QoF: Towards Comprehensive Path Quality Measurement in Wireless Sensor Networks

Jiliang Wang, *Member, IEEE*, Yunhao Liu, *Senior Member, IEEE*,
Yuan He, *Member, IEEE*, Wei Dong, *Member, IEEE*, and Mo Li, *Member, IEEE*

Abstract—Due to its large scale and constrained communication radius, a wireless sensor network mostly relies on multi-hop transmissions to deliver a data packet along a sequence of nodes. It is of essential importance to measure the forwarding quality of multi-hop paths and such information shall be utilized in designing efficient routing strategies. Existing metrics like ETX, ETF mainly focus on quantifying the link performance in between the nodes while overlooking the forwarding capabilities inside the sensor nodes. The experience on manipulating GreenOrbs, a large-scale sensor network with 330 nodes, reveals that the quality of forwarding inside each sensor node is at the least an equally important factor that contributes to the path quality in data delivery. In this paper we propose QoF, Quality of Forwarding, a new metric which explores the performance in the gray zone inside a node left unattended in previous studies. By combining the QoF measurements within a node and over a link, we are able to comprehensively measure the intact path quality in designing efficient multi-hop routing protocols. We implement QoF and build a modified Collection Tree Protocol (CTP). We evaluate the data collection performance in a testbed consisting of 50 TelosB nodes, and compare it with the original CTP protocol. The experimental results show that our approach takes both transmission cost and forwarding reliability into consideration, thus achieving a high throughput for data collection.

Index Terms—Path quality, node quality, quality of forwarding

◆

1 INTRODUCTION

A wireless sensor network (WSN) is typically designed to span in a large field for data collection. Data delivery is usually achieved with multi-hop transmission along a sequence of nodes. Many multi-hop routing protocols have been proposed for WSN data collection and they usually incorporate special path estimation metrics to select "good" paths for delivering data packets.

There have been many estimation metrics proposed to measure the forwarding quality of a multi-hop path, such as ETX [1], ETF [2], PRR, ETOP [3] etc. Existing metrics mainly focus on estimating the packet delivery quality on links in between the nodes. The quality of forwarding capacity along a path is estimated by the aggregate of the forwarding qualities of all the links on the path. Those link-based metrics while reflect the link performance of the path, however, overlook the forwarding capabilities inside the sensor nodes, thus resulting in an incomplete measurement of the path quality. Using the incomplete path indicators will lead to suboptimal routing decisions and degraded routing performance.

Such an effect has been revealed in our experience in manipulating GreenOrbs [4], a large-scale sensor network with 330 nodes. In current routing implementation in GreenOrbs, we use a modified Collection Tree Protocol (CTP) that relies on path ETX estimation for routing selection. During the field test of the system, we observe a portion of packets drops on some nodes. They are due to a variety of causes, such as forwarding queue overflow under high traffic pressure, software bugs in the CTP implementation, and etc. Those nodes, however, still respond with ACKs at the radio hardware. The bad fact is that with current path indicators the inability of packet forwarding within the individual nodes cannot be shared among the network, yet there is not a metric to quantify the packet forwarding quality at each node. As a result, the path estimation not always truly reflects the path quality and the data delivery performance is severely degraded.

The packet drops on the problematic nodes introduce intrinsic unreliability in data delivery. As a matter of fact, even a single link itself can hardly achieve full reliability. ETX over a link measures the expected number of transmissions for successfully delivering a packet, but transmitting the packet at the expected number of times does not guarantee it will be successfully received at the receiver end. In practical systems, a maximum number of retransmissions are usually set on a link to prevent sending a packet on a "bad" link infinitely, that exhausts the finite communication resources. The packet will be eventually dropped by the sender after a maximum number of transmission retries. The network is thus rendered unreliable due to both node unreliability and link unreliability. ETX of a path is estimated as the summation of ETX values over all links constituting the path. Using path-ETX for path selection minimizes the transmission cost and achieves a high throughput. However, ETX presumes that end-to-end

---

- *J. Wang, Y. Liu and Y. He are with the School of Software and TNLIST, Tsinghua University, and with the Hong Kong University of Science and Technology. E-mail: {jiliang, yunhao, he}@greenorbs.com.*
- *W. Dong is with the College of Computer Science, Zhejiang University. E-mail: dongw@zju.edu.cn.*
- *M. Li is with the School of Computer Engineering, Nanyang Technological University, Nanyang Avenue, Singapore. E-mail: limo@ntu.edu.sg.*

Manuscript received 9 Nov. 2012; revised 12 Mar. 2013; accepted 22 Mar. 2013; date of publication 2 Apr. 2013; date of current version 21 Feb. 2014.
Recommended for acceptance by K. Li.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TPDS.2013.98

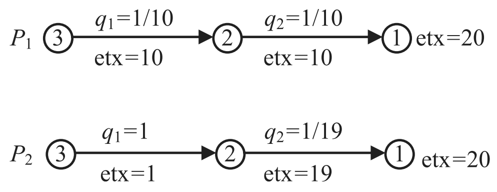


Fig. 1. Two paths with the same ETX but different PRRs.

delivery is reliable which, however, is not always the truth as we see from the above.

For data delivery within the network of inherent unreliability, a metric that better measures the data productivity is the amount of successful data delivery to the destination, i.e., data yield [5]. Data yield over the actual number of data transmissions, measures both transmission cost as well as achieved throughput. Existing path-ETX does not capture such a parameter. Consider a simplified example depicted in Fig. 1. There are two paths, both of which have path-ETX of 20. Suppose the link layer transmission retries is set to 1. In path 1, the probability that a packet passes the first link is $\frac{1}{10} + (1 - \frac{1}{10}) \cdot \frac{1}{10} = 0.19$. Similarly, the probability that a packet passes the second link is 0.19. Therefore, the probability that the packet passes the path (path reliability) is $0.19 \times 0.19 = 0.0361$, i.e., 361 packets will be received if the source sends 10,000 packets. In path 2, however, the path reliability is $1 \times (\frac{1}{19} + (1 - \frac{1}{19}) \cdot \frac{1}{19}) = 0.1025$, i.e., 1,025 packets will be received if the source sends 10,000 packets. This implies that ETX fails to capture the path reliability [6]. The situation will be similar if we further consider a higher number of transmission retries as well as node unreliability. Routing based on path-ETX does not give the optimal delivery path in terms of the data yield per transmission.

In this work, we comprehensively investigate the unreliability in both links and nodes. We present QoF, a new metric which estimates the chances for a packet to pass both a link and a node. The link-QoF not only considers the transmission cost at the sender but also considers the data delivery ratio at the receiver. The node-QoF estimates the quality of forwarding within a node, and it plays an important role in differentiating the problematic nodes. Based on link-QoF/node-QoF, we aggregate the QoF measure over a path (path-QoF). The path-QoF metric estimates the intact path forwarding quality and it considers both transmission cost and end-to-end data delivery ratio. The QoF metric measures the data yield over the actual number of data transmissions. Hence using such a metric can greatly improve the data yield while having a low transmission overhead.

The contributions of this paper are summarized as follows.

First, we reveal the limitations of existing link-based indicators like ETX and ETF in estimating the intact path forwarding quality. In a practical system, routing selection based on ETX may lead to severely degraded data yield.

Second, we propose a new metric QoF to measure the path quality. QoF can be used to estimate the forwarding quality over a link or within a node. Using QoF, we are able to characterize both the transmission cost and the data delivery ratio along a forwarding path.

Third, we implement QoF based on TinyOS 2.1 [7] and incorporate it into CTP [8]. We evaluate the QoF based routing performance in a testbed consisting of 50 TelosB nodes [9]. The results show that using the QoF metric improves the data yield while reducing the per-successful delivery cost.

