Reliable Load-Balancing Routing for Resource-Constrained Wireless Sensor Networks

---

A thesis submitted for the degree of

Doctor of Philosophy

by

Khaled Daabaj

Murdoch University

2012

Declaration

To the best of my knowledge, this thesis contains no material previously published by any other person except where due acknowledgment has been made.

This thesis contains no material which has been accepted for award of any other degree in any other university.

Signature:

Date: February 07, 2012

Dedicated to my parents, family and friends for their support…

Abstract

Wireless sensor networks (WSNs) are energy and resource constrained. Energy limitations make it advantageous to balance radio transmissions across multiple sensor nodes. Thus, load balanced routing is highly desirable and has motivated a significant volume of research. Multihop sensor network architecture can also provide greater coverage, but requires a highly reliable and adaptive routing scheme to accommodate frequent topology changes. Current reliability-oriented protocols degrade energy efficiency and increase network latency. This thesis develops and evaluates a novel solution to provide energy-efficient routing while enhancing packet delivery reliability. This solution, a *reliable load-balancing routing* (RLBR), makes four contributions in the area of reliability, resiliency and load balancing in support of the primary objective of network lifetime maximisation. The results are captured using real world testbeds as well as simulations. The first contribution uses sensor node emulation, at the instruction cycle level, to characterise the additional processing and computation overhead required by the routing scheme. The second contribution is based on real world testbeds which comprises two different TinyOS-enabled senor platforms under different scenarios. The third contribution extends and evaluates RLBR using large-scale simulations. It is shown that RLBR consumes less energy while reducing topology repair latency and supports various aggregation weights by redistributing packet relaying loads. It also shows a balanced energy usage and a significant lifetime gain. Finally, the forth contribution is a novel variable transmission power control scheme which is created based on the experience gained from prior practical and simulated studies. This power control scheme operates at the data link layer to dynamically reduce unnecessarily high transmission power while maintaining acceptable link reliability.

Publications

Journal Articles:

- K. Daabaj, M. Dixon, T. Koziniec and K. Lee, Reliable Routing for Low-Power Smart Space Communications, *Journal of IET Communications*, 2011.
- K. Daabaj, M. Dixon and T. Koziniec, Avoiding Routing Holes in Wireless Sensor Networks, *Journal of Lecture Notes in Engineering and Computer Science*, 2010.

Book Chapters:

- Khaled Daabaj and Shubat Ahmeda, Real-Time Cross-Layer Routing Protocol for Ad Hoc Wireless Sensor Networks, Advances in Computer Science and Engineering, Matthias Schmidt (Ed.), ISBN: 978-953-307-173-2, InTech, 2011.
- K. Daabaj, M. Dixon and T. Koziniec, LBR: Load Balancing Routing for Wireless Sensor Networks, *IAENG Transactions on Engineering Technologies*, American Institute of Physics (AIP) Conference Proceedings, Vol. 4, 1247, 2010.

Conference Papers:

- K. Daabaj, M. Dixon, T. Koziniec and P. Cole, Reliable Routing Scheme for Indoor Sensor Networks, (IEEE PIMRC'10): *21st Annual IEEE International Symposium on Personal, Indoor and Mobile Radio Communications*, 2010.
- K. Daabaj, Energy-Aware Reliability-Oriented Scheme to Deliver Time-Sensitive Data in Sensor Networks, (IEEE SOCA'10): *IEEE International Conference on Service-Oriented Computing and Applications*, 2010.
- K. Daabaj, M. Dixon, T. Koziniec and K. Lee, Trusted Routing for Resource-Constrained Wireless Sensor Networks, (IEEE/IFIP TrustCom'10): *6th IEEE/IFIP International Symposium on Trusted Computing and Communications*, 2010.

- K. Daabaj, M. Dixon and T. Koziniec, Traffic Eavesdropping Based Scheme to Deliver Time-Sensitive Data in Sensor Networks, (IEEE IPCCC'10): *29th IEEE International Performance, Computing, and communications conference*, 2010.
- K. Daabaj, M. Dixon, T. Koziniec and D. Murray, Reliable Data Delivery in Low Energy Ad Hoc Sensor Networks, (APCC 2010): *16th Asia-Pacific Conference on Communications*, 2010.
- K. Daabaj, Load-Balanced Routing Scheme for TinyOS-based Wireless Sensor Networks, (IEEE ICWITS 2010): IEEE International Conference on Wireless Information Technology and Systems, 2010.
- K. Daabaj, M. Dixon and T. Koziniec, Reliable Load-Balancing Routing Algorithm for Wireless Sensor Networks, (IEEE ICCCN'10): *19th IEEE International Conference on Computer Communications and Networks*, 2010.
- K. Daabaj, M. Dixon and T. Koziniec, Avoiding Routing Holes in Wireless Sensor Networks, (WCECS'09): *World Congress of Engineering and Computer Science*, 2009.
- K. Daabaj, M. Dixon and T. Koziniec, Experimental Study of Load Balancing Routing for Improving Lifetime in Sensor Networks, (WiCOM'09): *5th IEEE Inernational Conference of Wireless Communications, Networking and Mobile Computing*, 2009.

Posters and Demos:

- K. Daabaj, M. Dixon, T. Koziniec and K. Lee, Reliable Routing for Low-Power Smart Space Communications, (FTRA WTA'10): *6th International Symposium on Wireless sensor network Technologies and Applications for Smart Space*, 2010.
- K. Daabaj, Integrating wireless Sensor Network with Networked Control Systems, (SensorNets'09): *1st International School on Cyber-Physical and Sensor Networks*, 2009.

Table of Contents

List of Figures

List of Tables

Acknowledgements

All praises are due to Almighty God “Allah”, Who provided me with the strength and willingness to undertake this work and the opportunity to contribute a drop in the sea of knowledge.

I am most grateful to my supervisors, Mike Dixon and Terry Koziniec. Their open doors and persistent encouragement was invaluable for the completion of this research.

Finally, these acknowledgements would not complete without appreciating the unwavering support of my family including my father, my wife and children, and the memory of my mother.

Chapter 1
Introduction

1.1. Background

A Wireless Sensor Network (WSN) consists of a large number of distributed, autonomous, devices with inherently constrained resources. These devices consist of sensing, computation and wireless communication capabilities, but with limited energy source, processing speed, storage capacity, and communication bandwidth. They collaborate to perform sensing tasks in an unattended environment, collecting data from a monitored area and routing the aggregated data using a multihop approach; from node to node toward the base station (sink node) which provides the received data to the interested user. In recent years, WSNs have become a key component for many applications. This technology is being used in many areas such as battlefield communication, homeland security, environmental monitoring, habitat monitoring, agriculture, disaster relief networks, medical care, pollution sensing, industrial automation, and transportation [1].

WSNs have some unique characteristics that differentiate them from traditional wireless ad hoc networks such as high scalability and resource constraints. Furthermore, the operation of large scale sensor networks still requires solutions to numerous technical challenges that stem primarily from the constraints imposed by resource-limited sensor devices. Among these challenges, the power constraint is the most critical one because it involves not only reducing the energy consumption of a single sensor but also maximising the lifetime of an entire network. Prolonging the network lifetime can be effectively achieved by incorporating energy awareness into every stage of a sensor network's design and operation, thus enabling the system with the ability to make dynamic trade-offs among energy consumption and system performance [39,51,52,53].

A sensor node is constructed from four primary units with limited performance capabilities: embedded processor, memory, low power radio transceiver, and sensing unit. Among these aforementioned units it has been documented that the transceiver unit is the major energy consumer [1,2,3]. One fact is that the energy cost to transmit one bit is typically around 500 to 1000 times greater than a single 32-bit computation [4,5]. In WSNs, the significant resource constraints of the sensor nodes combined with the irregularity of a many-to-one traffic pattern have discouraged the use of traditional ad hoc wireless routing protocols [6].

1.2. Challenges and Key Design Issues

The emergence of wireless sensor networks (WSNs) has led to many open issues for network researchers and developers. WSNs are composed of a scalable number of wirelessly networked sensor nodes, where a sensor node is typically battery-powered. These networks are often deployed in unfriendly and unattended remote environments. This isolation makes changing the battery or replacing the failed nodes impractical and ongoing maintenance may not be feasible; thereby the progressive reduction of the available residual power needs to be considered as a crucial factor in the route selection process to control nodes' energy drain for the extension of the lifetime of the individual nodes and for the achievement of energy balancing in the entire network. Since network lifetime is a performance metric and communication is a major cost compared to computation and sensing [4,5], energy is a major design constraint for battery operated sensors and available resources are considered in favor of application objectives by taking into account different key factors of control redundancy, traffic load balancing, and data aggregation.

In addition to energy constraint, a reliable delivery of data is a classical design goal for reliability-oriented collection routing protocols for ad hoc WSNs. In low power WSNs, the unreliability of the links and the limitations of all resources bring considerable complications to the routing scheme. Even with a static topology of sensor nodes, the channel conditions may vary due to many factors such as the irregularity of radio transmission range, antenna orientations, and multipath fading effects [38].

Best-effort reliability performance can be ensured through the careful selection of error free links, quick recovery from packet losses, and avoidance of overloaded relay sensor nodes. Due to limited resources of individual sensor nodes, there is usually a trade-off between energy expending for packet transmissions and the appropriate level of reliability. Since link failures and packet losses are unavoidable, sensor networks may tolerate a certain level of reliability without significantly affecting packet delivery performance and data aggregation accuracy in favor of efficient energy consumption. However, a certain degree of reliability is needed because network reliability drops considerably as hop count increases and a single lost packet may result in loss of a large amount of aggregated data along longer hops [56,120].

In this thesis, two effective hybrid solutions are proposed. The first solution is a Reliable Load-Balancing Routing (RLBR) which is an energy-efficient data collection routing scheme proposed in chapter 3. The second is a new per link transmission power control scheme proposed in chapter 7. These two solutions jointly make a trade-off between energy, reliability, cost, and agility while improving packet delivery, maintaining low packet error ratio, minimizing unnecessary packets transmissions, and adaptively reducing control traffic in favor of high success reception ratios of representative data packets. Based on this approach, the proposed solutions can achieve moderate energy consumption and high packet delivery ratio even with the occurrence of high link failure

rates as shown from the extensive performance evaluation of these two solutions. Finally, other design issues are also considered including self-configuration, resilience to node failures, responsiveness, robustness, scalability, flexibility, loop prevention, fault tolerance, minimal overhead, and adaptability to link and topology changes.

1.3. Scope of the Thesis

Given the nature of the experimental work conducted here, this thesis will mainly focus on senor mote-oriented many-to-one routing protocols that rely on TinyOS-enabled tree-based network architecture [25,33,43,67] and have been implemented on TinyOS-compatible real sensor network platforms. TinyOS is an open source event-driven component-based operating system designed for low-power wireless sensor devices [25]. The scope of this thesis will be restricted to a representative set of mote-dominated TinyOS-enabled routing schemes [24,25] that use similar routing reliability metrics and share similar features with the RLBR scheme. It adopts a flexible approach that combines some of the advantages of the energy-aware routing protocols such as in [49,71,88] on the top of the cost-based reliability-oriented proactive and reactive collection protocols such as in [33,43,67,113,130,131].

Among many different routing protocols studied in mote-dominated WSNs, the TinyOS-based collection protocols offer simpler implementation and reliable routing [25]. Recently, variants of these collection protocols have been investigated for enhancing the data delivery in time varying communication channels. Some recent examples of TinyOS collection protocols are Minimum Number of Transmissions (MintRoute) [33], Multihop Routing Tree Protocol based on Link Quality Information (MultihopLQI) [43], and Collection Tree Protocol (CTP) [67]. Since the most critical wireless routing-related issue is the quality of the underlying links, these protocols make use of link reliability metrics in their routing schemes. While most routing protocols are formulated in a

graph-theoretical manner, it is often by no means clear which sensor nodes are connected by a wireless link [103,146]. Links reliabilities vary with time and can have relatively high packet error rates. Ad hoc sensor networks require a highly dynamic adaptive routing scheme to deal with the high rate of topology changes. Besides that, the energy consumption rate needs to be consistently distributed among sensor nodes, and efficient utilization of battery power is essential.

While the existing reliability-oriented routing protocols for wireless sensor networks (WSNs) are steadily improving for forming a reliable tree-based data gathering, it is still inferior over custom solutions concerning energy consumption [115]. Recent reliability-oriented routing protocols based on TinyOS-2.x [25] such as MultihopLQI [43] merely relies on the physical layer's link information, or CTP [67] which employs multifarious parameters that leads to complex configurations. As stated in [88,109,113], such reliability-oriented routing protocols also vary widely as a function of packet relaying workload and are not as consistent in terms of energy and load balancing across different topologies, and do not scale well for large networks. The existing reliability-oriented routing protocols are unaware of the energy status of nodes, and may divert load to low energy capacity nodes.

Since the wireless links in low-power WSNs are not stable, and the loss of packets happens frequently in communications, the Expected Transmission Count (ETX) routing metric [46] is mainly used by most reliability-oriented routing protocols, e.g., MintRoute [33] and CTP [67] to select the optimal link. However, WSNs are mainly powered by AA batteries and the resources are limited. If the reliability of communication is purely deemed as a routing cost metric to select the best quality route, a number of sensor nodes along this route will be exhausted quickly. Consequently, this number of dead sensor nodes is extremely essential to the lifetime of the entire network, if these important

nodes fail to relay packets, the network's functionality will be ruined. If only the link reliability metrics are considered in routing decisions of WSNs, it may create a route with more hops, and the high quality paths will be frequently used. This leads to a shorter lifetime of the high quality routes; consequently the entire network's lifetime will be significantly reduced. As a result, on the basis of reliability metrics, a reliable energy-efficient load-balancing routing is a key issue for maximising functional lifetime of the low-power WSNs.

1.4. Problem Definition

The ability of resource-constrained WSNs to function properly and recover quickly from lossy links or failed nodes depends largely on the performance of the wireless routing protocols employed. This thesis contributes to scholarly knowledge in the growing field of reliable and energy-efficient routing schemes for wireless sensor networks through the development of a distributed load-balancing routing protocol that ultimately aims to maintain an acceptable network performance and maximise the network lifetime.

The Reliable Load-Balancing Routing (RLBR) scheme is based on a per-hop load balancing strategy of the routing layer. It leverages recent advancements over the standard network layer components provided by the TinyOS operating system for WSNs [25]. RLBR provides a careful selection of valid parents for a routing tree toward the base station using locally overheard information that the different layers can provide. In addition, the determined routes evidently influence the lifetime of the network. Hence, the RLBR scheme goes further in that it attempts to maximise the lifetime of the network. In other words, the RLBR scheme appropriately adapts such situations of selecting optimal routes as in reliability-oriented protocols and does its best effort to be aware of energy levels of the relay sensor nodes along the routing paths. It also aims for load balancing between relay sensor nodes in terms of balanced energy usage, and

minimising energy dissipation for packet transmissions by means of adaptive beaconing and in-network aggregation of relayed data packets.

This thesis has five primary requirements to maintain a tolerable network performance and maximise network lifetime: (1) Minimising energy consumption of each individual sensor node, as malfunctioning of some critical forwarding sensor nodes due to power failure can cause significant topological changes and may require rerouting of packets and network reorganization. (2) If certain sensor nodes, usually the nodes in the path to the sink node, have much higher workload than others, then these nodes will drain off their energy rapidly and adversely impact the overall network lifetime. Hence, the workload of sensors is distributed evenly in order to achieve balanced energy usage, thereby resulting in longer system lifetime. (3) Since the communications overheads including transmission, receiving and/or overhearing are the major energy consumer during a sensor node's operation [44]; the RLBR scheme is a simple networking protocol that require minimal communication overheads for network configuration and sensed data dissemination. (4) The most critical wireless routing-related issue is the quality of the underlying links. The RLBR scheme is an adaptive routing scheme for highly dynamic WSNs that deals with the high rate of topology changes. (5) Studying network lifetime with variable transmission power is addressed to further reduce the energy dissipation for overhearing and collisions in dense WSN.

To that end, the RLBR scheme addresses the essential mechanisms to achieve the requirements for power-efficient wireless sensor networks in order to prolong network lifetime. Fault tolerance and adaptability are also accommodated to link and topology changes, while minimising communications overheads. The RLBR scheme is implemented using a testbed network as well as large-scale simulations for evaluating its performance. Experimental observations and simulation results show that RLBR has

significant improvements in packet delivery performance and energy savings. Finally, these experimental observations have led to proposing a new adaptive transmission power control scheme. This scheme aims to dynamically change the transmission power output to the lowest possible level that can provide acceptable link reliability while minimising the energy dissipation for packet transmission.

1.5. Research Methodology

The majority of the existing routing schemes for WSNs have not yet been experimentally tested in real environments and their performance evaluations have been left at simulation level. In this thesis, a realistic approach is employed to routing and the results are captured using real world testbeds as well as large-scale simulations. The obtained results show that this approach is promising in practice and more important and effective than pure simulation based approaches. This empirical research approach followed by the thesis has given a good understanding of the complex irregular behaviour of low-power wireless links in WSNs.

Since testing and debugging routing protocols on different platforms, testbeds and environments is vital for experimental validation of the routing efficiency. The added computation and communication process overhead introduced by the RLBR scheme is primarily analysed based on a cycle-accurate simulator [111] for TinyOS applications followed by experimental investigation using indoor and outdoor testbeds. In addition, intensive computer simulations are also conducted to validate the experimental results for large-scale WSNs.

Due to motes availability, the experimental testbeds range in size from 20 to 30 wireless sensor motes and comprise two different platforms of Crossbow's Mica2 [14,35] and TelosB [42,86], and two different low power listening (LPL) link layers provided by

Chipcon's CC1000 [20] and CC2420 [78] radios, i.e., B-MAC [21] and IEEE802.15.4 [87] in two disparate environments of indoor interference-free and outdoor interference-prone channels respectively. The RLBR is compared at the first stage of its design with the official updated version of TinyOS-2.x implementation of MintRoute on 20 Mica2 motes. After that, an other well-known collection protocol was also used to evaluate the RLBR scheme, which is the TinyOS-2.x implementation of MultihopLQI on 30 TelosB motes.

The reason of selecting MintRoute [33] and MultihopLQI [43] as evaluation benchmarks that they are well-established, well-experienced and most used routing layer protocols that are part of the TinyOS releases. MintRoute and MultihopLQI have been heavily used by a large number of research groups with good success. For example, MultihopLQI has been used in recent deployments [67,76], e.g., on a volcano in Ecuador [51]. Since TinyOS 1.x and 2.x have different packet scheduling and MAC layers, the recent stable version of TinyOS-2.0.2 [25] that supports different generations of wireless platforms is used all indoor and outdoor experiments.

