

On Exploiting Asymmetric Wireless Links via One-way Estimation

Lifeng Sang
Department of Computer
Science & Engineering
The Ohio State University
sangl@cse.ohio-
state.edu

Anish Arora
Department of Computer
Science & Engineering
The Ohio State University
anish@cse.ohio-
state.edu

Hongwei Zhang
Department of Computer
Science
Wayne State University
hzhang@cs.wayne.edu

ABSTRACT

A substantial percentage of links in wireless networks, especially low-power ones, is asymmetric. For the low-quality direction of asymmetric links, we observe based on testbed experiments that the reliability of synchronous acknowledgments is considerably higher than that of asynchronous messages. Thus the norm of estimating link quality in both directions via asynchronous beacons such as in ETX-based routing potentially underestimates the link reliability of asymmetric links. This leads us to investigate how to exploit asymmetric links in order to improve network functions such as convergecast routing in sensor networks via one-way link estimation. We propose a new one-way link metric ETF (for the *expected number of transmissions over forward links*) and present a local procedure for its estimation. We use ETF to identify high reliability forward links and use dynamic retransmission thresholding for error control and observe an improvement in convergecast routing over ETX. This is quantified with experimental testbed results with respect to reliability, number of transmissions per packet, latency, duplicates and average hops. We also study the comparative performance improvement of ETF over ETX when no special mechanism is employed to discover asymmetric links.

Categories: Network Protocols

General Terms: Algorithm, Performance, Design, Implementation

Keywords: Sensor Network, Link Estimation, Routing

1. INTRODUCTION

The bidirectional link model has been widely adopted in designing routing protocols for wireless networks [7, 19, 15, 11, 6, 16, 20]. Most routing metrics, such as ETX (for *expected number of transmissions*), evaluate the quality of a link based on its quality in both directions [7]. Due to variations in radio hardware, transmission power, interference, and environmental factors, however, a substantial percentage of links in wireless networks, especially low-power ones, are asymmetric. Selecting links based on bidirectional quality eliminates

asymmetric links and so discounts a potential resource. For example, consider a case in which source A has two routes to destination B : in one route the link reliability is 50% in both directions while in the other route the link reliability is 100% in the links towards B but 0% in the links towards A . ETX will favor the first route despite the fact that the second is better for moving data from A to B .

Ignoring asymmetric links can also lead to network disconnection if these links happen to be the cut set of the network [18]. So it is desirable to exploit asymmetric links for both the performance and the basic connectivity of wireless networks.

Asynchronous-beacon based link estimation in bidirectional metric-based routing protocols further exacerbates the problem. As we will show in Section 4, the reliability of synchronous acknowledgments in the low quality direction of asymmetric links is considerably higher than that of asynchronous messages. Since link estimation is usually based on periodic exchange of asynchronous messages among neighbors, the actual link quality may well be underestimated in those protocols using bidirectional routing metrics. This error of underestimating link quality can also degrade network performance, because potential good links may not be discovered.

Contributions of the paper. To address these issues, we propose a one-way routing metric ETF (for the *expected number of transmissions over forward links*) to exploit asymmetric links for improved network performance. In ETF, the quality of an outgoing link from a node is solely based on its forward quality, towards the destination. To justify the potential benefits of exploiting asymmetric links in routing, we analyze the physical length and stability properties of asymmetric links, and we find that most asymmetric links, especially those with high and stable forward reliability, are longer than symmetric links. We also experimentally evaluate the reliability of synchronous acks and asynchronous messages, and find that the transmission of asynchronous messages is much less reliable than that of synchronous acks, especially in the presence of interference.

To enable ETF-based routing, we address the following two challenges posed by link asymmetry:

- *Link discovery.* One challenge of link asymmetry is how a node detects asymmetric links that have good forward reliability but poor backward reliability. For instance, if the link from a node A to its neighbor B has very good reliability, then B can detect this by calculating the ratio of beacons that are successfully delivered over link $A \rightarrow B$. But if the backward reliability from B to A is very low, then B cannot inform A of the high quality of link $A \rightarrow B$. To address this challenge, we design a simple yet effective mechanism where nodes in

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

MobiHoc'07, September 9–14, 2007, Montréal, Québec, Canada.
Copyright 2007 ACM 978-1-59593-684-4/07/0009 ...\$5.00.

a neighborhood collaborate with one another to relay link estimation information for asymmetric links.

- *Error control.* Another challenge of link asymmetry is how to deal with ack loss (especially for asymmetric links with low backward reliability) which, if not handled correctly, will cause unnecessary retransmissions. To address this challenge, we design a simple mechanism that adapts the maximum number of allowable per-hop retransmissions to the forward link reliability, to ensure reliable packet delivery while reducing the number of unnecessary retransmissions at the same time.

Using a high fidelity sensor network testbed, realistic event-driven sensor network traffic trace, and synthesized periodic traffic, we run experiments to quantify the improvement in routing performance of ETF-based approach as compared with ETX-based routing. We find that exploiting asymmetric links in routing via ETF significantly improves the reliability and reduces the number of transmissions, latency, number of duplicate packets, and average routing hops in data delivery.

Organization of the paper. In Section ??, we briefly review the literature that is closely related to this paper. In Sections ?? and 4, we investigate in detail link asymmetry and the reliability of synchronous acks, which justify the effort of exploiting asymmetric links. Then, we present ETF, asymmetric-link discovery, and dynamic retransmission thresholding in Section ?. Finally, we present our experimental results in Section ?? and make concluding remarks in Section ??.

2. RELATED WORK

Several experimental study have shown that wireless links can be asymmetric and that link asymmetry can negatively impact network performance [21, 17, 12, 14, 13]. In consequence, one common approach is to avoid using asymmetric links in routing for wireless networks. This is reflected in many existing routing metrics and protocols that have been adopted in the community. To the best of our knowledge, there has not been any work that systematically studies the issue of exploiting asymmetric links in wireless networks, especially for convergecast routing in wireless sensor networks.

Based on the observations that wireless links tend to be dynamic, unreliable, and asymmetric [4, 21, 13], various routing metrics have been proposed to estimate wireless link quality. For instance, end-to-end success rate (SR) [11], round trip time (RTT) [9], packet pair delay (PktPair) [9], expected number of transmissions (ETX) [7, 19], required number of packets (RNP) [6], and the expected MAC latency per unit-distance to destination (ELD) [20] have been proposed and deployed in different wireless networks. Among these routing metrics, ETX has been shown to perform well in a variety of wireless networks [9, 19], and has been widely used in the research community. ETX is also most closely related to our one-way routing metric ETF, since the only difference between ETX and ETF is that ETX considers the quality of a link in both directions and ETF considers the quality of a link solely in the forward direction. To characterize the impact of only considering forward-direction link quality, we use ETX-based routing in our comparative study in this paper.