The remainder of this paper is organized as follows. Section 2 presents the related work. In Section 3, we introduce our basic observations in a real working system that motivate this study. In Section 4, we present the detailed design aspects. Section 5 describes the implementation details. Section 6 shows the evaluation results. We conclude this work in Section 7.

2 RELATED WORK

The quality of packet forwarding is a fundamental factor in sensor networks, which has been studied in a number of works. Such a factor can be estimated on different dimensions over the communication in between nodes, for example, received signal strength (RSS), transmission delay, packet reception ratio (PRR), etc. Those parameters are measured at different layers across the communication stacks.

At the physical layer, RSSI and LQI are two most widely used parameters that describe the communicational quality between nodes. Both RSSI and LQI reflect the physical quality of the wireless channel in between the nodes, though as suggested in [10], the two parameters are not adequate to represent the quality of packet forwarding over the link.

At the link layer, many other metrics have been proposed. ETX measures the expected number of transmissions for successfully delivering a packet over the link. Specifically, if we denote d_f as the probability that a packet is successfully received and d_r as the reverse probability that the link ACK can be successfully received. The ETX value over a link is then calculated as $\frac{1}{d_f \times d_r}$. Another metric ETF [2] is designed for links of high asymmetry. ETF suggests that though the reverse link quality is low, the ACK still has a high probability being received by the sender. ETOP [3] considers the impact of link positions to the path quality. It presents an algorithm to find the path with minimal ETOP value. Designed for wireless mesh networks, ETOP does not consider the unreliable forwarding quality of sensor nodes. In addition, it requires that the source determines the entire forwarding path to the destination, which is not suitable for the WSNs with unreliable and time-varying links. There are some other link layer metrics, such as expected transmission time, competence [11], L-NT [12], ENT [13], end-to-end success rate [14], required number of packets [15], EDR [16], etc. Among the aforementioned link estimation metrics, ETX is the most widely used one. ETX has worked as the de facto link quality indicator and has been used for a variety of wireless protocols [17], [18]. Currently, ETX estimation has been integrated into CTP [8], for reliable and efficient data collection in WSNs.

Besides those estimators at separate layers, the 4-bit link estimator uses 4 bits to combine information at PHY layer, link layer and network layer. After using PHY signal strength to do the first step link filtering, it also considers link layer packet reception quality and network layer congestion information.

There are different metrics proposed to characterize the forwarding quality of a path. Minimal hop count can

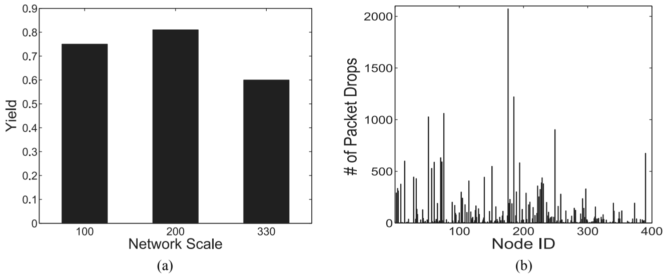


Fig. 2. Observation in a 330-node outdoor testbed. (a) Network yield at different scales. (b) Packet loss on different nodes.

be used to select a path. Summing up all link ETX values along the path gives the path-ETX. As we found in our testbed, however, the path-ETX is not always adequate in practical systems because it only gives an incomplete description of the path quality. Other metrics [13], [19], [12] overlook the node forwarding quality as well. Simply aggregating the estimated link qualities does not give comprehensive description of the intact path quality. There are also power aware geographic forwarding techniques [20] with location information.

There are many works focusing on node's capability and stability. For example, Schmid et al. [21] investigate the relation between the timer stability and the power and other environment factors such as temperature, humidity, the cut of the oscillator, etc. For an event driven embedded OS, such as TinyOS, a large portion of events are driven by timers. The timer instability largely renders the OS instable.

Queue length is another important factor that affects the forwarding quality of individual nodes. Backpressure routing uses local queue length information to select a node with the largest positive differential backlog. It is designed to be throughput optimal. BCP [22] is a recent realization of backpressure routing protocol for sensor networks. There are also a lot of other works for congestion control such as [23], [24]. While those approaches are effective in handling network congestions, they do not consider other causes affecting the node forwarding quality. As we found in GreenOrbs, there are various causes affecting the node forwarding quality. Our work not only considers packet loss due to congestion but also considers packet loss due other factors inside the node.

3 MOTIVATION AND BACKGROUND

This work is mainly motivated from the experience in manipulating a real-world system, GreenOrbs, deployed in the wild. During the experiment, we observe substantial packet loss on a number of nodes that cannot be characterized by existing path estimation metrics. We first give a brief description on the system architecture of GreenOrbs. We then present our basic observations on packet loss, which reveals the gray zone of packet drops within the sensor node. We take a close look at the packet losses with the work flow of the packet forwarding in the software implementation on a sensor node.

3.1 GreenOrbs System

GreenOrbs is a sensor network system for supporting a variety of forestry applications, such as canopy closure estimates, fire risk evaluation, microclimate monitoring, and carbon dioxide measurement. The latest deployment of GreenOrbs system consists of 330 nodes in the woodland. It has been in operation since December 2009.

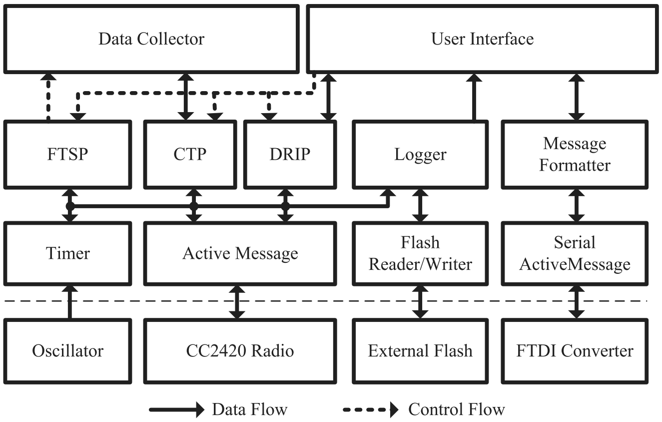


Fig. 3. Component graph of GreenOrbs implementation.

GreenOrbs uses the TelosB mote with MSP430 processor and CC2420 transceiver. We develop the program based on TinyOS 2.1. Currently, GreenOrbs operates with a synchronized low duty cycling mechanism and employs CTP [8] for data collection. A maximum of 30 retransmissions are provided to improve link layer reliability. In order to record the behaviors and performance of packet forwarding within a sensor node at a fine granularity, we modified the CTP component, recording all related events as well as some statistical information. Fig. 3 shows the component graph of the GreenOrbs system. The solid arrows stand for the data flow and dashed arrows stand for the control flow.

3.2 Basic Observations

In this section, we present the outdoor testbed results over 330 nodes that motivate our work. As used in [5], we use network yield to measure the quality of data collection of the network. The network yield measures the quantity of data received at the sink with respect to the total data generated by all nodes in the network. The network yield can be calculated by

$$yield = \frac{\#\text{ of data pkts received at the sink during } w}{\#\text{ of data pkts sent by all nodes during } w}.$$

The network yield gives us the goodput of the network, reflecting both forwarding reliability and the throughput.

