ERTP: Energy-Efficient and Reliable Transport Protocol for Data Streaming in Wireless Sensor Networks

Tuan Le[12] Wen Hu[2] Peter Corke[2] Sanjay Jha[1]

[1] University of New South Wales, Australia
{dtle,sanjay}@cse.unsw.edu.au
[2] Autonomous Systems Laboratory, CSIRO ICT Centre, Brisbane, Australia
{wen.hu, peter.corke}@csiro.au

Technical Report
UNSW-CSE-TR-0802
January 2008

THE UNIVERSITY OF
NEW SOUTH WALES

School of Computer Science and Engineering
The University of New South Wales
Sydney 2052, Australia

Abstract

Emerging data streaming applications in Wireless Sensor Networks require reliable and energy-efficient transport protocols [17] [18] [19] [14]. Our recent Wireless Sensor Network deployment in the Burdekin delta, Australia for water monitoring [14] is one such example. Our application involved streaming sensed data such as pressure readings, water flow rate, and salinity readings periodically from many scattered sensors to the sink node which in turn relayed them via an IP network to a remote site for archiving, processing and presentation. While latency is not a primary concern in this class of application (the sample rate is usually in terms of minutes or hours), energy-efficiency is. Long-term operation and reliable delivery of the sensed data to the sink are also desirable.

In this paper, we discuss ERTP, an Energy-efficient and Reliable Transport Protocol for Wireless Sensor Networks. ERTP is designed for data streaming applications, in which sensor readings are transmitted from one or more sensor sources to a base station (or sink). ERTP uses a statistical reliability metric which ensures the number of data packets delivered to the sink exceeds the defined threshold. Using a statistical reliability metric when designing a reliable transport protocol guarantees the delivery of adequate information to the users, and reduces the number of transmissions when compared to absolute reliability.

To reduce energy-consumption, ERTP uses hop-by-hop Implicit Acknowledgment with dynamically updated retransmission timeout for loss recovery. In multihop wireless networks, the transmitter can overhear a forwarding transmission and interpret it as an Implicit Acknowledgment. However, Implicit Acknowledgment timeout depends on the time taken a packet to be forwarded by the downstream node. Thus, a dynamic retransmission timeout estimation is crucial for the class of Hop-by-Hop Implicit Acknowledgment transport protocol.

By combining statistical reliability and hop-by-hop Implicit ACK loss recovery, ERTP can provide the reliability to application users with the minimal energy-expense. Our extensive discrete event simulations and experimental evaluations show that ERTP is significantly more energy-efficient than current approaches and can reduce energy consumption by more than 50% when compared to current approaches. Consequently, sensors are more energy-efficient and the lifespan of the unattended WSN is increased.

1 Introduction

Many applications in Wireless Sensor Networks (WSNs) produce streaming data [14] [17] [18] [19]. In this class of applications, each sensor node periodically samples and relays the sensed data to a central data collection node, referred to as the base station or the sink. In such applications, two important requirements are: end-to-end reliability and long-term operation. A data packet needs to be reliably relayed to the sink. Typically, the end-to-end transmission latency is not a primary concern in this class of applications, but energy-efficiency is. WSNs are expected to operate independently for weeks, or even for months. Our recent Wireless Sensor Network deployment in the Burdekin delta, Australia, for water monitoring [14] is one such example. Our application involved streaming sensed data such as pressure, water flow rate, and salinity periodically from many scattered sensors to the sink node which in turn relayed them via an IP network to a remote site for archiving, processing and presentation. In our deployment, we observed that the channel quality of radio links between sensor nodes is unreliable and the packet error rate in each link varied considerably over time [14]. Similar observation has been reported in literature [11] [12] [20].

Often, the reliability requirements for data streaming applications are not absolute but rather statistical in their nature. That is, the reliability is determined by the quantity of data packets delivered to the sink rather than the reliability of each data packet. For example, in the Burdekin deployment, we would like to study sensor readings over a reasonable scale of time such as a week or a day, but not over the scale of a minute or a second. In fact, we require that at least 75% of data packets per day from each sensor node are received at the sink [14] for offline processing. Using a statistical reliability metric when designing a reliable transport protocol guarantees delivery of enough information to the users, and also reduces the number of transmissions as compared to an absolute reliability metric. Previous studies in [3] [15] have also shown that the statistical reliability approach can significantly reduce energy-consumption.

While many transport protocols for WSNs have been studied in the literature, most of them do not address both requirements, i.e., transmission reliability and energy efficiency, for data streaming applications. Current work focuses on either providing reliability for data transmission, or minimizing energy-consumption, but not both. In this paper, we discuss the design and implementation of an Energy-efficient Transport Protocol (ERTP) that ensures statistically reliable delivery of sensor data to the sink for data streaming applications in WSNs. To reduce energy-consumption, we achieve the end-to-end reliability by controlling the reliability at each hop dynamically. ERTP uses Stop-and-Wait Hop-by-Hop Implicit Acknowledgment (SW HBH iACK) for loss recovery. In wireless links, the transmitter can overhear forwarding transmissions and interprets them as iACKs. Obviously, when a packet reaches the sink, there will be no further forwarding so the sink node needs to send an explicit ACK (eACK). The transmitter retransmits the packet if, after a certain timeout, no acknowledgment has been received.

The primary contributions of the paper are summarized as follows:

- We present an analysis of the trade-off between energy consumption and end-to-end reliability for ERTP, in which HBH iACK approach and duplicate detection are used at each sensor node. To balance energy consump-

Table 2.1: Sensor Network Transport Protocols

Protocol Name	Main Approach	Reliability	Energy-Aware	Type of data flows
ESRT	Centralized Rate Control	Yes	No	Continuous
RMST	Hop-by-hop NACK	Yes	Yes	Continuous
PSFQ	Hop-by-hop NACK	Yes	No	Bulk
RBC	Windowless block ACK	No	No	Bulk
Flush	Distributed Rate Control	Yes	No	Bulk
Wisden	End-to-end NACK	Yes	No	Continuous
RCRT	Centralized Rate Control	Yes	No	Continuous
ERTP	Hop-by-hop iACK	Yes	Yes	Continuous

tion and reliability, ERTP dynamically controls the maximum number of retransmissions at each sensor node.

- We propose a distributed algorithm for retransmission timeout estimation in ERTP. Determining how long the node should wait for an iACK is non-trivial since iACK timeout depends on the time it takes a packet to be forwarded by the downstream node. The simulation results in Section 4 show that the proposed retransmission timeout algorithm is significantly more energy-efficient than other approaches.
- We design, implement, and evaluate ERTP in TinyOS for real-world sensor networks. Our extensive evaluations show that ERTP can reduce energy consumption by more than 50% when compared to current approaches. Consequently, sensor nodes are more energy-efficient and the lifespan of the unattended WSN is increased.

The remainder of the paper is organized as follows. Related studies are described in Section 2. The protocol details are described in Section 3. The simulation and implementation results are presented in Sections 4 and 5. The paper is concluded in Section 6.

2 Related Work

A summary of relevant related work on transport protocols for WSNs is given in Table 2.1. We distinguish the transport protocols by three different characteristics: reliability, energy-awareness, and the type of data flows that they support (continuous data flows or a bulk data flow). As shown in Table 2.1, ERTP and RMST [6], to the best of our knowledge, are the transport protocols for continuous data flow WSNs that takes reliability and energy-constraints into account. RMST [6] uses Hop-by-Hop Negative Acknowledgment (NACK) for loss recovery. However, RMST is tightly bound to Directed Diffusion routing protocol [21] in which packet losses are recovered hop-by-hop using caches in the nodes along the path to the sink. Furthermore, RMST is not scalable because it requires each intermediate nodes to cache all packets received from each upstream source. Memory limitation on resource-constrained sensor nodes requires intelligent caching strategies to be considered. However, RMST is the closest in spirit to our work in that it attempts to control the hop-by-hop reliability to achieves end-to-end reliability. Unlike RMST, ERTP achieves the end-to-end reliability through hop-by-hop loss recovery using HBH iACK approach and it is independent to the routing protocols. Thus, ERTP has greater flexibility than RMST.

In [4], Akan et al. proposed Event to Sink Reliable Transport (ESRT) for end-to-end reliability based on the notion of event-to-sink reliability. ESRT achieves the reliable detection of an event and congestion avoidance by controlling the transmission rate of each source at the sink. Although ESRT does not require packet retransmissions, it is not as energy-efficient as hop-by-hop loss recovery schemes since the rate decision is controlled centrally. Moreover, ESRT assumes that the sink can communicate with all sources directly, which may not be a reasonable assumption in practical WSN deployments. PSFQ [5] is a reliable dissemination protocol aimed for reprogramming WSNs from a sink, i.e., a bulk data flow, a large finite bulk of data packets which needs to be transmitted to the sink, not for the transport of streaming data from the sources to the sink.

To overcome the memory constraints, an end-to-end NACK loss recovery scheme is used in [7] [8] [9] to provide transmission reliability in WSNs. In this scheme, the sink detects packet losses and requests end-to-end retransmissions from the source nodes. Although this scheme alleviates the memory burden on sensor nodes, it is not as energy-efficient as hop-by-hop loss recovery [3]. Moreover, using end-to-end NACKs may cause feedback explosion when the links are lossy. Xu et.al proposed Wisden [7], a reliable data collection protocol for structural monitoring. However, Wisden uses end-to-end NACKs for loss recovery, and thus is not energy-efficient. Kim et.al proposed Flush [8], a reliable, single-flow bulk transport protocol for large diameter WSNs. However, Flush only supports one data flow and targets bulk traffic. Paek et.al proposed RCRT [9], a rate-controlled reliable transport protocol for WSNs. Both Flush and RCRT focus on achieving 100% reliability and high throughput via congestion control without consideration of energy-efficiency. In contrast, ERTP explores the characteristics of statistical reliability in data streaming applications to reduce energy-consumption in packet transmissions, and achieves end-to-end reliability through hop-by-hop loss recovery using the iACK approach.

Woo et al. proposed an adaptive rate control mechanism which passively adapts the transmission rates of both original and forwarding traffic for the purpose of fair bandwidth allocations [24]. Snooping is used to reduce the control packets in communication handshakes. Zhang et al. proposed Reliable Bursty Convergecast (RBC) protocol to transport bulk traffic reliably in WSNs [25]. RBC uses a windowless block acknowledgment scheme to improve channel utilization and to reduce the number of nodes competing for channel access. However, RBC is not designed for continuous data flows, and does not guarantee statistical end-to-end reliability. Rangwala et al. proposed IFRC [27], an Interference-aware Fair Rate Control for WSNs. IFRC is a distributed rate allocation scheme that uses the local queue size to detect congestion. Bian et al. proposed QCRA (Quasi-static Centralized Rate Allocation) [28], a centralized rate allocation scheme which aims to achieve fair and near optimal rate allocation. Although both IFRC and QCRA aim to achieve high throughput, they do not guarantee statistical end-to-end reliability.

