On Link Asymmetry and One-way Estimation in Wireless Sensor Networks

Lifeng Sang and Anish Arora and Hongwei Zhang

Link asymmetry is one of the characteristic challenges that wireless sensor networks pose in the design of network protocols. We observe, based on testbed experiments, that a substantial percentage of links are asymmetric, many are even unidirectional. We also find that the reliability of synchronous acknowledgments is considerably higher than that of asynchronous messages. Thus the norm of estimating link quality bidirectionally via asynchronous beacons 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 reliable forward links, and we use dynamic retransmission thresholding for error control. Via experiments on testbeds of CC1000 radios and CC2420 radios (an IEEE 802.15.4-compliant radio), we quantify the performance improvement in ETF as compared with ETX. We also study the performance improvement of ETF over ETX when no special mechanism is employed to discover asymmetric links or to control retransmissions.

Categories and Subject Descriptors: C.2.2 [**Computer-Communication Networks:**]: Network Protocols

General Terms: Algorithm, Performance, Design, Implementation

Additional Key Words and Phrases: Sensor Network, Link Estimation, Routing

1. INTRODUCTION

Link asymmetry is a characteristic challenge when designing network protocols for wireless sensor networks. Previous experimental studies have shown that wireless links are nonisotropic, and can be asymmetric [Zhao and Govindan 2003; Srinivasan et al. 2006; Kim and Shin 2006; Ramasubramanian et al. 2002; Kotz et al. 2003]. To circumvent the challenge of link asymmetry, the conventional approach is to avoid using asymmetric links in routing, as is indicated by many existing routing metrics and protocols that have been widely used in the community [Couto et al. 2003; Woo et al. 2003; Dube et al. 1997; Gnawali et al. 2004; Cerpa et al. 2005; Keshav 1991; Zhang et al. 2006]. For instance, the routing metrics ETX (for *expected number of transmissions*) evaluates the quality of a link based on its quality in both directions [Couto et al. 2003]. Selecting links based on their bidirectional quality eliminates the possibility of using asymmetric links which are potentially critical network resources. For example, consider the case where a source node A has two routes to its destination node B: in one route the link reliability is 50% in

An extended abstract containing some preliminary results of this paper has appeared in ACM MobiHoc 2007.

Lifeng Sang and Anish Arora are with the Department of Computer Science and Engineering, The Ohio State University, Columbus, Ohio, 43210. Email: {sangl, anish}@cse.ohio-state.edu.
Hongwei Zhang is with the Department of Computer Science, Wayne State University, Detroit, Michigan 48202. Email: hzhang@cs.wayne.edu.

both directions while in the other route the link reliability is 100% towards B but 0% towards A. ETX favors 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 [Stann et al. 2006]. 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, where synchronous acknowledgements refers to link level acknowledgements (e.g. on the Mica2 motes) and hardware-level acknowledgements (e.g. on the TelosB motes), while the asynchronous acknowledgements generally refers to application level acknowledgements. 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. The underestimation of link quality can also degrade network performance, because potentially 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 toward 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. This again motivates the design of ETF.

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 present a simple mechanism in which nodes in 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 present 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 (with both *CC1000* radio-enabled motes and *802.15.4* radio-enabled motes), realistic event-driven sensor network traffic trace, and synthesized periodic traffic, we ran experiments to quantify the performance improvement in ETF-based routing 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 2, we briefly review the literature that is closely related to this paper. In Sections 3 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 5. Finally, we present our experimental results in Section 6 and make concluding remarks in Section 7.

2. RELATED WORK

Based on the observations that wireless links tend to be dynamic, unreliable, and asymmetric [Aguayo et al. 2004; Zhao and Govindan 2003; Kotz et al. 2003], various routing metrics have been proposed to estimate wireless link quality. For instance, end-to-end success rate (SR) [Gnawali et al. 2004], round trip time (RTT) [Draves et al. 2004], packet pair delay (PktPair) [Draves et al. 2004], expected number of transmissions (ETX) [Couto et al. 2003; Woo et al. 2003], required number of packets (RNP) [Cerpa et al. 2005], and the expected MAC latency per unit-distance to destination (ELD) [Zhang et al. 2006] 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 [Draves et al. 2004; Woo et al. 2003], 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 while 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 [Couto et al. 2003]. Therefore, ETX-based routing, like other routing protocols, tends to avoid asymmetric links in routing.

An earlier work [Zhou et al. 2006] reported that about 30% links were asymmetric in their deployed system. While they focused on modeling the radio irregularity, we attempt to exploit this fact to increase the routing performance. [Zuniga and Krishnamachari 2007] provided a comprehensive analysis of the root causes of link unreliability and asymmetry, while we focus on how to exploit link asymmetry. [Srinivasan et al. 2008] found that most intermediate links are bursty and they shift between poor and good delivery, which might be a reason for asymmetry.

[Marina and Das 2002] evaluated the benefit from utilizing unidirectional links for the AODV routing protocol. [Kim and Shin 2006] proposed a framework for reliably and efficiently estimating wireless link quality. They also briefly discuss the potential benefits of exploiting asymmetric links in routing for IEEE 802.11 mesh networks. Our work in this paper complements [Kim and Shin 2006] 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 [Ramasubramanian et al. 2002; Afek and Gafni 1994]. 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 for discovering asymmetric links.