During the measurement, we vary the network size. The network yield for different network sizes is shown in Fig. 2a. We see from the statistics that there are about 22∼40 percent data lost in the multi-hop data collection when the network scales from 100 to 330. We further look into the lost packets. The packet loss on different nodes is shown in Fig. 2b. We find that packet loss is quite common and a small portion of nodes experience excessively high packet losses. The packet loss here is mainly due to two reasons. The first reason is transmission timeout on the links (exceeding the retransmission threshold); and the second reason is local packet drops within the node which are mainly due to receive/transmit queue overflow, memory

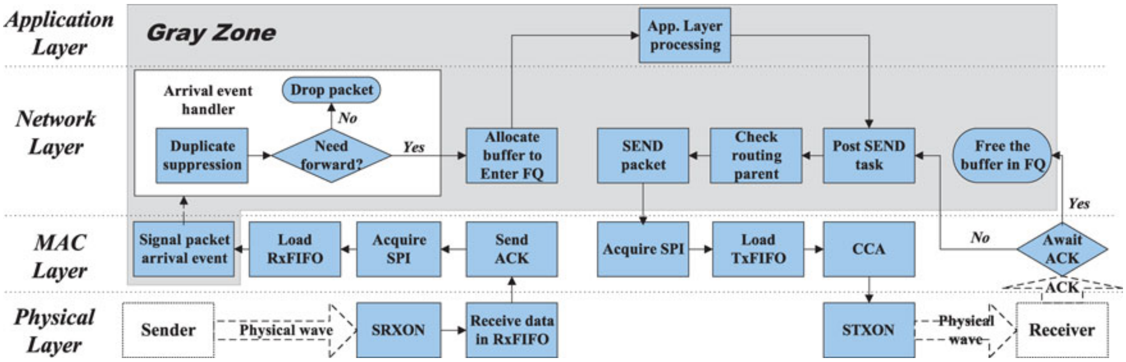


Fig. 4. The work flow of packet forwarding on a sensor node.

corruption, routing loops, packet duplication, and program bugs (e.g., race conditions) etc. We will take a closer look at the causes of packet loss at individual sensor nodes.

3.3 Anatomy of Packet Loss

3.3.1 Packet loss within a node

To understand the causes of packet loss, we take a close look into the work flow on a node that forwards a packet. As depicted in Fig. 4, the flow starts with the node perceiving the physical wave that carries the packet from the sender. It ends when the node gets an ACK from the next-hop receiver or the number of retransmissions reaches the limit.

We can see from Fig. 4 that there exists a gray zone in the work flow, spanning across the network, and application layers. Existing path estimators only measure the forwarding quality of packets outside the gray zone. The packet is presumed passing through the gray zone 100 percent successful. According to aforementioned observations, however, there is a probability for a packet to get lost when it is processed in the *gray zone*. Though CTP uses software ACK and beacon message to measure link quality, it cannot effectively capture the gray zone and its impact to the entire path. Though the working flow is for CTP protocol, this can also be applied to routing protocols with similar steps.

We conduct experiments on a 50-node testbed. In Fig. 5, we summarize the occurrence of several causes that lead to packet drops. The receiver queue overflow is due to the resource constraint on sensor nodes. The packets duplicate suppression is due to the fact that routing layer information is not timely updated. Task failure is caused by the OS mechanism that does not allow the same task to be posted twice if the former one has not been finished. The sendDone failure is due to the mismatch of number of successfully send operations and number of sendDone events.

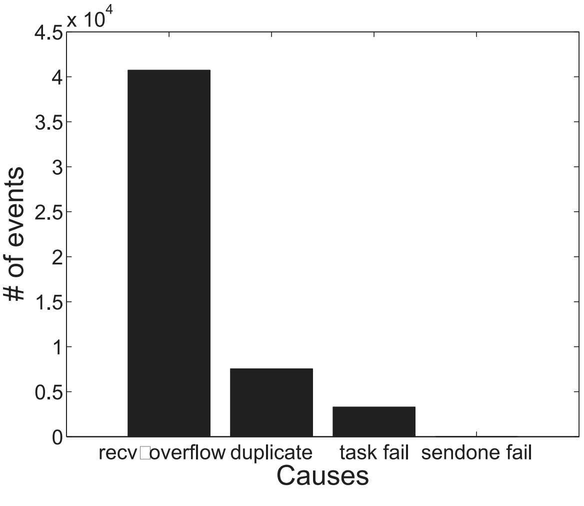


Fig. 5. Different causes of packet loss on sensor nodes.

The above anatomy reveals that overlooking the existence of the gray zone inside the sensor node is likely to cause mismatch between the path estimation and the real forwarding capacity of the path. Thus we shall carefully estimate such an unreliable factor in the gray zone and use it as an important indicator to select "good" paths for data delivery.

A node can be rendered unreliable by many factors, for example,

- *Network congestion*. The receive/transmit queues will overflow, resulting in packet drops inside the node.
- *Software bugs*. For example, on 10th Jan 2010, we observed a number of nodes that accept much traffic without forwarding it. After many rounds of verifications, we found it is due to a software bug in receiving buffer contention, which causes a node drop portion of received packets. We also found that many nodes choose those malfunctioned nodes as parents as they are unaware of the intra-node packet drops. Such a fact reveals the limitation of existing ETX-based routing in overlooking node unreliability.
- *Hardware failures*. As reported in [21], [25], the sensor clock becomes more unstable at a higher temperature. The atrocious weather, clock drift [26] and environment may also increase the chance of packet losses on a relaying node.

Indeed, there are many effective approaches to address software bugs, race conditions, routing loops, congestion etc. They can, however, hardly address all the problems inside the node, and they may not be effective to address these problems in real time. For example, blacklisting and backpressure are effective to address network congestion, but they may not be effective in alleviate packet losses due to software bugs. Therefore, there is still a probability that a packets gets lost inside the node. Combining all these factors into an integral metric can be beneficial to routing protocols in presence of node unreliability. Therefore, in this paper, we propose node-QoF to measure the instant node forwarding quality and take it into consideration in building optimized routing paths.

3.3.2 Packet Loss over Link

Another observation is that transmission timeout is common and accounts for a large portion of packet drops. For a 100-hour data set collected from the 330-node network, the transmission timeout accounts for 61.08 percent of all packet drops while the remaining packet loss is due to intra-node unreliability. This is the result collected from a network with a relatively high retransmission threshold of 30 in CTP. For lower values of the retransmission threshold, the problem will be more severe. As the example shown in Fig. 1 illustrates, ETX-based routing will treat both paths equally good because both paths have a path-ETX of 20. However, in considering data delivery reliability, path 2 is obviously better than path 1 as it delivers about two times more packets. ETX, which works well in reducing the transmission cost when end-to-end delivery is presumed highly reliable, may not be necessarily appropriate for improving the end-to-end reliability of data delivery.

4 QOF METRIC DESIGN

Current metrics for route selection have either of the following two limitations. (1) They only consider the transmission cost without necessary consideration of the data delivery ratio along a forwarding path. (2) They only consider packet losses over the links while overlooking packet drops on a forwarding node. To address these limitations, we propose QoF, Quality of Forwarding, which is defined to be the data yield over the actual number of transmissions. Therefore, the QoF metric comprehensively characterizes both the data delivery ratio and transmission cost of the forwarding paths.

4.1 Generic Link Model

We use a generic link model as the basis of QoF design. A generic link from A to B in the model represents either a physical link or a traversing path of the packet inside a node (so called a virtual link). When it is a physical link, A and B are the corresponding sending and receiving nodes. When it is a virtual link, A and B respectively denote the starting (packet received) and ending (packet successfully sent) points of packet forwarding on a relaying node.

Associated with a link AB, there are two attributes.

- The link quality q, which denotes the probability of a packet to successfully go through the link.
- The limit of retransmissions r. The sender is allowed to retransmit the packet for at most r times before giving up.

Then the packet delivery ratio on a generic link is

$$PDR = 1 - (1-q)^{r+1}. \tag{1}$$

Equation (1) indicates the probability of a packet to successfully go through the link with quality q and retransmission limit r.