In order to validate the experiments, large-scale network simulations were implemented in NS-2.33 simulator with the aid of Matlab using a maximum number of 100 sensor nodes deployed randomly in a sensor field of 100x100 meters square with a single stationary base station. Analytical and simulation results are also derived. These simulations are based on TinyOS-2.x with CC2420 implementation of IEEE802.15.4. Simulations-based performance comparisons are conducted between the RLBR scheme and the state-of-the-art benchmark schemes, the reliability-oriented MultihopLQI protocol [43] in order to keep the evaluation sensible. RLBR is shown to be more robust and energy efficient than the current collection layer of TinyOS2.x.

1.6. Organisation of the Thesis

This thesis is organized into 7 chapters including introduction in chapter 1 and conclusion in chapter 8. Chapter 2 presents an overview introduction to TinyOS-based reliability-oriented collection protocols and the most widely used energy-aware routing protocols. Chapter 3 presents a technical description of the proposed RLBR scheme and provides the primary analysis of its added computation overhead. The overhead analysis is performed before the routing scheme is deployed onto the target wireless sensor mote system in the subsequent chapters 4 and 5 where the RLBR scheme is experimentally evaluated based on medium-scale indoor and outdoor testbeds respectively. In chapter 6, extensive computer simulations are performed for large-scale network of randomly organised wireless sensor nodes. In chapter 7, a new adjustable transmission power control scheme is proposed by the guidance of the experience gained form the experimental work of the preceding chapters. Finally chapter 8 concludes the thesis and outlines the future work.

Chapter 2
Literature Review

2.1 Background

This chapter discusses in detail the routing metrics and routing protocols used as reference benchmarks for the proposed routing scheme. Given the nature of the experimental work conducted, this thesis will mainly focus on mote-oriented routing protocols that rely on TinyOS-enabled tree-based network architecture [25] and have been implemented on TinyOS-compatible real sensor network platforms. The discussion in this chapter will be restricted to a representative set of routing protocols [24,25] that use similar routing reliability metrics and share similar features with the proposed routing scheme. The interested reader should consult [106,114,115] for an exhaustive survey of other routing protocols that are targeted at Wireless Sensor Networks (WSNs).

In the literature, there are an enormous number of routing protocols for WSNs [106,115]. One of the simplest implementations of multihop routing is flooding broadcast packets to all connected sensor nodes in the network but it is not suitable for resource-constrained WSNs and does not assure the maximum lifetime in the network [47,115]. Alternatively, minimum cost-based routing protocols [40,130,131] (e.g., using minimum number of hops) and the reliability-oriented routing protocols [33,43,67,113] (e.g., using optimal link status) are typically used in wireless networks. Furthermore, while the majority of reliability-oriented routing protocols employ link quality metrics to define the best hop towards the base station based on link quality estimation as in [33,43,46,67,113], the traditional energy-wise routing protocols utilize the available energy to determine the most energy efficient path towards the base station as in [49,73,88]. In the subsequent paragraphs, widely-used well-established reliability and energy metrics will be discussed as well as those routing protocols that use such metrics in their routing schemes.

2.2 Reliable Data Delivery

2.2.1 Link Reliability Metrics

This section provides an introduction to the most popular link quality estimation schemes used for WSNs reliability-oriented routing protocols. As the link quality estimation scheme is the core component of the reliability-oriented routing protocols, its role is to provide the routing protocol with a valid set of neighbouring 1-hop nodes from which the best hop towards the base station can be selected from a link quality and connectivity perspective. Link quality estimation is the fundamental tool for the computation of reliability-oriented route selection metrics. Also connectivity discovery and route maintenance are carried out with the help of control packets or beacons that disseminate global state information used locally for route selection process. In the literature [59,79,80,81], the majority of the existing reliability-oriented routing protocols for WSNs rely on link quality estimators which can be classified in two categories: *hardware-based* using Channel State Information (CSI) from broadcast control traffic; and *software-based* using delivery cost estimates from unicast traffic.

***Hardware-based link quality estimators*:** These estimators are directly obtained from the radio chip module built on the wireless platform, such as the Chipcon, CC1000 [20] on Mica2 [14] or CC2420 [78] on TelosB [86], and require no computation overhead. These CSI estimators include the Link Quality Indicator (LQI), the Received Signal Strength Indicator (RSSI), and the Signal-to-Noise Ratio (SNR). The earlier common form of CSI is the *Received Signal Strength Indicator* (RSSI) which represents the amount of signal energy received by the sensor node. It can be measured by most radio transceivers. The RSSI has been recognized as a predictor of link quality of some wireless platforms [39] such as Mica2 CC1000 radio [20]. It has been shown in [59,65] that the RSSI correlates well with the packet reception ratio if RSSI level is higher than the sensitivity threshold of approximately -87dBm. However, RSSI does not cope well with asymmetric links. In

addition, using RSSI independently may not be an adequate indicator of the link quality for reliable connectivity; even with high RSSI there might be severe interference. The link quality needs to be computed based on bit or packet error. Therefore, for better understanding of low-power wireless link reliability, a newer CSI estimator specific on the IEEE 802.15.4 stack, namely *Link Quality Indicator* (LQI) [87], is used with RSSI for improved link quality estimations. IEEE 802.15.4 is a standard which specifies the physical layer and Media Access Control (MAC) for Low-Rate Wireless Personal Area Networks (LR-WPANs) [87]. LQI is a hardware-based link reliability metric introduced by 802.15.4 [87], which measures the error in the incoming modulation of successfully received packets which pass the Cyclic Redundancy Check (CRC) sums. LQI can be measured by Chipcon's CC2420 radio chip [78] on TelosB motes [86]. It is actually Chip Correlation Indicator (CCI) and its values are related to the chip error rate. Every received packet must be stamped with LQI value as stated in IEEE802.15.4. This value indicates the quality of the link at the time of packet reception. Recently, a new Resource-Aware and Link Quality based (RLQ) routing metric is presented in [153] to address energy limitations, link quality variations, and node heterogeneities in WSNs. The RLQ metric is a combined link cost metric, which is based on both energy efficiency and link quality statistics. This metric was proposed to adapt to varying wireless channel conditions while exploiting the heterogeneous capabilities of WSNs [153]. However, as observed in [59,79,80,81], hardware-based estimators are inaccurate as the link quality readings are calculated over 8 symbol periods of a received packet and does not consider the whole packet [87]. Also they are merely measured for successfully received packets at the receiver node [20,78]. Therefore, when a radio link suffers from excessive packet losses, the transmission performance is overestimated, by not considering the information of lost packets. These drawbacks can be resolved using software-based estimators.

***Software-based link quality estimators*:** These estimators are either based on reception ratio or the average number of packet transmissions/retransmissions before its successful reception [79]. The Packet Reception Ratio (*PRR*) and the Acquitted Reception Ratio (*ARR*) are based on the reception ratio. While *PRR* is calculated at the receiver, *ARR* is performed at the transmitter. These link quality estimators are simple and have been widely used in routing protocols for WSNs [39,65]. The Window Mean with Exponentially Weighted Moving Average (*WMEWMA*) [44], the Kalman filter-based link quality estimator [85] and the Packet Success Probability (PSP) [8] are based on PRR approximation after a successful reception. Conversely, the Required Number of Packet transmissions (*RNP*) estimator [82] is based on the average number of packet transmissions/retransmissions that is required before its successful reception. According to [82], *RNP* is better than *PRR* for characterizing the link quality because *PRR* provides a coarse-grain estimation of the link quality since it does not take into account the underlying distribution of losses. In addition, the Link Inefficiency metric (*LI*) [81], Expected Transmission Count (*ETX*) [46], and four-Bit [84] are based on *RNP* approximation before a successful reception. The following paragraphs discuss the most-widely software-based link quality estimators used for TinyOS-based routing protocols.

Packet Reception Ratio (*PRR*):

PRR metric can be computed at the receiver by taking the average of the ratio of the number of successfully received packets to the number of transmitted packets as in Equation 2.1 [39,65]. The number of lost packets is determined using the packets sequence number. The *PRR* is based on passive monitoring, which means that useful statistical data is collected from received/sent data packets over that link.

$$PRR = \frac{Number\ of\ successfully\ received\ packets}{Number\ of\ sent\ packets} \times 100 \qquad (2.1)$$

Window Mean with Exponentially Weighted Moving Average (*WMEWMA*):

WMEWMA [44] is a filter-based estimator that approximates the *PRR* estimator. This estimator is based on passive monitoring and is updated at the receiver for each time window of received packets. Although WMEWMA is the most stable estimator, it has the worst performance in supporting reliable data collection and multihop routing in the unreliable links of WSNs [79]. In the literature, some experiments [76,77] demonstrate that, instead of using link quality estimation of each route message or beacon packet individually as in MultihopLQI [43], the average link quality estimation is better to reflect packet delivery ratio and to apply the WMEWMA filter [44] on link quality values in each time frame window (w) to calculate the averaged link quality as in Equation 2.2, where α is the history control factor ranges between 0 and 1. It controls the effect of the previously estimated value on the new one.

$$WMEWMA(\alpha, w) = \alpha \times WMEWMA + (1+\alpha) \times PRR \qquad (2.2)$$

Required Number of Packet Transmissions (*RNP*):

RNP [82] calculates the average number of packet transmissions/retransmissions required before a successful reception. Based on passive monitoring, this metric is evaluated at the sender for each window transmitted and retransmitted packets, as in Equation 2.3. The number of successfully received packets is determined by the sender as the number of acknowledged packets.

$$RNP = \frac{Number\ of\ transmitted an\ and\ retransmitted\ \ packets}{Number\ of\ successfully\ received\ \ packets} - 1 \qquad (2.3)$$

The main drawback of the aforementioned link quality estimators that they are passive unidirectional estimators and unaware of the link asymmetry problem. They provide a passive estimate of the quality of the unidirectional link. The subsequent paragraphs will

address the active bidirectional link quality estimators used in TinyOS-based routing protocols, namely, ETX [46] and Four-bit [84].

Expected Transmission Count or *Expected Number of Transmissions* (*ETX*):

ETX metric [46] is a receiver-initiated bidirectional estimator that approximates *RNP values*. It uses active monitoring, which means that each node explicitly broadcasts probe packets to collect statistical information to consider the link asymmetry. The *ETX* metric provides bidirectional link quality estimations by estimating both the uplink quality from the sender to the receiver (*PRR* of received probe packets at the receiver node) and the downlink quality from the receiver to the sender (*PRR* of sent probe packets at the sender node) as expressed in Equation 2.4.

$$ETX = \frac{1}{PRR_{forward} \times PRR_{backward}} \tag{2.4}$$

By means of delivery cost estimates, the ETX link metric is proposed in [46]; ETX estimates the total number of transmissions needed to get a packet across a link, and uses the route with the minimum transmissions. ETX has been shown to be very robust, especially on top of an Automatic Repeat Request (ARQ) scheme [60] which strengthens low quality links. However, using the ARQ scheme in the link layer, the child sensor node will retransmit the unacknowledged packet and degrade the network throughput. The traditional way of estimating the ETX relies on link symmetry assumption, which is not accurate in typical WSN deployments where packets losses on the direct and reverse channel are not correlated even though sensor nodes are static [58]. For example, the observations in [120,121,122] show that MintRoute [33] and MultihopLQI [43] experience the asymmetric link problem inappropriately as child sensor nodes might not get their packets acknowledged from their current parents even though the maximum number of successive transmission failure is reached. To solve this problem, the

observations in [33,82,62,63,64] states that it is essential to use link layer acknowledgments to evaluate the ETX metric.

Four-bit:

Four-bit metric [84] is a sender-initiated bidirectional estimator that approximated *PRR* and *RNP*. It also uses both passive and active monitoring. During active monitoring, nodes periodically broadcast probe packets to estimate the quality of the unidirectional link from the receiver to the sender using *PRR*. During passive monitoring, the sender computes *RNP* based on transmitted and retransmitted data packets to the receiver to estimate the quality of the unidirectional link from sender to receiver. However, four-bit estimator heavily depends on the tuning of its parameters, e.g., using high beaconing rate to improve the estimation of the upstream link quality.

Weighted Round Robin Forwarding (*WRRF*):

WRRF metric is presented in [154] where the authors present a load-balancing congestion avoidance protocol that includes *source count* based hierarchical medium access control (HMAC) for reliable event detection. This scheme aims to avoid packet drop due to buffer overflow under bursty traffic conditions.

Finally, although all these studies provide a valuable contribution in WSNs routing, the problems of load balancing routing mechanisms jointly based on energy metrics and link quality are yet to be addressed. Based on observations in previous studies such as [59,79,81,82], the existing link quality estimators are limited in the sense that they provide only partial views of the real quality of the link. Each estimator computes only one metric, with an exception of four-bit [84], which combines *PRR* and packet retransmissions-based estimation techniques. However, in order to better estimate link quality, it is essential to employ several metrics simultaneously and exploit the broadcast

nature of wireless links [82,84]. This results in an estimator that is a function of several metrics, thus giving a more meaningful state of the link quality level. For example, the combination of RSSI, LQI, and ETX would be more reliable and accurate for describing the link status [58,60,79].

2.2.2 Mote-Dominated TinyOS-Enabled Reliability-Oriented Routing

In mote-specific WSNs, MultihopRouter [40], MintRoute (Minimum Number of Transmissions) [33,52], Hyper [107], RBC [108], MultihopLQI [43], Pull Collection Protocol (PCP) [69,70] and Collection Tree Protocol (CTP) [66,67] are multihop, reliability-oriented, routing protocols and successive evolutions of TinyOS-based collection tree routing layers. These protocols are widely used in TinyOS-based WSNs [25], where all nodes construct a multi-hop routing tree to a centralized root (e.g., gateway or base station). In typical collection tree routing protocols, the final destination of each data packet is the root of the tree (i.e., the base station). The base station advertises its existence by periodically broadcasting route advertisement messages (e.g., beacons). Sensor nodes receiving a route advertisement message set the sender of that message as the next hop “parent” towards the base station. These sensor nodes also periodically broadcast these route advertisements towards their reachable neighbours in the vicinity. As a result, the entire network can be identified and connected by using a limited form of *flooding*. The subsequent paragraphs will explain in details the most widely-used mote-oriented TinyOS-based collection tree multihop routing protocols, namely, MultihopRouter [40], MintRoute [33], MultihopLQI [43], and CTP [67].

MultihopRouter [40]

MultihopRouter (also known as TinyOS-1.1 Multihop Routing) [40] is one of the earliest and simplest cost-based routing protocols for TinyOS. It is included with TinyOS-1.1 and later. It is a *proactive shortest-path-first* algorithm mainly based on hop-count distance information with a single base station, which means it gives priority to routes that are the least number of hops to its destination or the base station by building the *shortest path tree* (SPT). The MultihopRouter routing protocol is a part of the Extensible Sensing System (ESS) deployment [41,45] and combines a simplified Directed Diffusion [47,48] and MintRoute [33,52] routing protocols. In MultihopRouter, a senor node broadcasts a packet every five seconds, including route information: hop-count to the base station and its node *id*. Once the neighbour sensor node gets the packets, it chooses a parent based on the minimal hop count and link cost. The link cost is computed using a network link quality estimator. MultihopRouter uses two-way link estimation, which is an estimation based on both the quality of the communication coming from the sensor node and the communication going towards it. These are referred to as receive quality and send quality. The receive quality is calculated using an Exponentially Weighted Moving Average (EWMA) filter which uses the fraction of successfully received packets [44]. The receive quality values are sent to the neighbours who use them as their own send quality. These link quality estimates are only used as a tiebreaker for routes with the same number of hops. For example, a route directly to a base station with very high packet loss, which is a likely case in a real WSN, will be chosen in favour of a two hop route with no packet loss at all. This is the major drawback of MultihopRouter's route selection process.

In the *MultihopRouter* routing protocol, route selection is both event-based and triggered upon reception of a route advertisement packet. In both cases however, paths are maintained by fixed periodic transmission of route advertisement packets with a

beaconing rate of a packet per five seconds sent by every sensor node in the network. However, in any *proactive* routing protocol, there is a fundamental trade-off between the control overhead of the periodic maintenance packets and the rate of adaptation to topological changes. MultihopRouter supports multiple base stations which announce their presence to the rest of the sensor nodes by periodically transmitting broadcast packets. Those packets get flooded to the entire network and then the sensor nodes themselves decide to which destination they want to send their data. However, the reliability-focused observations in [33,39,57] indicate that cost-based routing schemes that are based on hop-count distance as in MultihopRouter protocol might not be viable for low-power WSNs as they do not apply a metric that considers energy balancing in their routing scheme.

MintRoute [33,52]

MintRoute "Minimum Number of Transmissions Routing" [33,52] is a distributed, proactive, tree collection routing protocol that has been used extensively as its part of the official TinyOS1.1 distribution (i.e., an adaptation of the early TinyOS Multihop routing protocol [40]) and as such is considered by the research community to be the defacto standard mote routing protocol that supports Mica2 motes [24,25,27,32,52]. In the literature, MintRoute has been successfully used in various experiments by researchers from UC Berkeley, which seemed quite simple and would use a small data memory footprint using memory efficient time averaged link estimator [33,44]. The official MintRoute implementation supports only a single base station "Many-to-One" scenario. Sensor nodes can only connect to their pre-determined base station which is defined at compile-time, however multiple base station modifications exist as explained in [50].

MintRoute is a single-path routing protocol which heavily relies on link estimations for path selection and uses point-to-point transmissions of packets through the network [33].