3. EMPIRICAL STUDY OF LINK ASYMMETRY

Besides temporal and spatial dynamics of link properties, link asymmetry, resulting partly from variations in hardware and environmental noise, is widely observed in sensor networks but is typically not utilized because of its complexity. Instead of investigating the physical causes for asymmetry, we will study a simple characterization of link asymmetry and seek to model it in a way that we can exploit.

3.1 Testbed

We have conducted all the experiments of this paper in Kansei [Ertin et al. 2006], a high fidelity, indoor sensor network testbed. For the empirical study as described in this section, 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. 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 an earlier field sensor network [Arora et al. 2004] for which we have representative traffic traces.

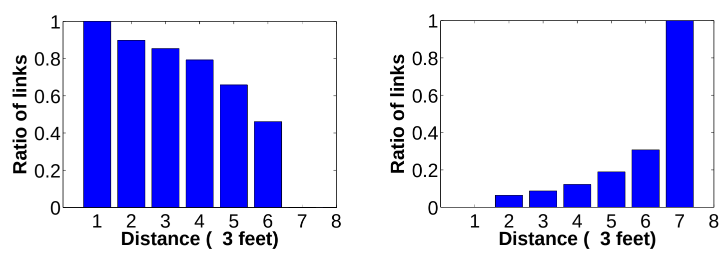
3.2 Link Asymmetry

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) was calculated over a relatively short period, we repeated such experiments 4 times to capture relatively long term variation: (1) **EXP1**: once in the morning; (2) **EXP2**: once in the afternoon; (3) **EXP3**: once in the evening; (4) **EXP4**: once after midnight. In each experiment, every node was 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 measured the PRR between each pair of nodes. We only considered *connected link*s that have non-zero PRR at least on one direction in the following measurements. In particular, we refer to a link with less than 10% PRR difference on both directions as a *symmetric link*, and the rest are *asymmetric*. And we refer to 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 refers only to links with excellent reliability on one direction and low reliability on the other.

PRR difference	< 10%	10-90%	> 90%	# of links
Power level 1 (-19dBm)	50%	43%	7%	500
Power level 3 (-14dBm)	65%	22%	13%	1038
Power level 9 (-5dBm)	88%	6%	6%	1155

Table I. Link asymmetry at different power levels.

Table I lists the average ratio of asymmetric links at different power level. In this experimental setting, the majority of the links are symmetric, especially at higher power levels. Specifically, half of the links are symmetric at the lowest power level 1, 65% at power level 3, and 88% at power level 9. In contrast, the ratio of asymmetric links decreases when the power level increases. This is consistent with the theory that the higher the transmission power, the more likely two nodes can communicate each other if they are physically close. Nonetheless, we do not observe a similar trend for the ratio of unidirectional links. Instead, both power level 1 and 9 have lower ratios of unidirectional links, which might be due to edge effects and link diversity at different power levels. For the Telosb motes, 35% links is symmetric, while around 15% is unidirectional at power level 2. At power level 3, we do not observe many asymmetric links because the distance between any two nodes is relatively small (constrained by the warehouse) so that most links at power level 3 are very good in terms of reliability.



(a) *symmetric links* (b) *unidirectional links*

Fig. 1. The ratio of *symmetric links* and *unidirectional links* as a function of physical link length at power level 3.

The next aspect that we explored 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). As shown in Figure 1(a), symmetric links are likely to occur between nodes nearby. The reason might be that the shorter the links, the lesser the propagation noise (due to different local environment conditions such as walls and hurdles) that 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 may not be a general case, but it does indicate that more links experience asymmetry with longer distance. This observation implies that unidirectional links (if properly exploited) are likely to result in more efficient routes since they tend to be longer.

Considering the scenario of convergecast, we then explored the occurrence of *unidirectional down links* that point to the base station, i.e., links with excellent reliability in the forward direction to the base station and bad quality in the reverse direction. The reason to explore such links is that upper layer routing protocols may conceivably benefit from these *unidirectional down links*. The observed topology of these links is shown in Figure 2. Let N_i denote node i (and hereafter). As

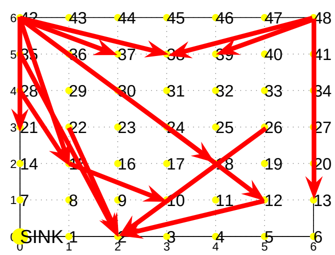


Fig. 2. Topology of the *unidirectional down links* at power level 3.

we mentioned earlier, a single link may exhibit different characteristics at different power level. Some of these *unidirectional down links* (e.g. $link(N_{42} \rightarrow N_{12})$) become disconnected at power level 1, while some become symmetric at power level 9. For example, $link(N_{42} \rightarrow N_{21})$ is unidirectional at power level 3, but $PRR(N_{42} \rightarrow N_{21}) = 99\%$ and $PRR(N_{21} \rightarrow N_{42}) = 98\%$ at power level 9. This is reasonable because reachability becomes much better at the higher power level. However, we find that $link(N_{42} \rightarrow N_{15})$ and $link(N_{42} \rightarrow N_{18})$ remain unidirectional at power level 9, which indicates that N_{42} may be an inherent *bad receiver* from some senders' view, even when it uses a higher transmission power. There are many reasons that make the receiver *bad*. For example, the device might be bad; the location where the device is placed might be bad so that the multi-path effects result in bad reception.