Note that the physical and virtual links comprise the entire path of packet delivery from source to destination. We examine the PDR on both types of links.

PDR over a physical link. For a physical link, q is the link quality and r is the retransmission limit.

PDR over a virtual link. For a virtual link inside the node, q denotes the forwarding quality of a forwarding node, and $r = 0$, i.e., no retransmission is executed inside a node. For a typical sensor node, PDR measures the forwarding quality from where the packet is received (at the MAC layer) to where the packet is passed downwards to the MAC layer for transmission.

PDR on a virtual link characterizes the forwarding quality of a node. For example, if a faulty node receives a large number of packets without forwarding them, its PDR equals to 0.

4.2 QoF of a Generic Link

In order to consider both the transmission cost and the packet delivery ratio, we define ETC of a generic link to be the expected transmission count for a unique packet over the link,

$$\begin{aligned} ETC &= \left(\sum_{k=1}^{r+1} kq(1-q)^{k-1}\right) + (r+1)(1-q)^{r+1} \\ &= \frac{1-(1-q)^{r+1}}{q}, \end{aligned} \tag{2}$$

where $q(1-q)^{k-1}$ denotes the probability that a packet passes the link at kth time. Therefore, the term $\sum_{k=1}^{r+1} kq(1-q)^{k-1}$ calculates the expected number of transmissions that the packet passes the link and the term $(r+1)(1-q)^{r+1}$ calculates the expected transmission count that a packet fails to pass the link.

ETC represents the actual transmission count a packet experiences. Note that ETC differs from ETX in that it not only considers link quality but also retransmission limit. When $r \to \infty$, $ETC = ETX$. When r equals to a fixed number, ETC is different from ETX. ETX overestimate the transmission count while ETC reflects the actual transmission count under retransmission limit r.

We define QoF of a generic link as ratio of the data delivery ratio to the actual transmission cost,

$$QoF = \frac{PDR}{ETC}.$$

QoF calculates the expected $successful$ end-to-end delivery per transmission cost. For a single link, substituting equations (1) and (2) into the above equation, we have

$$QoF = \frac{PDR}{ETC} = q.$$

For a single general link, $QoF^{-1} = ETX$.

4.3 QoF of a Forwarding Path

To calculate the QoF of a forwarding path, we use the following notations as shown in Fig. 6,

- $LinkPDR_{n,n-1}$ is the PDR over the link from node n to node n-1.
- $NodePDR_n$ denotes the PDR on node n.

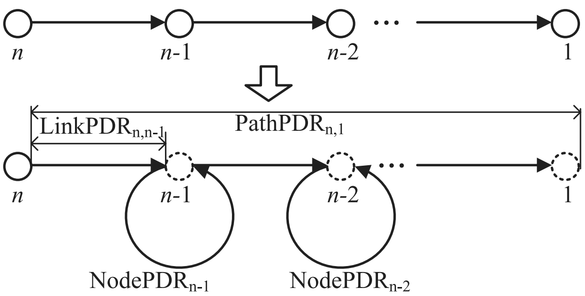


Fig. 6. QoF computation.

- $PathPDR_{n,i}$ denotes the PDR of path (n, i), with node PDR on the starting node n excluded. As shown in Fig. 6, $PathPDR$ can be calculated as $PathPDR_{n,1} = LinkPDR_{n,n-1} \times NodePDR_{n-1} \times LinkPDR_{n-1,n-2} \times NodePDR_{n-2} \times \cdots \times NodePDR_1$.
- $LinkETC_{n,n-1}$ is the ETC over the link from node n to node $n-1$. It can be calculated according to equation (2). $PathETC_{n,1}$ is the expected number of transmissions of a packet along the path from n to 1.
- $QoF_{n,1}$ is the QoF for the n-hop path from n to 1. $QoF_{n,1}$ is the number of packets received at the destination over the actual number of transmissions along the path. $QoF_{n,1} = \frac{PathPDR_{n,1}}{PathETC_{n,1}}$.

We do not count the ETC within a node because it does not incur actual communication cost. $QoF_{n,1}$ is calculated as,

$$\begin{aligned} QoF_{n,1} &= \frac{PathPDR_{n,1}}{PathETC_{n,1}} \\ &= \{PathPDR_{n,1}\}/\{LinkETC_{n,n-1} \\ &\quad + LinkPDR_{n,n-1} \cdot NodePDR_{n-1} \cdot PathETC_{n-1,1}\}, \end{aligned} \tag{3}$$

where $LinkPDR_{n,n-1} \cdot NodePDR_{n-1}$ denotes the probability that a packet from n can be forwarded by $n-1$. We can also see that even when $r \rightarrow \infty$, the path QoF^{-1} is significantly different from ETX.

Since $PathETC_{n-1,1} = \frac{PathPDR_{n-1,1}}{QoF_{n-1,1}}$, $QoF_{n,1}$ can also be calculated in a hop-by-hop recurrence as follows:

$$\begin{aligned} QoF_{n,1} &= \{PathPDR_{n,1}\}/\{LinkETC_{n,n-1} + LinkPDR_{n,n-1} \\ &\quad \cdot NodePDR_{n-1} \cdot PathETC_{n-1,1}\}, \\ &= \frac{PathPDR_{n,1}}{LinkETC_{n,n-1} + \frac{PathPDR_{n,1}}{QoF_{n-1,1}}}, \end{aligned} \tag{4}$$

where $PathPDR_{n,1} = LinkPDR_{n,n-1} \cdot NodePDR_{n-1} \cdot PathPDR_{n-1,1}$. $QoF_{2,1}$ can be calculated by equation (3).

The distributed computation of $QoF_{n,1}$ is as follows.

- Node 1 broadcasts its $NodePDR_1$.
- Node 2 calculates $QoF_{2,1}$ by equation (3). $LinkPDR_{2,1}$ and $LinkETC_{2,1}$ can be calculated locally; $NodePDR_1$ is broadcasted by node 1. Node 2 broadcasts (1) $QoF_{2,1}$ (2) $NodePDR_2$ (3) $PathPDR_{2,1} = LinkPDR_{2,1} \cdot NodePDR_1$.
- Node n ($n > 2$) calculates $QoF_{n,1}$ by equation (4). (1) $PathPDR_{n,1} = LinkPDR_{n,n-1} \cdot NodePDR_{n-1} \cdot PathPDR_{n-1,1}$. $LinkPDR_{n,n-1}$ can be calculated locally. $NodePDR_{n-1}$ and $PathPDR_{n-1,1}$ are broadcasted by node n-1. (2) $LinkETC_{n,n-1}$ can be calculated locally. (3) $QoF_{n-1,1}$ is broadcasted by node n-1.

Each node only needs to perform two kinds of operations. 1) Upon receiving a broadcast message, it updates the QoF and corresponding information. 2) If the QoF is updated, it broadcasts the QoF and corresponding information. When the QoF is calculated and not updated, the node does not need to perform any operations. The calculation of QoF is based on link quality and NodePDR. Thus when those parameters change, a node will update the QoF and broadcast the information to other nodes. The path QoF considers both data delivery ratio and transmission cost. If the data delivery ratios of two paths are the same, QoF favors the path with lower transmission cost. If the transmission cost of two paths are the same, QoF favors the path with high data delivery ratio. The QoF metric differs from ETX in three aspects. First, it calculates the transmission cost more accurately. As mentioned above, ETX overestimate the transmission cost. Second, it considers data delivery ratio which is important for data collection protocols. Third, it also considers node unreliability. When only the reliability is considered instead of the cost, the PDR metric can be used.