MintRoute, as well as the majority of other sensor network routing protocols, performs a distributed distance-vector-like routing algorithm to determine routes back from the source sensor nodes to the base station by exchanging periodically the route setup messages (i.e., beacon packets) among neighbour sensor nodes, and employs link estimation and neighbour tables to dynamically discover the topology of the entire WSN. With distance-vector routing algorithms, each sensor node independently selects a next hop from its neighbours that reduces its distance towards the base station. Being proactive, MintRoute performs a periodic parent selection process to actively maintain the established path entries in the routing table. Parent sensor nodes are chosen by evaluating the costs of routing data through different neighbours. When a parent sensor node is selected, it also broadcasts a route advertisement packet, as it currently has a reliable path towards the base station. MintRoute uses a base station-initiated periodic beacon, called a route advertisement, to construct and maintain the spanning-tree. Similar to shortest-path-routing tree protocols, route advertisements originate at the base station and are forwarded by every sensor node that receives them in order to cover and build the entire network. MintRoute employs neighbourhood routing tables for the parent selection process based on Minimum Transmissions (MT) [52] weights on links as an effective cost metric to minimize the total number of transmissions and/or retransmissions. This works with a link estimator based on the Expected Number of Transmissions (ETX) [46] as a cost metric along the routing path. A neighbour quality estimator running at each sensor node provides the required link quality to the MT metric and used to avoid dynamic asymmetric links. MintRoute applies a sequence number for each packet to detect packet loss and thus evaluate link quality; this sequence is shared by both control and data packets [33,52]. Backward links are also important for acknowledgments. For each link, the nonlinear MT routing cost metric is estimated by Equation 2.5 [33,52].

$$MT\ Cost = \left(\frac{1}{Forward\ Link\ Quality} \right) \times \left(\frac{1}{Backward\ Link\ Quality} \right) \qquad (2.5)$$

MintRoute can also employ *packet reception ratio* (PRR) estimates based on sequence numbers. MintRoute adopts a neighbourhood management policy based on the frequency algorithm as a link blacklisting mechanism [53], thereby allowing a given sensor node to only maintain a subset of its neighbours with reliable links. However, the blacklisting mechanism can cause partitioning if the thresholds are not set properly [60,113]. In dense sensor networks, MintRoute prevents the neighbourhood routing table from growing beyond a given threshold size by using a memory efficient time averaged link estimator, which is represented in "Window Mean with Exponential Weighted Moving Average" (WMEWMA) filter [44] as expressed in Equation 2.6, where t is a time window, and α is a tuning parameter or filter constant.

$$WMEWMA(t, \alpha) = \frac{(Packets\ \ received\ \ in\ \ t)}{Max\ (Packets\ \exp ected\ \ in\ t, Packets\ \ received\ \ in\ t)} \qquad (2.6)$$

MintRoute mainly employs link-quality estimation [44] using Received Signal Strength Indicator (RSSI) to evaluate the link qualities from both directions between sensor nodes. Since the data rate in WSNs is typically low, route messages do not need to be exchanged frequently and the rate of route message exchanges is very low as in MintRoute. In terms of energy dissipation cost, this helps MintRoute to reduce its energy consumption in low traffic rates. However, MintRoute is more expensive at high traffic rates. MintRoute protocol improperly assumes that intermediate links are stable with independent packet losses, and uses this assumption to derive the necessary sampling window for an inaccurate link quality estimations [33,67]. In addition, MintRoute takes a long time to convey the topological changes to the whole network (i.e., due to node failure or damage). During this period, many packets are routed through optimal paths. This results in additional energy consumption and thus offsets the benefit of energy balancing in such reliability-based routing scheme for propagating the topological

changes. Although MintRoute protocol balances the traffic load with occasional switches of nodes' parents which is a direct consequence of the Minimum Transmissions (MT) metric [52], MintRoute protocol does not explicitly apply a metric that considers transmissions balancing in its routing scheme [56,120].

MultihopLQI [43,51]

In the literature, newer tree-based collection protocols have been designed based on MintRoute to support IEEE802.15.4-compliant sensor motes. One of the state-of-the art collection protocols in TinyOS-2.x [25] is MultihopLQI [43,51], which has been used in recent WSN deployments [51,76]. MultihopLQI is a variant of MintRoute [33] that uses the IEEE802.15.4-compliant Link Quality Indicator (LQI) metric to select routing paths. While the original MintRoute protocol was designed for Mica1 and Mica2 motes as a part of the official TinyOS distribution, MultihopLQI [43] (the newer version of MintRoute [33]) was designed to support CC2420-(802.15.4)-based motes like MicaZ and TelosB [78]. There are two major differences between MultihopLQI and MintRoute. Firstly, MultihopLQI uses LQI values provided by the radio hardware instead of link estimator using RSSI to estimate link quality to its neighbours as in MintRoute. Secondly, MultihopLQI is based on MintRoute but without routing tables. In other words, MultihopLQI maintains only a state for one parent node at a time, neither routing tables nor blacklisting are used as in MintRoute, and a new parent is adopted if it advertises a lower cost than the current parent. Link Quality Information is used as a link metric with Channel State Information (CSI) to obtain the cost of a given route. MultihopLQI avoids routing tables by only keeping state for the best parent at a given time; this measure significantly reduces memory usage and control overhead. The size of each control packet size is 12 bytes and the data packets have eight bytes header. MultihopLQI uses a constant rate for transmitting beacons. The beaconing rate of the current implementation of MultihopLQI is fixed at one beacon every 32 seconds. Thus,

similar to MultihopRouter protocol [40], the energy dissipation cost of MultihopLQI protocol is a function of the beaconing rate.

MultihopLQI is a distance-vector routing protocol and employs LQI values as the routing metric to estimate link quality between a senor node and its neighbours. In CC2420 radio, the LQI measurement is a characterization of the strength and quality of a received packet. However, real-world sensor network experiments [51,76] state that, LQI fluctuates over the time where MultihopLQI uses only LQI of each beacon individually instead of taking the average LQI [76]. MultihopLQI only uses physical layer information provided by CC2420 radio and does not consider other reliability metrics to estimate link quality. MultihopLQI, as a distance vector routing protocol, intrinsically has the count-to-infinity problem. Although the use of CSI broadcast beacons is crucial to link estimation, MultihopLQI reliance on one form of CSI (i.e., LQI metric) is the main reason behind its inferior performance [51,76].

***Collection Tree Protocol* (CTP)** [66,67,68]

The most recent collection tree-based routing protocol in TinyOS-2.x [25] is the implementation of the Collection Tree Protocol (CTP) [66,67,68] (also known as CTP Noe [67]). CTP is the updated version of MintRoute architecture designed to support tree-based distributed data collection in homogeneous WSNs with multiple base stations [66,67]. CTP employs a data-path topology validation for loop detection and performs an adaptive beaconing with transmission deferrals in case of parent sensor node congestion [67]. CTP uses ETX [46] as a routing cost metric of the single-hop sender and performs an anycast routing to a single or a small number of designated base stations. CTP chooses the route with the lowest ETX. While ETX value at the tree root is zero, ETX at any sensor node in the tree is given in Equation 2.7.

$$ETX_{node} = ETX_{parent} + ETX_{link-to-parent} \quad (2.7)$$

Routing loops occur in CTP when a sensor node chooses a new route with a higher ETX than its old one. The routing loop problem was addressed by not considering routes with an ETX higher than a reasonable constant [68]. Packet duplication occurs in CTP when a sensor node receives a packet successfully but the acknowledgment is not received by the sender and the sender retransmits the packet and the receiver receives it a second time. Packet duplication problem is solved by using data frames stamped with a frame time that the frame has lived and incremented by the routing layer. If a sensor node generates a data frame, its time is set to zero. When a sensor node receives a data frame, the frame time is incremented [67,68]. CTP has also a number of added features such as link estimation from both control and data traffic, and employing routing tables for a quick parent selection process. Link quality estimation based on sequence numbers of control packets and data acknowledgement render CTP platform independent which allows CTP to adapt the amount of control traffic to the fluctuations in network connectivity. CTP can achieve more than 97% reliability with a non-duty cycled MAC [67]. However, CTP's reliability is based on miscellaneous parameters which make it complex to configure. It also varies widely as a function of existing load and is not as consistent as load-balancing across different topologies, and does not scale well for large networks [58,77,113]. Similar to other reliability-oriented MAC and network layers of TinyOS distributions, TinyOS-2.x implementation of CTP is inferior with respect to energy consumption [58,88,109]. Compared to MultihopLQI, CTP achieves a good throughput under light load traffic but it does not scale well under heavy load [58,77,109,113].

2.3 Energy-Balancing for Network Lifetime Maximization

2.3.1 Energy Cost Metrics

Some of the early works on energy efficient unicast routing has been done in [71,90,91], which is a modified Dijkstra's shortest path algorithm [103] that obtain routes with minimal total transmission power. In fact, there are various aspects how energy or power efficiency can be considered of in a routing context [61]. Figure 2.1 shows an example route scenario for communication between nodes A and H including energy costs per packet for each link and available battery capacity per node.

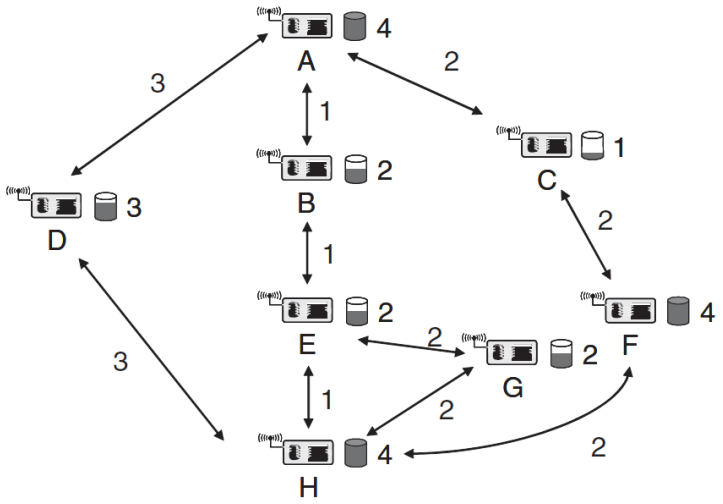


Figure 2.1 Various routes for communication between nodes A and H, labelled with energy costs per packet for each link and available battery capacity for each node [106]

One solution to minimise the total dissipated energy is to use the *adjustable transmission power-based routing scheme*, e.g., *Minimum Total Transmission Power Routing* (MTPR) [92] by considering the situation of several sensor nodes transmitting to farther sensor nodes or directly to the base station. MTPR Calculates the total transmission power for all routes between source and destination. It selects the route with the minimum total transmission power among all routes. The goal is to find an assignment of transmission power values for each transmitter, given that all transmissions are successful and that the sum of all power values is minimized. However, the MTPR

scheme might cause undesirable interference and does not reflect directly on the lifetime of each sensor node as if the selected routes are via a specific sensor node, the battery of this node will be exhausted quickly.

An alternative straightforward solution is *minimising the total dissipated energy per packet*, by selecting an energy efficient route. Minimising energy per packet is to consider the total energy required to transport a packet over a multihop path from source to destination including all overheads. This energy cost metric has been included in many standard routing algorithms. However, this can lead to widely variant energy consumption on sensor nodes.

The task of WSN is not only to transport data, but to monitor and possibly control unattended environments. Hence, *maximizing network lifetime* is essential in order to enable the network to fulfil its duty for as long as possible. In the literature, many definitions of network lifetime exist: time until the first node fails [61]; time until there is a spot that is not covered by the network (loss of coverage, a useful metric only for redundantly deployed networks) [61]; or time until network partition (when there are two nodes that can no longer communicate with each other) [61,91,93]. While these aspects are related, they require different solutions. However, maximizing the time to network partition is reported as NP-complete in [91]. it is not immediately obvious how to reach this goal using observable parameters of an actual network. As the finite energy supply in sensor nodes' batteries is the limiting factor to network lifetime, it stands to reason that information about battery status must be included in routing decisions to maximise network lifetime.

One of the routing schemes that consider the available battery energy is the *Maximum Total Available Battery Capacity* by choosing the route where the total of the available

battery capacities (reciprocal battery levels) along a given route is maximized, without using unnecessary detours. Alternatively, instead of looking directly at the sum of the available battery capacities along a given route, the *Minimum Battery Cost Routing* (MBCR) scheme considers the reluctance of a sensor node to route traffic [71,91]. This unwillingness increases as its battery is drained; for example, reluctance or routing cost can be measured as the reciprocal of the battery capacity. Then, the cost of a route is the sum of this reciprocals and the rule is to pick that route with the smallest cost. Since the reciprocal function assigns high costs to sensor nodes with low battery capacity under predefined threshold, this will automatically shift traffic away from routes with sensor nodes about to run out of energy. However, because only summation of values of battery cost functions is considered in MBCR, a route containing sensor nodes with low remaining battery capacity may still be considered. To make sure that no sensor node will be overused and to avoid sensor nodes with low energy battery resources along the selected route, the objective function of the MBCR algorithm is modified in *Min–Max Battery Cost Routing* (MMBCR) scheme [71,91] by selecting the battery cost which is maximum among all nodes of route, instead of summing the battery cost function of all sensor nodes of the individual routes. The largest reciprocal battery level of all sensor nodes along a route is used as cost metric for this route. Then, again the route with the smallest cost is selected. In this sense, the most efficient route is chosen by minimizing over a maximum. The same effect is achieved by using the smallest battery level along a route and then maximising over these route values as in [83]. This is then a maximum/minimum formulation of the problem. Using MMBCR, the battery of each sensor node will be used more fairly than in MBCR scheme. Nevertheless, MMBCR doesn’t guarantee that minimum total transmission path will be selected as it can consume more power to transmit user traffic and reduces lifetime of all sensor nodes.

Maximising lifetime of each sensor node and use the battery fairly cannot be achieved simultaneously by any of the abovementioned schemes. As a result, a further modification is to choose the shortest path if all sensor nodes in all possible routes have sufficient battery capacity as in *Conditional Max-Min Battery Capacity Routing* (CMMBCR) [71]. CMMBCR uses battery capacity instead of a cost function as a route selection metric. If there are routes along which all sensor nodes have available battery power levels exceeding a given power level threshold, CMMBCR selects the route that requires the lowest energy per bit. If there is no such route, then it picks that route which maximises the minimum battery level. However, It is by no means obvious that this energy metric in fact maximises network lifetime; other factors like deployment pattern, event patterns, and battery discharge/recharge mechanisms also have to be considered

The routing metrics discussed so far try to construct a single energy-efficient route between a source sensor node and the base station, typically by giving a sensible meaning to the "cost" of a link [71,91]. These costs typically try to balance energy required for communication across a link against the battery capacity of the sensor nodes involved. Focusing on choosing the most energy efficient route limits the opportunities for making such trade-offs and render a selected single-path vulnerable to node or link failure due to energy depletion. As a result, extending the single path routing scheme to a multipath routing scheme and trying to balance energy consumption across multiple routes is therefore a worthwhile solution by constructing several routes between a given sender and receiver [129]. A multiple-path routing scheme provides redundancy in that they can serve as "hot standbys" to quickly switch to when a sensor node or a link on a primary route fails [150,151].

Most of the existing multipath routing protocols based on the classic on-demand single-path routing schemes as in [79,99]. The multipath routing protocols aim to find a number

of disjointed paths that do not have either links or nodes in common. In the literature, applying multipath routing to wireless networks, both general ad hoc and sensor networks, is a well-studied problem as in [73,83,95,96,97,98,99,100,101,102,,129,132, 150,151]. Once the paths have been established by the multipath routing protocol, the forwarding phase can then dynamically choose a path or set of paths to transmit a packet. This can increase the robustness of the forwarding process in the case of link or sensor node failures. Other objectives of multipath routing protocols are to provide fault tolerance, reliable communication, and to ensure load balancing as well as to improve quality of service (QoS) of WSNs [112,156]. Sensor node's residual energy has been considered in some of the existing multipath routing schemes such as in [73,150] where the energy efficient multipath routing protocol is formulated as a linear programming problem with an objective to maximise the time until the first sensor node runs out of energy as the source sensor nodes are assumed to be transmitting data packets at a constant rate. Although multipath routing can positively influence energy dissipation in WSNs, there is the possibility of an increase in total overhead, route maintenance, and packet disorder, which negatively influences some QoS services [101]. Also the nature of the shared radio medium impacts the proper running of multipath techniques since the paths should be disjointed that do not have either links or nodes in common, which makes the mechanisms, employed much more complex in comparison to single-path routing [101,129,150].

2.3.2 Many-to-One In-Network Data Aggregation

When the relaying of aggregated data packets is taken as a routing metric it has a direct impact on the distribution of the traffic flows, and in particular on energy consumption. Therefore, a great deal of research focuses on reducing the total number of messages sent throughout the network by using in-network data aggregation [36,94]. In data

aggregation, data packets from neighbouring sensor nodes are combined into a single packet on their way towards the base station as the energy cost of starting up packet transmissions individually can be significant. However, using this approach independently is not enough when the relaying sensor nodes are overloaded with the traffic from downstream source sensor nodes to deliver their data to the base station. To maximize the lifetime, the energy consumption rate of each relay sensor node must be evenly distributed by means of energy and load balancing. In addition, the overall transmission power for each connection request must be minimized by means of data aggregation. This hasn't been achieved concurrently using the existing routing layers [36,58,77,88,94,109,113]. Hence, supporting energy-efficient routing in a wireless sensor network is a crucial optimisation task and its solution draws upon insights from many different disciplines. For WSNs, these mechanisms are important but they are complemented by mechanism that deal with the collection and dissemination of data directly and multihop routing to bring data from multiple sources through *convergecast* (reversed-multicast) tree [88] by means of *collaborative in-network information processing* (i.e., *data aggregation*) to reduce routed data volume. *Data aggregation* is a way to combine data from different sources. The simplest data aggregation function is *duplicate suppression* - if multiple sources all send the same data and only one is required, these data items can be aggregated together. Other aggregation functions could be *min*, *max*, *count*, *average*, or even a user-defined function with multiple inputs, as long as the function is *decomposable*. A function is decomposable if it can be computed by another function, for example, $f(v_1, v_2, ..., v_n) = g[f(v_1, ..., v_k), f(v_{k+1}, ...v_n)]$. Using decomposable functions, the value of the aggregate function can be computed for disjoint subsets, and these values can be used to compute the aggregate of the whole set, using the merging function g. In this approach, after the queries are distributed across the

network, aggregate results are sent back to the aggregator sensor node over a spanning tree, with each sensor combining its own data with results received from its child nodes.

Gathering data offers an additional energy-efficiency optimization. Often, it is not necessary that all data from all sensor nodes arrives at the base station. Frequently, an aggregate of the data, e.g., maximum, average, or minimum, is to be computed anyway. In such a case, performing these aggregation operations within the network is a viable option to reduce the amount of data that has to be transported; thereby minimising the energy required for data dissemination. If the data sources are all nearby, for example, when they all observe an event at a certain place, and they are located far away from the base station and their routes to the base station merge early on, the expected benefits of aggregation are large [94]. Recent studies shows that whether the source sensor nodes are clustered near each other or located randomly, significant energy gains are possible with data aggregation [58,77,88,94,109,113]. These benefits are greatest when the node density is high, and when the data sources are located relatively close to each other and far from the base station. The actual benefits of such aggregation depend on other factors affecting performance, such as node density, the communication network topology, and the placement of data sources relative to the base station. When all data sources are spread out, the routes to the base station do not intersect, and there is little if any opportunity to aggregate data at some intermediate nodes [94]. Additionally, the latency caused by data aggregation could be significant and should be taken into consideration during the design stage of parent selection process. WSN routing protocols should permit real-time applications to affect desired tradeoffs between latency and energy [88].