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 in relatively dense networks.

(3) Symmetric links tend to be short, while asymmetric links (especially unidirectional ones) are likely to be longer. This implies that effective exploitation of asymmetric links would lead to more efficient routing.

Note that we use the same transmission power for both the senders and receivers in each experiment. In reality, if transmission power is allowed to be different at senders and receivers, the above observations may vary, but the degree of link asymmetry is likely to become even higher.

4. RELIABILITY OF SYNCHRONOUS ACKS

[Cerpa et al. 2005] have studied temporal properties of low power wireless links and indicated that for improved ACK reliability it is better to send acknowledgments soon after packet reception. Packets in their experiments were sent in a round robin fashion with at least one second separation. Here we go one step further. We investigate the performance of synchronous acknowledgment, hypothesizing that link quality for synchronous ACK packets is significantly better than that for beacons, especially in the presence of interference. This is because: 1) nodes in the neighborhood are in the process of "backing off" so the channel is likely to be clear to successfully send 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 or without interference. We consider explicit replies as asynchronous ACK because they are sent preceded by a random back-off in the link layer. To reduce temporal variation, we collected data for synchronous ACK (using the default B-MAC in TinyOS) and asynchronous ACK in the same experiments, as follows:

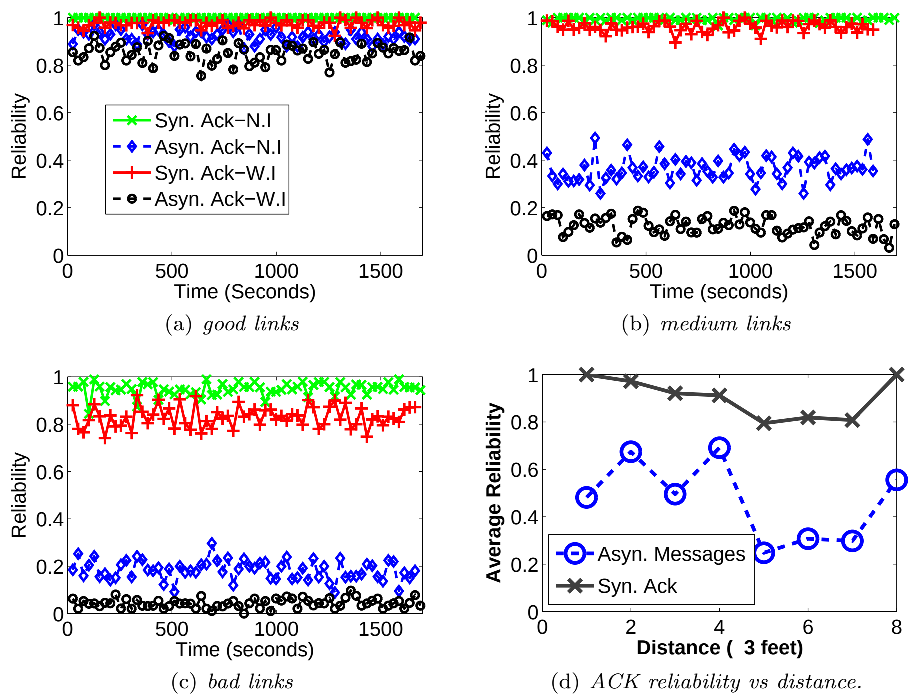
(1) *Synchronous/asynchronous ACK without interference.* One node sends 20000 unicast packets (one per 128 ms) to its destination with link layer ACK mechanism enabled. When the application at the destination receives a packet, it immediately replies a data packet as an explicit ACK. The reliability 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. The reliability of asynchronous ACK is calculated as the 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.* The experiments above is are repeated, but now with interference. We let all other nodes (except the sender) each periodically send a packet after a random delay (1 per second on average). These interfering packets may either compete for the channel or possibly collide with our target packets. [1]

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 above 90% on average, and some are close to 100%. Yet the reliability of asynchronous ACKs becomes very low (e.g., below 20% in the presence of interference).

[1] Our previous study [Sang et al. 2007] applied a less realistic interference scenario, yet we find similar conclusions with a more representative interference model.

Another surprising observation is that even bad links have medium reverse reliability for synchronous ACKs in the low quality direction, as shown in 3(c), although asynchronous ACK reliability is quite low (e.g. close to 0 in the case of 'with interference'). 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 (1) the reliability of synchronous ACK is much better than that of our asynchronous ACK (with short backoff); and (2) experiments in [Cerpa et al. 2005] showed that the reliability of asynchronous ACK are better to be sent sooner, it is clear that the reliability of synchronous ACK is generally better than that of periodic beacons. This means that asynchronous message based estimation significantly underestimates link quality for synchronous ACK.



(a) *good links* (b) *medium links*

(c) *bad links* (d) *ACK reliability vs distance.*

Fig. 3. Experiments for ACK reliability. **N.I**: No interference; **W.I**: With interference. First three figures have the same legends.