ETX has been proposed to minimize the number of transmissions required to deliver data packets to their destinations. The ETX metric of a link is calculated as $\frac{1}{d_f \times d_r}$, where d_f and d_r are the forward and backward reliability of the link respectively. d_f and d_r are estimated mainly via asynchronous broadcast beacons [7]. Therefore, ETX-based routing, similar

to other routing protocols, tends to avoid asymmetric links, leaving them for routing.

Kim et al. [12] proposed a framework for reliably and efficiently estimating wireless link quality. They also briefly discussed the potential benefits of exploiting asymmetric links in routing for IEEE 802.11 mesh networks. Our work in this paper complements [12] by systematically studying the properties of asymmetric links and addressing the challenges of exploiting asymmetric links for convergecast routing in sensor networks.

There has been extensive work on addressing the challenges of routing (and distributed computing in general) in unidirectional networks [14, 3]. Sensor networks with omnidirectional radios but asymmetric links are different from general unidirectional networks, since the end-points of an asymmetric link may still be close by geographically in most cases. Therefore, the issue of sharing control information in sensor networks with asymmetric links is usually simpler than that in general unidirectional networks, and we show this in Section 5.2 by designing a simple method to discover asymmetric links.

3. EMPIRICAL STUDY OF LINK ASYMMETRY

Many studies have shown that radio connectivity on low power devices is complex and non-ideal ([19, 8, 21]), especially in the presence of concurrent transmissions. Besides temporal and spatial dynamics of link properties, link asymmetry, resulted partly from variations in hardware and environmental noise, is a widely observed but less utilized complexity in wireless sensor networks. Instead of investigating the causes of asymmetry in physics, we seek a simple characterization of link asymmetry to motivate our design and development of ETF for exploiting asymmetric links.

3.1 Testbed

All the experiments in this paper were conducted on Kansei [10], a high fidelity sensor network testbed. We used the testbed's eXtreme Scale Mote (XSM), an enhanced version of Mica2 motes. Each XSM consists of a 4MHz ATmega128L microcontroller and a Chipcon CC1000 radio operating at 433MHz. It has 128KB of flash and 4KB of RAM, and runs TinyOS [1]. 210 XSMs are currently deployed on Kansei, with each XSM elevated 3 feet above the ground. In our experiments, we selected a 7 × 7 subgrid, with a 3 feet separation between neighboring nodes, from the testbed to mimic the setup of a field sensor network[5].

3.2 Link Asymmetry

Using round-robin broadcast experiments on the 7 × 7 grid, we collected empirical data to characterize link asymmetry in the target testbed. Since packet reception rate (PRR) is calculated over a relatively short period, we repeat such experiments 4 times to capture temporal variation: (1) **EXP1**: once in the morning; (2) **EXP2**: once in the afternoon; (3) **EXP3**: once in the evening; (4) **EXP4**: once at midnight. In each experiment, each node is given a turn in a round-robin manner to transmit 100 broadcast messages (one per second). With source address and sequence numbers in all traces, we measure the PRR between each pair of nodes. We only consider *connected links* that have non-zero PRR at least on one direction in the following measurements. Particularly, a link with less than 10% PRR difference on both directions is considered as a *symmetric link*, and the rest are *asymmetric*. We also consider a link with more than 90% PRR difference as

a *unidirectional link*, a special case of *asymmetric link*. Note that the notion of *unidirectional links* in this paper only refers to links with excellent reliability on one direction and close to zero reliability on the other.

Table 1 lists the average ratio of asymmetric links at dif-

PRR difference	< 10%	10-90%	> 90%	# of links
Power level 1	50%	43%	7%	500
Power level 3	65%	22%	13%	1038
Power level 9	88%	6%	6%	1155

Table 1: Link asymmetry at different power levels.

ferent power level. We see that the majority of the links are symmetric, especially at higher power levels. Half of the links are symmetric at the lowest power level 1, 65% at 3, and 88% at 9. In contrast, the ratio of asymmetric links increases when the power level decreases. Nonetheless, we do not observe similar trend for the ratio of unidirectional links. Instead, both power level 1 and 9 have lower ratio of unidirectional links, which might be due to edge effects and link diversity at certain power level. Overall, we observe that a substantial percentage of links are asymmetric, and many of them are even unidirectional. The next aspect to explore is the distribution of symmetric links and unidirectional links as a function of the physical link length (i.e., distance between the sender and receiver of a link). We see that symmetric links are likely to occur between nodes nearby, as shown in Figure 1(a). The

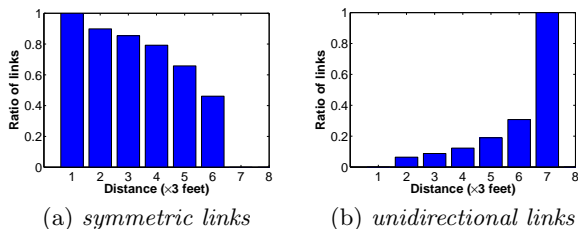


Figure 1: The ratio of *symmetric links* and *unidirectional links* as a function of physical link length when the transmission power level is 3.

reason might be that the shorter the links, the less propagation noise they may experience. On the other hand, as shown in Figure 1(b), distance seems to have positive impact on the ratio of unidirectional links. One extreme example is that links with distance 7 at power level 3 are all unidirectional. This might not be a general case, but it does indicate that more links may experience asymmetry with longer distance. In other words, if unidirectional links tend to be long, they could be exploited to result in more efficient routes.

Considering the scenario of convergecast, we are particularly interested in *unidirectional down links* that point to the base station, i.e., links with excellent reliability in the forward direction closer to the base station and bad quality in the reverse direction. The reason to explore such links is that the upper layer routing protocol would substantially benefit from these *unidirectional down links*. The topology of these links are shown in Figure 2. Compared with the results at the other two power levels, we find that some of these links (e.g. *link*(42, 12)) become disconnected at power level 1; and many links become symmetric at power level 9. For example, *link*(42 \rightarrow 21) is unidirectional at power level 3, but $PRR(42 \rightarrow 21) = 99\%$ and $PRR(21 \rightarrow 42) = 98\%$ at power level 9, which is reasonable because reachability becomes much better with the

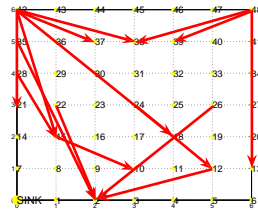


Figure 2: Topology of the *unidirectional down links* at power level 3.

higher power level. However, we find that *link*(42 \rightarrow 15) and *link*(42 \rightarrow 18) are still unidirectional at power level 9, which indicates that node 42 seems to be an inherent *bad receiver* from certain senders' view, even with relatively high transmission power.

Here is a summary of our observations as they pertain to our study.