Due to delay and computation issues, the intermediate sensor node, at which the data packet is aggregated, has to decide how long to wait for data from each of its children in a *convergecast* tree. In the simplest case, a sensor node recognises its children nodes

during the routing tree formation process while waiting for responses from them. This can, however, take a long time because of fluctuations in the link reliability with ensuing high error rates, temporary node failures, or simply because of a very imbalanced tree. Waiting a long time will result in more data entering the computation of the aggregate and thus to higher accuracy but it will also increase delay and, potentially, energy consumption because of the required holding time at the relaying sensor node [88].

If there are no failures within the sensor network, the in-network aggregation technique is both effective and energy-efficient. However, a high loss ratio is unavoidable on low-power wireless links, and this effect accumulates quickly as the number of hops increases which makes the packet loss ratio worse [88,94,109,113]. For example, when the loss rate is 3% per hop, the loss rate after 10 hops becomes 30%. The effect becomes more severe when aggregation is used because a single packet loss results in the loss of all aggregated data in that packet. Thus a reliable transmission scheme is critical for efficient *data aggregation* [58,77,88,94,109,113].

Finally, the existing energy metrics are trying to fulfil different objectives. While these objectives are fairly easy to formulate individually, it is not trivial to implement them effectively in a distributed routing protocol that wisely balances the overhead necessary to collect routing information with the performance gained by an intelligent and reliable routing scheme.

2.3.3 Energy-Efficient Load-Balancing Routing

There are many recent studies that investigate the idea of using energy and load balancing metrics in routing protocols in order to maximize network lifetime. However, most of these studies focus on Mobile Ad Hoc Networks (MANETs) and are not customized further to be suitable for WSNs and have shortcomings in that they do not

overcome the weaknesses of the existing energy-aware routing protocols which don't guarantee the selection of the minimum total transmission path [71,91]. These protocols lack a route monitoring function which is essential for achieving energy efficiency. Data packets need to be routed based on the state of individual nodes by taking into account the energy and the relaying load at each sensor node and selecting the route with the maximum estimated lifetime to forward any pending traffic of aggregated data packets.

Although the reliability-oriented routing protocols covered so far in the literature within the research community always show an improvement in the energy balance, they do not explicitly consider an energy metric in addition to a reliability metric or the design of the cost functions implemented is unable to achieve energy balance of network activities among the constituent sensor nodes [61,88,113,121,123]. As a result, there is a large number of energy-efficient routing protocols proposed in the literature [4,5,34,49,61,73,88,93,140,141] that use the minimum required energy as a routing metric as an attempt to prolong network lifetime by distributing the workload among the relay sensor nodes. The performance objective of energy-efficient routing for maximizing the network lifetime was mainly considered in [49] and [141]. In [49], one of the state-of-the art energy-wise routing protocols for WSNs is proposed which is known as the Energy Aware Routing (EAR) [49]. EAR is a reactive, destination-initiated routing protocol where the base station initiates the route request and maintains the route subsequently. Thus, it is similar to the Directed Diffusion routing protocol [47] by maintaining multiple paths from source sensor nodes to the base station. However, EAR was intended as an energy efficient routing for WSNs. The primary goal of EAR is to improve the survivability of the networks. The EAR occasionally uses suboptimal paths rather than always using the optimal path. These paths are evaluated and selected using energy-based probabilities to slow the energy depletion of the sensor nodes along the optimal path. Every time data is to be sent

from the source sensor node to base station, one of the paths is randomly chosen depending on the probabilities. This means that none of the paths is used all the time; thereby preventing energy depletion. As a result, the network lifetime is prolonged.

The EAR uses localized flooding of request messages to find all possible routes between the source and destination, as well as the energy costs and the transmission probabilities associated to these routes. The protocol may encounter routing loops, in which data packets are being transmitted backward and forward between two neighbouring nodes, if both nodes select each other as the next hop continuously. Such a dead-lock situation is costly in terms of energy consumption for the resource constrained sensor network [49]. Another energy-aware collection layer protocol is proposed in [88] (known as a Dozer), a data gathering protocol meeting the requirements of periodic data collection and ultra-low power consumption. The protocol comprises MAC-layer, topology control, and routing all coordinated to reduce energy wastage of the communication subsystem. Using a tree-based network structure, packets are reliably routed towards the base station parents schedule precise rendezvous times for all communication with their children. However, the Dozer protocol is a closed source and specific to the Tinynode platform [89] which is different from Crossbow's TinyOS-based wireless sensor platform used in this thesis. Dozer protocol is a proprietary collection protocol running exclusively on Shockfish hardware. Moreover, since Tinynode platform [89] has different packet scheduling and MAC layers compared to Crossbow's platform, the benchmarking of the proposed routing scheme is only conducted on Crossbow's Berkeley motes.

In addition to energy-efficient routing, load balancing routing can alleviate and even prevent the effects of congestion, such as longer packet latency, poor packet delivery and high routing overhead [113,124,133,147,148]. The bandwidth and power limitations in WSNs mean that the consequences of relayed traffic congestion further worsen the

unbalanced energy dissipation of relay nodes. This is due to an excessive consumption of resources of such relays, which results in a rapid depletion of batteries and the consequent partitioning of the network. Therefore, load balancing is advantageous for avoiding traffic congestion and for ensuring the even distribution of traffic, which translate into more efficient energy utilization and maximised network lifetime.

Recent research has focused on load balancing routing for WSNs and ad hoc networks [133,147,148,149,150,151,152]. In [147], it has been shown that the traffic generated by each source sensor node can be balanced through multiple paths instead of using a single path. Authors in [148] introduce a network optimization problem used for performing the load balancing in wireless networks with a single type of traffic using the idea of Kirchhoff's voltage law. In [149], it has been observed that using only short paths allows for minimizing the latency and achieving load balancing. While a collision awareness node-disjoint multipath routing algorithm has been proposed in [150], authors in [151] propose a multipath routing protocol to reduce the congestion effects in wireless networks. In [152], the challenge of maximizing the network lifetime by load balancing the traffic has been addressed by deploying multiple sinks simultaneously to balance the energy consumption among sensor nodes, where sinks are connected through hybrid wired or wireless networking.

Specifically for mote-oriented WSNs, there are many routing protocols that consider load balancing in their routing schemes such as in [58,77,101,113]. In [113], Arbutus achieves load balancing by using the traffic load on the immediate links of a relay node rather than accumulating link costs for the route cost computation algorithm. Although the main objective of load balancing routing is the efficient utilization of network resources, it does not jointly consider communication patterns with link reliability and energy metrics in determining an optimal load balanced topology. However, this

approach is not always optimal as a route is only as good as its lowest quality hop. For example, a child node decided to select its parent based on its immediate link quality; it would pick the neighbour sensor node with the highest link quality as its next hop to the base station. However, this child node cannot either deduce the dynamics of link qualities between predecessor parents and the base station. This approach of route cost computation will be discussed later in Section 3.2 of the next chapter.

2.4 Summary

Mote-specific tree-based routing protocols, namely MintRoute [33,52] MultihopLQI [43] and CTP [66], are still the reference collection protocols for the stable implementation versions of TinyOS-2.x distribution. Although CTP protocol is the latest evolution of TinyOS collection protocols, MintRoute and MultihopLQI protocols have been intensively tested on mote-based sensor networks and are heavily used by a large number of research groups as reliability-oriented benchmarks [51,58,76,77,101,113]. For example, MultihopLQI has been used in recent deployments [67,76] (e.g., on a volcano in Ecuador [51]). To keep the performance evaluation of the routing scheme proposed in this thesis reasonable, MintRoute and MultihopLQI protocols have been selected to be the best effort benchmark that falls in the category of reliability-oriented collection protocols. Although the reliability-based routing protocols covered so far in the literature within the research community always show an improvement in the energy balance, they are not sufficiently representative as they are either only compared with proposals that do not consider energy metric in addition to reliability metric or because the design of the cost functions implemented is unable to achieve energy balance of network activities among the constituent sensor nodes. For more meaningful benchmarking, a state-of-the art energy-aware benchmark that consider energy metric (i.e., Energy-Aware Routing (EAR) protocol [49]) is also compared with the proposed routing scheme.

Chapter 3
Reliable Load-Balancing Routing (RLBR)

3.1 Background

The routing problem in ad hoc Wireless Sensor Networks (WSNs) is a nontrivial issue because of frequent link failures or sensor node malfunction due to environment conditions or restricted recourses (e.g. low battery power and limited memory). Thus, the routing protocols of WSNs encounter two conflicting issues: on the one hand, in order to optimise routes, frequent topology updates are required, while on the other hand, frequent topology updates result in imbalanced energy dissipation and higher message overhead. In mote-dominated WSNs, several authors have presented routing algorithms for WSNs that consider purely one or two metrics at most in attempting to optimise routes while attempting to maintain small message overheads and balanced energy dissipation [33,43,64,67,88].

In this chapter, various routing metrics are selectively integrated to examine their joint benefit on the lifetime of individual sensor nodes and the functional lifetime of the entire network. Recent studies on energy efficient routing in multihop WSNs greatly rely on radio link quality in the path selection process [67,113]. If sensor nodes along the routing path and closer to the base station advertise a high quality link to forwarding upstream packets, these sensor nodes will experience a faster depletion rate in their residual energy. This results in a topological routing hole or network partitioning [104,105]. This chapter presents a routing scheme that aims to improve energy efficiency for reliable multihop communication by developing a cross-layer lifetime-oriented routing scheme and integrating useful information from different layers. The proposed scheme aims to redistribute the relaying workload and the energy usage among relay sensor nodes to achieve balanced energy dissipation, thereby maximising the functional network

lifetime. The obtained experimental results presented in the subsequent chapters show that the proposed scheme has a responsive recovery from link failures, higher node energy efficiency, lower control overhead, and fair average delay.

3.2 Related Work

Since the most important challenge of a reliability-oriented routing protocol is how it can achieve efficient route propagation and how available routes to the base station are maintained, a common characteristic of the existing collection tree protocols is the use of network layer beacons to propagate route information. This can be performed by means of either an immediate or an accumulative link cost approach for route cost computation. However, these approaches are not always optimal, as routes are only as good as the lowest quality link. An example of the immediate cost approach is shown on *route* r_1 of Figure 3.1. If a source node S decided to select its parent based on its current one-hop link quality; it would pick the neighbor sensor node 1 with the highest link quality as its next hop parent towards the base station *BS*. However, since the link quality is time-varying, the source node S cannot deduce the dynamics of upstream link qualities of the parents towards the base station as the link broken between node 3 and the base station.

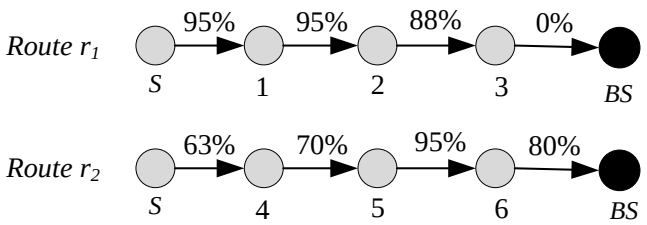


Figure 3.1 Route Cost Computation

On the other hand, the accumulative link cost approach uses the sum of the link quality values along a route and then averaging these values. However, this approach is also not the best. For example, although *route* r_1 has a broken immediate link with a quality of 0% between sensor node 3 and the base station, the child sensor node would still select *route* r_1 by averaging its link qualities which results in higher link quality than *route* r_2.

As these challenges are common in low-power radio WSNs, the routing protocol must infer the quality of all links on a route from a source to a destination with minimal overhead, and this condition should be reflected in the route metric that qualifies the selected route [120,121,122,157].

In mote-specific TinyOS-enabled WSNs [24,25], MintRoute [33], MultihopLQI [43] and Collection Tree Protocol (CTP) [67] are multihop reliability-oriented collection routing protocols. These collection tree protocols can be either classified as *proactive* distance vector routing protocols as in MintRoute [33] or *reactive* distance vector routing protocols as in MultihopLQI [43] and CTP [67]. The advantages and disadvantages of the traditional ad-hoc routing protocols that were originally designed for ad-hoc networks are well investigated and discussed within the context of WSNs in [114,115,116]. For example, in reactive protocols, sensor nodes do not need to maintain route entries to the base station as routes are requested on demand, thus saving memory space. Associated with this benefit are some drawbacks, including the fact that route request messages use a broadcast mechanism which can easily lead to a flooding problem. The unique communication architecture of WSNs creates the potential for the selection of a suboptimal route. This is due to the limited topological information available to the sensor node [117], the delay that is incurred in acquiring a route [118], and the energy profile of relay sensor nodes [49,88]. Consequently these are factors that should be considered when using a reactive routing protocol. As a result, the proposed routing scheme adopts a similar mechanism to route propagation but using jointly ad-hoc proactive and reactive approach.

From the reliability point of view, the collection tree protocols vary in the way how the route cost metric is calculated. MintRoute [33] employs the Expected Number of Transmissions (ETX) reliability metric [46]. ETX represents the cost in terms of the ratio

of the expected number of received packets to the number of packets actually received on the immediate link. MultihopLQI [43] and CTP [67] have been developed as variants of MintRoute [33]. While CTP attempts to improve upon MintRoute by adding the link costs across all hops, MultihopLQI uses a cumulative function of the CSI-based Link Quality Indicator (LQI) as a cost metric. This hardware-based LQI is provided by IEEE802.15.4-compliant RF transceivers such as those found on TelosB motes [87]. MintRoute and CTP use ETX [46] as a routing cost metric of the single-hop sender and Window Mean Exponentially Weighted Moving Average (WMEWMA) estimator [44] as an average filter. However, the aforementioned collection protocols are reliability-oriented protocols and do not explicitly employ energy or load balancing in their routing schemes [113,123]. Arbutus [113] is also a collection tree protocol but load balancing is its primary objective. It achieves load balancing by using the traffic load on the immediate links of a relay sensor node as an input to the cost computation algorithm. Although the main objective of load balancing routing is the efficient utilization of network resources, it does not jointly consider communication patterns with link reliability and energy metrics in determining an optimal load balanced topology. There is no doubt that a better distribution of relayed load leads to the more efficient use of bandwidth, leading to less contention and consequently lower energy consumption.

Another important challenge in low power WSNs deals with balanced energy usage for packet transmissions as it has been shown in [104,105,110,119] that the network lifetime can be extended if the rate of energy across the network is uniformly dissipated. For example, if packets are frequently relayed through relay sensor nodes along a selected route, these relay sensor nodes will deplete their batteries faster and fail earlier than their peers on other routes. The proposed routing scheme, namely, Reliable Load-Balancing Routing (RLBR), appropriately adapts to such situations through awareness of the relaying loads and the energy level of the relay sensor nodes. The scheme also aims for

load balancing between relay sensor nodes in terms of balanced energy usage and minimized energy dissipation for packet transmissions via adaptive beaconing and in-network aggregation of data packets. To that end, the RLBR scheme adopts a flexible approach that combines some of the advantages of the energy-aware protocols [49,71,88] on the top of the reliability-oriented proactive [33] and reactive protocols [43,67,113]. The proposed scheme also accommodates fault tolerance and adaptability to link and topology changes, while minimising communications overheads.

3.3 Routing Framework

Since the communications overheads are the major energy consumer during a sensor node's operation, the RLBR scheme, a simple but reliable routing protocol, aims to add minimal communication overheads for network configuration and multihop data dissemination. In low power WSNs, the unreliability of the links and the limitations of all resources bring considerable complications to routing. Even though most deployed WSNs use stationary nodes or have low mobility, the channel conditions vary because of various effects such as irregular low-power radio performance, or multipath fading effects which modify the patterns of radio wave reflections [38]. As shown in the framework of the routing scheme in Figure 3.2, the RLBR mutually employs hardware-based CSI to evaluate the wireless channel quality and software-based link quality estimations of adjacent neighbours to provide an estimate of the number of transmissions and retransmissions it takes for the sensor node to successfully receive a unicast packet. This improves delivery reliability and keeps RLBR adaptive to dynamic traffic and topology changes. RLBR also exploits the benefit from in-networking processing mechanisms (i.e., data aggregation) which can pack multiple small packets into a single data packet with the aim of minimising energy consumed for communications while considering the delay-sensitivity of the relayed aggregated packets. RLBR requires each

sensor node to switch among multiple parents for load-balancing purposes. Taking the load-balancing optimization into consideration at the MAC layer will significantly complicate the design and implementation of MAC protocols. Therefore, RLBR is designed to perform the dynamic adaptation at the routing layer of the network stack.

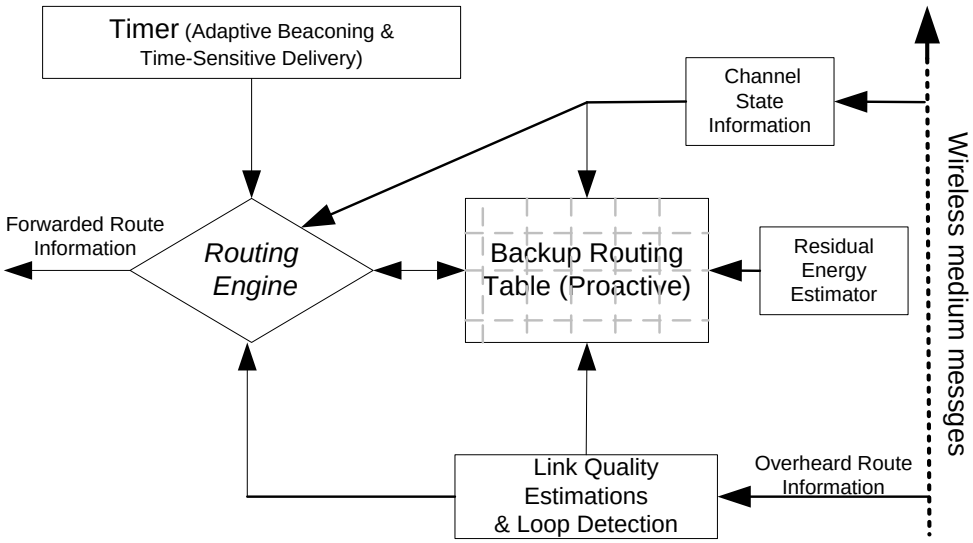


Figure 3.2 The RLBR Scheme Framework

Based on the recent existing work in [120,121,122,157], the RLBR uses multiple metrics including both hardware-based Channel State Information (CSI) and software-based link quality estimations. The CSI includes the Received Strength Signal Indicator (RSSI) and Link Quality Link (LQI) [87] that can be directly provided by the radio transceiver. The link quality estimations are calculated based on packet transmissions/retransmissions such as Packet Reception Ratio (PRR) and can be obtained statistically over a long period of time. The RLBR also employs energy, load-aware and latency metrics (explained in Section 3.5). Although overhearing is an aspect of contention-based MAC protocols that have a negative impact on energy consumption in WSNs where nodes can receive packets that are destined to other nodes, overhearing can also be a beneficial aspect of contention in updating the route information of routing protocols [122]. As a result, the RLBR makes an integration of locally overheard route information into the

routing cost function. The route information includes parent's *id*, routing tree level or depth, aggregating load status and relaying deadlines.