The QoF metric also considers the impacts of link positions [3] and node positions as shown in calculation of PathETC. Consider path 2 shown in Fig. 1. Path 2's QoF equals to $\frac{1}{56}$. If the two links were exchanged, path 2's QoF would be $\frac{1}{38}$. The QoF metric selects path with good links near the destination. If bad links are near the destination, the packet loss on such links will waste transmission efforts on previous hops while not contributing to the data yield at the destination.

Now we look back to the example shown in Fig. 1, the following are three scenarios to examine the QoF for forwarding paths:

Scenario 1. Assume $r = 0$ and all nodes have $PDR = 1$ (nodes will never drop packets).

For path P1, we have $LinkPDR_{3,2} = 1/10$, $LinkPDR_{2,1} = 1/10$, and $PathPDR_{3,1} = 1/100$. The expected transmission count for link between nodes 2 and 3 can be computed as $LinkETC_{3,2} = 1$, and $LinkETC_{2,1} = 1$. The path QoF for P1 is $QoF_{2,1} = 1/10$ and $QoF_3 = 1/110$, which means a successful end to end delivery incurs 110 transmissions for P1. For path P2, similarly we have $QoF_{3,1} = 1/38$, which means a successful end to end delivery incurs 38 transmissions for P2.

This scenario shows that although the paths have equal ETX values, the cost for a successful end to end delivery can be quite different. The cost for P1 is about two times larger than the cost for P2. Such a difference cannot be measured by ETX. By simultaneously considering the transmission cost and the packet delivery ratio QoF provides more comprehensive estimation.

Scenario 2. Now we consider the second scenario where $r = 0$ and there are nodes with PDR less than 1. For simplicity, we assume only one node with PDR value less than 1. If $NodePDR_2 = 1/2$, we have $QoF_{3,1} = 1/57$ on P2. If

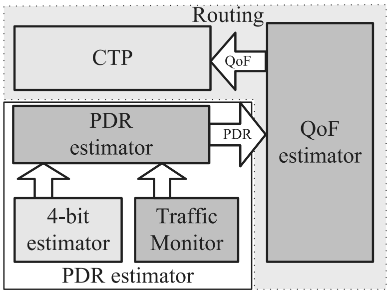


Fig. 7. Integrating QoF with CTP and Layer structure of QoF implementation.

$NodePDR_1 = 1/2, QoF_{3,1} = 1/76$ on P2. This scenario shows that the node's forwarding quality indeed affects the transmission cost.

Scenario 3. In this scenario, we assume $r = 1$ for all links and $PDR = 1$ for all nodes. For P1, $QoF_{3,1} = 19/1190$. For P2, $QoF_{3,1} = 37/1064$, which is about two times of P1's QoF, while originally QoF of P2 is almost three times of that of P1. Such a result further implies increasing retransmission count can improve the path quality while the improvements brought by retransmissions are different for different paths.

5 IMPLEMENTATION

Our implementation is based on TinyOS 2.1 in NesC. We implement the QoF and incorporate it in state-of-the-art data collection protocol, CTP. The hardware platform is the TelosB mote with MSP430 MCU and CC2420 radio.

The overview architecture is shown in Fig. 7. The dark grey components are the components we implemented. The light grey components are the components provided in TinyOS. The layer structure is shown in Fig. 7. More details of the structure and interfaces can be found in Appendix A, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TPDS.2013.98.

5.1 Measuring PDR within a Node

The PDR within a node can be computed as

$$PDR = outCtr/inCtr,$$

where $inCtr$ is the number of packets received by NodePdrC during a packet window of length w, $outCtr$ is the number of packets that is passed down to NodePdrC for transmission. As shown in Fig. 8, the component, NodePdrC, sits above the MAC layer and below the networking layer, and it is responsible for monitoring the incoming traffic and outgoing traffic across the two layers. When NodePdrC receives a packet from the MAC layer, it increases the incoming traffic counter ($inCtr$), and then forwards the packet to the upper layer. When NodePdrC receives a packet from the upper layer, it increases the outgoing traffic counter ($outCtr$), and then transfers the packet to the lower layer for actual transmission. More details to deal with aggregation and packet loss inside a node in NodePdrC can be found in Appendix B, available in the online supplementary material.

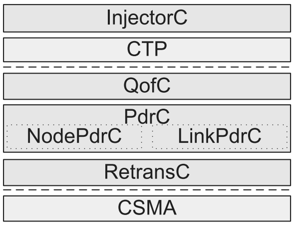


Fig. 8. Layer structure of QoF implementation.

The short-term PDR value, PDR_w, is calculated according to the number of incoming and outgoing packets in the current window w. The long-term PDR value is calculated by $PDR = (1-\alpha) \times PDR + \alpha \times PDR_w$. We set $\alpha = 9/10$ in our current implementation. Since the node PDR estimation requires a sufficient number of incoming packets, if there is few incoming traffic. We actively inject packets by a component, InjectorC, which is above the CTP component. The InjectorC component is triggered when there are few incoming packets. It periodically injects packets to the network layer, i.e., CTP in our case.

5.2 Measuring PDR over a Link

In order to measure the PDR over a link, we first need to measure the PRR over a link. Then we use the specified link retransmission threshold r to obtain the link PDR as follows: $PDR = 1 - (1 - PRR)^{r+1}$. The LinkPdrC component provides link PDR estimations, relying on state-of-the-art link estimation methods (for estimating link PRR). In our current implementation, the link PRR estimation is provided by the 4-bit link estimation component in the TinyOS distribution.

5.3 QoF Calculation

To calculate the QoF of the path, we use the distributed computation method described in Section 4. The beacon interval of the broadcast is controlled by a Trickle timer [27], which increases the time interval exponentially when the network is steady and decreases the interval to the minimum when there is new information for update. To address the problem of nodes with software bugs that will not report information, we also propose collaborative reporting in Appendix C, available in the online supplementary material.

6 EVALUATION

In this section, we evaluate effectiveness of our design extensively. Section 6.1 introduces the evaluation methodology. Section 6.2 examines how QoF improves the routing performance. Section 6.3 reveals further observations on QoF. Section 6.4 summarizes the results.

6.1 Methodology

We use a testbed network consisting of 50 TelosB nodes to evaluate the efficacy of our design. Fig. 9 depicts the testbed we conduct experiments on. Each node is attached with a USB wire to reprogram the node. In our experiments, data are collected from the network to the sink, the left-bottom node, and then sent to the base station, which is usually a PC. We integrate QoF with CTP (CTP-QoF) for evaluating

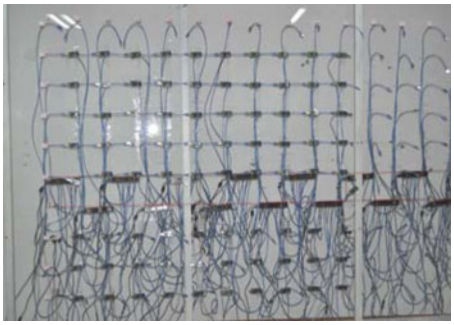


Fig. 9. The picture of testbed.

the performance of QoF in supporting the routing. In the network, we set the power level of the transmissions to 1 to construct a multi-hop network. We mainly compare CTP-QoF with the original CTP protocol (CTP-ETX) in following two cases.

Case I. Streaming application case. In this case, we set the retransmission threshold to 1 and the transmission frequency per node to 3 Hz. In such a case, we explore the performance of QoF in supporting data streaming applications that pursue low latency and high traffic throughput without reliability guarantee on individual packets.