Scheuermann et al. presented a hop-by-hop congestion control protocol in wireless multihop networks [34] using HBH iACK. The protocol ensures that the input rate of a given flow does not exceed the output rate in all intermediate nodes. To avoid redundant retransmissions, several heuristics to handle packet loss are discussed. However, this work primarily addresses transmission rate and congestion control for absolute reliability and only considers the packet loss caused by buffer overflows whereas in reality, packet loss is mostly caused by lossy wireless channels. Moreover, a fixed retransmission timeout scheme (three times the HBH transmission time) is used in this work. Our simulation results in Section 4 show that a fixed retransmission timeout scheme is not energy-efficient when the link loss rates are high. We propose a distributed algorithm for retransmission timeout estimation, which adapts to the environment, i.e., lossy wireless channels. To the best of our knowledge, ours is the first work which investigates adaptive retransmission timeout estimation for the class of HBH iACK protocols in WSNs.

Surge Reliable [10] is a state-of-the-art reliable multi-hop routing protocol for continuous data flows that uses expected number of transmissions as the routing metric. Surge Reliable dynamically forms a spanning tree that covers every node in the network, using link connectivity estimation and neighborhood table management techniques. Each node periodically measures the link qualities between itself and its neighbors by link layer active snooping. A node obtains bi-directional link qualities by exchanging neighborhood tables with its neighbors. Link layer hop-by-hop eACK and retransmissions improve end-to-end transmission reliability. A node selects the best neighbor, the one with the minimum expected number of transmissions, as its parent to which it forwards data packets. The performance of Surge Reliable has been shown to be superior to other routing protocols such as Destination-Sequenced Distance-Vector Routing (DSDV) [22], and Ad hoc On-Demand Distance Vector Routing (AODV) [23] in unreliable wireless environments. However, a fixed number of link layer retransmissions is used, and therefore, it does not guarantee statistical end-to-end reliability when the link loss rates change. Further, Surge Reliable is not energy-efficient when the link quality is good, since it introduces a significant number of eACKs.

3 ERTP: An Energy-Efficient and Reliable Transport Protocol

Definition 1 *The application layer end-to-end reliability for each sensor node* α *(*$0 < \alpha < 1$*), is described by probability of data packets to be delivered to the sink.*

Definition 2 *The hop-by-hop reliability requirement* β_k *for flow* k *(*$0 < \beta_k < 1$*), is described by probability of data packets of node* k *to be delivered from one node to its next-hop node along the routing path between source* k *to sink.*

In this section, we firstly provide an overview of ERTP that includes the requirements and our assumptions. We then discuss the details of the components of ERTP: the Hop-by-Hop Reliability Control, and the Hop-by-Hop Retransmission Timeout (RTO) Control. Finally, we discuss other details of ERTP that include link quality estimation, duplicate packet detection, and a distributed algorithm for RTO updating.

Table 3.1: Notation

Symbol	Meaning
α	Application layer end-to-end reliability requirement
β_k	Hop-by-Hop reliability requirement for flow k
$N(\beta_k, i)$	The maximum number of retransmissions for a packet of flow k at node i to be delivered successfully with β_k reliability
$X(\beta_k, i)$	The expected number of transmissions from node i to $i+1$ for a packet of flow k to be delivered successfully with β_k reliability
$Y(\beta_k, i)$	The expected total number of transmissions from node i to $i+1$ for the iACK of a packet of flow k received successfully by node i with β_k reliability
p_i	Link error rate between nodes i and $i+1$
q_i	Link error rate between nodes $i+1$ and i
E_k	The expected total number of transmissions for a packet of flow k received at the sink with α reliability
$\xi(k, i)$	The expected overhearing time for a packet of flow k from node i after sending the packet
$T(k, i)$	The retransmission timeout for a packet of flow k at node i

3.1 Overview of ERTP

ERTP is a transport protocol for data streaming applications in WSNs, in which sensor readings are transmitted from one or more sensors (sources) to a base station (or sink). Two requirements of ERTP are:

- **End-to-End Reliability**: Our primary goal is to achieve an application layer end-to-end reliability of all data transmitted by each sensor to a sink.
- **Energy-Efficiency**: While end-to-end transmission latency is not a pressing concern in many WSN data streaming applications, energy-efficiency often is. For long-term unattended operation of the network, the transport protocol should minimize sensor energy consumption.

ERTP makes three assumptions about the link layer below and the application layer above:

- **Low data rates**: ERTP assumes that transmission rate is low such that network congestion is negligible. This is a reasonable assumption for deployed data streaming applications in practice [14] [17] [18] [19].
- **Low cost snooping**: A node is able to overhear packet transmission by single hop neighbours. Estimation through snooping comes at a cost, since a node needs to listen for packets that are not addressed to it (idle listening). We assume that a low power listening (LPL) mechanism [29] [30] [31] [32] is used in the underlying MAC layer. LPL MAC protocols operate at a low duty cycle in which sensor nodes periodically sleep, wake-up, listen to the channel, and then return to sleep instead of idle listening. As a result, the snooping cost is very low [10]. Therefore, the communication cost, i.e., the number of transmissions of data packets, is the dominant factor in sensor energy consumption [36].
- **Low transmission contention**: Transmission collisions happen if at least two neighbouring nodes, which lie within the interference range of

each other, transmit at the same time. However, for low data rate applications, transmission collisions are negligible because the probability that at least two neighbouring nodes transmit at the same time is small. For example, if there are N interfering neighbor nodes and M number of packets that can be transmitted in a period, the probability that two or more nodes transmit a packet simultaneously is

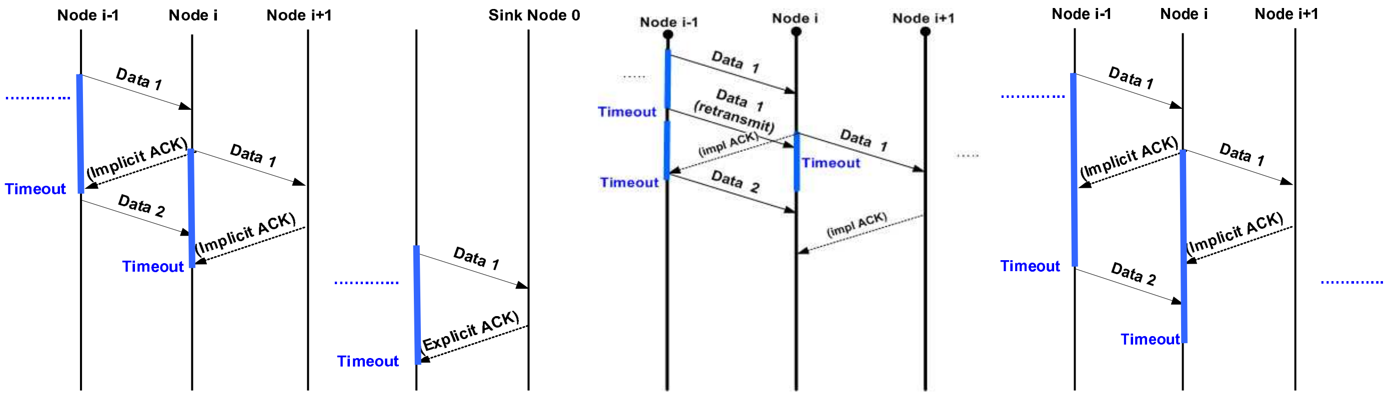
$$1 - 1.\frac{M-1}{M}\frac{M-2}{M}....\frac{M-N+1}{M} = 1 - \prod_{k=1}^{N-1}\frac{M-k}{M} \qquad (3.1)$$

For example, in our Fleck-3 [16] platform, the bandwidth is $W = 50$ kbps and the size of each data packet is $L = 40$ bytes. The transmission rate at each node is $D = 0.017$ packet per second (1 packet per minute). So, the number of packets that can be transmitted in the period $M = \frac{1}{D}\frac{W}{L} = 9192$ packets. If a node has $N = 20$ neighbouring nodes, for a medium density network, the probability that two or more nodes transmit a packet simultaneously is less than 0.02 (Equation 3.1).

ERTP consists of two components: ***hop-by-hop reliability***, and ***hop-by-hop retransmission timeout***.

- ***The Hop-by-hop reliability component*** ensures the required application layer end-to-end reliability by dynamically controlling the maximum number of retransmissions for each data packet in all intermediate nodes. Obviously, a sensor node can not allow a very large number of retransmissions because of packet freshness and fairness concerns. In most transport protocols, a preset number of retransmissions is used [8] [10] [9]. To achieve both end-to-end reliability and energy-efficiency, ERTP dynamically determines the maximum number of retransmissions at each node. An insufficient maximum number of retransmissions may cause packet to be lost as it travels to the sink, wasting energy and network resources, as well as degrading end-to-end reliability. Conversely, there will be energy-inefficiency when the maximum number of retransmissions is too high. To balance energy consumption and end-to-end reliability, the ***hop-by-hop reliability component*** dynamically determines and updates a near optimal maximum number of retransmissions for data packets at each node.

- ***The Hop-by-hop RTO component*** ensures application layer end-to-end reliability by dynamically adjusting the RTO at each node. The hop-by-hop iACK mechanism operates by the transmitter overhearing the packet being forwarded by the receiver to its next hop and considers this as an iACK. The transmitter will retransmit the packet if it has not received the iACK after a time-out interval. Determining how long the node should wait for an iACK is non-trivial [34] and depends on the time it takes a packet to be forwarded by the downstream node. Fig. 3.1(a) shows the normal operation of the HBH iACK protocol. When node i forwards a packet of node $i-1$ to node $i+1$, node $i-1$ overhears this forwarding and considers it as an iACK. A "premature" RTO value for HBH iACK may increase sensor energy-consumption because transmitters will send duplicate packets. This is energy-inefficient since the packet has already been received (Fig. 3.1(b)). On the other hand, a large RTO value tends



(a) Correct *RTO* (b) Premature *RTO* (c) Large *RTO*

Figure 3.1: HBH iACK operation

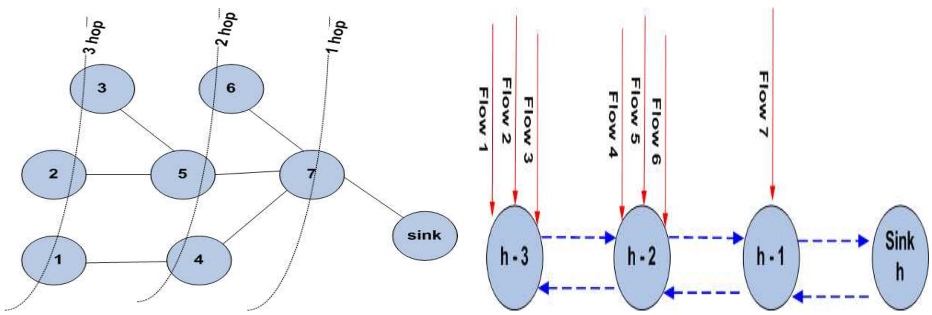
to increase transmission latency and thus reduces network goodputs (Fig. 3.1(c)). Therefore, in order to achieve energy efficiency, the ***hop-by-hop RTO component*** of ERTP is responsible for adjusting the retransmission timeout dynamically. Obviously, when a packet reaches the sink, there will be no further forwarding. Therefore, the sink node needs to send an eACK. Since the eACK is sent immediately by the receiver, the eACK timeout is primarily based on the Hop-By-Hop Round Trip Time. Each node maintains a duplicate packet detection list to prevent duplicate packets being propagated over the network.