Next, we study the relationship between ACK reliability and channel distance in order to understand whether the improvement of synchronous ACK is distance sensitive. In this particular study, one node sends unicast packets to all the other nodes in the testbed without interference in a round robin fashion. We collected data with sender N_0 and N_1, and computed the average ACK reliability illustrated in Figure 3(d). It is obvious that the improvement of synchronous ACKs is not limited to short links. Instead, long links enjoy the improvement too. These observations indicate 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 link asymmetry and quality of synchronous acknowledgment observed in our empirical study lead us to investigate how to exploit asymmetric links in order to improve network functions such as multi-hop convergecast routing, via one-way link estimation. In this section, we propose ETF (for the *expected number of transmissions over forward links*), 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 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

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

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

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

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 4, for instance, N_{42} selects N_{30} as its parent if ETX is the metric, because $ETX_{N_{42}\to N_{30}\to N_{15}} = 5.69$ and $ETX_{42\to 15} = 33.3$. It takes 5.4 transmissions on average to deliver a data packet from N_{42} to N_{15} through N_{30}. However, only 1 transmission is actually needed to deliver a data packet directly from N_{42} to N_{15}. In this particular case, N_{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* [Stann et al. 2006] 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.

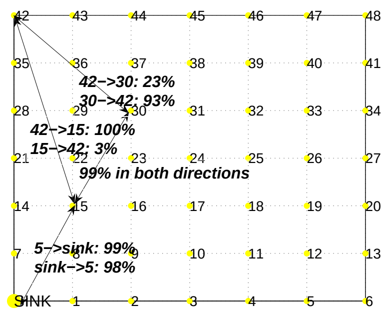


Fig. 4. An example of route selection. $link(N_{42}, N_{15})$ is a *unidirectional link* in all the experiments. Consider a scenario where N_{42} is the source, and N_{15} and N_{30} are potential forwarders to the SINK. ETX selects path $N_{42} \rightarrow N_{30} \rightarrow N_{15} \rightarrow SINK$ while ETF selects path $N_{42} \rightarrow N_{15} \rightarrow SINK$.

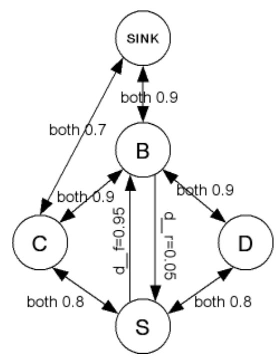


Fig. 5. An example of discovering asymmetry.

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 5, 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 quality. Moreover, some links may be completely unidirectional, as we find in our testbed, then the source nodes would have no way of knowing the existence of such a unidirectional link. This motivates us to design a 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 5, 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 constraints, however, it is impossible for C to keep information about all neighboring links, 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 5, 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: i) 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 ii) 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 its 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, 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 not need be reported every time a node detects asymmetry. A node needs to sharing information about asymmetric links only once every estimation period.

Listing 1. Asymmetry Discovery

```
AsymmetryDiscovery(ngbr1, ngbr2, r) {
 /*r is the reliability of Link(ngbr1->ngbr2) */
 if r>85%
  if exists(ngbr2,ngbr1)
     remove entry(ngbr2,ngbr1);
  elseif similar fact has been heard 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;
 }
}
```

Next, we discuss the overhead of control packets. For each asymmetric link that needs to be discovered indirectly, the number of extra transmissions has an upper bound of Δ, the number of detectors in every neighborhood. In fact, it is also upper bound 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 other's 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, and not all neighbors are eligible for delivering discovery information due to our reliability-based selection, so the number of transmissions would be fewer than $\min(\Delta, 5)$. Moreover, 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 much lower compared with the overhead introduced by 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.

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 a 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. A 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,

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

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

where MTC is the pre-defined maximum transmission count for the whole system, and $\theta(reliability)$ 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 in both directions might be better than a unidirectional link with 98% forward reliability and so ETF may not be a wise choice in this particular case. As we explain below, ETF is a better indicator of link quality than ETX on more than 80% of the links. This is because 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 $\sum_{i=1}^{k-1}(i \times (1-u_2 \times v_2)^{i-1} \times (u_2 \times v_2)) + k \times (1-u_2 \times v_2)^{k-1} = \frac{1-(1-u_2 \times v_2)^{k-1}}{u_2 \times v_2} + k \times (1-u_2 \times v_2)^{k-1}$ 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,

$$\varphi(u_2, v_2, k) < \varphi(u_1, v_1, MTC) \tag{5}$$

where φ is a function,

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

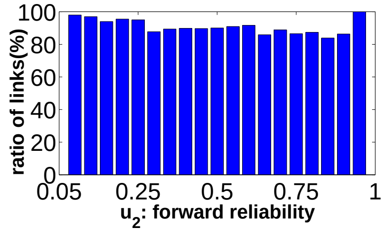
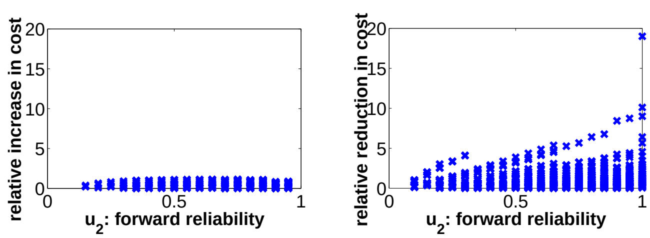


Fig. 6. The percentage of cases where ETF is better.

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 $MTC \to \infty$, 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 particularly 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 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 6. 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. In particular, 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 the reverse link reliability is. A deeper question is "how much is the improvement/degrade of using ETF?". 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 7(a). We see that all the extra relative cost is less than 1.2, and most



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

Fig. 7. Impact of using ETF in different cases.