Case II. Real-world deployment case. In this case, we set the retransmission threshold to 30 and the transmission frequency per node to 3 Hz, which is the same with our settings in the real-world deployment. Based on our experience with GreenOrbs, we use a program version before Jan 10th 2010, with some "faulty" nodes that drop a portion of incoming packets. Some faulty nodes can still report QoF to its neighboring nodes while other nodes may keep silent all the time.

We use three key metrics to compare CTP-QoF and CTP-ETX:

1. Data yield: the number of successfully received packets at the sink over the total number of generated packets.
2. Transmission cost: the total number of packets transmitted in the network.
3. Transmission cost per data delivery: the number of transmissions normalized by the data yield.

6.2 Performance Comparison

We present the experimental results in this section, comparing the performance of CTP-QoF and CTP-ETX.

6.2.1 Data Yield

We first investigate the improvement of using QoF to the data yield in the network. In this experiment, we use the setting in Case I and Case II. Fig. 10 depicts the node yield of CTP-QoF and CTP-ETX for Case I. For brevity, we refer CTP to CTP-ETX and QoF to CTP-QoF. The node yield gives the data yield from a specified node. We find from Fig. 10 that most nodes have a higher node yield in CTP-QoF than nodes in CTP-ETX. There are only four nodes (ID 9, 12, 41, 45) which have lower node yield. We find that link qualities of those nodes in CTP-QoF are very low and among all outgoing links the node cannot find an alternative path to avoid packet drops. The average improvement is around 12 percent. The improvement of CTP-QoF in Case I is mainly due to the fact that as the retransmission limit is low, PDR of links is low and thus packets are likely to be dropped over links. This type of packet drops cannot be captured by ETX estimation. The packet drops due to limited retransmissions affect the entire path and such an effect, while is not quantified by existing methods, can be captured by the QoF metric.

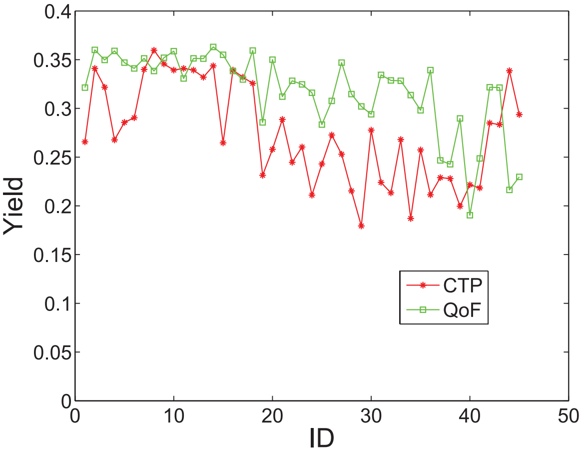


Fig. 10. Comparison of node yield for CTP-ETX and CTP-QoF for Case I.

Fig. 11 shows the node yield of CTP-ETX and CTP-QoF for Case II. During the experiment, the faulty nodes randomly drop around 30 percent of the received packets. Again we find from Fig. 11 that almost all nodes in CTP-QoF have higher node yields than those in CTP-ETX. This is because that QoF measures the forwarding qualities of both the nodes and the links. We find from Fig. 11 that the yield of some nodes in CTP-ETX are only one third of those in CTP-QoF. We look into the experimental data and find that with ETX estimation, most faulty nodes exist on the optimal data delivery paths chosen by CTP-ETX. In CTP-QoF, however, most of the faulty nodes are excluded from the optimal routing paths. For example, as shown in Fig. 12, we select node 20 near the faulty nodes on the testbed and investigate its behavior. Links from 20 to nodes 88, 89, 14, 15 are

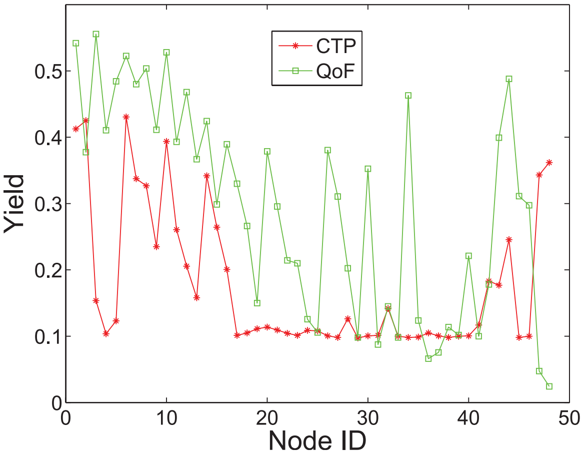


Fig. 11. Comparison of node yield for CTP-ETX and CTP-QoF for Case II.

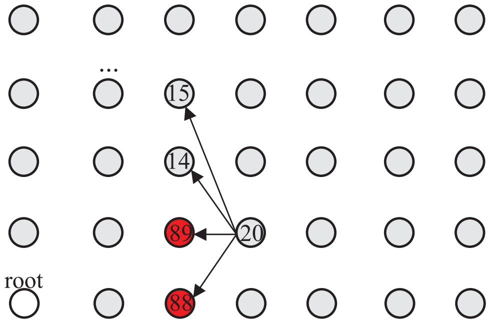


Fig. 12. Topology of the nodes on the testbed. Node 20 has links to node 14, 15, 88, 89. Nodes 88 and 89 are two faulty nodes.

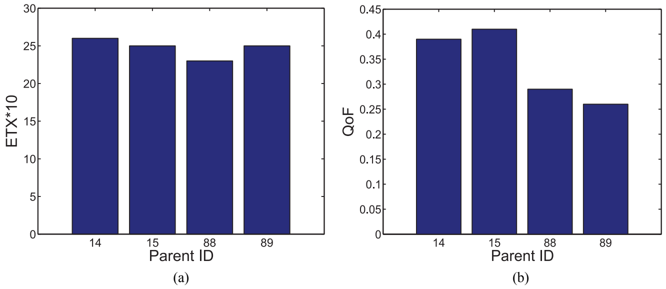


Fig. 13. Path-ETX and path-QoF from node 20 through nodes 14, 15, 88, 89 (a) ETX comparision. (b) QoF comparision.

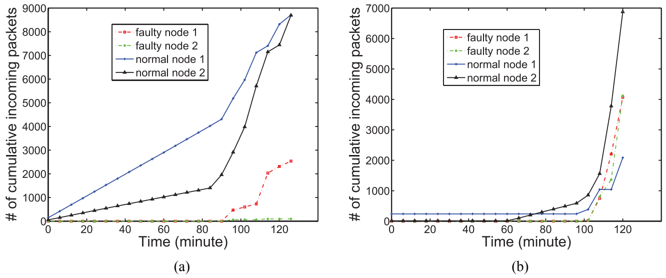


Fig. 14. Incoming packets for QoF and ETX in case II. (a) CTP-QoF, (b) CTP-ETX.

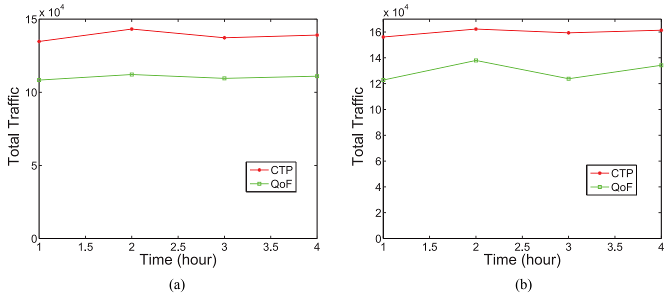


Fig. 15. Total traffic comparison of CTP-ETX and CTP-QoF for different cases. (a) Case I, (b) Case II.