The remainder of this section describes each component in detail. Let us denote $0 \leq \alpha \leq 1$ as the desired application layer end-to-end reliability. We first present an idealized model with simplifying assumptions. We then lift these assumptions as we present how ERTP dynamically estimates the maximum number of retransmissions and RTO at each node.

3.2 Hop-by-Hop Reliability Component

Network Model

We model the network as a graph $G = (V, E)$, where V is a set of nodes and E is a set of edge links. Each sensor node periodically transmits its sensed packet to the sink at the rate of D packets per second. Let W (bits per second), L (bits), respectively, denote the network bandwidth and the size of a packet. The packets are served by the sensor nodes on first come first serve basis. We assume that the shortest path routing protocol is used [22] [10]. Thus, sensor nodes are aware of their next-hop neighbors along the routing path to the sink. The network consists of a set of data flows $k \subset V$ ($k \neq$ the sink node), where k represents the ID of the sensor node from which the flow originated. Fig. 3.2(a)-3.2(b) show an example of network topology. The data traffic of the network in Fig. 3.2(a) can be represented as the graph of 7 data-flows as shown in Fig. 3.2(b).



(a) Network Topology (b) The Corresponding Data Flows

Figure 3.2: An example of Network Topology

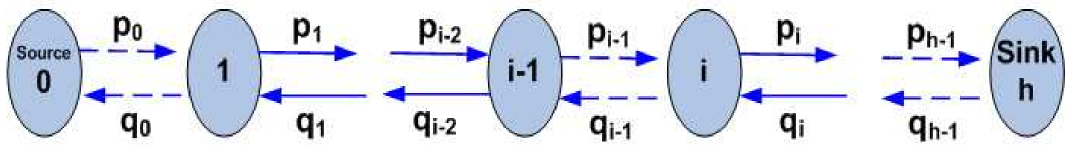


Figure 3.3: Single data flow

Maximum number of retransmissions

Let us consider a data flow which involves $h + 1$ nodes, where node h is the sink and node 0 is the source, as depicted in Fig. 3.3. Node 0 sends packets to the sink through $\{1, 2, ..., h - 1, h\}$. Nodes forward a packet as soon as possible after receiving it. We assume the random channel error model for each link. We denote the probabilities that a transmission from node i to node $i+1$ and from node $i+1$ to node i is received successfully as $1 - p_i$ and $1 - q_i$, respectively. For notational brevity, for every probability p and q, let $\bar{p}$ and $\bar{q}$ denote $1 - p$ and $1 - q$, respectively.

To achieve application layer end-to-end reliability α, the required maximum number of retransmissions N_0^i at node i for a packet of flow 0 is [3]:

$$N_0^i = \frac{log(1 - \alpha^{1/h})}{log(p_i)} \tag{3.2}$$

where $\alpha^{1/h}$ is the HBH reliability requirement for each node. For multi-flow WSNs, each sensor node could be a source node or a relay node. Similarly, we have,

$$N(\beta_k, i) = \frac{log(1 - \beta_k)}{log(p_i)} \tag{3.3}$$

where, $\beta_k = \alpha^{\frac{1}{h-k}}$ is the HBH reliability requirement for flow k, and
$N(\beta_k, i)$ is the maximum number of retransmissions for a single packet of flow k at node i.
Assume that node i transmits a packet of flow k to node $i + 1$. Let us denote $X(\beta_k, i)$ as the expected number of transmissions made by node i for a packet of flow k so that node $i + 1$ receives the packet successfully. This event is

a truncated geometric distribution with the successful probability of $\bar{p}_i$ taken from the set $\{1, 2, ..., N(\beta_k, i)\}$. Thus, its expected value $X(\beta_k, i)$ is:

$$X(\beta_k, i) = \sum_{j=1}^{N(\beta_k,i)} j(\bar{p}_i)(1-\bar{p}_i)^{j-1} + N(\beta_k, i)(1-\bar{p}_i)^{N(\beta_k,i)-1} \tag{3.4}$$

By simplifying (3.4) and re-arrange the terms, we have,

$$X(\beta_k, i) = \frac{1 - p_i^{N(\beta_k,i)}}{\bar{p}_i} \tag{3.5}$$

However, it is possible that node $i+1$ receives the packet from i successfully, but the forwarding by $i+1$ is not overheard by node i (the iACK is lost). Let us denote $Y(\beta_k, i)$ as the expected number of transmissions for a single packet of flow k made by node i so that node i overhears the iACK successfully. Therefore, $Y(\beta_k, i)$ is greater than or equal to $X(\beta_k, i)$ (recall that $X(\beta_k, i)$ is the expected number of transmissions made by node i for a packet of flow k so that node $i+1$ receives the packet successfully, and an iACK will be "sent" by node $i+1$ after node $i+1$ receives the packet successfully).

Proposition 1 *For ERTP, the expected number of transmissions $Y(\beta_k, i)$ for a packet of flow k to be delivered successfully from node i to node $i+1$ is*

$$Y(\beta_k, i) = \begin{cases} \frac{1-(1-\bar{p}_i\bar{q}_i)^{N(\beta_k,i)}}{\bar{p}_i\bar{q}_i} & , i = h-1 \\ X(\beta_k, i) + q_i^{Y(\beta_k,i+1)}(N(\beta_k, i) - X(\beta_k, i)) & , k \le i \le h-2 \end{cases} \tag{3.6}$$

Proof

- For $i = h-1$, eACK is used. Node $h-1$ transmits a packet of flow k until both the packet received at the sink h and the ACK from the sink h is received successfully by $h-1$. The probability of this event is $\bar{p}_{h-1}\bar{q}_{h-1}$. This is a truncated geometric distribution with the successful probability of $\bar{p}_{h-1}\bar{q}_{h-1}$ taken from the set $\{1, 2, ..., N(\beta_k, h-1)\}$. Thus, its expected value $Y(\beta_k, h-1)$ is given by the first term of Equation (3.6).

- For $k \le i \le h-1$, with the optimal iACK timeouts, the transmitter node, i, transmits a packet, either from itself (when $i = k$) or forwarded from node $i-1$ (when $i \ge k$), until the packet is successfully received by node $i+1$. The successful probability of this event is $\bar{p}_i$ and the expected number of transmissions for this event is given by $X(\beta_k, i)$ in Equation (3.5). After this event occurs, node $i+1$ will forward the packet to node $i+2$ with the expected $Y(\beta_k, i+1)$ number of transmissions. Note that each sensor node maintains duplicate detection to prevent redundant packets from being propagated over the network. If node i overhears the forwarding from node $i+1$ to node $i+2$ in one of the $Y(\beta_k, i+1)$ forwarding times, it will transmit the next packet. Otherwise, it will retransmit an additional $(N(\beta_k, i) - X(\beta_k, i))$ times. This event occurs with a probability of $q_i^{Y(\beta_k,i)}$. Therefore, the expected number of transmissions is given by the second term of Equation (3.6).

Proposition 2 *The expected total number of transmissions E_k for a single packet of flow k to be delivered successfully to the sink is*

$$E_k = Y(\beta_k, h-1)(1 + \bar{p}_{h-1}) + \sum_{i=k}^{h-2} (Y(\beta_k, i)) \tag{3.7}$$

Proof The expected total number of transmissions for a packet of flow k is the summation of expected number of transmissions $Y(\beta_k, i)$ at each intermediate node i ($k \leq i \leq h-1$) along the routing path from the source k to the sink h. The expected number of transmissions for a packet to be transmitted successfully from node $h-1$ to h is given by $Y(\beta_k, h-1)$ derived from Proposition 1, regardless of the ACK outcome. In addition, the sink h needs to send eACKs to node $h-1$, so the expected number of transmissions of ACKs in the backward route from the sink h to node $h-1$ is reduced by a factor of $\bar{p}_{h-1}$. Therefore, the expected total number of transmissions is given by the Equation (3.7).

3.3 Hop-by-Hop RTO Component

First, we need to estimate the time required to transmit a packet from node i to node $i+1$. With the channel bandwidth of W and for low data rate applications, the packet collision can be ignored. Thus, the average transmission time for a packet of L bits can be approximated by

$$\overline{T_{tx}} = \frac{L}{W} \tag{3.8}$$

To estimate the RTO value in node $i-1$, assume that node $i-1$ forwards a packet of flow k to node i. Let us denote $\xi(k,i)$ as the expected "overhearing" time in node i. Once node $i-1$ sends a packet of flow k, $\xi(k,i)$ represents the expected time in which node $i-1$ is expected to "overhear" the forwarded packet.

Proposition 3 *For ERTP, the* ***retransmission timeout*** *$T(k,i)$ and the* ***expected "overhearing" time*** *$\xi(k,i)$ for a single packet of flow k ($0 \leq k < h-1$) at node i is given by*

$$T(k,i) = \frac{L}{W} + \xi(k, i+1), k \leq i < h-1 \tag{3.9}$$

$$\xi(k,i) = \begin{cases} \frac{1-(1-\bar{p}_i\bar{q}_i)^{N(\beta_k,i)}}{\bar{p}_i\bar{q}_i}\frac{L}{W} & , i = h-1 \\ \frac{1-(1-\bar{q}_{i-1})^{N(\beta_k,i)}}{\bar{q}_{i-1}}T(k,i) & , k \leq i < h-1 \end{cases} \tag{3.10}$$

where, $T(k,i)$ is the ***retransmission timeout*** for a packet of flow k at node i.

Proof

- For i = h - 1, since eACK is used from node $h-1$ to the sink h, node $h-1$ is expected to send $Y(\beta_k, h-1)$ transmissions for a packet of flow k as given by Proposition 1. Therefore, the expected overhearing time $\xi(\beta_k, h-1)$ is,

$$\xi(\beta_k, h-1) = Y(\beta_k, i)\overline{T}_{tx} \tag{3.11}$$

Substituting (3.6) and (3.8) into (3.11), we can obtain the first term of (3.10). Note that node $h-1$ does not need overhearing time $\xi(\beta_k, h-1)$ since the eACK scheme is used between node $h-1$ and h, but node $h-2$ does need overhearing time $\xi(\beta_k, h-1)$ to estimate its retransmission timeout $T(k, h-2)$.

