. 2005.
- [15] D. S. D. Couto, D. Aguayo, J. Bicket, and R. Morris, "A high-throughput path metric for multi-hop wireless routing," in *Proceedings of ACM MobiCom*, San Diego, CA, Sep. 2003.
- [16] B. M. Maggs, "Asymmetric wireless networks," <http://www-2.cs.cmu.edu/bmm/wireless.html>.
- [17] "IEEE 802.11, wireless LAN medium access control (MAC) and physical layer (PHY) specifications," Standard, IEEE, Aug. 1999.
- [18] P. Bahl, A. Adya, J. Padhye, and A. Wolman, "Reconsidering wireless systems with multiple radios," *ACM SIGCOMM Computer Communication Review*, no. 5, Oct. 2004.
- [19] A. Adya, P. Bahl, J. Padhye, A. Wolman, and L. Zhou, "A multi-radio unification protocol for IEEE 802.11 wireless networks," in *Proceedings of IEEE BroadNets*, San Jose, CA, Oct. 2004.
- [20] D. Kotz, C. Newport, R. S. Gray, J. Liu, Y. Yuan, and C. Elliott, "Experimental evaluation of wireless simulation assumptions," Dept. of Computer Science, Dartmouth College, Tech. Rep. TR2004-507, 2004.
- [21] S. Haykin, "Cognitive radio: Brain-empowered wireless communications," *IEEE JSAC*, vol. 23, no. 2, Feb. 2005.
- [22] D. D. Clark, C. Partridge, J. C. Rammung, and J. T. Wroclawski, "A knowledge plane for the internet," in *Proceedings of ACM SigComm*, Karlsruhe, Germany, Aug. 2003.
- [23] "Multiband atheros driver for wifi," <http://madwifi.org/>.
- [24] C. Dovrolis, P. Ramanathan, and D. Moore, "What do packet dispersion techniques measure?" in *Proceedings of IEEE InfoCom*, Anchorage, AK, Apr. 2001.
- [25] A. B. Downey, "Using pathchar to estimate Internet link characteristics," in *Proceedings of ACM SigComm*, Cambridge, MA, Sep. 1999.
- [26] C. Hedrick, "Routing information protocol," Internet Request for Comments 1058 (rfc1058.txt), Jun. 1988.
- [27] J. Moy, "OSPF version 2," Internet Request for Comments 2328 (rfc2328.txt), Apr. 1998.
- [28] D. B. Johnson and D. A. Maltz, "Dynamic source routing in ad hoc wireless networks," in *the Book of Mobile Computing*. Kluwer Academic Publishers, 1996, vol. 353.
- [29] C. Perkins, E. Belding-Royer, and S. Das, "Ad-hoc on-demand distance vector routing," Internet Request for Comments 3561, Jul. 2003.
- [30] T. Goff, N. B. Abu-Ghazaleh, D. S. Phatak, and R. Kahvecioglu, "Preemptive routing in ad hoc networks," in *Proceedings of ACM MobiCom*, 2001, pp. 43–52.
- [31] C. Ho, K. Ramachandran, K. Almeroth, and E. Belding-Royer, "A scalable framework for wireless network monitoring," in *Proceedings of ACM WMASH*, Philadelphia, PA, Oct. 2004.
- [32] S. Seshan, M. Stemm, and R. Katz, "SPAND: Shared passive network performance discovery," in *Proceedings of USENIX Symposium on Internet Technologies and Systems*, Monterey, CA, Dec. 1997.
- [33] C. Williamson, "Internet traffic measurement," *IEEE Internet Computing*, vol. 05, no. 6, pp. 70–74, Nov. 2001.
- [34] "Linux-WLAN project," <http://www.linux-wlan.com/>.
- [35] "Atheros Communications," <http://www.atheros.com>.
- [36] "ns-2 network simulator," <http://www.isi.edu/nsnam/ns>.
- [37] "Netfilter," <http://www.netfilter.org>.
- [38] D. Giustiniano and G. Bianchi, "Are 802.11 link quality broadcast measurements always reliable?" in *Proceedings of ACM CoNEXT*, Lisboa, Portugal, Dec. 2006.
- [39] "Atheros WLAN Chipset," <http://www.atheros.com/pt/AR5006EXS.htm>.
- [40] C. Perkins and P. Bhagwat, "Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers," in *ACM SigComm*, London, UK, Sep. 1994, pp. 234–244.
- [41] R. Sivakumar, P. Sinha, and V. Bharghavan, "CEDAR: Core extraction distributed ad hoc routing," *IEEE JSAC*, vol. 17, no. 8, 1999.
- [42] S. Chen and K. Nahrstedt, "Distributed quality-of-service routing in ad hoc networks," *IEEE JSAC*, vol. 17, no. 8, pp. 1488–1505, 1999.
- [43] A. Jain, D. Qiao, and K. G. Shin, "RT-WLAN: A soft real-time extension to the orinoco linux device driver," in *Proceedings of IEEE PIMRC*, Beijing, China, Sep. 2003.
- [44] K.-H. Kim and K. G. Shin, "On accurate measurement of link quality in multi-hop wireless mesh networks," in *Proceedings of ACM MobiCom*, Los Angeles, CA, Sep. 2006.



**Kyu-Han Kim** (S'02) received the BS from the Department of Computer Science and Engineering at Korea University, Seoul, Korea in 2000 and the MS from the College of Computing at Georgia Institute of Technology, Atlanta, GA in 2003. Since 2003, he has been pursuing Ph.D degree in the Real-Time Computing Laboratory in the Department of Electrical Engineering and Computer Science, The University of Michigan, Ann Arbor, Michigan. His research interests include Quality-of-Service support in wireless networks, mobile computing, and distributed systems. He is a recipient of the ACM MobiCom Best Student Paper Award (Co-author, 2003) and the government scholarship from the ministry of information and communication, Republic of Korea.



**Kang G. Shin** (S'75-M'78-SM'83-F'92) is the Kevin and Nancy O'Connor Professor of Computer Science and Founding Director of the Real-Time Computing Laboratory in the Department of Electrical Engineering and Computer Science, The University of Michigan, Ann Arbor, Michigan.

His current research focuses on QoS-sensitive networking and computing as well as on embedded real-time OS, middleware and applications, all with emphasis on timeliness and dependability. He has supervised the completion of 58 PhD theses, and authored/coauthored about 700 technical papers (more than 240 of which are in archival journals) and numerous book chapters in the areas of distributed real-time computing and control, computer networking, fault-tolerant computing, and intelligent manufacturing. He has co-authored (jointly with C. M. Krishna) a textbook "Real-Time Systems," McGraw Hill, 1997.

He has received a number of best paper awards, including the IEEE Communications Society William R. Bennett Prize Paper Award in 2003 and an Outstanding IEEE Transactions of Automatic Control Paper Award in 1987. He has also received several institutional awards, including Distinguished Faculty Achievement Award in 2001, and Stephen Attwood Award in 2004 from The University of Michigan; a Distinguished Alumni Award of the College of Engineering, Seoul National University in 2002; 2003 IEEE RTC Technical Achievement Award; and 2006 Ho-Am Prize in Engineering (the highest honor bestowed to Korean origin engineers). He is Fellow of IEEE and ACM, and member of the Korean Academy of Engineering.