1. A substantial percentage of links in wireless networks, especially low-power ones, are asymmetric.
2. Lower transmission power is likely to result in more asymmetric links.
3. Symmetric links tend to be short, while asymmetric links (especially unidirectional ones) are more likely to be long. This implies that effective exploitation of asymmetric links could lead to more efficient routing.

Note that the above conclusions do not include scenarios where senders and receivers use different transmission power. In these cases, the observation may not be exactly the same as we have from the testbed, but the degree of link asymmetry is likely to become even higher.

4. RELIABILITY OF SYNCHRONOUS ACKS

Cerpa et al.[6] studied temporal properties of low power wireless links and indicated that acknowledgments are better to be sent soon right after packet reception (for improved ACK reliability). In their experiments, however, they regarded asynchronous broadcast packets (at least 1 second after packet reception) as acknowledgments. Here we go one step further. We investigate the performance of synchronous acknowledgment. Our conjecture is that the link quality for synchronous ACK packets would be significantly different from that for beacons, especially in the presence of interference. This is because 1) everyone is backing off so that the channel is likely to be clear for a short acknowledgment immediately after a successful data transmission, and 2) the size of a synchronous ACK is usually much smaller than that of a normal data packet. To verify the conjecture, we select different classes of links: good, medium and bad reliability on the reverse direction at power level 3 based on the previous round robin experiments, to study the reliability of synchronous acknowledgments with and without interference. We consider explicit replies as asynchronous ACK because they are sent after a random back-off in the link layer. To reduce temporal variation, we collect data for synchronous ACK (using the default B-MAC in TinyOS) and asynchronous ACK in the same experiments as follows:

1. Synchronous/Asynchronous ACK without interference. One node sends 20000 unicast packets (one per 128 ms) to its destination with link layer ACK mechanism enabled. When the application at the destination receives

PRR difference	$\leq 10\%$	10-20%	20-30%	30-40%	40-50%	50-60%	60-70%	70-80%	80-90%	$> 90\%$
EXP1	660	56	22	16	15	12	13	33	69	139
EXP2	648	49	25	23	20	20	16	22	63	143
EXP3	692	54	27	23	15	10	16	36	47	126
EXP4	685	49	21	7	18	15	22	31	59	135
Intersection	572	3	0	0	0	0	0	0	3	65

Table 2: Detailed statistics of PRR difference at power level 3. EXP1: morning; EXP2: afternoon; EXP3: evening; EXP4: midnight; Intersection: number of links that fall in the same category in all the experiments.

a packet, it immediately replies a data packet as an explicit ACK. The reliability of synchronous ACK is calculated as the number of synchronous ACKs received at the sender over the number of data packets received at the receiver during a window of 100 packets. And the reliability of asynchronous ACK is calculated as number of explicit replies received at the sender over the number of data packets received at the receiver on a window of 100 packets interval.

2. Synchronous/Asynchronous ACK with interference. Repeat the above experiments with interference. Nodes 0, 6, 42 and 48 at the four corners each send a broadcast interfering packet immediately after they overhear a packet from the sender. Since the acknowledgment packet and the interfering packets are sent at almost the same time, they will compete for channel access.

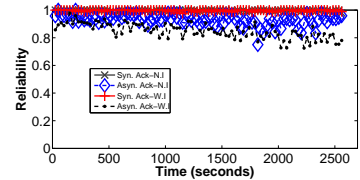
The experimental results are shown in Figure 3. We see that synchronous ACKs for good links have almost 100% reliability whether or not there is interference, while asynchronous ACKs experience some loss, which increases in the presence of interference. In the case of medium links, the reliability of synchronous ACKs is around 90% on average, and some are close to 100%. Yet the reliability of asynchronous ACKs becomes very low (e.g., around 20%), especially in the presence of interference. Another surprising observation is that even bad links have reasonable reverse reliability for synchronous ACKs in the low quality direction, as shown in 3(c). If we compare the interference data with non-interference data, we see that synchronous ACKs are much less affected by potential channel contention and collision. Since the reliability of synchronous ACK is better than that of asynchronous ACK and experiments in [6] showed that the reliability of asynchronous ACK have higher reliability than those of beacon packets, we see that the reliability of synchronous ACK is generally better than that of periodic beacons.

We also study the relation between ACK reliability and sender-receiver distance in the absence of interference. In this particular study, one node sends unicast packets to all the other nodes in the testbed in a round robin fashion. We collect data for nodes 0 and 1, and compute the average ACK reliability, as shown in Figure 4.

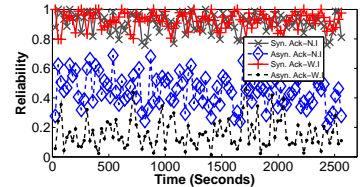
From this discussion we see that the improvement that synchronous ACK could achieve is significant (even for long links). This indicates that synchronous ACK is useful in guaranteeing the performance of ETF-based routing for both dense networks (where most links are short) and sparse networks (where links tend to be long).

5. ETF: A ONE-WAY ROUTING METRIC

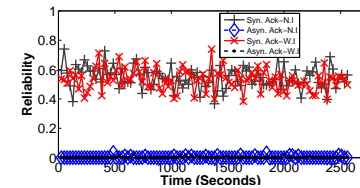
The empirical study of link asymmetry and quality of synchronous acknowledgment leads us to investigate how to exploit asymmetric links, to improve network functions such as



(a) good links



(b) medium links



(c) bad links

Figure 3: Experiments for ACK reliability. **N.I.**: No interference; **W.I.**: With interference

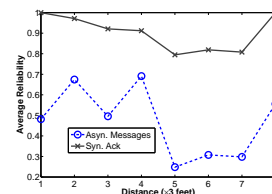


Figure 4: ACK reliability vs distance.

multi-hop convergecast routing, via one-way link estimation. In this section, we introduce ETF, followed by its implementation and discussion.

5.1 The ETF metric

ETF is the expected number of data transmissions required for a data packet over a forward link, without considering the delivery ratio of ACK packets. ETF is better than ETX in the scenario of convergecast for the following reasons:

1. The quality of forward links is the key factor for successful data delivery.
2. A substantial percentage of links are asymmetric, with a wide range of loss ratios.
3. The reliability of synchronous acknowledgments on the

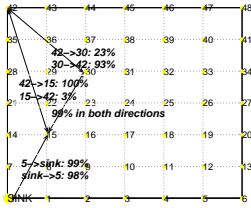


Figure 5: An example of route selection. $link(42, 15)$ is a *unidirectional link* in all the experiments. Consider a scenario where node 42 is the source, and node 15 and 30 are potential forwarders to the SINK. ETX selects path $42 \rightarrow 30 \rightarrow 15 \rightarrow SINK$ while ETF selects path $42 \rightarrow 15 \rightarrow SINK$.

reverse links is considerably higher than that of asynchronous messages.