- For $k \leq i < h-1$, if node i receives packet of flow k from node $i-1$ successfully, it will forward the packet to node $i+1$, but no more than $N(\beta_k, i)$ times. Node $i-1$ is expected to overhear the forwarding by node i with the successful probability of $\bar{q}_{i-1}$. This event is the truncated geometric distribution with the successful probability of $\bar{q}_{i-1}$ taken from the set $\{1, 2, ..., N(\beta_k, i)\}$. Thus, its expected value is:

$$\frac{1 - q_{i-1}^{N(\beta_k,i)}}{\bar{q}_{i-1}} \tag{3.12}$$

  Therefore, the expected overhearing time $\xi(\beta_k, i)$ for a packet of flow k is equal to the number of transmissions from node i to node $i+1$ that node $i-1$ can overhear multiplied by the retransmission timeout setting at node i ($T(k, i)$). Thus, we can obtain the second term of (3.10).

  The retransmission timeout $T(k, i)$ depends on the HBH transmission time from node i to $i+1$ (denotes by $\overline{T_{tx}}$), and the expected "overhearing" time $\xi(\beta_k, i+1)$ for the packet being served at node $i+1$ (denotes by $\xi(\beta_k, i+1)$). Thus, we can obtain (3.9).

3.4 Other Details

Equation (3.3) and Proposition 3 provide the estimation for the maximum number of retransmissions and retransmission timeout values in each node. We now describe how ERTP dynamically estimates the maximum number of retransmissions and RTO values in the real environment.

Link Quality Estimation

So far, we have not discussed how to obtain the link quality p and q. Link quality indicates the packet reception probability of the link and is an important parameter in our protocol. One of the main differences between WSNs and wired networks is that the link quality in WSNs may vary greatly with time as a consequence of interference and propagation dynamics.

Link quality can be obtained from the Link Quality Indicator (LQI) defined by IEEE standard 802.15.4 which is readily used on MicaZ and Telos sensor network devices [1] [2]. For those platforms that do not support LQI, link quality can be estimated by observing packet success and loss events. In our implementation, we use the exponentially weighted moving average (EWMA) [10] for link quality estimation. The EWMA estimator is simple and memory efficient, requiring a constant amount of storage for prior quality estimates. EWMA uses a linear combination of prior estimates, and weighted exponentially. The forwarding probability p over a link l is calculated using the ratio of the number of data packets received to the total number of data packets transmitted over the link l. The link quality on the reverse link l, i.e., q is calculated as $p(\bar{l})$

by the node at the other end of link l. The nodes at both ends of link l exchange the information to obtain the bi-directional link quality of link l.

Duplicate Packet Detection and Avoidance

Due to the the existence of asymmetric links, it is possible that the forwarding packet is successfully received but the iACK is lost. For example, in Fig. 3.4, node $i+1$ may receive the packet from node i successfully, while node $i-1$ does not overhear this forwarding. In this case, node $i-1$ will retransmit the packet, even though node i has already received it. Following the same principle, node i will also have to repeat the transmission to node $i+1$. Therefore, once an iACK is lost, duplicate packets would be generated and propagated over the network to the sink.

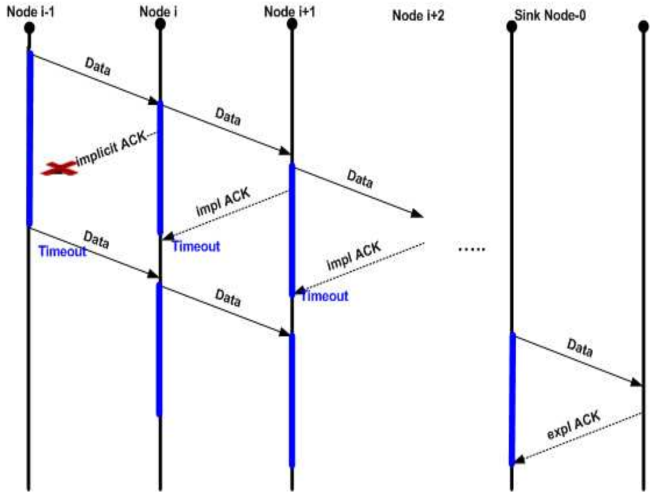


Figure 3.4: The Impact of Loss of Implicit Acknowledgment

We handle this case by a simple duplicate detection mechanism. Each node maintains a list of the M-most recent packets it has received. If a node receives a duplicate packet, it will drop the packet to reduce unnecessary forwarding. In our implementation, we select $M = 5$.

Dynamic Maximum Number of Retransmission Control and Dynamic Retransmission Timeout Control

As the link quality varies with time, sensor nodes need to control the number of retransmissions and RTO values dynamically.

Once the link quality is updated, the hop-by-hop reliability component estimates the maximum number of retransmissions using Equation (3.3). These estimates are noisy so we apply a smoothing filter. Each node maintains the most-recent set of m values and the weight w_m is given to each value in the history. Intuitively, the choice should give greater weight to the recent estimations. The smoothed estimate $\bar{x}$ is calculated from the raw estimates x_i by:

$$\bar{x} = \frac{\sum_{i=0}^{4} w_i x_i}{\sum_{i=0}^{4} w_i} \tag{3.13}$$

where x is the new estimation of either the maximum number of retransmissions or retransmission timeout. We used $m = 5$ and $w = [1, 1, 0.8, 0.6, 0.4]$ in our protocol.

While the maximum number of retransmissions can be calculated locally, the RTO estimation in a sensor node depends on the RTO of its parent node (Proposition 3). This information could be sent explicitly, but such a mechanism would incur additional communications and therefore energy. We use a distributed algorithm to update the RTO as follows. When a new link quality is estimated, node $h-1$ calculates its new timeout T_{h-1} locally by Equation (3.10). Node $h-1$ does not use the T_{h-1} value, but node $h-2$ does need T_{h-1} to estimate its timeout T_{h-2}. To minimize overheads, node $h-2$ snoops T_{h-1}, which is embedded in the forwarded data packet of node $h-2$ from node $h-1$, and estimates its timeout T_{h-2} by Equation (3.9). Similarly, node $h-3$ snoops the new timeout T_{h-2} and estimates its timeout T_{h-3}. Eventually, all the nodes in the network will update their own timeout values.

Since ERTP uses snooping for all the control and update information, it does not explicitly increase overhead. An extra 2-byte field is used for RTO information in the ERTP header.

4 Simulation

In this section, we evaluate the performance of the ERTP through extensive simulations.

4.1 Summary

We conducted the simulations for a 50-node network. The nodes are uniform-randomly deployed in an area of 90m×90m, as shown in Fig. 4.1. The simulations were run in the discrete-event network simulator ns-2 [33] using a modified-version (to enable iACK) of the Carrier Sense Multiple Access (CSMA) MAC protocol, and DSDV as the routing layer protocol. We selected node 0 as the sink and the other nodes generate packets of 40 bytes at the rate of 0.1 packet per second. The simulation parameters are summarized in Table 4.1 based on Fleck-3 platform [16].

The application layer end-to-end reliability requirement is $\alpha = 0.95$. Different channel conditions are considered, i.e., HBH loss error rates are 1%, 5%, 20%,40%, and 50%. For brevity, we only present the results of the following cases: $p = q = 20\%$, $p = q = 50\%$, $p = 50\%$ and $q = 70\%$. We observed similar behaviours in other cases. Each data point in the figures is the average of 10 simulations. The average values are plotted along with their 95% confidence intervals.

4.2 Goals, Metrics, and Methodology

In order to evaluate the performance of hop-by-hop reliability component, each node adjusts the maximum number of transmissions dynamically based on Equation (3.3). We compare the actual achieved end-to-end delivery ratio to the delivery requirement ($\alpha = 0.95$).

Table 4.1: Simulation setup

Parameter	Value
Bandwidth W	50 Kbps
Packet Size L	40 bytes
Statistical Reliability Requirement α	0.95
Arrival Rate	0.1 pps
Radio Transmit Current	31.8 mA
Radio Receive Current	13.4 mA
Supply Voltage	3.3V
Simulation Time	1000 seconds

In order to evaluate the performance of the hop-by-hop RTO component in ERTP introduced in Section 3.3, we compare it with the following algorithms:

- **Theoretical Result:** Proposition 2 shows the expected energy consumption (E_k) required for a packet of flow k ($0 \leq k < h-1$) to be delivered successfully to the sink. The theoretical result provides the lower bound to compare the performance of different RTO algorithms.

- **Fixed RTT:** The RTO value is assigned to a fixed value, i.e., a multiple of RTT. We consider three cases: short timeout such as $RTO = 1 * RTT$ and $RTO = 2 * RTT$, medium timeout such as $RTO = 10 * RTT$ and $RTO = 20 * RTT$, and long timeout such as $RTO = 50 * RTT$ and $RTO = 100 * RTT$. As the obtained results are similar, we only present the results of three cases: $RTO = 1 * RTT$, $RTO = 10 * RTT$, and $RTO = 100 * RTT$.

- **Jacobson's Algorithm:** Jacobson's Algorithm estimates a future Round-Trip Time (RTT) by linearly filtering previous measured $RTTs$, and the RTO value is obtained by adding a scaled mean absolute deviation to the estimated future RTT [26]. The RTO values obtained by Jacobson's algorithm, similar to those obtained by ERTP, change continually as channel conditions vary. Specifically, the estimated RTT g_u for packet u is calculated by

  $$g_u = (1-a) * g_{u-1} + a * h_{u-1} \tag{4.1}$$

  where h_{u-1} is the actual RTT, g_{u-1} is the estimated RTT for packet $u-1$, and a is a constant ($0 < a < 1$). The mean absolute deviation of the estimated RTT v_u is then calculated by

  $$v_u = (1-b) * v_{u-1} + b * |g_{u-1} - h_{u-1}| \tag{4.2}$$

  where v_{u-1} is the mean absolute deviation for the packet $u-1$, and b is a constant ($0 < b < 1$). The RTO T_u for the packet u is

  $$T_u = g_u + 4 * v_u \tag{4.3}$$

  In an IP network, $a = 1/8$, and $b = 1/4$ are used [26]. To study the performance of Jacobson's algorithm, we simulated five different cases of (a, b): $(1/8, 1/4)$, $(1/8, 3/4)$, $(1/8, 19/20)$, $(3/4, 1/4)$, and $(3/4, 3/4)$.

We use the following metrics:

- **Reliability (Delivery Ratio):** This metric characterizes the end-to-end application layer delivery ratio achieved. The application layer reliability requirement is $\alpha = 0.95$.