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 7(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. Since ETF does not perform too badly in the few worse cases while it does perform significantly better in many other cases, the overall improvement that ETF can achieve is substantial. We will demonstrate the improvement via testbed experiments in the next section.

6. EXPERIMENTAL EVALUATION

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

6.1 Experiment Design

6.1.1 *Protocols studied.* We have implemented a distance-vector routing framework in TinyOS so as to evaluate different aspects of routing (e.g., routing metric and neighbor table management mechanisms) in a common setting. To study the impact of exploiting asymmetric links on routing as well as to evaluate the components of ETF, we instantiate the routing framework in different manners to get the following routing protocols:

—*ETX*: convergecast using routing metric ETX.

—*ETF*: convergecast using routing metric ETF.

—*ETF-NU*: same as ETF, except that the explicit asymmetric-link discovery is disabled. This protocol lets us study whether ETF will work well without detecting links with asymmetry (e.g., unidirectional links).

—*ETF-ND*: same as ETF, except that the the dynamic retransmission threshold control is disabled. This protocol lets us study the impact of our error control method.

(Note that, for simplicity, we name the protocols simply 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 used the 7×7 grid testbed discussed in Section 3.1 for the performance evaluation, setting the mote at the left-bottom corner of the grid be the base station. To validate ETF on different platforms, we used a [TMote] testbed in addition to XSM testbed. Each TMote consists of an $8MHz$ MSP430 microcontroller, and a Chipcon *CC2420* radio (*IEEE 802.15.4* compatible). It has $48K$ bytes of flash and $10K$ bytes of RAM. Our implementations were in TinyOS 1.x.

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. Figure 8 shows the boxplots of packet reception rate (PRR) as a

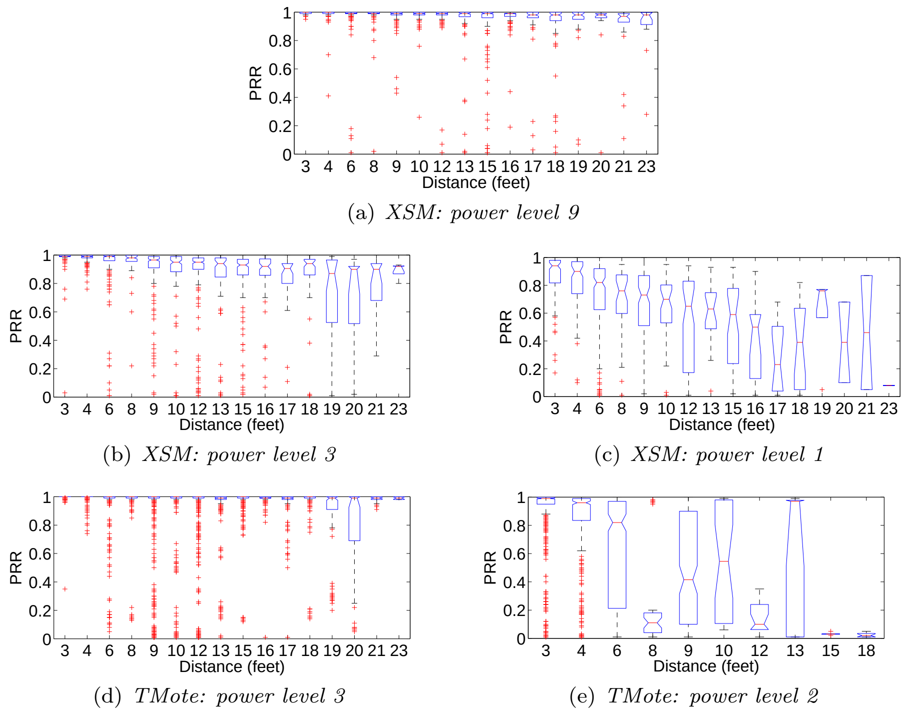


Fig. 8. Testbed density. XSM with power level 9, 3 and 1, and TMote with power level 3 and 2.

function of the distances between senders and receivers, when the power level is 9, 3 and 1 (out of a range of [1, 255]) for XSM, and 3 and 2 for TMote respectively. Power level 3 at TMote is about -25dBm, and power level 2 is about -30dBm. Note that the power level for XSM and TMote is not comparable because the transceiver is quite different.

For the XSM, the network is sparse at power level 1, most nodes can only reliably 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

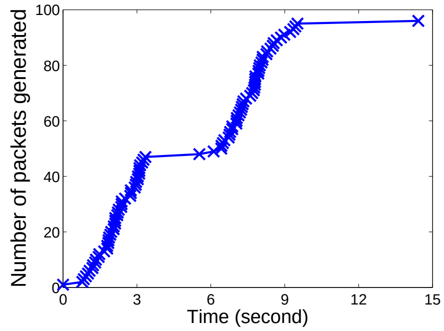


Fig. 9. Traffic trace of Lites.

are reliable while some are not. Thus, power level 3 gives us a typical multi-hop network (about 4-5 hops).

For the TMote, the density is quite high at power level 3, while it is fairly sparse at power level 2. There is no intermediate power level between 2 and 3 we can adopt to constitute a relatively dense network in this setting. Therefore, only power levels 2 and 3 are used in the TMote evaluation.