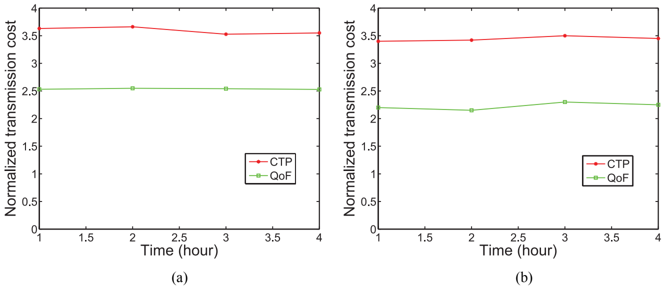


Fig. 16. Normalized traffic comparison of CTP-ETX and CTP-QoF for different cases. (a) Case I, (b) Case II.

shown in the Fig. 12. Here nodes 88 and 89 are two faulty nodes. Fig. 13a shows the ETX values of node 20 to four neighbors. By using the ETX values, the node cannot distinguish faulty nodes from normal nodes. Fig. 13b shows that paths containing the faulty nodes have a lower QoF. Therefore, using QoF values can avoid selecting path containing those faulty nodes.

We further investigate the behaviors of faulty nodes in comparison with the normal nodes in CTP-QoF. The normal nodes are selected close to faulty nodes such that they have similar external conditions.

We find from Fig. 14a a clear trend that in CTP-QoF the incoming traffic for faulty nodes does not accumulate much, as the QoF information help to choose paths that avoid the faulty nodes of much lower forwarding quality. On the other hand, CTP-ETX overlooks the internal problems on the faulty nodes and chooses delivery paths simply according to the link estimation. As a result, the incoming traffic on faulty nodes grows similarly with the traffic on normal nodes as shown in Fig. 14b. It implies that other nodes are unaware of the packet loss inside the faulty nodes and keep sending packets to them, resulting in degraded routing performance.

6.2.2 *Transmission Cost*

In this set of experiments, we evaluate the total data transmissions incurred within the entire network for CTP-QoF and CTP-ETX. We conduct experiments for both Cases I and II. Fig. 15a shows the network traffic cost for Case I, and Fig. 15b shows the network traffic cost for Case II. In both two cases, CTP-QoF saves nearly 30 percent transmission cost compared with CTP-ETX.

6.2.3 *Normalized Transmission Cost*

In this section, we evaluate the average number of transmissions for a successful end-to-end delivery (i.e. normalized transmission cost), which the QoF metric tries to minimize. Fig. 16 compares CTP-QoF with CTP-ETX in both Cases I and II. The average cost for the end-to-end data delivery in CTP-QoF is much less than the cost in CTP-ETX. For Case I, the CTP-QoF reduces the average cost/yield by 28 percent. For Case II, the CTP-QoF reduces the average cost/yield by 34 percent. This is mainly because of the more comprehensive estimation on the path quality in CTP-QoF.

6.3 Observations on QoF

In order to further explore the detailed reasons for the improvement in CTP-QoF, we investigate the details of PDR on node. We test the network at high traffic pressure (3 packets/second) and (low traffic 1 packet every 3 seconds) at different retransmission thresholds ($r = 1$ and $r = 30$).

When the network traffic is low, most nodes are with a high forwarding quality, as suggested in Fig. 17. Fig. 17 also shows the PDR on nodes with high traffic pressure. We find

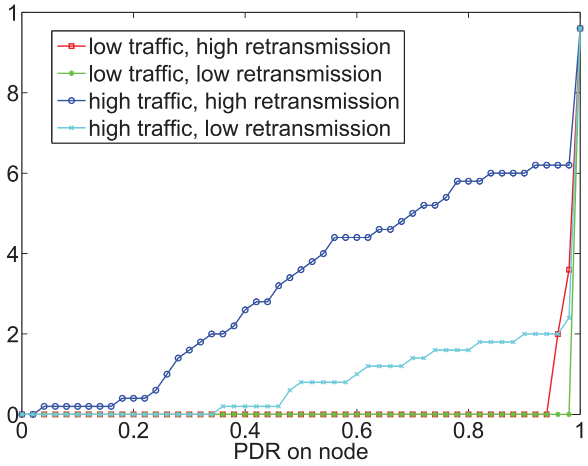


Fig. 17. CDF of node PDR for under high traffic and low traffic.

that the PDR values are more diversified. For the network with high traffic and high retransmission threshold, there are about 40 percent of nodes with PDR lower than 60 percent. This is mainly because more packets accumulate within the nodes and thus more likely a node suffers from resource constraints (e.g., receive queue overflow). The QoF approach will be more effective in such a diversified situation, and is able to select nodes of better PDR in building the routing paths.

6.4 Summary of Results

The above experimental results reveal the following findings.

1. The QoF metric minimizes the benefit/price ratio. The evaluation results show that QoF-based routing reduces the cost/delivery by 28-34 percent.
2. The QoF metric can effectively capture the impact of both the data yield and the transmission cost, improving the data yield by 12-15 percent and reducing the transmission cost by about 30 percent.
3. There is a tradeoff in determining the retransmission limit. With a higher retransmission limit, the link PDR is supposed to be improved while the node PDR may decrease as retransmission consumes system resources. Higher number of retransmissions may also increase channel contentions, affecting other links' reliability. Therefore, the retransmission limit should be carefully chosen so that we can obtain a satisfactory PDR along a path.

7 CONCLUSION

Comprehensive and accurate measurement of path quality is an essential and crucial factor in founding an efficient routing mechanism for multihop wireless sensor networks. Existing approaches for path quality estimation emphasize link quality in between in nodes, but overlook the end-to-end data delivery ratio and the node unreliability. This paper presents our experience with GreenOrbs, which suggests that the existing metrics fail to make comprehensive measurement of path quality. Our proposal called QoF, overcomes this limitation by simultaneously considering node forwarding quality and link quality based on a generic link model. As analyzed and demonstrated by the experiments, routing decisions yielded with QoF achieve low traffic cost as well as high end-to-end delivery ratio.

ACKNOWLEDGMENTS

This study was supported in part by NSFC/RGC Joint Research Scheme N_HKUST602/08, National Basic Research Program of China (973 Program) under Grants No. 2010CB328000 and 2011CB302705, NSFC under Grant No. 61202359 and China Post doctoral Science Foundation under grant 2012M520013. A preliminary work has been presented in IEEE INFOCOM 2011 [28].