- **Energy consumption:** This metric characterizes the average energy required for a packet to be delivered to the sink successfully. Ideally, the energy consumption should be as small as possible. Based on the Table 4.1, we can calculate the communication cost, which is 0.477 mJ per packet. To compare sensor energy consumption, we measure the **normalized energy consumption** (R_E) as a ratio of actual energy consumption ($E^{measure}$) and a lower bound of energy consumption (E^{theory}) achieved from Proposition 2. Namely, $R_E = \frac{E^{measure} - E^{theory}}{E^{theory}}$. Thus, the lower R_E is, the more energy-efficient the RTO estimator is.

- **Average Packet Delay:** This metric characterizes the average latency for a data packet to travel from its source to the sink. Ideally, this metric should be as small as possible to indicate timely data transfer.

4.3 Results

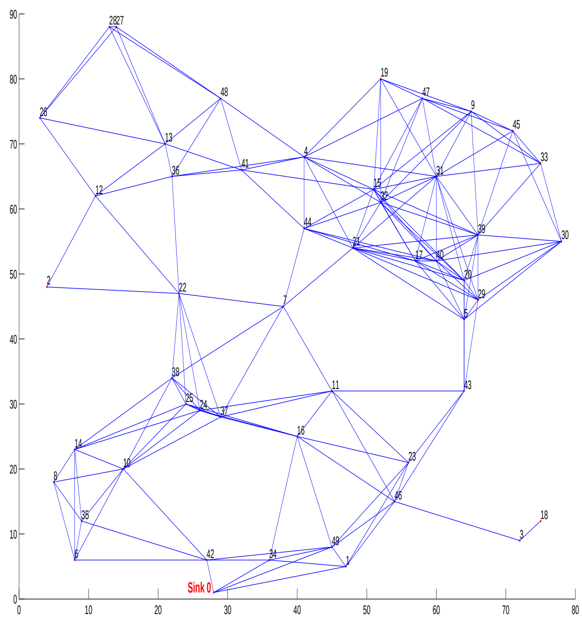
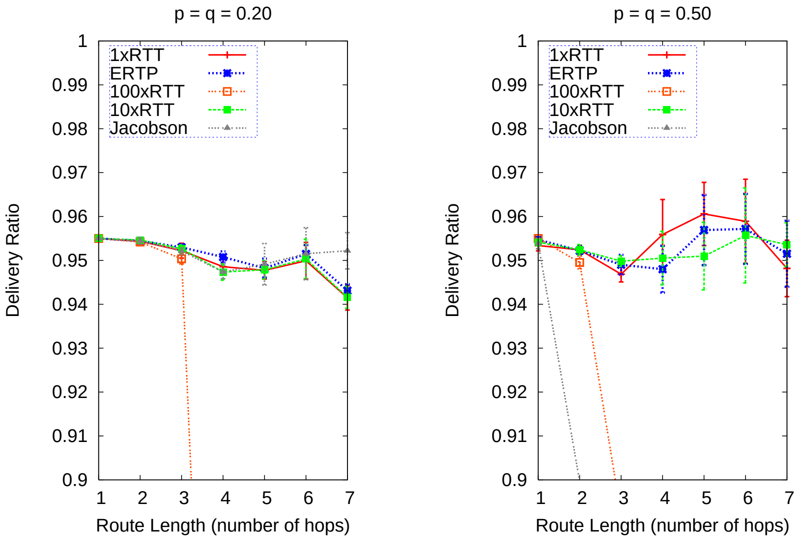
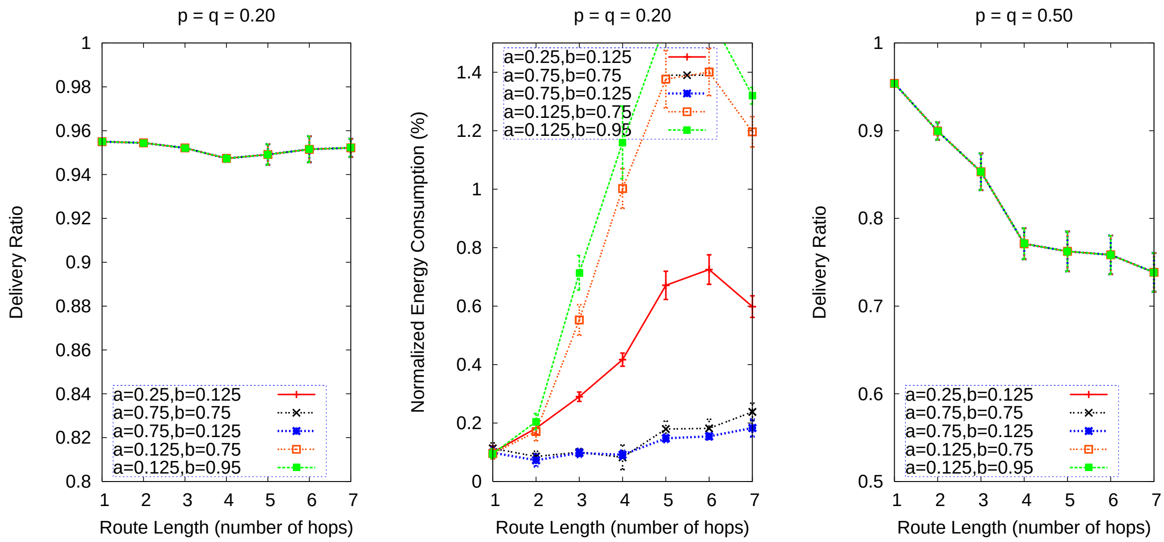


Figure 4.1: Simulation Network Topology



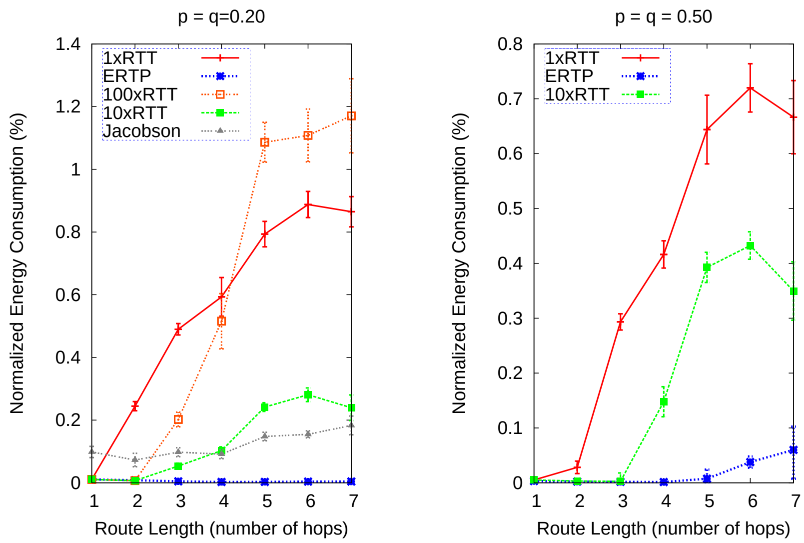
(a) Loss rate p=20% (b) Loss rate p= 50%

Figure 4.2: Average Delivery Ratio



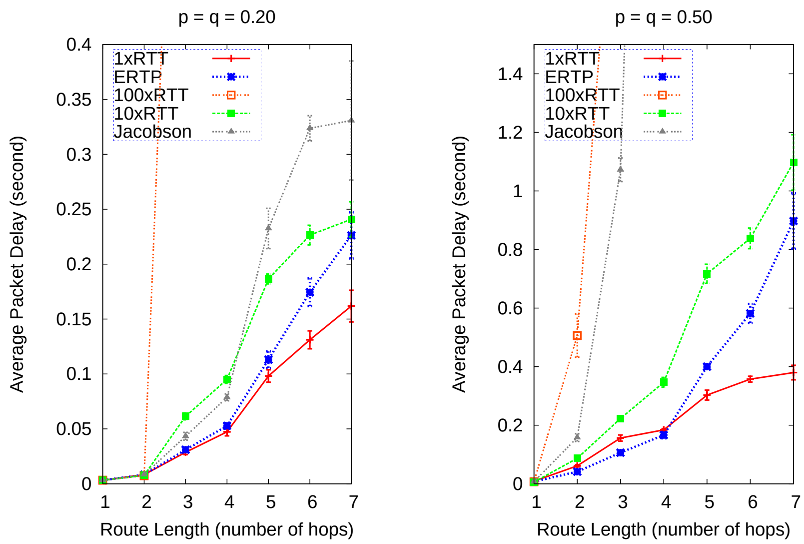
(a) Loss rate p=20% (b) Loss rate p=20% (c) Loss rate p= 50%

Figure 4.3: The performance of Jacobson's algorithm with different values of parameters



(a) Loss rate p=20% (b) Loss rate p=50%

Figure 4.4: Normalized Energy Consumption



(a) Loss rate p=20% (b) Loss rate p= 50%

Figure 4.5: Average Packet Delay

Reliability

Fig. 4.2(a)-4.2(b) show the simulated end-to-end delivery ratios. We observe that ERTP can achieve 95% end-to-end reliability with the small error range of ± 0.01. We observe that the packets were dropped in DSDV routing discovery phase in the simulation. As packet losses by routing are not considered in our theoretical model, the obtained reliability is slightly lower than the reliability requirement for a few nodes, i.e., 7-hop nodes. However, the difference is very small (1%) and on average, ERTP can achieve $94-96.5\%$ end-to-end reliability. The results validate the correctness of the Equation (3.3) as well as the hop-by-hop reliability component.

Note that the Jacobson algorithm and $RTO = 100 * RTT$ do not satisfy the reliability requirement when the link loss rates are high ($p = q = 50\%$) for the reason explained next. Therefore, we do not compare the energy consumption of Jacobson's algorithm and $RTO = 100 * RTT$ to the other cases when their reliability is lower than the design criteria of 95%, i.e., $p = 50\%$.

Jacobson's algorithm

Fig. 4.3 shows the delivery ratios and the normalized energy consumption of Jacobson's algorithm for different sets of a and b. We observe that the performance of Jacobson's algorithm varies significantly with different parameters. When $p = 20\%$, the normalized energy consumption for the case of $a = 0.75, b = 0.125$ is 5 times less than the one for the case of $a = 0.125, b = 0.95$ for 7-hop nodes. When $p = 50\%$, Jacobson's algorithm no longer satisfies the reliability requirement for the $\geq$ 2-hop nodes because Jacobson's algorithm chooses a very long value of RTO when the link loss rates are high. Since Jacobson's algorithm with $a = 0.75, b = 0.125$ provides the best normalized energy consumption in our simulation, we compare this case to the other algorithms.