We conducted experiments at different power levels to study the impact of network density on the performance of routing protocols.

6.1.4 *Traffic Trace.* To study performance for representative sensor network traffics, we used both bursty-event traffic and periodic traffic.

The event traffic pattern was collected in a real-world intrusion detection sensor network [Arora et al. 2004]. In the traffic trace, each node of a 7×7 grid, except for the base station, generated two packets denoting the start and the end of its local detection of an intrusion event. A total of 96 packets were generated each time the event occurs. The cumulative distribution of number of packets generated during the event is shown in Figure 9 (interested readers can find the details in [Arora et al. 2004]).

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

The periodic traffic pattern was selectd to be representation of data collection scenarios. More specially, each node generates a packet every 30 seconds on average in the periodic traffic trace. The maximum number of total entries for storing potential asymmetric links is set to be 32, which corresponds to about 200 bytes.

6.1.5 *Maximum Transmission Count (MTC).* To improve packet delivery reliability, the typical 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, named the maximum transmission count (MTC), number of times. 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 to 8 (a typical value that is also used in IEEE 802.15.4) unless explicitly mentioned otherwise.

6.2 Experimental Results On XSM

Since our XSM platform has more diversity in network density, we begin with our evaluation results for the XSM platform. Experimental results on the TMote platform are summarized in Section 6.3.

The general observations are quite similar for experiments based on event traffic and those based on periodic traffic, we focus on presenting results for experiments with event-traffic in this section. We briefly discuss summary data for periodic traffic based experiments at the end of this section. For event traffic based experiments, we first present the results for networks of different densities, and then present the impact of MTC on the performance of different routing protocols.

	Reliability	TXPP	Delay	Dup.	Hops
ETX	84.63%	1.45	0.097	14	2.53
ETF-NU	92.02%	1.52	0.093	149	1.81
ETF-ND	92.21%	2.10	0.105	124	2.02
ETF	92.60%	0.99	0.072	40	1.83

Table II. Performance of ETF and ETX at power level 9 (on XSM).

6.2.1 *Dense Network: Transmission Power Level=9.* Table II shows the summary performance data for ETX and ETF respectively. We see that, even though ETX seems to perform well at power level 9, it is still not as good as ETF and its variants. Compared with ETX, ETF improves the reliability from 84.63% to 92.60%, and the TXPP decreases by one third. Note that if the value of TXPP for ETF is 0.99, which seems to be an anomaly because TXPP value should never be smaller than 1 by definition. We carefully checked the logged data, and found that one node (ID 84) did not record the transmissions for the last two packets. Such an error rarely happened on the testbed. The real TXPP for ETF should be around 1 in this experiment. Because of the better routes that it establishes, ETF has lower latency than ETX. Note that ETF is observed to have more duplicates in these experiments. The reason could be that fewer packets in ETX go through the whole path to the base station, and those lost packets do not experience any duplicates later. The reliability of ETF and its variants is comparable. In particular, we see that ETF-ND has a larger TXPP, probably due to ACK loss since no dynamic thresholding is applied.

6.2.2 *Regular Network: Transmission 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 III.

	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-ND	85.46%	2.57	0.080	234	2.17
ETF	88.39%	2.48	0.096	197	2.02

Table III. A comparison of reliability, TXPP, latency, duplicates and average hops at power level 3 (on XSM).

The reliability exceeds that of ETX in this scenario as well. 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 and ETF-ND incur comparable TXPP (2.83 and 2.57 respectively), 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, and the dynamic retransmission threshold is disabled. (Note that links with moderate asymmetry can still be detected and used in ETF-NU.) The latency in ETF-NU, ETF-ND and ETF is slightly better than that in ETX, but would be much better if lost packets were taken into account. Again, ETF has many fewer duplicates than ETX. It is not surprising that ETF-ND has more duplicates than ETF, since ETF-ND is likely to experience more acknowledgment loss along the low-quality direction. As to average hop count, ETX and ETF are comparable, and ETF-NU and ETF-ND have a little larger hop count. 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 10. As we can see, initially, the reliability of ETF is around 72%. With the help of

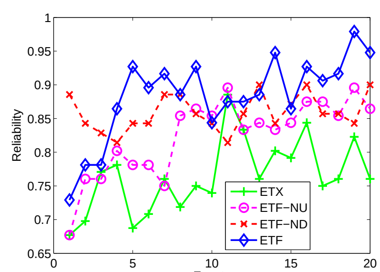


Fig. 10. Comparison of event reliability at power level 3 (on XSM).

asymmetric links, it becomes better. ETF-NU and ETF-ND also outperform ETX in this measurement. Note that the reliability for the first few events is relatively low, largely because the routing structure has not been stabilized yet.

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 IV 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

	ETX	ETF-NU	ETF-ND	ETF
link reliability	70.4%	72.1%	76.96%	78.3%
C.O.V. of reliability	0.142	0.109	0.04	0.107
link length (feet)	9.05	9.22	9.26	9.90

Table IV. 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.