Due to the highly dynamic nature of WSNs, the inherent advantages (e.g., overhearing) of contention-based MAC protocols, contention-based MAC protocols have been widely used in WSNs [106,115]. Hence, B-MAC [21] and IEEE802.15.4 [87] were the preferred choice as underlying layers providing MAC services to the RLBR scheme at the routing layer. Contention-based MAC protocols allow any sensor node to overhear packets transmitted by its neighbours; this allows the upper routing layer to employ snooping for the sake of link quality estimation, in-network processing and data aggregation. In addition to the intended receiver sensor nodes, neighbours of the sender node possibly will receive sent packets, even the packets are not addressed or destined for them (i.e., overhearing). In this situation, the routing information used for routing decisions in the RLBR scheme can be embedded in the header of the sent packets. When a sensor node receives a packet not addressed to itself, it can retrieve this helpful routing information from the packet header before dropping the packet to be used by the routing engine. Hence, the underlying MAC protocol is chosen in order to be aware of the upper routing protocol and offers control to the routing protocol that sit on top of it, allowing the routing layer to change parameters (e.g., the number of retransmissions).

In the RLBR scheme, the remaining energy capacity in the forwarding sensor nodes and the link or channel quality between communicating sensor nodes are the key factors that shape the network topology: the hardware-based CSI can be measured directly from the radio hardware circuitry of the wireless platform in form of signal quality; the packet transmissions-based link estimations are computed by software at the receiver based on the successfully received packets; and the residual energy capacity is estimated after deducting the estimated dissipated energy based on the current consumption model of

the mote system (processor and radio) during its operations. The presence of a time constraint requires the network to favour routes that minimise the number of hops at network layer and delay-sensitive data aggregation at application layer in order to minimize the average end-to-end data transfer latency.

The RLBR scheme is a hybrid, reactive and proactive, routing protocol designed to adaptively provide enhanced balanced energy usage on reliable routes and to employ ready-to-use neighbourhood routing tables in order to allow sensor nodes to quickly find a new parent upon parent loss due to link degradation or sensor nodes run out of energy. The RLBR scheme is a tree-based routing protocol where a child sensor node forms a routing tree to its upstream parent towards the perimeter base station and is also address-free in that a sensor node does not send a packet to a particular sensor node; instead, it implicitly chooses a parent sensor node by choosing a next hop based on the routing selection parameters including link quality, energy level, hop count, aggregating load status and relaying deadlines [120,121,122,157].

3.4 Network Configuration and Maintenance

3.4.1 Routing Tree Formation

The routing tree is a directed acyclic graph [103] in which packets are relayed towards the base station over multiple routes. The routing tree is built by assigning a *level number* to each sensor node depending on its distance (e.g., number of hops) to the base station, and delivers sensing data packets from higher-level to lower-level sensor nodes. The base station is at *level* 0. Each sensor node at level i can select a valid parent from its level i or from lower level i-1 towards the base station as shown in Figure 3.3. The resulting routing tree starts with the easily-constructed shortest path tree (SPT) [103], and then allows each sensor node to pick a new parent node if it has a better routing cost with a higher link quality and energy level, and with minimum latency.

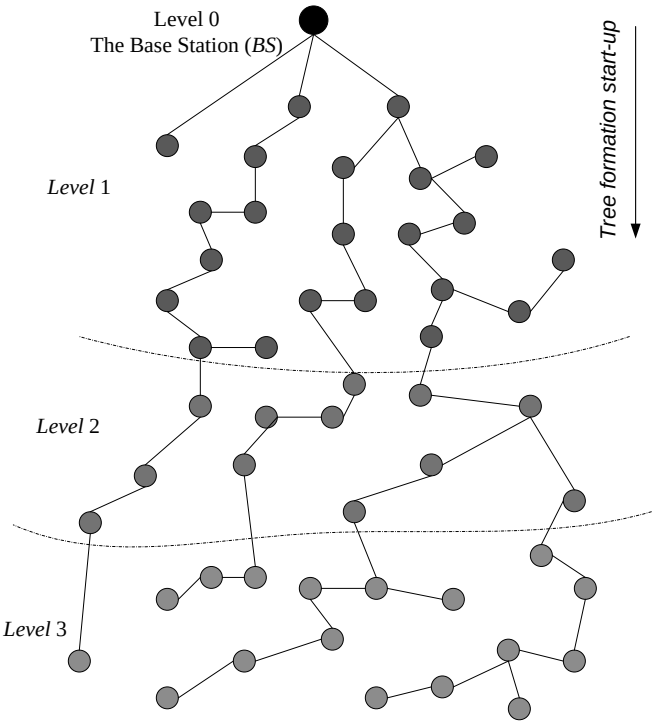


Figure 3.3 Example of Routing Tree Formation

Using the broadcast nature of the contention-based wireless medium, a sensor node can easily observe its neighbourhood by either receiving or overhearing route messages or beacon packets within an update period. Route information is inserted on the top of the payload segment of the outgoing message or beacon. Figure 3.4 shows the RLBR message structure which has 8 bytes (64 bits) header (routing frame) on the top of the payload (data frame). The routing frame has control and route information field that sensor nodes use to exchange topology and route information. The routing frame advertises the status of sensor node's current parent and routing cost. It has two control bits and the reserved segment is kept for potential future extensions of the RLBR scheme. The first control bit is the f_j bit which is used in the adaptive beaconing mechanism discussed later in Section 3.4.3. The second control bit is L_i bit is used for load-aware aggregation as discussed later in Section 3.5.1.

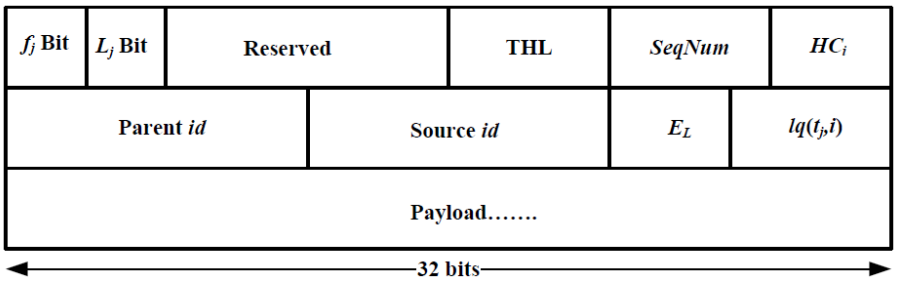


Figure 3.4 Frame Format of the RLBR Message

The routing information contained in the jth outgoing route beacon b_j sent by a sensor node n_i includes the *beaconing control bit* f_j, the *bottleneck relaying/aggregating load control bit* L_i, the *current parent's id*, the *link quality estimation* $lq(t_j,i)$, the *depth* or *tree level* HC_i (*hop count* to the base station), the *parent's residual energy level* E_L, data packet's sequence number (*Origin SeqNum*), the source sensor node's *id*, and the time-has- lived (THL) field. The THL field commences with the value of zero at the origin and increments by one with each hop. The *Origin SeqNum* and source *id* are initiated at the source sensor node and used to avoid routing loops and data packet duplications.

State information cached in the routing table of a sensor node n_i includes the *parent's id*, the *link quality estimation* $lq(t_j,i)$, the *bottleneck load bit control* L_i, the *hop count* HC_i, the *data packets relaying count* C_i of n_i', and a *parent loss time* $T_{ParentLoss}$ (time spent since the current parent became lost). Once the beacon is received, the receiver sensor node then proceeds to extract the routing information from the beacon for local processing in the parent selection process discussed in Section 3.4.3. The rate of the neighbourhood beacon reception depends on the update period of the beaconing rate B_rate_j. In every update period, the backup forwarding routing table is updated. The RLBR beaconing intervals are adjusted (the beaconing rate (B_rate_j) is accelerated or decelerated) according to the frequency of topological changes. RLBR accelerates its beacon interval whenever a sensor node receives a packet with the f_j bit set ($f_j = 1$) and

vice versa. A sensor node sets the f_j bit when it does not have a valid parent. For example, when a sensor node disjoins from the current routing tree and sets its routing cost to infinity due to parent loss, it transmits beacons with the f_j bit set. Setting the f_j bit will allow a sensor node with lost parent to rediscover its neighbours in order to update its routing table entries and quickly recover from link failures. For load-aware routing, a parent sensor node sets the L_i bit when its forwarding queue is overloaded with aggregating data packets. For example, if a beacon is received with the L_i bit set $(L_i = 1)$, overloaded aggregators "bottlenecks" can be quickly identified and avoided.

3.4.2 Phases of Routing Tree Construction

The construction of the routing tree is mutually performed in three phases: *Network setup*, *Data transmission*, and *Route searching/maintenance*.

Network Setup Phase

In order to start-up the ad hoc networking and to build the *proactive* routing tables, the base station acts as a tree root which initially disseminates a route setup message into the network to find all possible routes and to measure their costs back from the source sensor nodes to base station. The routes costs are kept updated by using adaptive beaconing during the *reactive* route maintenance phase by sending fewer beacons in order to adapt with link dynamics while simultaneously minimising the route repair latency. Therefore, the receiving sensor nodes determine all routes with their updated cost parameters to be used in the parent selection process. The base station is assigned with a tree level or depth equal to *zero*, it is also set with the cost parameters to *zero* before sending the route setup message. The intermediate sensor nodes at level one (e.g., one-hop from the base station) that can receive the route setup message from the base station, forward the route setup packets to the reachable sensor nodes at level two (e.g.,

two-hops from the base station). Sensor nodes that have a higher cost compared with other peer sensor nodes (e.g., lower residual energy level or lossy link are discarded from the routing table). Sensor nodes at level three repeat the previous steps and all information travels until it reaches the leaf source sensor nodes and all nodes identify their depth and the routing tree is fully defined.

Data Transmission Phase

In this stage, the source sensor nodes start to transmit data packets towards the base station through the preselected least-cost route based on the parent selection parameters. Intermediate sensor nodes aggregate and relay the data packets to the upstream parents toward the base station. This process continues until the data packets of interest reach the base station. *Data aggregation load* is considered in this phase in order to maintain timely data delivery and to avoid misplacing deadlines for delay-sensitive data packets. Hence, each sensor node must decide when to stop waiting for more data to be aggregated based on a preset maximum waiting time. For example, at time 0, an aggregating parent sensor node starts aggregating data from its own packets, if any, and from its children that have participated in aggregation. Later, at time t this aggregator node will forward the so far aggregated data up to time t to its parent. The amount of aggregated data is a function increasing in participating sensor nodes and decreasing in waiting time t. Moreover, a sensor node within the vicinity can exploit unavoidable overhearing or eavesdropping on neighbouring nodes' traffic to improve the selection of parent nodes and data aggregators. This will be further explained in Section 3.5 in terms of load-balanced and real-time packet relaying.

Route Searching/Maintenance Phase

Route maintenance is the most important phase, which is performed using adaptive beaconing (explained in Section 3.4.3) to handle topology dynamics due to link failures

to *reactively* maintain a set of "on-demand" available paths upon any data packet transmissions. Hence, the routing tree is sustained and the backup neighbour routing tables are also kept updated to avoid relays with lower energy and unreliable links. To achieve reliable data packet delivery and parent selection process, each sensor node maintains a neighbour routing table indicating one hop sensor nodes it can reach. This table contains the links quality to such parent nodes' *id*, residual energy, hop count and other helpful routing information. The rationale behind maintaining a neighbour table is to *proactively* keep track of possible efficient routes to the base station and to order them on the basis of a joint metric favouring high-quality links, relays with good energy resources above predetermined threshold, and minimum delays. By keeping track *reactively* and *proactively* of the channels with minimum link quality and the sensor nodes with the lowest residual energy, overloaded relays "bottlenecks" can be promptly identified and avoided during network operations.

3.4.3 Adaptive Parent Selection Process

This section presents the two mechanisms used by the RLBR scheme during the route searching/maintenance phase. These mechanisms make the routing scheme reliable and resilient to link dynamics while having low cost (e.g., energy). The first mechanism is the adaptive beaconing which achieves both rapid route repairs and low routing overhead when the network topology is converging. The second mechanism is the blacklisting which reduces the possibility of using unreliable asymmetric links and allows a given sensor node to only maintain a subset of its neighbours with good link qualities.

Adaptive Beaconing of Route Messages

In order to reduce unnecessary energy consumption, the routing scheme transmits fewer beacons when link dynamics are low while keeping the network responsive to topological changes and maintaining fast recovery from links failures. Each sensor node

n_i starts to transmit route messages or control beacons at regular intervals of beaconing rate B_rate_j to update route information. B_rate_j is initially set to a beacon packet per 7sec then the beaconing rate of route messages adapts to topological changes using the *beaconing control bit* f_j. As shown in Algorithm 3.1, the beaconing rate of sensor node n_i is decelerated upon successful transmission of at least β data packets to the same parent (i.e., $(C_i \geq \beta)$) and sensor node n_i successfully advertises its route to its current parent using beacons with $(f_j = 0)$. Where C_i is *the data packets relaying count* of n_i; and β is *the relaying threshold* from n_i to the same parent. Otherwise, if the condition of initialising the route searching/maintenance phase $(\mathrm{T}_{\mathrm{ParentLoss}} > \mathrm{T}_{\mathrm{waiting}})$ is met or $(f_j = 1)$, the beaconing rate is immediately accelerated to inform any child sensor nodes that n_i's parent is lost and n_i will be blacklisted (blacklisting mechanism is initiated). n_i will wait for $\mathrm{T}_{\mathrm{waiting}}$ until a new valid parent is *reactively* discovered during the route searching/maintenance phase of the parent selection process. Where $\mathrm{T}_{\mathrm{ParentLoss}}$ is the time spent since the current parent became lost which is a function of the number of route beacons sent by sensor nodes n_i since its parent was lost. $\mathrm{T}_{\mathrm{ParentLoss}}$ is incremented every time a route beacon b_j is sent until a new parent is joined. $\mathrm{T}_{\mathrm{waiting}}$ is the timeframe allowed to n_i to wait while *reactively* searching for a new parent. $\mathrm{T}_{\mathrm{waiting}}$ is used as a route searching/maintenance timeout threshold. If this threshold is reached and no parent is *reactively* joined, n_i immediately disjoins from the current routing tree and sets its routing cost to infinity to avoid possible routing holes. This leads to parent reselection as the condition of initialising the route maintenance phase $(\mathrm{T}_{\mathrm{ParentLoss}} > \mathrm{T}_{\mathrm{waiting}})$ is met. To manage the frequency of parent reselection, the threshold values in routing cost function are set higher so that sensor nodes change their parents less often. To reduce the delay of link recovery, the routing scheme *proactively* looks up an alternative valid parent or path that has been recently cached into the routing table based on overheard neighbourhood

information embedded into the foregoing route beacon b_{j-1}. The lowest cost route will be immediately selected in preference to waiting other beacons to rediscover a route. This will allow sensor nodes to quickly find a new valid parent upon parent loss and the sensor network will quickly self-organise during the route searching/maintenance phase by maintaining a reliable set of valid parent nodes in the built-in routing table.

Algorithm 3.1 Adaptive Beaconing

```
Initialisation: route maintenance
Input: B_rate_j                                   //Current beaconing rate
Input: C_i                                        //Relaying count of n_i
Output: B_rate_{j+1}                              //Adjusted beaconing rate
For each sensor node n_i
    While (b_j sent at B-rate_j)                  //Route advertisements being sent
        If (C_i ≥ β) Then                         //At least β data packets are sent
        {
            If (f_j = 0)                          //n_i successfully advertises its route.
                B_rate_{j+1} = B_rate_j - d ;     //Beaconing decelerated by constant d
        }
        Else                                      //Route or parent is lost
            If (T_ParentLoss > T_waiting) or (f_j = 1) Then   // Lost parent (Maintenance Condition)
            {
                B_rate_{j+1} = B_rate_j + d ;     //Beaconing accelerated by constant d
                Blacklisting Algorithm Initialisation   // n_i is blacklisted
                Update Route information
            }
            Endif
        Endif
    Endwhile
Endloop
```

Upon parent loss due to link failure or low residual energy, a sensor node with an invalid parent waits for $T_{waiting}$ to restore a new valid parent. During this waiting period, a number of relaying sensor nodes may possibly discard some of their received data packets due to aggregation load overflow (load-aware aggregation is discussed later in Section 3.5.1). The waiting period $T_{waiting}$ at a relaying/aggregating sensor node is bounded by delay constraints in order to reduce the latency of end-to-end data packet delivery (relaying and aggregating deadlines are bounded by the *holding time* Δt_{agg} discussed later in Section 3.5.2).

To perform the algorithm, routing information can be initially acquired through *route messages beaconing* or packet *overhearing* to be maintained in the routing tables during the *network setup phase*. In the *network setup phase*, the network is initially considered as fully identified and the values of route metrics are initially obtained in the routing table and ready for use in *route searching/maintenance phase*. While the routing table entries are used for the quick *proactive* rerouting, broadcasting route messages is also necessary for updating routing tables and the *reactive* route searching due to link dynamics. The routing information required for the routing tree formation is added into the original beacon packets' headers as shown earlier in Figure 3.4, so that sensor nodes can have the necessary neighbour information to modify the routing path up on request.