Energy consumption

Fig. 4.4(a)-4.4(b) show the normalized energy consumption versus the route length when p = 20% and p=50%. Not surprisingly, the long iACK timeout is more energy-efficient than the short one. For 7-hop nodes, the normalized energy consumption in $1 * RTT$ is 60.62% and 30.72% more than $10 * RTT$ for $p = 20\%$ and $p = 50\%$, respectively. This is due to the RTO value of $10 * RTT$ reducing unnecessary retransmissions.

Counter-intuitively, the long RTO scheme, i.e., $100 * RTT$, is not always the most energy-efficient. Fig. 4.4(a) shows that the normalized energy consumption of $100 * RTT$ is significantly more than that of $10 * RTT$. Note that a sensor node will not forward the next packet in its routing queue until either the current packet is successfully forwarded to the next hop or the number of retransmissions for the current packet exceeds the threshold (a characteristic of the Stop-and-Wait iACK protocol). However, retransmissions and very long RTO in the receiver cause very long serving time for the current packet (significantly longer than the RTO value) when the loss occurs. As a result, the forwarding rate of the next packet in the queue is significantly low. Consequently, the transmitter times out well before its receiver forwards the packet, thus it will retransmit the packet.

We observe that the normalized energy consumption with Jacobson's algorithm is about $10-18\%$ higher than the theoretical values. The results show that the linear filter for RTO value it not energy-efficient in lossy wireless multihop networks.

Finally, Fig. 4.4(a)-4.4(b) show that ERTP significantly outperforms other approaches[1]. The normalized energy consumption in ERTP is very close to the theoretical results and is up to 28.4% less than that of $10 * RTT$ when $p = 50\%$. The results validate the correctness of the analysis in Section 3.

Average Packet Delay

As shown in Fig. 4.5(a) and Fig. 4.5(b), a long RTO value has a packet delay penalty. The average packet delay in $RTO = 10 * RTT$ is significantly higher than that of $RTO = 1 * RTT$. When $p = 50\%$, the average packet delay in $RTO = 10 * RTT$ could be three times more than that of $1 * RTT$ (1.1 seconds and 0.38 second respectively, when the nodes are 7 hops away from the sink).

Moreover, we observe that Jacobson's algorithm and $RTO = 100 * RTT$ incur significantly high average packet delay because of the long RTO. For $RTO = 100 * RTT$, when $p = 50\%$, the average packet delay is 49.45 seconds (about 5 times the arrival rate) for the 4-hop nodes and about 261.97 seconds (more than 26 times the arrival rate) for the 7-hop nodes (not shown in the figure). For Jacobson's algorithm, the average packet delays are 44.92 seconds and 83.86 seconds, respectively.

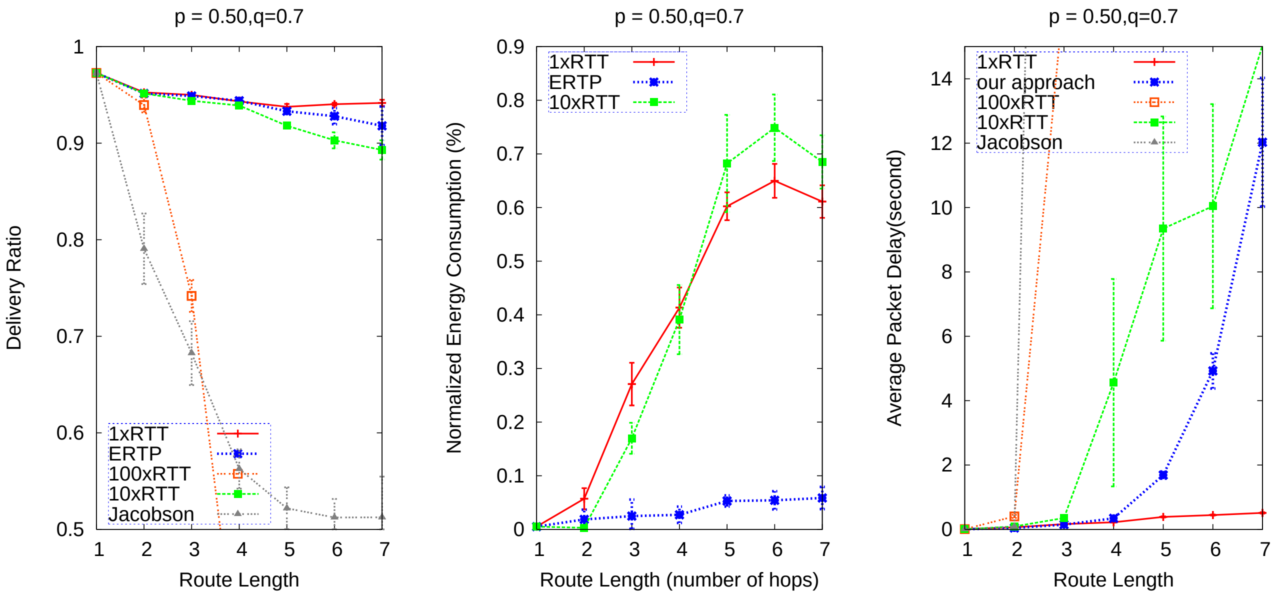
Finally, we observe that the average packet delay in ERTP is as low as that in $RTO = 1*RTT$ for $\leq$ 4-hop nodes and higher than that in $RTO = 1*RTT$ from 5-hop nodes. However, ERTP has lower average packet delay when compared to other approaches.

The impact of asymmetric links

Asymmetric links in sensor networks have been observed and reported by many in the research community [10] [13] and in our experiment described in Section 5. We study the impact of asymmetric links on the performance of ERTP and other RTO estimation approaches. We set upstream link quality $p = 50\%$ and downstream link quality $q = 70\%$. Fig. 4.6(a) shows the delivery ratios achieved by different approaches. Similar to the previous cases, the delivery ratios of Jacobson's algorithm and $100 * RTT$ are relatively low for $\geq$ 3-hop nodes. ERTP achieves the design of reliability requirement of 95% with the error of 3% due to the significant packet losses by routing maintenance. Furthermore, we observe that the delivery ratios in ERTP are slightly higher than those of $10 * RTT$.

Fig. 4.6(b) shows that $10 * RTT$ **is no longer as energy-efficient as** $1 * RTT$ for ≥ 5 hop-away nodes when the links are asymmetric, which validates the necessity for the adaptive RTO component in ERTP. Fig. 4.6(b) also shows that the normalized energy consumption in ERTP closely matches the theoretical results. We observe that ERTP can reduce energy consumption by more than 50% compared to the results of the best heuristic approach.

[1] Note that $RTO = 10 * RTT$ is a static approach which is not adaptive to changes in the environment such as different topology or traffic patterns. Therefore, it is not always the "best" heuristic solution (see Fig. 4.6(b))



(a) Delivery Ratio (b) Normalized Energy Consumption (c) Average Packet Delay

Figure 4.6: Performance with asymmetric links

5 Implementation and Experimental Evaluation

Having validated the performance of ERTP by simulations in Section 4, we implement ERTP in TinyOS 1.x and compare it with state-of-the-art reliable WSN communication protocol, Surge Reliable [10] in a 16 node Fleck-3 [16] testbed.

Fig. 5.1 is a snapshot of the routing tree in one of our experiments. As link changes quality over time, the routing tree also changes. We selected node 0 as the sink and the other nodes generate packets of 40 bytes at the rate of 0.1 packet per second. Each experiment was run for half an hour. Each node logged the energy consumptions for handling each data packet.

5.1 Link Quality

We observed the well-known phenomena in wireless communication such as link asymmetry and dynamic link qualities. Fig. 5.2 and Fig. 5.3 show the link qualities of nodes 8 and 2 during one of our experiments. We discovered that the link quality of node 2 varies a lot because of its low quality antenna.

5.2 Energy Consumption

The average upstream link quality, the average downstream link quality, and the average number of hops from the sink are obtained for each node. Based on these parameters, we can calculate the expected total energy consumption by Proposition 2.

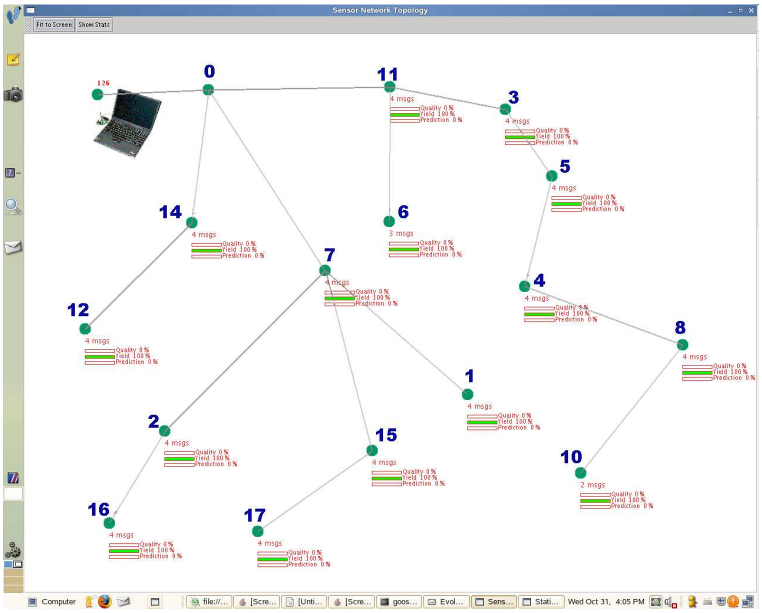


Figure 5.1: Network Topology

Fig. 5.4 compares the predicted and the actual average energy consumption for a data packet to be delivered successfully to the sink. There are slight differences between the theory and the actual measurements because the *RTO* update value is slower than the changes of link qualities. Especially, the energy consumptions of node 16 and node 2 are significantly higher than the theoretical results. Because node 2 is downstream of node 16 in the routing tree, the performance of node 16 is impacted by the poor link quality at node 2 (see Fig. 5.3). Despite these differences, there is a consistent trend between theoretical results and experimental results.

5.3 Delivery Ratio

Fig. 5.5 shows the average delivery ratio of all nodes. Apart from node 2 and 16, the other nodes achieved more than 93% average application layer delivery ratios. The reliability achieved is slightly lower than the requirement for the following reasons. First, the update of the maximum number of retransmissions is slower than the changes of link qualities. Thus, old estimate for maximum number of retransmissions is not accurate when link changes quality. Moreover, we observe that the routing path broke down for a small period of times in our experiments and causes significant packet loss.

Although affected by poor link quality between node 2 and node 7 (see Fig. 5.3), node 2 and node 16 achieved reasonable delivery ratios (83% and 81%, respectively).