If each attempt of transmitting a packet from node A to node B is considered a Bernoulli trial with probability $p(t)$, then

$$ETF(t)_{A \rightarrow B} = \sum_{i=1}^{\infty} (i \times (1 - p(t))^{i-1} \times p(t)) = \frac{1}{p(t)} \quad (1)$$

and the delivery ratio of $p(t)$ is measured over a period of time,

$$p(t) = \frac{recv(t - w, t)}{send(w)} \quad (2)$$

where $recv(t - w, t)$ is the number of probes received during the window w at node B , and $send(w)$ is the number of probes sent by node A , i.e., the number of probes that should have been received by node B . The ETF of a route is the sum of the ETF of each forward link along the path.

The advantage of ETF can be illustrated by a realistic example from the testbed. In Figure 5, for instance, node 42 selects node 30 as its parent if ETX is the metric, because $ETX_{42 \rightarrow 30 \rightarrow 15} = 5.69$ and $ETX_{40 \rightarrow 15} = 33.3$. It takes 5.4 transmissions on average to deliver a data packet from 42 to 15 through 30. However, only 1 transmission is actually needed to deliver a data packet directly from node 42 to 15. In this particular case, node 42 would spend much less number of data transmissions if ETF is the metric. If these links happen to be some of the *important links* [18] that join well connected components in certain network topologies, the choice of ETF would substantially improve the network performance. The next question is how to obtain ETF, especially for asymmetric links.

5.2 Discovering Asymmetric Links

Unlike routing implementations in wireless ad hoc networks, most sensor network applications prefer low-rate beaconing (e.g. one beacon every 30 seconds) due to energy constraints. Thus connectivity updates along low-quality links may be delivered only infrequently. In Figure 6, for instance, $link(S \rightarrow B)$ is an asymmetric link with 95% forward reliability and 5% backward reliability. Delivery ratio of $S \rightarrow B$ is recorded at node B . In basic distance-vector routing, node S knows the forward quality only when it receives some reports from B . However, chances for such a direct report to be received at node S is extremely low, only 5%. Statistically speaking, node S may know the forward delivery ratio eventually, but it might be too late to take advantage of the excellent forward

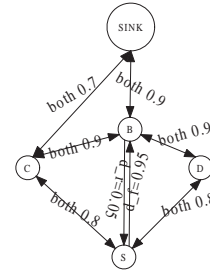


Figure 6: An example of discovering asymmetry.

quality. Moreover, some links may be completely unidirectional, as we find in our testbed, then the source nodes would have no way of knowing such a fact. This motivates us to design an effective strategy for locally discovering asymmetric links. Forward quality of asymmetric links can be obtained in an indirect manner based on the simple density argument as follows. In Figure 6, for instance, node C learns the quality of $Link(S \rightarrow B)$ and $Link(B \rightarrow S)$ when node B and S broadcast their reception reliability. C can then help convey this information. Certainly it requires C to keep track of information of $Link(S \rightarrow B)$ and $Link(B \rightarrow S)$. Due to memory constraint, however, it is impossible for C to keep information about all the links around (especially in a dense network). To address this issue, we can design a memory-efficient asymmetric link discovery method based on the following observations: (1) A node only needs to help a neighbor with which a high-quality bidirectional link exists. In Figure 6, for example, it is reasonable for C to keep information about the good forward link SB only if link quality of CS is good in both directions. This is because 1) it is difficult for C to learn from S that $Link(B \rightarrow S)$ is bad if the quality of $Link(S \rightarrow C)$ is low, and 2) it is difficult for C 's report to be received by S if $Link(C \rightarrow S)$ is bad. On the other hand, if C does not help because the quality of link SC is low in either direction, there exists with high probability another node (e.g., D) in the neighborhood who can help as long as the network connectivity (or density) is not too low. (2) A node only needs to keep information about asymmetric links and for asymmetric links, information about the direction where the link quality is high.

When a third node detects an asymmetric link of interest, it can either put this information in his beacon packets or it can send a control packet to the node that may not be able to directly discover the asymmetry. If beacon packets can accommodate the information of asymmetry discovery, the first 'piggyback' choice may be better since no additional control packets are introduced. In our current implementation, we adopt the second choice since 1) we want the discovery procedure to be independent of the link quality estimation protocol, 2) we want to ensure reliable delivery of asymmetry-discovery-information via link-level ACK, and 3) the payload size of each packet is limited in TinyOS (29 bytes by default), and beacon packets may need to accommodate other information such as reception reliability. Such an implementation introduces extra communication overhead due to control packets. To avoid control packet explosion, similar discovery reports are suppressed. In addition, such a discovery packet does not need to be reported every time a node detects asymmetry. A node needs to sharing information about asymmetric links only once every estimation period.

Next we discuss the overhead of control packets. For each asymmetric link that needs to be discovered indirectly, the number of extra transmission is upper bounded by Δ , the

number of detectors in every neighborhood. In fact, it is also upper bounded by $\min(\Delta, 5)$. If the communication model were a perfect unit disk, the worst case transmission overhead would occur when all reporters are on the disk boundary and just outside each others communication range. In this case, up to five messages could be sent to inform link asymmetry in the disk. In practice though, the common neighborhood of nodes having an asymmetric link is likely to be much smaller than this unit disk, so the number of transmissions would be fewer than $\min(\Delta, 5)$. Since control packets are introduced only for links of high degree of asymmetry, and the control packet is transmitted only once every estimation period, the communication overhead is low compared with that of beacon packets.

The pseudo code for current asymmetry discovery implementation is presented in Listing 1, where the threshold value could be adjusted according to application requirements.

Listing 1: Asymmetry Discovery

```

AsymmetryDiscovery(ngbr1, ngbr2, r){
  //r is the reliability of Link(ngbr1->ngbr2)
  if r>85%
    if exists(ngbr2, ngbr1), remove (ngbr2, ngbr1);
    elseif similar fact heard for 4 times
      schedule a report to ngbr1;
    elseif there is a space for Link(ngbr1->ngbr2)
      and quality(me, ngbr1) is good
      keep Link(ngbr1->ngbr2);
    elseif r<15%
      if exists(ngbr2, ngbr1)
        schedule a report to ngbr2;
    elseif exists(ngbr1, ngbr2) || exists(ngbr2, ngbr1)
      remove that entry;
}

TimerFired() {
  if a pending control packet is timeout
    send it;
}

OverhearControlPacket(msg){
  if msg for me
    update link information with msg;
  else if I have a similar control packet
    suppress mine;
}

```

With such a local procedure, ETF can be effectively estimated either directly or indirectly in the presence of link asymmetry. The potential cost of using ETF is that the sender may be unaware of the success of packet delivery, especially if unidirectional links are adopted in routing. To address this issue, one could use strategies such as indirect acknowledgments via common neighbors (similar to the indirect asymmetric link discovery). In this paper, however, we explore a simpler solution that is based on dynamic retransmission threshold, as to be discussed in the next subsection.