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. Comparing ETF with ETF-NU, we see that the explicit asymmetric-link discovery mechanism used in ETF also helps in identifying those links of extreme asymmetry (e.g., unidirectional links) and thus further improves the routing performance. Comparing ETF with ETF-ND, we see that dynamic retransmission thresholding seems to result in more reliable and efficient routes by minimizing extra overhead produced by ACK loss. Regarding the overhead introduced by link asymmetry discovery, we noticed that only a small portion of the nodes sent control packets each round. Therefore the total number of control packets is much less than 6.25% of the beacon packets (16 beacons a round in our experiments), which can be reasonably ignored in an application with a large volume of traffic.

6.2.3 *Sparse Network: Transmission Power Level=1.* The routing performance at power level 1 is shown in Table V. We see that the reliability of ETF has been

	Reliability	TXPP	Delay	Dup.	Hops
ETX	52.75%	15.36	0.333	327	7.20
ETF-NU	68.07%	4.05	0.134	500	4.19
ETF-ND	71.40%	6.18	0.161	529	3.26
ETF	73.15%	3.10	0.101	705	2.67

Table V. Performance of ETF and ETX at power level 1 (on XSM).

improved from 50% to 70% compared to ETX, and the TXPP has been decreased from 15 to 3. The improvement in latency is also substantial, showing the benefits of exploring asymmetric links even in sparse networks. Note that since a network at power level 1 represents a typical sparse network, the significant improvement of ETF over ETX showcases the promise of using ETF. Again, ETF-ND exhibits more TXPP compared to ETF due to ACK loss without error control, and ETF-NU does not perform as well as ETF since no link discovery is performed.

Summary. From the discussions above and the data shown in Tables II, III, and V, 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 in a variety of networks of different density. Note that the current results on TXPP do not include the extra transmissions incurred by asymmetry discovery in ETF. Although this number is much less than

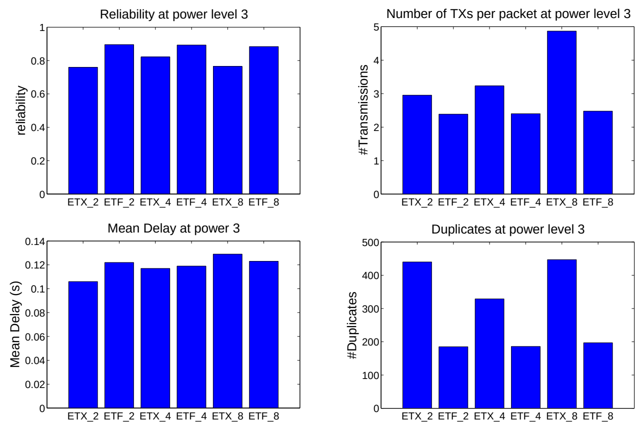


Fig. 11. The impact of MTC at power level 3 (on XSM). The number following ETF_/ETX_ is the value of MTC.

that of mass data transmissions, TXPP in ETF would be slightly larger if the number of control packets were counted. On the other hand, the overhead can be eliminated by embedding the discovery information into the beacon packets. We also find that ETF outperforms ETX even if no special mechanism is applied. On the other hand, comparing the performance of ETF-NU, ETF-ND and ETF, we find that link asymmetry discovery helps to establish more efficient routes, while dynamic error control reduces unnecessary retransmissions which further improve the reliability for the whole network.

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 11 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 of 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 a priori determine an MTC value that suits for 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 station are received in ETF routing and they contribute to the increase of average latency.

6.2.5 *Periodic Traffic.* Table VI summaries the experimental results using ETF and ETX at power level 3 with periodic traffic. 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. Note that ETF has more duplicates in this case, which could be because far more packets have traversed along the path to the base station, thereby yielding 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 VI. Performance of ETF and ETX with periodic traffic at power level 3 (on XSM).

6.3 Experimental Results On TMote

The experimental results for the event traffic on the TMote platform testbed are illustrated in Tables VII and VIII, and the results for the periodic traffic are illustrated in Tables X and XI. Although ETF generally outperforms ETX with different traffic trace at different power level, the improvement at power level 3 is limited. This is because the network is so dense that almost all the links are symmetric. Nonetheless, the improvement on the sparse network (where there are more asymmetric links) is still substantial. For example, the average reliability is almost doubled with event traffic at power level 2 using ETF, while its TXPP decreases more than half from 29.36 using ETX to 12.45.

We list the overall quality of all the links used in the routing for the event traffic case in Table IX. Again, we see that ETF tends to choose links with better reliability and stability, yet most likely they are longer. Note that the average link length for ETF is not longer than that for ETX at power level 2. Investigation of the trace log shows that more than 95% of the links are within 2 physical hops due to lower transmission power, so that the overall link length is on average lower although ETF does apply a few longer links in the routing.

	Reliability	TXPP	Delay	Dup.	Hops
ETX	94.31%	1.35	0.054	2	2.43
ETF	97.93%	1.23	0.052	1	2.41

Table VII. Performance of ETF and ETX with event traffic at power level 3 (on TMotes).

	Reliability	TXPP	Delay	Dup.	Hops
ETX	26.53%	29.36	0.341	302	8.32
ETF	49.54%	12.45	0.259	1300	7.62

Table VIII. Performance of ETF and ETX with event traffic at power level 2 (on TMotes).

	ETX, P.L.=3	ETF, P.L.=3	ETX, P.L.=2	ETF, P.L.=2
link reliability	88.8%	89.4%	38.2%	57.9%
C.O.V. of reliability	0.075	0.052	0.222	0.134
link length (feet)	10.8	12.1	5.59	5.58

Table IX. Quality of links used in routing in the event traffic case. The definition of link reliability, C.O.V. and link length is the same as in Table IV. P.L. abbreviates power level.