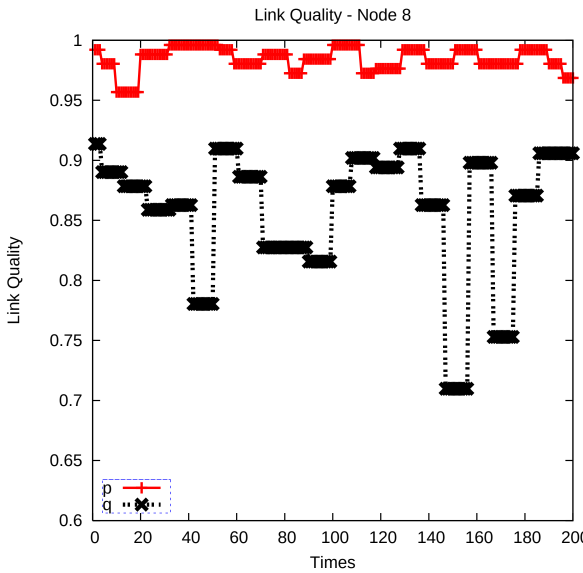


Figure 5.2: Link Quality at Node 8

5.4 Comparison between ERTP and Surge Reliable

We compare the performance of ERTP with state-of-the-art reliable WSN communication protocol, Surge Reliable together with eACK (we call it *Surge* for the purpose of brevity). To compare sensor node energy consumption of ERTP to that of *Surge*, we modified *Surge* so that it can dynamically control the maximum number of retransmissions to satisfy application layer end-to-end reliability requirements. We selected the longest route (the black line in Fig. 5.6) and measured the energy consumption. Note that the routing tree in Fig. 5.6 is slightly different from the one in Fig. 5.1 due to dynamic link qualities. Each experiment was run five times. The statistical reliability requirement is $\alpha = 95\%$.

Fig. 5.7 shows the average energy consumption of ERTP and *Surge*. We also observe that ERTP significantly outperforms *Surge*. As eACK is used for 1-hop nodes, the average energy consumption of ERTP is similar to that of *Surge*. However, for a 6-hop node, ERTP reduces by more than 42% energy consumption compared to *Surge*.

We also conducted experiments to assess the performance of ERTP when the link error rates are high. We introduced artificial losses for all links in the

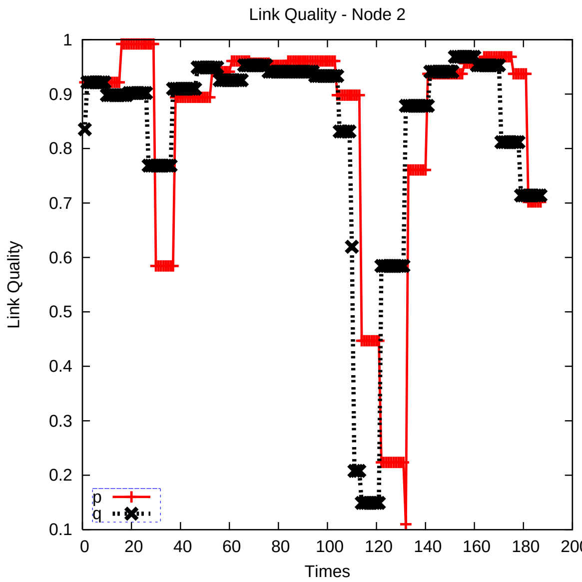


Figure 5.3: Link Quality at Node 2

testbed. The link layer dropped packets with a 25% probability. This effectively increased the expected energy consumption of the sensor nodes.

Fig. 5.8 shows the average energy consumption of ERTP and *Surge* when the network links are lossy. Similar to Fig. 5.7, we observe that ERTP is more energy-efficient than *Surge*. For a 6-hop node, ERTP reduces energy consumption by more than 33% compared to *Surge*. Similarly, we also observe that both protocols achieve 95% end-to-end reliability with the small error rate of ±3%.

5.5 Scalability

Finally, to evaluate the scalability of ERTP, we ran the protocol in a network consisting of 50 Fleck-3 nodes for 30 minutes. Fig. 5.9 shows the minimum, median, and maximum hop-counts and the delivery ratios achieved for each sensor node. We observe that the network was very dynamic with significant routing variability. As a result, packets were dropped more often in the routing maintenance phase. The longest route length was 9 hops observed from nodes 22, 24, and 39. This highlights the robustness and scalability of ERTP design and implementation. Despite the dynamics of the network topology, the average

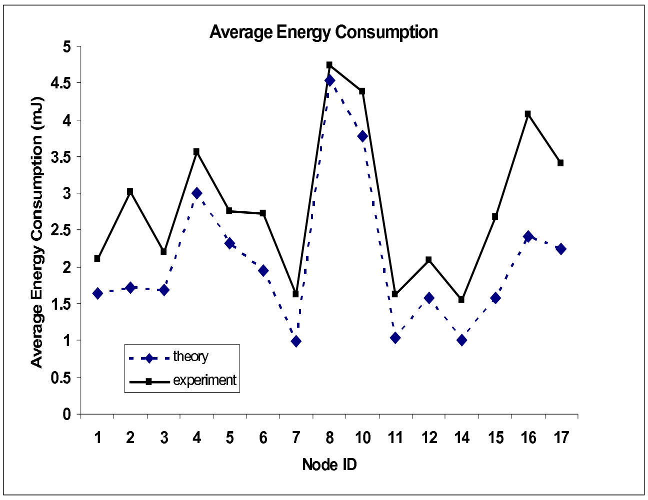


Figure 5.4: Energy Consumption

delivery ratio per node achieved by ERTP is more than 91%, which is close to the reliability requirement. High traffic levels, particularly added traffic for route maintenance, impact the performance of ERTP.

6 Conclusions

This paper has presented ERTP, an Energy-efficient and Reliable Transport Protocol for WSNs. ERTP targets the class of WSN data streaming applications. ERTP achieves the application layer end-to-end statistical reliability and energy-efficiency by dynamically controlling the maximum number of retransmissions, and exploring the wireless overhearing capability for Implicit Acknowledgment. We also proposed a distributed algorithm for retransmission timeout estimation in ERTP. The challenge in deciding the retransmission timeout is that premature timeout will cause redundant transmissions while a large timeout will cause poor capacity utilization. Our extensive simulation and experiment evaluation show that ERTP can reduce energy consumption by more than 50% when compared to current approaches for WSNs.

Acknowledgment

The authors would like to thank CSIRO for the hardware and technical supports, and Stephen Rothery for his help with protocol evaluation.

Bibliography

[1] “IEEE standard 802.15.4”, 2003.

[2] “CC2420 Product information and data sheet”, chipcon, available at: http://www.chipcon.com/.

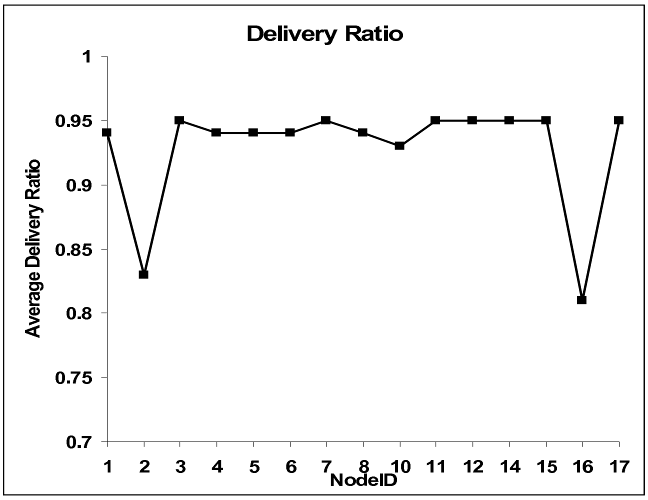


Figure 5.5: Delivery Ratios

[3] Z. Rosberg, R. Liu, L. D. Tuan, S. Jha, A. Y. Dong and J. Zic, "Energy Efficient Statistically Reliable Hybrid Transport Protocol for Sensed Data Streaming", CSIRO ICT Centre Pub. no. 07/213, June 2007. Available at: http://fairflows.com/rosberg/papers/eRDC.pdf.

[4] O. Akan and I. Akyildiz, "Event-to-Sink Reliable Transport in Wireless Sensor Networks", *IEEE/ACM Transactions on Networking*, vol. 13, no. 5, pp. 1003–1016, Oct. 2005.

[5] C-Y Wan, A. Campbell, L. Krishnamerthy, "PSFQ: A Reliable Transport Protocol for Wireless Sensor Networks", In Proceedings of the first ACM international workshop on Wireless sensor networks and applications, Atlanta, Georgia, pp. 1–11, 2002.

[6] R. Stann and J. Heidemann, "RMST: reliable data networks", In Proceedings of the First IEEE International Workshop on Sensor Network Protocols and Applications, Anchorage, Alaska, pp. 102–112, May 2003.

[7] N. Xu, S. Rangwala et al, "A Wireless Sensor Network for Structural Monitoring", In Proceedings of the ACM Conference on Embedded Networked Sensor Systems (Sensys), Baltimore, USA, 2004.

[8] S. Kim, R. Fonseca, P. Dutta, A. Tavakoli, David E. Culler, P. Levis, S. Shenker, and I. Stoica,"Flush: A Reliable Bulk Transport Protocol for Multihop Wireless Network", In Proceedings of the ACM Conference on Embedded Networked Sensor Systems (Sensys), Sydney, Australia, 2007.

[9] J. Paek, and R. Govindan, "RCRT: Rate-Controlled Reliable Transport for Wireless Sensor Networks", In Proceedings of the ACM Conference on Embedded Networked Sensor Systems (Sensys), Sydney, Australia, 2007.

[10] A. Woo, T. Tong, and D. Culler, "Taming the underlying challenges of reliable multihop routing in sensor networks", In Proceedings of the ACM

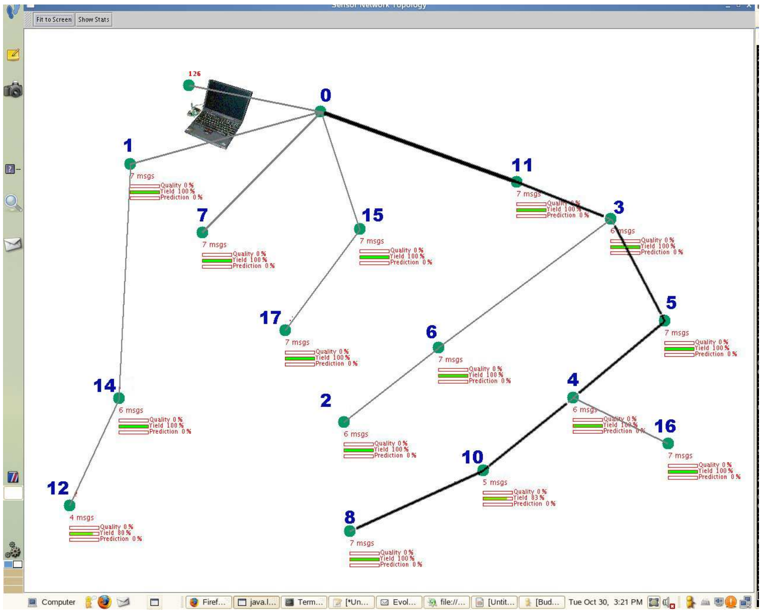


Figure 5.6: Network Topology for Comparison of ERTP and *Surge*

Conference on Embedded Networked Sensor Systems (Sensys), Los Angeles, USA, 2003.