5.3 Dynamic Retransmission Threshold

Retransmission is a commonly used technique to compensate for loss. It can be done either at the routing layer or link layer. In many protocols, a sender retransmits a packet that is not successfully acknowledged up to a pre-defined application-wide threshold. The use of retransmission scheme introduces interesting temporal properties. A high threshold increases the chance for a packet to be received, while it also introduces duplicates and interference because of ACK loss. Low threshold

reduces the extra attempts but a packet may never get through a bad link after a maximum number of attempts. Instead of using a static threshold, we prefer a dynamic thresholding that depends on the quality of forward links, or ETF,

$$threshold = \min\{\text{ceil}(\theta(\text{reliability})), MTC\} \quad (3)$$

$$\theta(\text{reliability}) = \begin{cases} 1 & \text{if reliability}=1 \\ \frac{\log(0.01)}{\log(1-\text{reliability})} & \text{otherwise} \end{cases} \quad (4)$$

where MTC is the pre-defined maximum transmission count for the whole system, and $\theta(\text{reliability})$ is the number of transmissions required to achieve at least 99% reliability. The $threshold$ gives a statistical fit for the number of transmissions required to deliver a packet without any waste of energy for unnecessary retransmissions.

5.4 Discussion

Effectiveness of ETF over ETX. One may argue that a link with 90% reliability on both directions might be better than a unidirectional link with 100% forward reliability and so ETF may not be a wise choice in this particular case. To answer this question, we explain in this section that ETF is a better indicator of link quality than ETX on more than 80% of the links by considering the fact that the reliability of synchronous ACKs is significantly under-estimated by asynchronous beacons. Let us consider a scenario where node A is a sender and nodes B and C are two potential forwarders. Both B and C have the same path quality to the base station. $Link(AB)$ has forward reliability u_1 , reverse reliability v_{1b} estimated via broadcast probes, and unknown true reverse reliability v_1 for ACKs. $Link(AC)$ has forward reliability u_2 , reverse reliability v_{2b} estimated via broadcast probes, and unknown true reverse reliability v_2 for ACKs. Let us assume $u_1 \times v_{1b} \geq u_2 \times v_{2b}$ without loss of generality.

Case 1: if $u_1 \geq u_2$, both ETF and ETX chooses node B as the relay node according to the definition.

Case 2: if $u_2 > u_1$, ETF would choose node C , while ETX would choose node B . For ETF, node A would spend $\sum_{i=1}^{k-1} (i \times (1 - u_2 \times v_2)^{i-1} \times (u_2 \times v_2)) + k \times (1 - u_2 \times v_2)^{k-1} = \frac{1 - (1 - u_2 \times v_2)^k}{u_2 \times v_2} + k \times (1 - u_2 \times v_2)^{k-1}$ sending $(1 - (1 - u_2)^k)$ of the packets over the forward link, where k is the *threshold* defined in Equation 3. So the average number of transmissions per packet received over $link(AC)$ would be $\varphi(u_2, v_2, k)$ (φ is defined at Equation 6), similarly, the true average number of transmissions per packet received over $link(AB)$ using ETX would be $\varphi(u_1, v_1, MTC)$, where MTC is the maximum number of transmissions for ETX. ETF is better than ETX if the following inequality holds,

$$\varphi(u_2, v_2, k) < \varphi(u_1, v_1, MTC) \quad (5)$$

where φ is a function,

$$\varphi(u, v, n) = \frac{1 + (n \times u \times v - 1) \times (1 - u \times v)^{n-1}}{u \times v \times (1 - (1 - u)^n)} \quad (6)$$

If $MTC = 1$, then $k = 1$, the left part of Inequality 5 becomes $\frac{1}{u_2}$ and the right part becomes $\frac{1}{u_1}$, which is obviously true for Inequality 5. If $MTC \rightarrow \infty$, then the left part remains $\varphi(u_2, v_2, k)$, while the right part becomes $\frac{1}{u_1 \times v_1}$. Regardless of the impact of MTC , we are more interested in how much improvement we could achieve via ETF instead of ETX. By applying the ACK error model obtained from testbed experiments in Section 4, we calculate, as a function of u_2 , the ratio

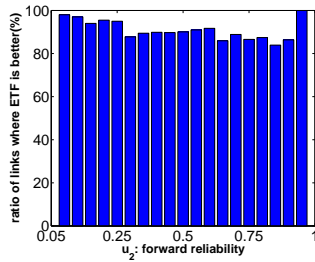
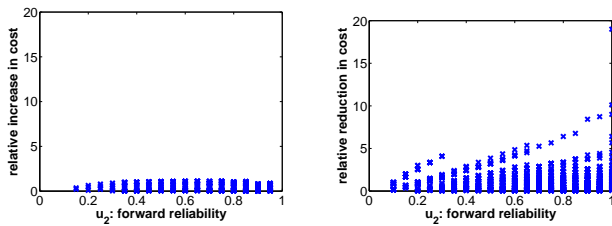


Figure 7: The percentage of cases where ETF is better.

of cases when ETF is better than ETX. For a fixed u_2 , the ratio is calculated by varying the other three variables uniformly between 0 to 1 with a 0.05 step. The results are shown in Figure 7. We see that ETF outperforms ETX for more than 80% of the links, with u_1 , v_1 , u_2 , and v_2 are uniformly selected. Particularly, it is not surprising to see that ETF is 100% better than ETX when $u_2 = 1$, because ETF spends only one transmission to successfully deliver a data packet while ETX spends more than one transmission on average no matter how the reverse link reliability is. Taking the number of transmissions required for each packet as the cost, we calculate the increased cost of using ETF instead of ETX on the remaining 20% links where ETF is worse than ETX (i.e., in *bad cases*), and the result is shown in Figure 8(a). We see that all the



(a) *Relative increase in cost when using ETF in the bad cases.* (b) *Relative reduction in cost when using ETF in the good cases.*

Figure 8: Impact of using ETF in different cases.

extra relative cost is less than 1.2, and most of them are less than 0.6. Nonetheless, if we check the improvement achieved by using ETF (i.e., in *good cases*) as shown in Figure 8(b), we see that benefit of using ETF is substantial, where many nodes only pay half of the price when using ETF instead of ETX.

Neighbor table management. When neighbor table size is limited, it may be impossible to maintain information about all the neighbors. In this case, an eviction policy should be carefully designed. In our implementation, an entry with the least priority is evicted when a new neighbor is discovered while the table is full. The priority is defined by a set of metrics such as path quality, neighbor’s path quality, current reception rate, and hop count. A subtle detail of the neighbor table management is that a newly added neighbor should not be replaced unless its quality has been reliably estimated (e.g., by at least a round of estimation). We conduct a simple experiment in Section ?? to show the importance of this subtlety.

6. EXPERIMENTAL EVALUATION

In this section, we evaluate, via testbed based experiments in convergecast routing, the performance of ETF and compare it with that of ETX.