	Reliability	TXPP	Delay	Dup.	Hops
ETX	99.03%	1.70	0.075	1	2.77
ETF	99.86%	1.08	0.048	1	1.68

Table X. Performance of ETF and ETX with periodic traffic at power level 3 (on TMotes).

	Reliability	TXPP	Delay	Dup.	Hops
ETX	41.94%	22.45	0.327	277	7.93
ETF	72.64%	7.16	0.213	575	6.30

Table XI. Performance of ETF and ETX with periodic traffic at power level 2 (on TMotes).

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 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 large amount 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 (on different platforms), 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 focused mainly 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. Note that we have so far focused on convergecast traffic, and whether our scheme is beneficial for other routing problems in general needs further investigation. 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. This material is based upon work supported by the National Science Foundation under Grants No. 0520222 and 0341703.

REFERENCES

Afek, Y. and Gafni, E. 1994. Distributed algorithms for unidirectional networks. *SIAM Journal on Computing 23,* 6, 1152–1178.

Aguayo, D., Bicket, J., Biswas, S., Judd, G., and Morris, R. 2004. Link-level measurements from an 802.11b mesh network. *ACM SIGCOMM*.

Arora, A., Dutta, P., Bapat, S., Kulathumani, V., Zhang, H., Naik, V., Mittal, V., Cao, H., Demirbas, M., Gouda, M., Choi, Y.-R., Herman, T., Kulkarni, S. S., Arumugam, U., Nesterenko, M., Vora, A., and Miyashita, M. 2004. A line in the sand: A wireless sensor network for target detection, classification, and tracking. *Computer Networks (Elsevier) 46,* 5.

Cerpa, A., Wong, J. L., Potkonjak, M., and Estrin, D. 2005. Temporal properties of low power wireless links: Modeling and implications on multi-hop routing. *CENS Technical Report 0044*.

Couto, D. S. J. D., Aguayo, D., Bicket, J., and Morris, R. 2003. A highthroughput path metric for multihop wireless routing. *ACM MobiCom*.

Draves, R., Padhye, J., and Zill, B. 2004. Comparison of routing metrics for static multi-hop wireless networks. *ACM SIGCOMM*.

Dube, R., Rais, C., Wang, K., and Tripathi, S. 1997. Signal stability based adaptive routing (ssa) for ad hoc mobile networks. *IEEE Personal Communication*.

Ertin, E., Arora, A., Ramnath, R., and Nesterenko, M. 2006. Kansei: A testbed for sensing at scale. *Proceedings of the 5th Symposium on Information Processing in Sensor Networks (IPSN/SPOTS Track)*, 399–406.

Gnawali, O., Yarvis, M., Heidemann, J., and Govindan, R. 2004. Interaction of retransmission, blacklisting, and routing metrics for reliability in sensor network routing. *IEEE SECON*.

Keshav, S. 1991. A control-theorectic approach to flow control. *ACM SIGCOMMM*.

Kim, K.-H. and Shin, K. G. 2006. On accurate measurement of link quality in mulit-hop wireless mesh networks. *ACM Mobicom*.

Kotz, D., Newport, C., and Elliott, C. 2003. The mistaken axioms of wireless-network research. Tech. Rep. TR2003-467, Dartmouth College, Computer Science. July.

Marina, M. K. and Das, S. R. 2002. Routing performance in the presence of unidirectional links in multihop wireless networks. *ACM Mobihoc*.

Ramasubramanian, V., Chandra, R., and Mosse, D. 2002. Providing a bidirectional abstraction for unidirectional ad hoc networks. *IEEE INFOCOM*.

Sang, L., Arora, A., and Zhang, H. 2007. On exploiting asymmetric wireless links via one-way estimation. *8th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, 11–21.

Srinivasan, K., Dutta, P., Tavakoli, A., and Levis, P. 2006. Understanding the causes of packet delivery success and failure in dense wireless sensor networks. *ACM SenSys*.

Srinivasan, K., Kazandjieva, M. A., Agarwal, S., and Levis, P. 2008. The β-factor: Measuring wireless link burstiness. *ACM Sensys*.

Stann, F., Heidemann, J., Shroff, R., and Murtaza, M. Z. 2006. RBP: Robust broadcast propagation in wireless networks. *ACM SenSys*.

TMote. http://www.moteiv.com/.

Woo, A., Tong, T., and Culler, D. 2003. Taming the underlying challenges of reliable multihop routing in sensor networks. *ACM SenSys*.

Zhang, H., Arora, A., and Sinha, P. 2006. Learn on the fly: Data-driven link estimation and routing in sensor network backbones. *IEEE INFOCOM*.

Zhao, J. and Govindan, R. 2003. Understanding packet delivery performance in dense wireless sensor networks. *ACM SenSys*, 1–13.

Zhou, G., He, T., Krishnamurthy, S., and Stankovic, J. A. 2006. Models and solutions for radio irregularity in wireless sensor networks. *ACM Transactions on Sensor Networks*.

Zuniga, M. and Krishnamachari, B. 2007. An analysis of unreliability and asymmetry in low-power wireless links. *ACM Transactions on Sensor Networks 3,* 2 (June).