Blacklisting Mechanism of Unreliable Links

As shown in Algorithm 3.2, RLBR employs a time-varying link blacklisting mechanism to reduces the possibility of using asymmetric links and allows a given sensor node to only maintain a subset of its neighbours with good link qualities. In dense sensor networks, this mechanism prevents the backup neighbourhood routing table from growing beyond a given memory size. However, due to the link asymmetry problem, the blacklisting mechanism can cause network partitioning if the thresholds are not accurately set [58,60,79]. The routing scheme calculates the average packet transmission and retransmissions approximations after a successful reception. When a beacon is received at sensor nodes n_i, link quality information is collected in order to avoid routing over unreliable asymmetric links. To improve routing reliability, link failure detection is addressed by collecting bidirectional link quality information at each node. This is performed by monitoring both sending and receiving link quality estimations.

Algorithm 3.2 Blacklisting of Unreliable Links

Initialisation: *route maintenance*

Input: $lq(t_j, i)$; *// link quality of a received route beacon* b_j.

```
Output: Φ_t(j) = Φ_t(j−1) + G_t(j);   //Blacklisting threshold.
For each n_i
   While (b_j received)
     If (lq(t_j,i) < Φ_t(j)) Then
     {
       If (T_ParentLoss > T_waiting) Then        // Route searching/maintenance condition.
       {
         Blacklist lq_(t_j,i);                   //Invalidate node n_i's link to its current parent.
         f_j = 1;                                //Beaconing rate is accelerated.
         If (b_j)                                // n_i successfully advertises its route using b_j.
            Φ_t(j) = Φ_t(j−1) − c;               // G_t(j) is set to constant (-c).
            If (b_j−1)                           // n_i successfully advertises its route using b_j−1.
               Φ_t(j) = Φ_t(j−1) + c;            // G_t(j) is set to constant (c).
            Endif
         Endif
       }
       Endif
     }
     Else
       G_t(j) = 0;
       Unblacklist lq_(t_j,i);                   //Validate node n_i's link to its current parent.
     Endif
   Endwhile
Endloop
```

For example, if the packet loss ratio (PLR) during a time window exceeds the threshold, the child sensor node reactively changes its current parent to an alternative valid parent, and temporally puts the old parent into a blacklist to be excluded from forthcoming parent selection. If we assume that $\Phi_t(j)$ is a blacklisting threshold of time-varying link quality estimation $lq(t_j,i)$ at sensor node n_i, t_j is the time at which the jth route beacon b_j is received, and $G_t(j)$ is the tuning function of the blacklisting algorithm. When a parent is not available due to a ruined link quality, sensor node n_i invalidates its current parent that has a link quality $lq(t_j,i) < \Phi_t(j)$ if there is an alternative parent in the backup routing table and the condition of initialising the route searching/maintenance phase $(\mathrm{T}_{\mathrm{ParentLoss}} > \mathrm{T}_{\mathrm{waiting}})$ is met. The blacklisting threshold $\Phi_t(j) = \Phi_t(j-1) + G_t(j)$ is tuned by inverting $G_t(j)$ and based on both the jth and the $(j\text{-}1)^{th}$ received beacons until a new parent is eventually selected and the routing table is updated.

To reduce control overhead of the routing scheme, link quality information provided by the underlying MAC layer is used effectively with packet transmission and retransmissions as reliability metric. The RLBR scheme takes the advantage of averaging the CSI-based link quality (*lq*) measured values (e.g., *RSSI* and/or *LQI* provided by the radio circuit) in each time frame *t* to calculate the averaged time-varying link quality estimation (lq_t). This can be calculated using equation 3.2 to better reflect lq_t measurements for short-term link quality estimations of time frame *t*. Where α is the history control factor ranges from 0 to 1 of time frame *t*. It controls the effect of the previously estimated link quality values (lq_{t-1}) in relation to the current values (lq_t).

$$lq_t(\alpha) = \alpha \times lq_{t-1} + (1+\alpha) \times lq_t \tag{3.2}$$

3.4.4 Routing Loops Prevention Strategy

During the routing tree formation process, specifically, in the tree setup phase, a sensor node can only pick its parent from the same level or lower level according to its communication range and routing metrics. As shown earlier in Figure 3.3, choosing a parent node from the same level gives the routing scheme more flexibility and unrestricted membership of parent candidates in the parent selection process. To prevent the formation of possible routing cycles or loops in the whole routing tree, the parent selection is restricted to neighbours which are not farther away than its tree level based on the routing information in route beacons. The route beacon frame contains useful information such as source node's *id* and data packet's sequence number (*Origin SeqNum*). This information is initialised at the source sensor node and also used to eliminate data packet duplications. Routing loops are prevented at the same level using a sensor node's *id* as a tiebreaker. Nodes in the same level have an *ascending* ordering in the priority of being selected as parents, i.e., a node with larger *id* is selected as a parent for nodes with smaller *id*. Therefore, no loop can be created within the same level as

well as the entire network. Without the tiebreaker, two sensor nodes in the same level may pick each other as their parents and form a routing loop at this level. If a routing loop is detected, the routing scheme will discard the packet and immediately invalidate the route and re-discover a new route towards the base station. Therefore, there is no possibility of creating a cycle in the parent selection process.

3.4.5 Resilience to Link Fluctuations

Since route selection process is distributed in the RLBR scheme, sensor nodes use local routing information for routing decision. When a sensor node receives a request, it will include a link quality estimate when forwarding the request towards its neighbour. These estimates computed in bounded time enable the base station to maintain the routing tree with link quality estimates along each edge. Sensor nodes use request beacons to get link quality estimates. The adaptive beaconing allows each sensor node to send several beacons in quick succession to grant the ability to quickly react to persistent changes in link quality. Hence, a better network response is achieved. In order to reduce control overhead, each relaying sensor node only forwards a cumulative join request beacons that contains the total number of broadcast requests received by that sensor node. Using this information, the sensor nodes have sufficient data to form instantaneous routes' link qualities estimates. Link failures of a sensor node on a routing path can be identified and allocated by using the end-to-end data packet sequence number and source sensor node *id* of each data packet. Link qualities estimates are updated and kept for as long as the sensor node is trying to join the routing tree using the adaptive beaconing. Hence, this improves the ability of the routing scheme to resist to transient variations or fluctuations in link quality using *short-term estimations* with faster beaconing. This is in contrast with *long-term estimations* with fixed beaconing in the traditional routing protocols where link estimates need to be kept for longer times and cannot be discarded unless a

sensor node is removed from the neighbourhood routing table. The impact of using long-term vs. short-term estimates is experimentally analysed in Chapters 4 and 5.

3.5 Balanced Data Dissemination and Collection

The main objective of load balanced routing is the efficient utilization of limited resources and evening out the distribution of traffic loads in terms of efficient energy consumption. As a result, maximising lifetime of each sensor node can be achieved with balanced battery usage. The routing scheme takes into account not only the current energy capacity of the sensor nodes and the channel state but also considering various aggregating loads resulted from different deployment and event patterns as explained in section 3.5.1. It also considers the overall distribution of the delay-aware in-network aggregation load along the selected route by means of load balancing benefits, which translates into more reliable real-time packet relaying as explained in section 3.5.2.

3.5.1 Load-Aware Aggregation

Using the broadcast nature of the contention-based wireless medium, a sensor node can easily observe its neighborhood by receiving and overhearing periodic beacon packets. The underlying MAC protocol is chosen to be a contention-based approach such as B-MAC and IEEE802.15.4, in order to allow each sensor node to overhear packets transmitted by its neighbours; this allows routing layer to employ snooping for link quality estimations, and in-network processing and data aggregation. Data aggregation as a form of in-network data processing or redundancy suppression is performed to enhance the load balancing paradigm by conserving the energy along the routing path from the source sensor nodes towards the base station using standard data aggregation functions such as max, min, and average. Aggregation points occur close to the source or event sensor nodes as early as possible to maximize the aggregation benefit by means of

opportunistic aggregation strategy. The RLBR scheme balances the relaying load when transmitting packets by embedding useful routing information into the overheard packets to allow for taking the advantage of traffic overhearing and also minimising control traffic. Source sensor node's *id* and data packet's sequence number are used to avoid data packet duplications. For load-aware aggregation, a parent sensor node sets the *bottleneck relaying/aggregating load control bit* L_i when its forwarding queue is overloaded with aggregating data packets. The overloaded aggregators "bottlenecks" can be quickly identified and avoided. As a result, low packet error rates can be maintained and packet delivery is improved while minimizing redundant packet transmission and retransmissions throughout the network.

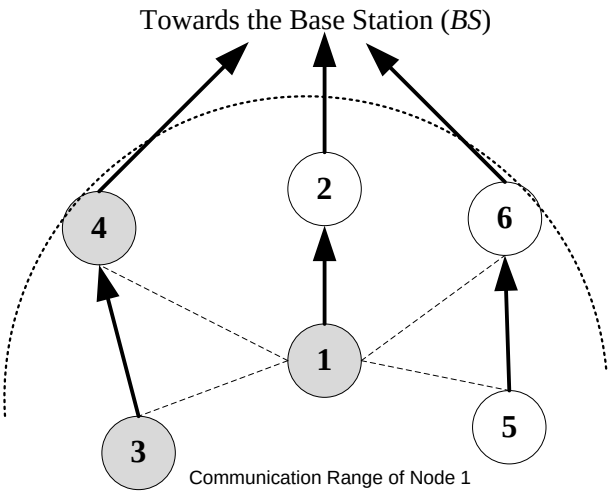


Figure 3.5 Load-Aware Aggregation

Figure 3.5 shows the communication range for a source sensor node 1. While node 1 is sending its packets to its current valid parent 2, it can overhear the packets sent from 3 to 4 and from 5 to 6. If we assume that parent node 2 is overloaded (i.e., $L_2 = 1$) using the overheard information, sensor node 1 can change its current parent from 2 to 4 or to 6 in order to reduce the aggregation load on 2. This reduces the likelihood that time-sensitive aggregated data will be dropped at the overloaded sensor node 2. Assuming the following are met: Sensor node 4 compared to node 2

has less aggregation load, better link quality with 1, higher residual energy; and node 4 has higher *id* compared to node 1; Node 3 sends its packets to 4 within its vicinity. In terms of energy dissipated for transmissions, it is more efficient for sensor node 1 to send its data packets to 4, where its data packets can be aggregated with 3 and 4's data packets. However, aggregating sensor node 1's data packets with 3's and 4's is dependent on the aggregation queue state information maintained in sensor node 4. Node 4 must not be overloaded with aggregated data packets in order to allow the routing scheme to ensure the time-sensitive deadlines of the forwarded data packets.

As various deployments could result in different data patterns, this feature of data aggregation is kept optional as it is application-specific. It can be enabled or disabled based on the application and physical topology. Since this distributed parent selection process is performed dynamically on a packet-by-packet basis, this approach is adaptive and the topology of aggregation can change to accommodate different situations based on the aggregation or relaying load. However, aggregating data packets at each sensor node of the selected route introduces extra processing overhead which increases energy consumption. Parent selection process also consumes energy. To achieve high success reception ratios of data packets, it needs control traffic transmission, which again demands extra energy. Considering all these factors, the *data packet delivery efficiency metric* (η) is proposed as a measure of the effectiveness of this approach in minimising packet transmissions throughout the network. *Data packet delivery efficiency* (η) accounts for the ratio of the total number of data packets received at the base station to the total number of all control and data packets sent throughout the network. This is expressed in Equation 3.1. The η is used as a benefit metric to gauge end-to-end packet delivery performance of the routing scheme in terms of route message transmission weight. Conversely, the reciprocal of *data packet delivery efficiency*, namely, *data packet delivery*

cost ($1/\eta$) is used as a *routing overhead metric* to give an overall estimation of the energy consumed by relay sensor nodes for delivering a data packet towards the base station.

$$Delivery\ Efficiency\ (\eta) = \frac{Number\ of\ received\ data\ packets}{Number\ of\ sent\ data\ and\ control\ packets} \tag{3.1}$$

3.5.2 Bounding Relaying Deadlines

The routing scheme should minimize the number of transmissions to improve the energy-efficiency and cost-effectiveness of low-power WSNs. Therefore, aggregating smaller, relayed, data packets into larger encapsulated packets bounded by the Maximum Transmission Unit (MTU) could significantly minimize the number of packet transmissions and improve energy savings. However, in real-time applications, these encapsulated data packets vary in their deadlines and sensitivity to end-to-end delay. These deadlines are governed by the importance of the sensing measurements in order to maintain a real-time packet delivery. As shown in Figure 3.6, the *average end-to-end delay* is the sum of all single-hop delays along the selected route r_j.

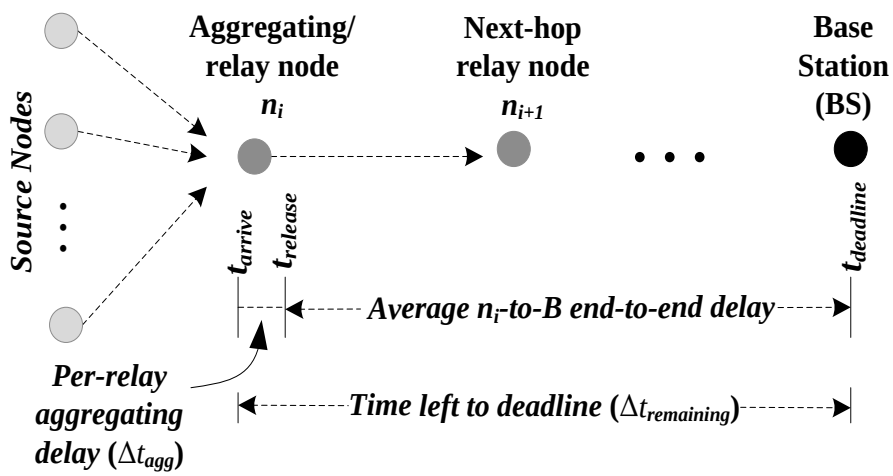


Figure 3.6 Bounding Relaying/Aggregating Deadlines

Due to on-flight aggregation when the delay calculations occur, encapsulated data packets tend to be delayed at each intended relaying sensor node waiting to be encapsulated with other arriving or locally generated data packets for a given *holding time* Δt_{agg}. This time is

known as the *per-relay aggregating or encapsulating delay*. The *average* (n_i*-to-BS*) *end-to-end delay* $\Delta t_{n_i,r_j,BS}$ is estimated on-flight on route r_j between sensor node n_i at the point of data encapsulation and the base station *BS* by summing the individual delays as stated in [112]. However, the total accumulated *per-relay encapsulating delay* including propagation on route r_j must not exceed the *remaining time* $\Delta t_{remaining}$ which is the time left before the associated *real-time deadline* $\Delta t_{deadline}$expires. In other words, *per-relay aggregating delay* Δt_{agg}needs to be bounded in order to avoid missing the application-specific packet delivery deadlines. If a data packet arrives at relay sensor node n_i at a time $\Delta t_{arrive}$to be aggregated with other data packets, Δt_{agg} must be bounded and the encapsulated packet sent at an appropriate dispatch or *release time* $\Delta t_{release}$. Subsequently, this dispatched, encapsulated, data packet might also be re-encapsulated on further hops and Δt_{agg}must permit receipt within the packets delivery deadlines. In the case where $\Delta t_{agg} \leq 0$, $\Delta t_{n_i,r_j,BS}$ is negative and the arriving packet must be relayed immediately without encapsulating delay. In other cases the arriving packet can be delayed for a bounded *holding time* Δt_{agg} as expressed in Equation 3.2.

$$\Delta t_{agg} = \Delta t_{remaining} - \Delta t_{n_i,r_j,BS} \tag{3.2}$$

Since the packet encapsulates more than one data element over the route of (*N*-*i*) relay sensor nodes, the encapsulated packet at relay node n_i must be dispatched once either sensor node n_i reaches its memory limit or one of these packets reaches the end of its minimum dispatch time of $\min(\Delta t_{release})$. This time must satisfy the accumulated condition in Equation 3.3 over route *k* of (*N*-*i*) sensor nodes.

$$\sum_{k=i}^{N}[\min(t_{release_k}) - t_{arrive_k}] \leq \sum_{k=i}^{N}\Delta t_{agg_k} \tag{3.3}$$

3.6 Reliable Energy-Balancing Routing

Since maximising network longevity is the focal ambition of the RLBR scheme, network lifetime needs to be estimated in terms of the ratio of residual energy level of the active or alive sensor nodes. However, the estimation of network lifetime is application-specific, and it is complicated to derive a general indicator to estimate the network life. Consequently, functional lifetime of relaying sensor nodes is estimated per hop over the selected route according to their importance and location in relation to the base station.

3.6.1 Route Average Dissipated Energy

From an energy usage viewpoint, the sensor nodes closer to the base station are the most critical nodes in the network as the load on them is significantly higher than their more distant peers. Without appropriate countermeasures to ensure network lifetime maximisation by balancing the energy dissipation, these nodes will deplete their residual energy faster, thereby making the network worthless. In Figure 3.7, it is supposed that an optimal multihop route r is constructed by N linearly adjacent sensor nodes transmitting with a given transmission power level of P_{tx}. A data packet is relayed over the route r with similar link reliabilities from source sensor node n_i towards the base station BS. The total average dissipated energy E_r required to forward one packet from each of the sensor nodes n_i at level (N+1-i) to the base station along the routing path r can be calculated based on the number of hops or hop count (HC) and average amount of energy consumed E_{ni} by node n_i at each hop. Equation 3.4 expresses E_r as a function of the *hop count* from the sensor node n_i at which the packet is generated along the route r towards the base station. Where HC = (N+1-i) and E_{n_i} is the average consumed energy by an individual node n_i.