6.1 Experiment Design

6.1.1 Protocols studied

We have implemented in TinyOS a distance-vector routing framework where different aspects of routing (e.g., routing metric and neighbor table management mechanisms) can be evaluated in the same setting. To study the impact of exploiting asymmetric links on routing and to examine some design decisions of ETF, we instantiate the routing framework in different ways to get the following routing protocols:

- *ETX*: convergecast using routing metric ETX.
- *ETX-pre*: same as ETX, but a newly added neighbor can be replaced even if its quality has not been reliably estimated (e.g., by a round of estimation) yet. This is to study the importance of proper neighborhood management, especially in high-density networks.
- *ETF*: convergecast using routing metric ETF.
- *ETF-NU*: same as ETF, except that the explicit asymmetric-link discovery is disabled. This is to study whether ETF will work well without explicit asymmetric-link discovery which is designed for detecting links with extreme asymmetry (e.g., unidirectional links).

(Note that, for simplicity, we name protocols primarily based on their routing metrics.)

6.1.2 Performance Metrics

We use the following metrics to compare the performance of different routing protocols:

- *End-to-End Reliability*: number of unique packets received at the base station divided by the number of packets originated.
- *Number of Transmissions Per Packet (TXPP)*: *TXPP* is the expected number of transmissions, including retries, required for delivering a packet from its source to the base station. It is calculated as the total number of transmissions divided by the total number of unique packets received at the base station. It reflects the reliability of selected relay nodes, path optimality, throughput, and energy efficiency in routing.
- *End-to-End Latency*: the time taken for a packet to be delivered from its source to the base station. Latency is a critical consideration in many mission-critical sensor networks such as those for security surveillance.
- *Duplicates*: the number of duplicate packets found at all the nodes in the network. Duplicate packets are mostly due to ack loss. Duplicates incur extra energy consumption and introduce extra interference.
- *Hop Count*: the average number of hops in the routes used in packet delivery.

6.1.3 Testbed

We use the 7×7 grid testbed as discussed in Section 3.1 for the performance evaluation, and we set the mote at the left-bottom corner of the grid as the base station.

Varying network density. The actual network density (or connectivity) in the testbed depends, to a great extent, on the radio transmission power level adopted by the motes. To illustrate this, we provide in Figure 9 the boxplots of packet

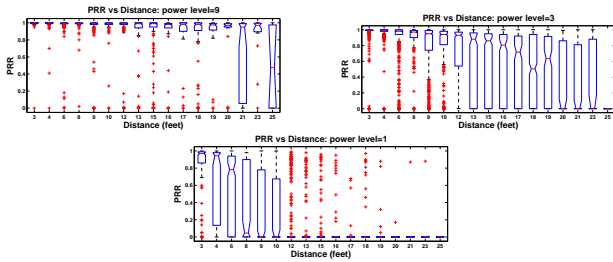


Figure 9: Testbed density at power level 9, 3 and 1.

reception rate (PRR) as a function of the distances between senders and receivers, when the power level is 9, 3 and 1 (out of a range of [1, 255]) respectively. Clearly, link connectivity generally falls off with increasing distance, but not uniformly though. Nodes geographically faraway may have excellent connectivity, whereas nodes nearby may have poor link quality.

From the figure, we see that the network is sparse at power level 1, where most nodes can only talk to a limited number of nodes nearby. At power level 9, the density is fairly high since almost any two nodes can talk to each other with high probability. At power level 3, a node is able to reach many nodes in the network, and some links are reliable while some are not. Thus, power level 3 gives us a typical multi-hop network (about 4-5 hops). We will conduct experiments at different power levels to study the impact of network density on the performance of routing protocols.

6.1.4 Traffic Trace

To mimic sensor network traffic pattern, we use both bursty-event traffic and periodic traffic.

The event traffic pattern is based on a publicly available traffic trace [2] collected in a real-world intrusion detection sensor network. In the traffic trace, each node of a 7×7 grid, except for the base station, generates two packets denoting the start and the end of its local detection of an intrusion event. A total of 96 packets are generated each time the event occurs.

For each experiment configuration (e.g., a fixed radio transmission power level) with the event traffic trace, we run 20 runs of the same experiments, with an interval of 3 minutes between two consecutive experiments.

To reflect a typical traffic pattern in data-collection sensor networks, we also evaluate the performance of different routing protocols using a trace of periodic traffic. More specially, each node generates a packet every 30 seconds on average in the periodic traffic trace.

6.1.5 Maximum Transmission Count (MTC)

To improve packet delivery reliability, one common approach is to retransmit lost packets. That is, when a node sends a packet to its next-hop forwarder (or the base station), the node will keep transmitting the packet until an ACK is received or the packet has been transmitted up to a threshold number of times, which we call the maximum transmission count (MTC). It is usually difficult to determine a perfect value for MTC. Large MTC may introduce high overhead because of ACK loss, while small MTC may lead to low packet delivery reliability. In our experiments, MTC is set as 8 (a typical value that is also used in IEEE 802.15.4) unless explicitly mentioned otherwise.

6.2 Experimental Results

In our study, we find out that the general observations are quite similar for experiments based on event traffic and those based on periodic traffic. Therefore, we focus on presenting

results for experiments with event-traffic in this section, and we only briefly discuss some summary data for periodic traffic based experiments at the end of the section. For event traffic based experiments, we first present the results for networks of different densities, and then we present the impact of MTC on the performance of different routing protocols.

6.2.1 Dense Network: Transmission Power Level=9

Neighbor table management is very important in dense networks, since a node can only keep state information about a selected set of its neighbors. Therefore, besides ETX and ETF, we also study the performance of ETX-pre to understand the impact of proper neighbor table management.

Table 3 shows the summary performance data for ETX-pre,

	Reliability	TXPP	Delay	Dup.	Hops
ETX-pre	66.88%	5.55	0.096	561	1.67
ETX	84.79%	3.69	0.160	177	3.06
ETF	94.84%	1.78	0.095	91	1.52

Table 3: Performance of ETF and ETX at power level 9.

ETX, and ETF respectively, and Figure 10 shows the packet

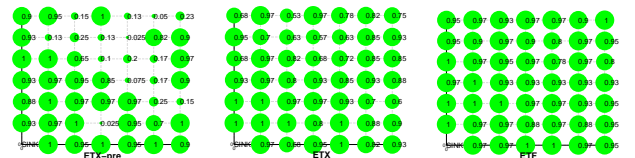


Figure 10: The packet delivery ratio when transmission power level is 9.

delivery reliability from individual nodes in different routing protocols. We see that the performance of ETX-pre is quite low (even compared with ETX), and that the packet delivery ratios for some nodes (e.g., those in the middle-upper part) are extremely low. After detailed data analysis, we find that most of the packet transmission failures are due to packets being directly sent from those nodes to the base station, even though those links are of low reliability. This is also part of the reasons why ETX-pre has lower average hop count than ETX. Looking deeper, we found that better potential links in ETX seem to be evicted from the neighbor table in the routing convergence process as a result of inappropriate neighbor table management. With the enhanced neighbor table management as described in Section 5.4, ETX has much better performance than ETX-pre. Therefore, we use the same enhanced neighbor table management mechanism in ETF.