REFERENCES

[1] D. Couto, D. Aguayo, J. Bicket, and R. Morris, "A High-Throughput Path Metric for Multi-Hop Wireless Routing," *Proc. ACM MobiCom*, 2003.
[2] L. Sang, A. Arora, and H. Zhang, "On Exploiting Asymmetric Wireless Links Via One-Way Estimation," *Proc. ACM MobiHoc*, 2007.
[3] G. Jakllari, S. Eidenbenz, N. Hengartner, S. Krishnamurthy, and M. Faloutsos, "Link Positions Matter: A Noncommutative Routing Metric for Wireless Mesh Networks," *Proc. IEEE INFOCOM*, 2008.
[4] L. Mo, Y. He, Y. Liu, J. Zhao, S. Tang, X. Li, and G. Dai, "Canopy Closure Estimates with GreenOrbs: Sustainable Sensing in the Forest," *Proc. Seventh ACM Conf. Embedded Networked Sensor Systems (SenSys)*, 2009.
[5] G. Werner-Allen, K. Lorincz, J. Johnson, J. Lees, and M. Welsh, "Fidelity and Yield in a Volcano Monitoring Sensor Network," *Proc. USENIX Seventh Symp. Operating Systems Design and Implementation (USENIX OSDI)*, 2006.
[6] CTP TEP, http://www.tinyos.net/tinyos-2.x/doc/html/tep123.html, 2013.
[7] TinyOS, http://www.tinyos.net, 2013.
[8] O. Gnawali, R. Fonseca, K. Jamieson, D. Moss, and P. Levis, "Collection Tree Protocol," *Proc. Seventh ACM Conf. Embedded Networked Sensor Systems (SenSys)*, 2009.
[9] J. Polastre, R. Szewczyk, and D. Culler, "Telos: Enabling Ultra-Low Power Wireless Research," *Proc. ACM/IEEE Fourth Int'l Symp. Information Processing in Sensor Networks (IPSN)*, 2007.
[10] K. Srinivasan and P. Levis, "RSSI is under Appreciated," *Proc. Third Workshop on Embedded Networked Sensors (EmNets)*, 2006.
[11] S. Lin, G. Zhou, K. Whitehouse, Y. Wu, J. Stankovic, and T. He, "Towards Stable Network Performance in Wireless Sensor Networks," *Proc. IEEE 30th Real-Time Systems Symp. (RTSS)*, 2009.
[12] H. Zhang, L. Sang, and A. Arora, "Comparison of Data-Driven Link Estimation Methods in Low-Power Wireless Networks," *IEEE Trans. Mobile Computing*, vol. 9, no. 11, pp. 1634-1648, Nov. 2010.
[13] R. Draves, J. Padhye, and B. Zill, "Comparison of Routing Metrics for Static Multi-Hop Wireless Networks," *ACM SIGCOMM Computer Comm. Rev.*, vol. 34, no. 4, pp. 133-144, 2004.
[14] O. Gnawali, M. Yarvis, J. Heidemann, and R. Govindan, "Interaction of Retransmission, Blacklisting, and Routing Metrics for Reliability in Sensor Network Routing," *Proc. IEEE First Ann. Comm. Soc. Conf. Sensor and Ad Hoc Comm. and Networks (SECON)*, 2004.
[15] A. Cerpa, J. Wong, M. Potkonjak, and D. Estrin, "Temporal Properties of Low Power Wireless Links: Modeling and Implications on Multi-Hop Routing," *Proc. ACM MobiHoc*, 2005.
[16] Y. Gu and T. He, "Data Forwarding in Extremely Low Duty-Cycle Sensor Networks with Unreliable Communication Links," *Proc. ACM Fifth Int'l Conf. Embedded Networked Sensor Systems (SenSys)*, 2007.
[17] S. Biswas and R. Morris, "ExOR: Opportunistic Multi-Hop Routing for Wireless Networks," *Proc. ACM SIGCOMM*, 2005.
[18] S. Chachulski, M. Jennings, S. Katti, and D. Katabi, "Trading Structure for Randomness in Wireless Opportunistic Routing," *Proc. ACM SIGCOMM*, 2007.

[19] K. Kim and K. Shin, "On Accurate Measurement of Link Quality in Multi-Hop Wireless Mesh Networks," *Proc. ACM MobiCom*, 2006.

[20] I. Stojmenovic and X. Lin, "Power Aware Localized Routing in Wireless Networks," *IEEE Trans. Parallel and Distributed Systems*, vol. 12, no. 11, pp. 1122-1133, Nov. 2001.

[21] T. Schmid, Z. Charbiwala, J. Friedman, Y. Cho, and M. Srivastava, "Exploiting Manufacturing Variations for Compensating Environment-Induced Clock Drift in Time Synchronization," *ACM SIGMETRICS Performance Evaluation Rev.*, vol. 36, no. 1, pp. 97-108, 2008.

[22] S. Moeller, A. Sridharan, B. Krishnamachari, and O. Gnawali, "Routing without Routes: The Backpressure Collection Protocol," *Proc. ACM/IEEE Ninth Int'l Conf. Information Processing in Sensor Networks (IPSN)*, 2010.

[23] B. Hull, K. Jamieson, and H. Balakrishnan, "Mitigating Congestion in Wireless Sensor Networks," *Proc. ACM Second Int'l Conf. Embedded Networked Sensor Systems (SenSys)*, 2004.

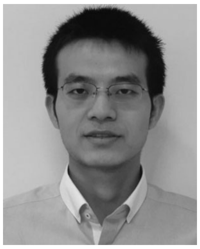
[24] A. Sridharan, B. Krishnamachari, "Explicit and Precise Rate Control for Wireless Sensor Networks," *Proc. ACM Seventh Conf. Embedded Networked Sensor Systems (SenSys)*, 2009.

[25] T. Schmid, R.S. Shea, Z.M. Charbiwala, J. Friedman, Y.H. Cho, and M.B. Srivastava, "On the Interaction of Clocks, Power, and Synchronization in Duty-Cycled Embedded Sensor Nodes," *ACM Trans. Sensor Networks* , vol. 7 article 24, 2010.

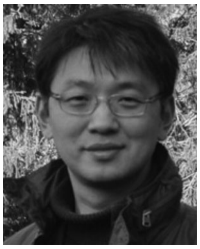
[26] Z. Li, W. Chen, C. Li, M. Li, X.yang Li, and Y. Liu, "Flight: Clock Calibration Using Fluorescent Lighting," *Proc. ACM MOBICOM*, 2012.

[27] P. Levis, N. Patel, D. Culler, and S. Shenker, "Trickle: A Self-Regulating Algorithm for Code Propagation and Maintenance in Wireless Sensor Networks," *Proc. USENIX First Conf. Symp. Networked Systems Design and Implementation (NSDI)*, 2004.

[28] J. Wang, Y. Liu, M. Li, W. Dong, and Y. He, "QoF: Towards Comprehensive Path Quality Measurement in Wireless Sensor Networks," *Proc. IEEE INFOCOM*, 2011.



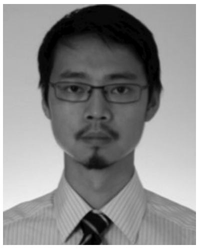
Jiliang Wang (M'09) received the BE degree from the Department of Computer Science, University of Science and Technology of China and the PhD degree from the Department of Computer Science and Engineering, Hong Kong University of Science and Technology. His research interests include wireless sensor networks, network measurement, and pervasive computing. He is a member of the IEEE.



Yunhao Liu (SM '06/ACM '06) received the BS degree in automation from Tsinghua University, China, in 1995, and the MS and PhD degrees in computer science and engineering from Michigan State University, East Lansing, in 2003 and 2004, respectively. He is currently an EMC chair professor at Tsinghua University, as well as a faculty member at the Hong Kong University of Science and Technology. His research interests include wireless sensor network, peer-to-peer computing, and pervasive computing. He is a senior member of the IEEE and a member of the ACM.

Yuan He received the BE degree from the University of Science and Technology of China, the ME degree from the Institute of Software, Chinese Academy of Sciences, and the PhD degree from the Hong Kong University of Science and Technology. He is a member of Tsinghua National Lab for Information Science and Technology. His research interests include sensor networks, peer-to-peer computing, and pervasive computing. He is a member of the IEEE and the ACM.

Wei Dong received the BS and PhD degrees in computer science from Zhejiang University, China, in 2005 and 2010, respectively. He was a postdoc fellow at the Department of Computer Science and Engineering, Hong Kong University of Science and Technology from December 2010 to December 2011. He is currently an assistant professor at the College of Computer Science in Zhejiang University. His research interests include networked embedded systems and wireless sensor networks. He is a member of the IEEE.



Mo Li (M'06) received the BS degree from the Department of Computer Science and Technology, Tsinghua University, China and the PhD degree in computer science and engineering from Hong Kong University of Science and Technology, in 2004 and 2010, respectively. He is currently an assistant professor in the School of Computer Engineering, Nanyang Technological University, China. His research interests include wireless sensor networking, pervasive computing, and peer-to-peer computing. He is a member of the IEEE.