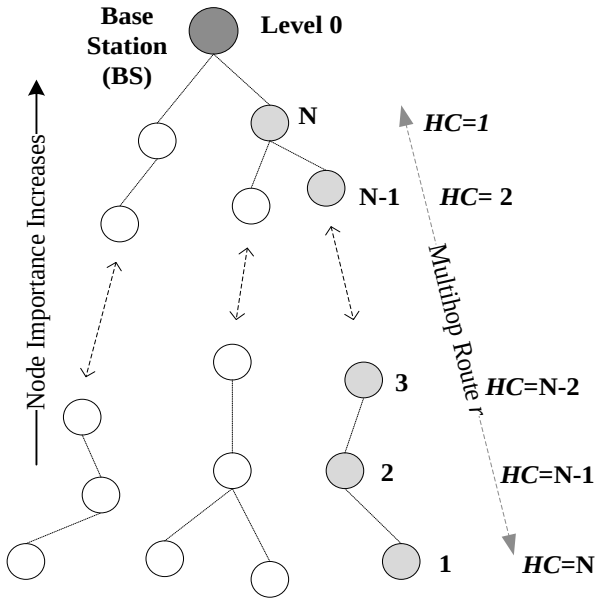


Figure 3.7 Calculating the Energy Cost over Route *r*

$$E_r = (N(P_{tx} \times t_{n_1,r,BS}) + (N-1)(P_{tx} \times t_{n_2,r,BS}) + \ldots$$
$$\ldots + 2(P_{tx} \times t_{n_{N-1},r,BS}) + 1(P_{tx} \times t_{n_N,r,BS}))$$

$$E_r = (N \times E_{n_1} + (N-1) \times E_{n_2} + \ldots + 2 \times E_{n_{N-1}} + 1 \times E_{n_N})$$

$$E_r = \sum_{i=1}^{N} [(N+1-i) \times E_{n_i}] = \sum_{i=1}^{N} [HC \times E_{n_i}] \tag{3.4}$$

In this work, the following assumptions are made: the packet transfer rate at all sensor nodes along the routing path *r* is the same; the time $t_{n_i,r,BS}$ required for forwarding the packet is the same at each relay node and the transmission power is fixed for all sensor nodes. However, E_{n_i} is increasing as the sensor node n_i becomes closer to the base station as it forwards more packets from its downstream nodes. For example, the most critical sensor node is node n_N, which is the closest sensor node to the base station and always consumes the maximum amount of energy as a result of relaying packets originated at all (N-1) sensor nodes, e.g., n_1, n_2, …, n_{N-1}, along the route *r* towards the base station.

To this point, total average energy dissipation E_r required to forward one packet from each of the sensor nodes n_i to the base station *BS* along the routing path *r* has been

considered as a function of hop count (*HC*) (also known as tree depth or level number). The next step focuses on the derivation of the average consumed energy E_{n_i} of node n_i as a function of the link reliability metric of the multihop route *r*.

3.6.2 Energy and Reliability Probability

The link reliability probability embodies the link quality metric of the probabilistic routing scheme used for parent selection. A sensor node n_i may forward a packet to its nearest neighbour node n_{i+1} with link reliability probability $P_{n_i,r,n_{i+1}}$ which is the readiness of a sensor node n_i to relay a data packet towards the base station *BS* through a selected route *r* of (*N*+1-*i*) hops. A sensor node n_i may also send directly to the base station *BS* with probability $P_{n_i,r,BS}$ based on its location, where $P_{n_i,r,BS} = 1 - P_{n_i,r,n_{i+1}}$. Therefore, the average dissipated energy of node n_i is E_{n_i} which is expressed by Equations 3.5 to 3.7. Assuming the following strategy is met: each sensor node generates an equal amount of traffic with a transmission power of P_{tx}. Using the nearest-neighbour routing approach, the traffic is relayed over a route *r* through a chain of *N* adjacent sensor nodes with equal spacing. Similar approaches were addressed in [34,139,144,145,155]. However, they neglect the complexity of the wireless channel. All energies in the following derivations are normalised by the transmitting power P_{tx}.

$$E_{n_i} = ((\sum_{j=1}^{i}[P_{n_j,r,n_{j+1}}]) + HC \times P_{n_i,r,BS}) \tag{3.5}$$

Recalling that $P_{n_i,r,n_{i+1}} = 1 - P_{n_i,r,BS}$.

$$E_{n_i} = (\sum_{j=1}^{i}[1 - P_{n_i,r,BS}] + (N+1-i) \times P_{n_i,r,BS})$$

$$E_{n_i} = (\sum_{j=1}^{i}[1] - \sum_{j=1}^{i}[P_{n_j,r,BS}] + (N+1-i) \times P_{n_i,r,BS})$$

$$E_{n_i} = (i - \sum_{j=1}^{i-1+1}[P_{n_j,r,BS}] + (N+1-i) \times P_{n_i,r,BS})$$

$$E_{n_i} = (i - (\sum_{j=1}^{i-1}[P_{n_j,r,BS}] + \sum_{j=1}^{1}[P_{n_j,r,BS}]) + (N+1-i) \times P_{n_i,r,BS})$$

$$E_{n_i} = (i - \sum_{j=1}^{i-1}[P_{n_j,r,BS}] - P_{n_i,r,BS} + (N+1-i) \times P_{n_i,r,BS})$$

$$E_{n_i} = (i - \sum_{j=1}^{i-1}[P_{n_j,r,BS}] + P_{n_i,r,BS}((N+1-i)-1) \quad (3.6)$$

Recalling that $i = N+1-HC$

$$E_{n_i} = (N+1) - (HC + \sum_{j=1}^{i-1}[P_{n_j,r,BS}]) + (HC-1)P_{n_i,r,BS} \quad (3.7)$$

In Figure 3.7, node n_N is the closest to the base station and consumes the maximum amount of energy for transmitting and relaying all packets from its downstream child sensor nodes to the base station *BS*. Sensor node n_N can also transmit directly to the base station with one-hop link reliability probability $P_{n_N,r,BS} = 1$. From an energy standpoint, the energy consumption of node n_N can be estimated in Equation 3.8 in terms of the single-hop link reliability probability $P_{n_i,r,BS}$ between node n_N (where, i=N and HC=1) and the base station *BS*.

$$E_{n_N} = (N - \sum_{j=1}^{N-1}[P_{n_j,r,BS}])$$

$$E_{n_N} = (N - [P_{n_1,r,BS} + P_{n_2,r,BS} + \ldots + P_{n_{N-1},r,BS}]) \quad (3.8)$$

In order to moderate the energy dissipation of all these N-1 sensor nodes, that are participating in constructing the preselected multihop route r from node n_1 node n_{N-1}, to the energy dissipation of node n_N, the sum of (N-1) one-hop link reliability probability of $P_{n_i,r,BS}$ or $1 - P_{n_i,r,n_{i+1}}$ must be smaller than the value of *order of N* "O(N)". The (N-i+1) link

reliability probabilities can be estimated by solving equation 3.6 using two dimensional matrices for (N-i+1) hops along the route r in Equation 3.9.

$$\begin{bmatrix} N & 1 & \dots & 1 \\ 0 & (N-1) & \dots & 1 \\ : & : & \dots & : \\ 0 & 0 & \dots & 2 \end{bmatrix} \begin{bmatrix} P_{n_1,r,BS} \\ P_{n_2,r,BS} \\ : \\ P_{n_{N-1},r,BS} \end{bmatrix} = \begin{bmatrix} N-1 \\ N-2 \\ : \\ 1 \end{bmatrix} \tag{3.9}$$

3.6.3 Packet Relaying Probability Model

If we assume that a given number of sensor nodes are distributed arbitrarily and each node n_i sends a packet at a given transmitting power P_{tx} and has a multihop route r_j of $HC_{n_i,r_j,BS}$ hops to the base station *BS*. $HC_{n_i,r_j,BS}$ is the hop count of the route r_j between n_i and *BS*, which is greater than or equal *zero*. If sensor node n_i can't reach the base station *b*, $HC_{n_i,r_j,BS}$ is set to *infinity*. Based on link reliability as a primary routing cost metric, the *likelihood of relaying a packet* originated at node n_i is expressed in Equation 3.10, which is the *probability* $P_{n_i,r_j,n_{BS}}$ of relaying a data packet towards the base station *BS* through the selected route r_j. Where $lq_{n_i,r_j,n_{i+1}}$ is the quality of the link between sensor node n_i and its current parent (upstream neighbour node) n_{i+1} along route r_j. $lq_{n_i,r_j,n_{i+1}}$ is characterised based on the channel gain of this link. The cost function is the total required transmitting power P_{tx} for transmitting the given number of packets with an acceptable link quality. From link reliability point of view, $P_{n_i,r_j,n_{BS}}$ counts for the readiness of node n_i of forwarding a packet based on link quality $lq_{n_i,r_j,n_{i+1}}$ to its intended upstream neighbour sensor node n_{i+1} that receives the packet and relay it towards the base station *BS*.

$$P_{n_i,r_j,BS} = \begin{cases} \prod_{HC_{n_i,r_j,BS}} P_{n_i,r_j,n_{i+1}}(lq_{n_i,r_j,n_{i+1}}, P_{tx}), & 0 < HC_{n_i,r_j,BS} < \infty \\ 0, & HC_{n_i,r_j,BS} = \infty \\ 1, & HC_{n_i,r_j,BS} = 0 \end{cases} \tag{3.10}$$

In case the base station is unreachable, $P_{n_i,r_j,n_{BS}}$ is approaching zero as the cost or route r_j in terms of hop count $HC_{n_i,r_j,BS}$ is perpetuating to infinity. Otherwise, $P_{n_i,r_j,n_{BS}}$ is normalised to one and the cost $HC_{n_i,r_j,BS}$ is zero; this means that no packets are being sent or relayed by sensor node n_i.

3.6.4 Energy Balancing Model

To consider the benefit of energy balancing of the RLBR scheme, the energy discharge behavior is gauged in terms of energy depletion rate $R(E_{n_i})$ of a sensor node n_i. Thus sensor node n_i has an average dissipated energy of E_{n_i}. The lifetime ratio of a sensor node n_i is proportional to its remaining energy capacity. The initial residual energy capacity of this sensor node's battery is set to $C_{n_i}(t_0)$ and evenly divided into L energy levels. Each energy level has energy of $C_{n_i}(t_0)/L$. A small number of energy levels leads to a low performance in energy balance, for instance, if there is only one energy level, all routes will always have the same energy level and the most reliable route will be used frequently until it is exhausted. At the beginning, it is assumed that the initial energy capacity of all sensor nodes is identical. When sensor node n_i with a given residual energy capacity level relays packets, its energy capacity decreases to lower levels by an energy depletion rate of $R(E_{n_i})$. The routing scheme reduces the workload on relay sensor nodes whose energy capacity reaches the lowest level. In the network initialization stage, sensor nodes with the best *link reliability probability* $P_{n_i,r,n_{i+1}}$ are considered first based on link quality estimated values, whereas sensor nodes with the highest residual energy capacity levels are considered afterwards. Thus, a parent is selected if it offers a reliable route, but when the aggregating or relaying load of a sensor node increases, the remaining battery capacity of this node is considered as the prime metric in the parent selection process. This allows for constructing paths along which all sensor nodes have the actual available battery

capacity levels exceeding the lowest energy level. The cost function selects the route that requires the lowest energy per packet. If it has no choice with equal cost paths, then it picks that path which maximizes the minimum battery level by utilizing the principle of max-min cost function of the *Conditional Max-Min Battery Capacity Routing* (CMMBCR) as stated in [71]. This strategy minimizes the variance in energy capacity levels for a uniform energy depletion to avoid relaying sensor nodes from sudden running out of energy and disrupting the network. Hence, routes should be chosen such that the variance in battery levels between different routes is reduced.

The energy depletion rate $R(E_{n_i})$ at which the residual energy capacity C_{n_i} of sensor node n_i is reduced can be expressed in Equation 3.11 which is only valid for $t_{n_i,r,i+1} > 0$. Where $t_{n_i,r,i+1}$ is the time spent for sensor node n_i for transmitting or forwarding a packet to sensor node n_{i+1} over route r. While $P_{n_i,r,n_{i+1}}$ is the probability of forwarding a packet to the next hop n_{i+1} through the route r, $P_{n_i,r,n_{i-1}}$ is the probability of receiving a packet from node n_{i-1} through the route r. Hence, $R(E_{n_i})$ is a bidirectional function of the energy expenditure for relaying the projected network traffic by transmitting and receiving packets at a given energy depletion rate of $(P_{n_i,r,n_{i+1}} \times E_{n_i})_{tx} / t_{n_i,r,n_{i+1}}$ and $(P_{n_i,r,n_{i-1}} \times E_{n_i})_{rx} / t_{n_i,r,n_{i-1}}$ respectively. $t_{n_i,r,i-1}$ is the time spent for node n_i for receiving or aggregating a packet from node n_{i-1}. Assuming that transmitting time $t_{n_i,r,n_{i+1}}$ equals receiving time $t_{n_i,r,n_{i-1}}$ for packets of the same size. As the link reliability estimations require link layer acknowledgements, the energy required for transmitting or receiving a packet will also include the energy required for acknowledging this packet. $R(E_{n_i})$ is measured in *energy unit per second*.

$$R(E_{n_i}) = \frac{\left((P_{n_i,r,n_{i+1}} \times E_{n_i})_{tx} + (P_{n_{i-1},r,n_i} \times E_{n_i})_{rx}\right)}{t_{n_i,r,n_{i+1}}} \tag{3.11}$$

From an energy efficiency point of view, the *functional lifetime* T_{n_i} of an individual sensor node n_i, in which sensor node n_i can participate in constructing the route r with sufficient energy, is obtained by dividing the *initial energy capacity level* $C_{n_i}(t_0)$ by *energy depletion rate* $R(E_{n_i})$ as in Equation 3.12.

$$T_{n_i} = \frac{C_{n_i}(t_0)}{R(E_{n_i})} \tag{3.12}$$

Given these assumptions, the maximum relay sensor node's lifetime T_{n_i} is achieved by minimising ($1/T_{n_i}$). Logically the maximum lifetime of a given route r is determined by the weakest intermediate or relaying sensor node, which is that with the highest cost. For a wireless sensor network of m randomly deployed sensor nodes, where every sensor node has k available routes towards the base station, the entire network's functional lifetime T_{WSN} can be maximized as in Equation 3.13. The aim here is to derive a general formula for network lifetime which holds independently of the underlying network model. It allows identifying key parameters that affect network lifetime without worrying about specific network settings or application. As a result, it can provide design guidelines applicable to various types of sensor networks. From an energy cost point of view, the *residual energy capacity level* C_{n_i} of a sensor node n_i defines the refusal or readiness of this node to respond to route requests and forward data traffic. The maximum lifetime of a given route is determined by the weakest intermediate sensor node, which is that with the highest cost.

$$T_{Network} = C_{n_i}(t_0) \sum_{j=1}^{k} \sum_{i=1}^{m} \left[\frac{t_{n_i, r_j, n_{i+1}}}{(P_{n_i, r_j, n_{i+1}} \times E_{n_i})_{tx} + (P_{n_i, r_j, n_{i-1}} \times E_{n_i})_{rx}} \right] \tag{3.13}$$

3.7 Preliminary Analysis of Routing and Computation Overhead

A preliminary analysis of the routing/computation overhead introduced by the RLBR scheme is conducted in this section before the routing scheme is deployed onto the target mote system. This preliminary analysis is based on sensor node emulation at the instruction cycle level. The communication process estimation of an individual mote is analysed and modelled to embody the additional processing/computation overhead required by the RLBR scheme on a sensor mote. The results of this estimation model will be used later in the following chapters to calculate the *total average dissipated energy* for communications of the network in the WSN testbed.

Since the existing network simulators such as NS-2 [7], Prowler [8], and OMNeT++ [75] do not model the hardware of targeted sensor platforms with sufficient accuracy, they need to explicit support for the hardware being simulated and evaluated network protocols need to be first implemented specifically for the simulation platform or the software development environment. This will significantly restrict the way in which the network protocols may be evaluated in real WSNs testbeds. Thus, emulating the implementations directly on the target platform appropriately reflected the operating constraints of the hardware of sensor platform. There are many existing cycle-based emulators, such as *Avrora* [111], *PowerTossim* and *Atemu* [16-19]. Based on sensor mote emulation at the CPU cycle level, the *Avrora* emulator [111] is the most recognized tool of this approach. *Avrora* is a *scalable cycle-accurate instruction-level emulator* allows the emulation of the Atmel AVR microcontroller based sensor node platforms such as Crossbow's Mica2 motes.

Therefore, the preliminary analysis of this section estimates the *CPU cycle profile* using *Avrora* emulator which is the AVR simulation and analysis tool for embedded sensing programs implemented in Java [111]. Using *Avrora* emulator, the code is cross-compiled

for the sensor node architecture and is executed on an emulated processor. This provides an accurate measure of protocol modelling fidelity. This approach has the advantage that it can run the sensor code-base without any modifications either to programming syntax, compiler or the binaries compared to would run on the hardware of the target sensor platform. The sensor node applications utilizing the radio communication standard such as Mica2's CC1000 radio model included in *Avrora* can be tested on the emulator.

However, the *Avrora simulator* [111] was developed for the AVR Atmel microcontroller on Mica2 motes and do not support the newer TelosB platform. At the time of conducting this work, no model compliant with the IEEE 802.15.4 standard [87] is available in Avrora. This means that sensor node applications utilizing compliant radio communication model, e.g., TelosB's CC2420 radio [78], cannot be directly tested on *Avrora* emulator. Therefore, in order to enable an accurate emulation of IEEE 802.15.4 standard for the RLBR scheme on TelosB motes, the link quality indicator (LQI) [87] module has been added into the original version of *Avrora* emulator.

LQI is a hardware-based link reliability metric introduced by IEEE802.15.4 standard specification [87], which measures the error in the incoming modulation of successfully received packets which pass the CRC check sums. LQI can be measured by IEEE802.15.4-compliant radio chips such as Chipcon's CC2420 RF transceiver [78] on TelosB motes used here in the outdoor experiments. LQI is actually the Chip Correlation Indicator (CCI) and its values are related to the chip error rate. Every received packet must be stamped with LQI value as stated by IEEE802.15.4 standard to indicate the quality of the link at the time of packet reception. Instead of using each beacon's LQI individually, LQI measured values are averaged in each time frame t to calculate the *averaged time-varying link quality*. This can be calculated using Equation 3.14 to better reflect LQI measurements as this controls the effect of the previously estimated on the

new estimated LQI values for short-term link estimations. Where α is the history control factor ranges from 0 to 1 in time frame t.

$$lq_t(\alpha) = \alpha \times lq_{t-1} + (1+\alpha) \times lq_t \tag{3.14}$$

According to the IEEE802.15.4 standard, the LQI value must be an integer that is calculated over a field 8 bits of the packet format following the start frame delimiter (SFD) and uniformly distributed and bounded within the interval [0, 0xFF]; with 0 being the lowest LQI level L_{low} and 0xFF, i.e., 255 being the highest LQI level L_{high}. The link quality value of a received packet is measured as a combination of received signal strength and signal-to-noise ratio. IEEE802.15.4 sets the nominal transmit power of the CC2420 RF transceiver to 0dBm and the receiver sensitivity to −94dBm. CC2420 RF transceiver provides LQI values ranges from 50 to 110dBm and correspond to minimum and maximum quality packets respectively. The relationship between the integer values of the time-varying link quality (L_i) at level i and the power of the received packets (P_{rx}) in dBm can be calculated by $L_i=2.712766P_{rx}$ + 255. For example, if the power of the received packet is set at the default value at which P_{rx} = 0dBm, L_i results in an integer value of L_{high} = 255; on the other hand, if the power of the received frame is set at the receive sensitivity of P_{rx} = -94dBm of the CC2420 RF transceiver, L_{low} = 0. The value of L_i for all received signal power levels in the range between the receiver sensitivity of −94dBm and the nominal transmit power of 0dBm is bounded within the interval [0, 255]. To validate LQI values, the resulted decimal fraction of L_i needs to be curved upward or downward in relation to the value of $(L_i\text{-}L_{i\text{-}1})/2$ to keep integer values of LQI, i.e., LQI = L_i if LQI > $(L_i\text{-}L_{i\text{-}1})/2$; otherwise LQI = $L_{i\text{-}1}$.