[11] S. Fitz, A. Gonzalez-Velazquez, I. Henning and T. Khan, "Experimental investigation of wireless link layer for multi-hop oceanographic-sensor networks", *Electronics Letters*, vol. 41, no. 24, pp. 1310-1311, Nov. 2005.

[12] R. K. Ganti, P. Jayachandran, H. Luo and Tarek F. Abdelzaher, "Datalink Streaming in Wireless Sensor Networks", In Proceeding of the 4th ACM conference on Embedded Networked Sensor Systems (SenSys), Boulder, Colorado, USA Nov. 2006.

[13] Kannan Srinivasan, and Philip Levis, "RSSI Is Under-Appreciated", In Proceedings of the Third Workshop on Embedded Networked Sensors (EmNets), Cambridge, USA, 2006.

[14] T. Le Dinh, W. Hu, P. Sikka, P. Corke, L. Overs, S. Brosnan, "Design and Deployment of a Remote Robust Sensor Network: Experiences from an Outdoor Water Quality Monitoring Network", Second IEEE Workshop on Practical Issues in Building Sensor Network Applications (SenseApp 2007), Dublin, Ireland, 2007.

[15] T. Le Dinh, W. Hu, P. Sikka, P. Corke, and Sanjay Jha, "Design and Deployment of a Remote Robust Sensor Network: Experiences from an Outdoor Water Quality Monitoring Network", Transactions on Computers (in submission).

[16] P. Sikka, P. Corke, and L. Overs, "Wireless sensor devices for animal tracking and control", In Proceedings of the First Workshop on Embedded Networked Sensors (EmNets), Tampa, USA, 2004.

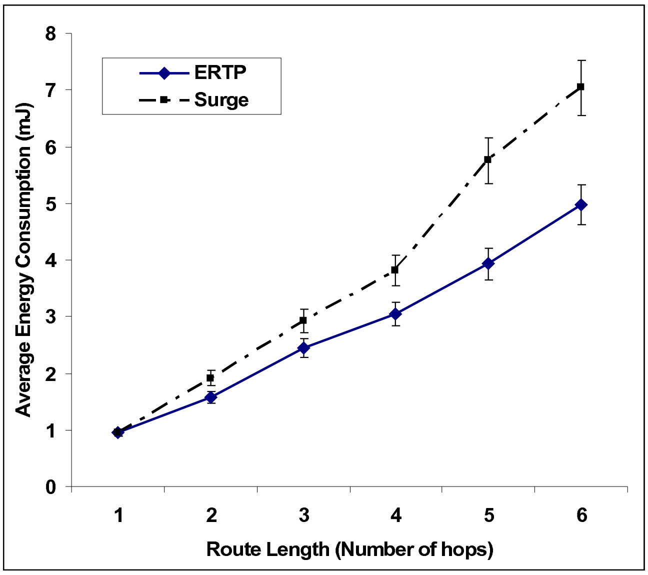


Figure 5.7: Energy Consumption

[17] Habitat monitoring on great duck island, http://www.greatduckisland.net/index.php.

[18] C. Hartung, R. Han, C. Seielstad, and S. Holbrook, "Firewxnet: a multi-tiered portable wireless system for monitoring weather conditions in wildland fire environments"', MobiSys 2006, pages 2841, New York, NY, USA, 2006.

[19] G. Werner-Allen, K. Lorincz, M. Ruiz, O. Marcillo, J. Johnson, J. Lees, and M. Welsh, "Deploying a wireless sensor network on an active volcano", Special Sensor Nets issue of IEEE Internet Computing, vol. 10, pp. 18-25, 2006.

[20] A. Willig and R. Mitschke, "Results of Bit Error Measurements with Sensor Nodes and Casuistic Consequences for Design of Energy-Efficient Error Control Schemes", In Proceedings of the 3rd European Workshop on Wireless Sensor Networks (EWSN), Zurich, Switzerland, Jan. 2006.

[21] Chalermek Intanagonwiwat, Ramesh Govindan, and Deborah Estrin, "Directed diffusion: a scalable and robust communication paradigm for sensor networks", In Proceedings of the Sixth Annual International Conference on Mobile Computing and Networking (Mobicom), Boston, Massachussetts, 2000.

[22] Charles Perkins and Pravin Bhagwat, "Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers", In Proceedings of the ACM Conference on Communications Architectures, Protocols and Applications (SIGCOMM), London, UK, 1994.

[23] Ad hoc On Demand Distance Vector (AODV) routing protocol, http://www.ietf.org/rfc/rfc3561.txt

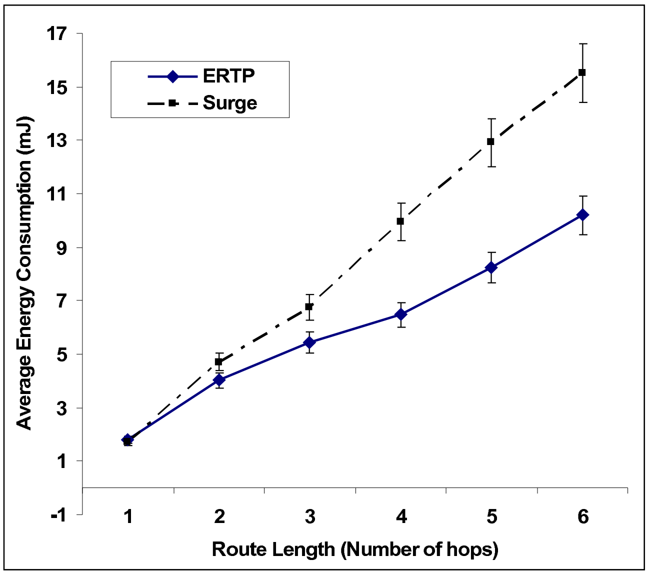


Figure 5.8: Energy Consumption for Lossy Links

[24] A. Woo and D. Cellur, "A Transmission Control Scheme for Media Access in Sensor Networks", In Proceedings of the Sixth Annual International Conference on Mobile Computing and Networking (Mobicom), Rome, Italy, 2001.

[25] H. Zhang,A. Arora,Y. Choi, and M. Gouda, "Reliable bursty convergecast in wireless sensor networks," In Proceeding of the Sixth ACM International Symposium on Mobile Ad Hoc Networking and Computing (Mobihoc), Urbana, USA, 2005.

[26] V. Jacobson, "Congestion avoidance and control", In Proceeding of SIGCOMM '88, Standford, CA, August 1988.

[27] S. Rangwala, R. Gummadi, R. Govindan, and K. Psounis, "Interference-Aware Fair Rate Control in Wireless Sensor Networks", In Proceedings of the ACM Conference on Communications Architectures, Protocols and Applications (SIGCOMM), Pisa, Italy, 2006.

[28] F. Bian, S. Rangwala, and R. Govindan, "Quasi-static Centralized Rate Allocation for Sensor Networks", In Proc. IEEE Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON), San Diego, USA, 2007.

[29] T. Dam, K. Langendoen, "An Adaptive Energy-Efficient MAC Protocol for Wireless Sensor Networks", In Proceedings of the ACM Conference on Embedded Networked Sensor Systems (Sensys), Los Angeles, USA, 2003.

[30] J. Polastre, J. Hill and D. Culler, "Versatile Low Power Media Access for Wireless Sensor Networks," In Proceedings of the ACM Conference on Embedded Networked Sensor Systems (Sensys), Baltimore, USA, 2004.

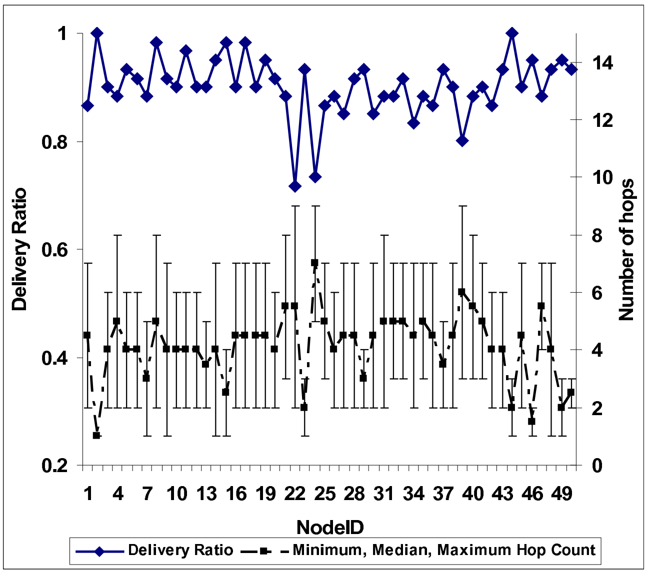


Figure 5.9: Delivery Ratio and Hop Count for 50-node network

[31] W. Ye, J. Heidemann, D. Estrin, "Medium Access Control With Coordinated Adaptive Sleeping for Wireless Sensor Networks", IEEE/ACM Transactions On Networking, Vol. 12. No. 3, June 2004.

[32] T. Zheng, S. Radhakrishnan, and V. Sarangan, "PMAC: An adaptive energy-efficient MAC protocol for Wireless Sensor Networks", In Proceedings of 19th International Parallel and Distributed Processing Symposium (IPDPS), Denver, Colorado, 2005.

[33] The Network Simulator - ns-2,http://www.isi.edu/nsnam/ns/

[34] B. Scheuermann, C. Lochert, and M. Mauve, "Implicit Hop-by-Hop Congestion Control in Wireless Multihop Networks,", Elsevier Ad Hoc Networks, Vol 6, pp. 260-286, 2008.

[35] Qing Cao, Tian He, Lei Fang, Tarek Abdelzaher, John Stankovic, and Sang Son, "Efficiency Centric Communication Model for Wireless Sensor Networks", In Proceedings of the 25th Conference on Computer Communications, Barcelona, Spain, 2006.

[36] Qing Cao, Tarek Abdelzaher, Tian He, and Robin Kravets, "Cluster-Based Forwarding for Reliable End-to-End Delivery in Wireless Sensor Networks", In Proceedings of the 26th Conference on Computer Communications, Alaska, USA, 2007.

[37] Tmote, http://www.moteiv.com/products/tmotesky.php