From Table 3, we see that, even though ETX seems to perform well at power level 9, it is still not as good as ETF. Compared with ETX, ETF improved the reliability from 84.79% to 94.84%, with less than half of the TXPP required and even lower latency. ETF also has better performance in terms of duplicates and average hops.

Recall that we have observed, in Section ??, that $link(42 \rightarrow 18)$ seems to be unidirectional at power level 9. We find that a fair number of packets from node 42 are successfully forwarded through $link(42 \rightarrow 18)$ in ETF. Yet in ETX, most packets from node 42 take the route of $42 \rightarrow 17 \rightarrow 0$. As shown in Figure 10, accordingly, the base station only receives 68% of the packets from node 42 in ETX, even though it receives 95% of the packets from node 42 in ETF.

6.2.2 Regular Network: Transmission Power Level=3

	Reliability	TXPP	Delay	Dup.	Hops
ETX	76.51%	4.87	0.115	467	2.08
ETF-NU	82.50%	2.83	0.104	336	2.13
ETF	88.39%	2.48	0.096	197	2.02

Table 4: A comparison of reliability, TXPP, latency, duplicates and average hops at power level 3.

Since the testbed at power level 3 mimics typical real-world sensor network connectivity, we discuss the results of this set of experiments in more detail. We also use power level 3 to investigate the impact of other factors such as MTC and traffic pattern later. To study whether ETF works without the explicit asymmetric-link discovery mechanism, we also measure the performance of ETX-NU when the power level is 3.

A comparison of reliability, number of transmissions per packet, latency, duplicates and mean hops at power level 3 is given in Table 4.

The reliability is 88.39% for ETF, 82.50% for ETF-NU, and only 76.51% for ETX in this scenario. As expected, ETF yields far fewer number of transmissions per packet, 2.48 compared to 4.87 with ETX, because it yields a better choice of forwarders. We also find that ETF-NU spends comparable TXPP (2.83), which indicates that ETF is more energy efficient than ETX, even when asymmetric-link discovery is not enabled, i.e., when links with extreme asymmetry (e.g., unidirectional links) are not used. (Note that links with moderate asymmetry can still be detected and used in ETF-NU.) The latency in ETF and ETF-NU is slightly better than that in ETX, but would be much better if lost packets could be taken into account in some way. Again, ETF has much less duplicates than ETX. As to average hop count, ETX and ETF are comparable, and ETF-NU has a little more. But this achievement does not help ETX to improve the reliability and reduce the latency. Since the traffic pattern is derived from an intruder detection application, we are also interested in the event reliability that is calculated based on each event, as shown in Figure 11. As

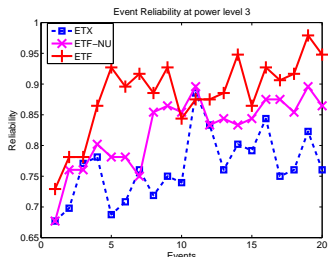


Figure 11: Comparison of event reliability at power level 3.

we can see, initially, the reliability of ETF is around 72%. With the help of asymmetric links, it becomes better. Again, ETF-NU also outperforms ETX in this measurement.

To understand the underlying reasons for the improved performance of ETF and ETF-NU, we analyze the properties of the links used in different protocols. Table 5 shows the mean reliability, coefficient of variation in reliability, and the mean length of the links used in different protocols. It is interesting to see that, even though the links used in ETF are longer than those in ETX, the links used in ETF are still more reliable and stable than those used in ETX. Thus exploiting asymmetric links in routing (via ETF-NU or ETF) can help find those long, reliable, and stable links in the network. By comparing ETF with ETF-NU, we can see that the explicit asymmetric-link discovery mechanism used in ETF also helps in identifying

	ETX	ETF-NU	ETF
link reliability	70.4%	72.1%	78.3%
C.O.V. of reliability	0.142	0.109	0.107
link length (feet)	9.05	9.22	9.90

Table 5: Quality of links used in routing. Here *link reliability* refers to the mean reliability of all the links used, *C.O.V.* refers to the coefficient of variation (i.e., standard deviation divided by mean) in link reliability and measures the stability of the used links, and *link length* refers to the mean length of the links used in routing.

those links of extreme asymmetry (e.g., unidirectional links) and thus further improves the routing performance.

6.2.3 Sparse Network: Transmission Power Level=1

The routing performance at power level 1 is shown in Table 6. We see that the reliability of ETX and ETF are both

	Reliability	TXPP	Delay	Dup.	Hops
ETX	56.51%	14.21	0.292	999	7.23
ETF	57.71%	7.25	0.197	945	4.39

Table 6: Performance of ETF and ETX at power level 1.

low, mostly because the link quality at the lowest power level is extremely bad. Nonetheless, TXPP in ETF is only about half of that in ETX, and the improvement in latency is also substantial, showing the benefits of exploring asymmetric links even in sparse networks.

Summary. From the discussions above and the data shown in Tables 3, 4, and 6, we see that, compared with ETX, ETF significantly improves the energy efficiency and reduces packet delivery latency irrespective of network density. ETF also improves packet delivery reliability, especially in regular to dense networks (e.g., at power levels 3 and 9). Note that current results of the TXPP does not include extra transmissions for asymmetry discovery in ETF. Although this number is much fewer than that of mass data transmissions, the TXPP in ETF would be a little bit more if the number of control packets are counted. Again, if the implementation puts the discovery information in the beacon packets, then no extra communication overhead is required.

6.2.4 Impact of MTC

To study the impact of MTC on the performance of routing protocols, we measure and compare the performance of ETF and ETX when MTC is set to 2, 4, and 8 respectively. Since the change of MTC does not affect the mean hop count much, here we only present in Figure 12 the performance results in terms of reliability, TXPP, latency and duplicates. We see that the reliability of ETF is higher than that of ETX in all the three cases. Change of MTC has more impact on the performance ETX than that of ETF. This is because ETF uses dynamic retransmission thresholding according to link reliability. One interesting result is that ETX achieves 82.19% reliability when $MTC = 4$, while it performs worse with both higher and lower MTCs. This again suggests that it may well be difficult to determine a unanimous MTC in advance to fit all network conditions. Besides reliability, ETF substantially outperforms ETX in terms of the number of transmissions per packet received at the base station and duplicates. As to the mean packet delivery latency, ETX is a little better than ETF, but this is mainly because more packets far away from the base

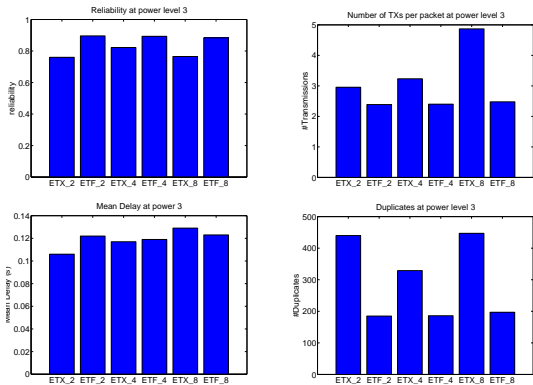


Figure 12: The impact of MTC at power level 3. The number following ETF_/ETX_ is the value of MTC.

station are received in ETF routing and they contribute to the increase of average latency.

6.2.5 Periodic Traffic

In addition to the bursty-event trace, we have also evaluated ETF and ETX at power level 3 with periodic traffic, where each node reports an event every 30 seconds preceded by a random delay. We summarize the experimental results in Table 7. We see that the reliability of ETF is considerably better than that of ETX, increasing from 70.76% to 90.33%. The TXPP and average latency are lower for ETF, and the average hops is comparable for both. We have also observed an exception that ETF has more duplicates in this case. One reason is that far more packets have traversed along the path to the base station, which has yielded duplicates. Another reason could be that ACK loss occurred on the low quality direction of some asymmetric links exploited by ETF. However, the overall performance of ETF is much better than ETX with this periodic traffic pattern.

	Reliability	TXPP	Delay	Dup.	Hops
ETX	70.76%	2.86	0.081	247	1.74
ETF	90.33%	2.03	0.075	365	1.68

Table 7: Performance of ETF and ETX with periodic traffic at power level 3.

Based on the above experimental results, we have observed that exploiting asymmetric links via ETF substantially improves routing performance, even when the explicit asymmetric-link discovery mechanism is disabled.

7. CONCLUDING REMARKS

Unlike existing work that tries to avoid asymmetric links, we have explored the benefits of exploiting asymmetric links in wireless networks. To this end, we have examined in detail the link asymmetry and the quality of synchronous acknowledgments based on testbed experiments. We have also proposed a one-way link metric ETF, and addressed the challenges of link asymmetry to routing protocol design by proposing the collaborative asymmetric-link discovery and the dynamic retransmission thresholding mechanisms. Despite the great deal of prior work in the context of link metrics, we believe that our work provides a unique one-way solution to the link estimation problem in wireless sensor networks. Through detailed experimental study in a high fidelity testbed, we find that exploiting asymmetric links (via ETF) significantly improves the

performance of functions such as convergecast routing in sensor networks. Even though we have mainly focused on sensor networks in this paper, we believe that asymmetric links can also be exploited for performance optimization in other networks such as wireless mesh networks, and we will explore this potential opportunity in our future work. We will also explore how to take advantage of the high reliability of synchronous ACKs in other routing paradigms such as opportunistic routing.

8. ACKNOWLEDGMENT

We thank Bill Leal and Vinodkrishnan Kulathumani for comments on the manuscript, Mukundan Sridharan for help on the testbed experiments, and the anonymous reviewers for their thoughtful suggestions. This material is based upon work supported by the National Science Foundation under Grants No. 0520222 and 0341703.

9. REFERENCES

- [1] <http://www.tinyos.net>.
- [2] Traffic trace from a field sensor network. <http://www.cs.wayne.edu/~hzhang/group/publications/Lites-trace.txt>.
- [3] Y. Afek and E. Gafni. Distributed algorithms for unidirectional networks. *SIAM Journal on Computing*, 23(6):1152–1178, 1994.
- [4] D. Aguayo, J. Bicket, S. Biswas, G. Judd, and R. Morris. Link-level measurements from an 802.11b mesh network. *ACM SIGCOMM*, 2004.
- [5] A. Arora and et al. A line in the sand: A wireless sensor network for target detection, classification, and tracking. *Computer Networks (Elsevier)*, 46(5), 2004.
- [6] A. Cerpa, J. L. Wong, M. Potkonjak, and D. Estrin. Temporal properties of low power wireless links: Modeling and implications on multi-hop routing. *CENS Technical Report 0044*, 2005.
- [7] D. S. J. D. Couto, D. Aguayo, J. Bicket, and R. Morris. A highthroughput path metric for multihop wireless routing. *ACM MobiCom*, 2003.
- [8] D. Ganesan, D. Estrin, A. Woo, D. Culler, B. Krishnamachari, and S. Wicker. Complex behavior at scale: An experimental study of low-power wireless sensor networks. *Technical Report CS TR 02-0013, UCLA*, 2002.
- [9] R. Draves, J. Padhye, and B. Zill. Comparison of routing metrics for static multi-hop wireless networks. *ACM SIGCOMM*, 2004.
- [10] E. Ertin and et al. Kansei: A testbed for sensing at scale. *IEEE/ACM IPSN/SPOTS*, 2006.
- [11] O. Gnawali, M. Yarvis, J. Heidemann, and R. Govindan. Interaction of retransmission, blacklisting, and routing metrics for reliability in sensor network routing. *IEEE SECON*, 2004.
- [12] K.-H. Kim and K. G. Shin. On accurate measurement of link quality in multi-hop wireless mesh networks. *ACM Mobicom*, 2006.
- [13] D. Kotz, C. Newport, and C. Elliott. The mistaken axioms of wireless-network research. Technical Report TR2003-467, Dartmouth College, Computer Science, July 2003.
- [14] V. Ramasubramanian, R. Chandra, and D. Mosse. Providing a bidirectional abstraction for unidirectional ad hoc networks. *IEEE INFOCOM*, 2002.
- [15] R. Dube, C. Rais, K. Wang, and S. Tripathi. Signal stability based adaptive routing (ssa) for ad hoc mobile networks. *IEEE Personal Communication*, Feb. 1997.
- [16] S. Keshav. A control-theoretic approach to flow control. *ACM SIGCOMM*, 1991.
- [17] K. Srinivasan, P. Dutta, A. Tavakoli, and P. Levis. Understanding the causes of packet delivery success and failure in dense wireless sensor networks. *ACM SenSys*, 2006.
- [18] F. Stann, J. Heidemann, R. Shroff, and M. Z. Murtaza. RBP: Robust broadcast propagation in wireless networks. *ACM SenSys*, 2006.
- [19] A. Woo, T. Tong, and D. Culler. Taming the underlying challenges of reliable multihop routing in sensor networks. *ACM SenSys*, 2003.
- [20] H. Zhang, A. Arora, and P. Sinha. Learn on the fly: Data-driven link estimation and routing in sensor network backbones. *IEEE INFOCOM*, 2006.
- [21] J. Zhao and R. Govindan. Understanding packet delivery performance in dense wireless sensor networks. *ACM SenSys*, pages 1–13, 2003.