In this preliminary analysis, the communication process overhead is estimated using a single mote as a base station surrounded with a number of neighbours in order to develop the average *dissipated energy model* of the mote system during its operations due to employing the RLBR scheme. The total network wide energy expenditure is due

to: *route searching*, *transmitting*, *receiving/overhearing*, *idle*, and *CRC failed packet reception*. The route searching phase includes both parent selection and routing table update processes. However, the amount of power used for radio communication in wireless sensor nodes typically dominates that used in computation [36]. The *Average Dissipated Energy* measures the ratio of total dissipated energy per sensor node in the network to the number of distinct events received by the base station. This metric computes the average work done by a participating sensor node in delivering data of interest to the base station. This metric also indicates the overall lifetime of sensor nodes. To calculate the routing scheme's energy expenditure per sensor node, the energy is calculated repetitively for multiple runs each run is conducted for a given period of time using Equation 3.15. Where V_{batt} is the battery voltage level; I_{drawn} is the current drawn and consumed by the RLBR on a mote system during different routing tasks; *Cycle Time* is the time spent per CPU cycle and depends on the type of the mote system, i.e., (1/7.3828)µs and (1/8)µs for Mica2 [14] and TelosB [86] respectively; and *Cycle count* is the number of CPU cycles spent during mote's tasks is counted using *Avrora* emulator based on the target mote system, i.e., Mica2 and TelosB motes.

$$Average\ Dissipated\ Energy = V_{batt} \times I_{Drawn} \times Cycle\ Time \times Cycle\ Count \tag{3.15}$$

Figures 3.8 and 3.9 present the *Avrora* emulation results of the routing/computation overhead caused by RLBR in term of cycle count for individual Mica2 and TelosB motes respectively in two separate runs. In these two runs, both Mica2 and TelosB motes are surrounded by 10 neighbours respectively. In terms of scalability, results stated in [111] shows that *Avrora* can achieve better real hardware system performance for networks of smaller numbers of nodes. The energy consumption is estimated during different tasks of transmitting and receiving processes in terms of CPU cycles profile. The number of cycles caused by the additional computation overhead is required for constructing the routing tree while transmitting a packet (including preamble), and for

acquiring the local routing information for route maintenance and updating routing tables while receiving a packet (including preamble). However, the cycle count required for parent selection and updating routing table is still very small compared to the actual cycle required for preamble and packet transmission/reception.

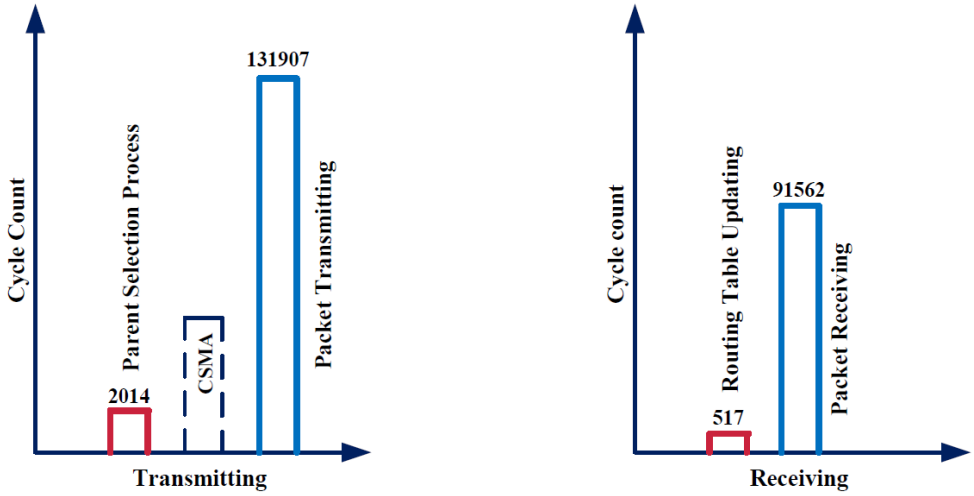


Figure 3.8 Routing/Computation Overhead Estimations on Mica2

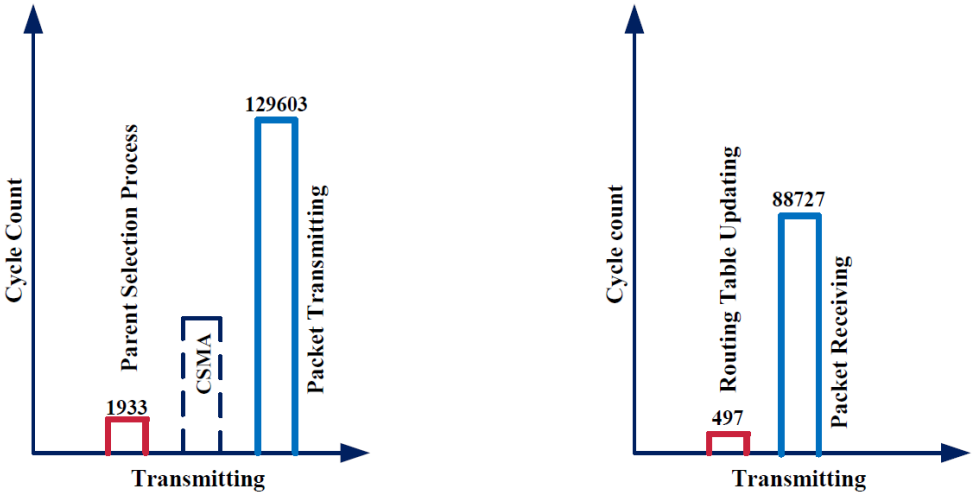


Figure 3.9 Routing/Computation Overhead Estimations on TelosB

The Mica2 mote system (e.g., CPU and radio), used in the indoor experiments of Chapter 3, with a 3V power supply draws a current (I_{drawn}) of approximately 25.4mA while transmitting at default power (0dBm), up to 15.1mA when receiving or overhearing, and 8mA with active CPU and idle/sleep radio [42]. Recalling equation

3.15, the estimated amounts of energy consumed by a Mica2 mote, as presented in Table 3.1, for parent selection process, route maintenance, transmitting, and receiving are 6.55, 1.68, 1361.45, 561.81μJoles respectively. Where V_{batt} is 3Volts and cycle time of Mica2's CPU is (1/7.3828)μs [14]. These results show that for each packet sent, the parent selection process of the routing scheme consumes a minimal amount of energy of 6.55μJoules compared to 1361.45μJoules requires for transmitting a packet. In addition, the process of route maintenance and updating the routing table needs only 1.68μJoules compared to 561.81μJoules needed for receiving a packet. On Mica2 mote, the RLBR scheme causes a minor computation overhead hence it introduces a little more energy consumption of %0.48 for parent selection and %0.30 for route maintenance.

TABLE 3.1 ESTIMATED ENERGY CONSUMPTION ON MICA2 MOTE SYSTEM

RLBR's Task	**Estimated Energy Consumption (μJoles)**
Parent Selection Process	6.55
Route Maintenance	1.68
Transmitting	1361.45
Receiving	561.81

Since the TelosB motes, used in the outdoor experiments of Chapter 4, use a different processor and transceiver than the Mica2 motes, the new components have influence on the energy consumption. The network communication specification followed by Mica2 and TelosB is different and is not completely compatible even within the same frequency band. Both have two different radio specifications. While Mica2 uses a CC1000 radio, TelosB uses the IEEE802.15.4-compliant CC2420 radio. These two radios use different encoding formats and different error correction schemes. For example, the CC2420 radio chip is more intelligent than CC1000 as it does some of the processing itself while the CC1000 requires the Mote's CPU to perform the same function. The TelosB mote

system (CPU and radio) with a 3V power supply draws a current (I_{drawn}) of approximately 19.5mA while transmitting at default power (0dBm), up to 21.8mA when receiving or overhearing, and 1.8mA with active CPU and idle/sleep radio which is much smaller than the one for Mica2 which is 8mA [42]. Recalling equation 3.15 again, the estimated amounts of energy consumed by a TelosB mote, as presented in Table 3.2, for parent selection process, route maintenance, transmitting, and receiving are 1.30, 0.34, 947.72, 725.34μJoules respectively. Where V_{batt} is 3Volts and cycle time of TelosB's CPU is (1/8)µs [86]. These results show that for each packet sent, the parent selection process of the routing scheme consumes a slightly small amount of energy of 1.30μJoules compared to 947.72μJoules requires for transmitting a packet. In addition, the process of route maintenance and updating the routing table needs only 0.34μJoules compared to 725.34μJoules needed for receiving a packet. On TelosB mote, the RLBR scheme causes a minor computation overhead hence it introduces a little more energy consumption of %0.14 for parent selection and %0.05 for route maintenance.

TABLE 3.2 ESTIMATED ENERGY CONSUMPTION ON TELOSB MOTE SYSTEM

RLBR's Task	**Estimated Energy Consumption (μJoles)**
Parent Selection Process	1.30
Route Maintenance	0.34
Transmitting	947.72
Receiving	725.34

3.8 Conclusion

Ad hoc sensor networks require a highly dynamic, adaptive reliable routing scheme to deal with the high rate of topology changes [157]. Besides that, the energy consumption rate needs to be evenly distributed among sensor nodes, and efficient

utilization of battery power is essential. A distributed, reliable, load balancing routing (RLBR) scheme is proposed in this chapter to face the dynamics of the real world of resource-constrained wireless sensor networks. The preliminary analysis of the computation overhead caused by the routing scheme shows that the RLBR scheme adds only a trivial additional computation overhead and a minimal increase in energy consumption for parent selection process and route maintenance. Since testing and debugging routing protocols on different platforms, testbeds and environments is vital for experimental validation of the routing efficiency, the RLBR scheme is implemented and evaluated in the subsequent chapters on a variety of experiments based on real medium-scale indoor and outdoor testbeds ranging in size from 20 to 30 wireless sensor motes as explained in details in the chapters 4 and 5 respectively. In addition, RLBR scheme is also further extended and tested using intensive computer simulations of large-scale networks in chapter 6.

Chapter 4
Indoor Testbed Experiments

4.1 Background and Motivations

Reliable and energy efficient routing is a critical issue in Wireless Sensor Network (WSN) deployments. Many approaches have been proposed for WSN routing, but sensor field implementations, compared to computer simulations and fully-controlled testbeds, tend to be lacking in the literature and not fully documented [72, 74]. Typically, WSNs provide the ability to gather information cheaply, accurately and reliably over both small and vast physical regions. WSNs are about collecting data from unattended physical environments. Although WSNs are being studied on a global scale, the major current research is still focusing on pure simulations experiments. In particular for sensor networks, which have to deal with very stringent resource limitations and that are exposed to severe physical conditions, real experiments with real applications are essential. In addition, the effectiveness of simulation studies is severely limited in terms of the difficulty in modeling the complexities of the radio environment, power consumption on sensor devices, and the interactions between the physical, network and application layers.

While the majority of WSN-related research activities have used open-source network simulators such as NS-2 [7], Prowler [8], and OMNeT++ [75], others have used well-controlled indoor remote access testbeds such as Motelab [72] and Tutonet [74] to demonstrate the benefits of employing various routing algorithms' scalable performance. However, simulations and remote access testbeds have limitations in fully emulating real-world low power WSN characteristics. In addition, sensor nodes are prone to failure and various adverse factors that are unpredictable and difficult to capture in simulations. Therefore the work done in this chapter has been conducted on a real-world WSN by

taking in account the unpredictable behaviour of wireless signal propagation, and how it changes spatially, temporally, or with certain environmental conditions; and how the real sensor device's inconsistent or erroneous behaviour affects a routing protocol's performance or even a device's rate of energy consumption.

Standalone evaluation of routing efficiency is impracticable, as link dynamics prevent knowing what the best route would be for data dissemination. Therefore, routing efficiency is evaluated as a comparative measure. In this chapter, the proposed routing scheme is mainly benchmarked with the TinyOS routing layer implementation of MintRoute [33,52] as well as with other benchmark protocols on Crossbow's Mica2 platform. The experimental work is conducted in a low-interference indoor environment on a testbed of 20 sensor motes of Crossbow's Mica2 868/916MHz wireless platform [14,35]. Mica2 motes represent the lowest cost wireless sensor platform based on commercial off-the-shelf hardware components. Mica2 motes run the TinyOS programming environment [25]. The standard TinyOS-2.x CSMA B-MAC layer [20] for CC1000 Radio Frequency (RF) transceivers [21] is used as an underlying low-power listening (LPL) MAC protocol.

4.2 Related Work

The indoor experiments conducted in this chapter focus on a distributed, multipath, proactive, mote-oriented routing that mainly employ link-quality estimation [44] using the Received Signal Strength Indicator (RSSI) provided by CC1000 radio on Mica2 motes to evaluate the link qualities from both directions between sensor nodes [20,32]. In the literature, researchers have proposed a number of cost-based reliability-oriented routing protocols specifically for WSNs [83,115]. Among these protocols, MintRoute [33,52] and TinyAODV [134] are widely used protocols built on top of TinyOS [25]. MintRoute has been successfully studied and used in various experiments by researchers

from UC Berkeley [25,52] and in [33,126]. MintRoute focuses on improving link and route reliability through neighbourhood table management. MintRoute is a shortest path protocol and uses a distance-vector approach. It periodically sends out control packets at fixed rate for route maintenance. MintRoute is known to depend on the snooping capability of the MAC layer [33,127] but improperly assumes that intermediate links are stable with independent packet losses. It uses this assumption to derive the necessary sampling window resulting in inaccurate link quality estimations [33,67]. Furthermore, TinyAODV [134] is a WSN multipath routing protocol version of the Ad-hoc On-Demand Distance Vector (AODV) protocol [132]. It sets up a route only when there is a demand and invalidates the route upon receiving an error message or a parent is lost.

In addition, since WSNs are resource-constrained, other routing protocols have long been well-recognised as efficient broadcasting and flooding schemes of data throughout the network. Most of these protocols use a single optimal path for every communications [47,49]. One of the early single-path routing schemes known as Directed diffusion [47] employs application-level data dissemination, path reinforcement and in-network data aggregation in order to improve energy efficiency. However, single-path routing protocols are vulnerable to rapid sensor node depletion of energy and link failures. In the literature, many multipath routing protocols have also been proposed for WSNs. One of these protocols known as Directed Transmission Routing Protocol (DTRP) [128,129] which is a parametric and probabilistic multipath cost-based routing protocol. It is designed based on hop distance and was implemented on Mica2 motes for the purpose of scalability [130,131]. DTRP considers random reliability probability similar to Gossiping protocol [125] which is a probabilistic reliable multicasting Gossip-based routing protocol for ad hoc WSNs, but for non-constant relaying probability per packet. DTRP consumes less energy and has a smaller memory footprint as it requires only the hop count information between the source sensor nodes and the base station, which is

available from the beacons initiated by the base station and relayed by each sensor node [128,129]. However, the above mentioned multipath routing schemes cause a flooding of route requests to the entire network, which increases frequent routing updates, creates considerable communication overhead and results in high packet loss ratio due to collisions [133]. Conversely, the RLBR scheme aims to uniformly balance the traffic load and energy depletion on per-hop basis; thereby improving packet delivery performance and maintaining a better network functionality.

4.3 Implementation Platform: *Mica2 Motes*

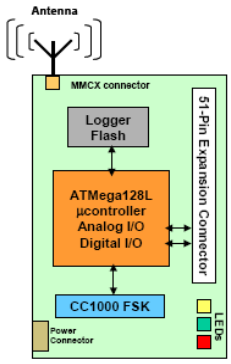
This section investigates the hardware implementation challenges in the tiny resource-constrained wireless sensors platforms in addition to the underlying layers (i.e., the physical and MAC layers).

4.3.1 Platform Details and Experimental Features

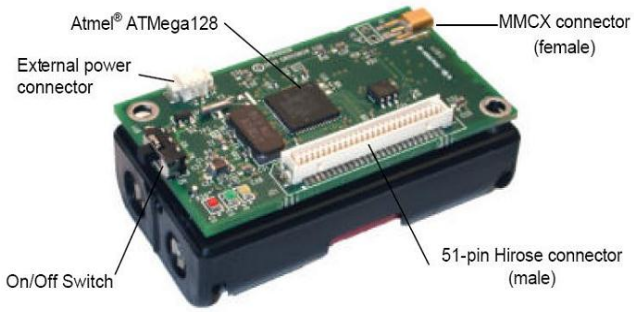
The implementation is based on a real world testbed of wireless sensor nodes, specifically, the UC Berkeley's Mica2 motes which are popular due to their simple architecture, open source development and their commercial availability from Crossbow Technology. The Mica2 Mote Module is the third generation tiny, wireless platform for smart sensors designed specifically for deeply embedded low power sensor networks. Table 4.1 reveals the specifications of a typical radio/processor platform Mica2 868/916MHz (MPR400CB) [14,35] which is powered by AA batteries. Mica2 is built with an 8-bit, 7.3828MHz low-power Atmel ATmega128L processor, 128Kbytes of in-system program memory, 4Kbytes of in-system data memory, and 512Kbytes of external flash (serial) memory for measurements storage. Mica2 motes are equipped with CC1000 radio transceiver and Omni-directional whip antennas. Figure 4.1 shows the overall block diagram of Mica2 mote [14].

TABLE 4.1 CROSSBOW MICA2 MOTE (MPR400CB) SPECIFICATIONS [14,35]

Component	Feature
Processor	8-bit Atmel® ATmega128L Processor (7.3828 MHz)
In-System Program Memory	128 Kbytes
In-System Data memory	4 Kbytes
External Serial Flash Measurements Memory	512 Kbytes
Radio Chip Transceiver	Chipcon CC1000 Radio
Centre Frequency	868/916 MHz, 4 channels
Modulation Format	FSK modulation
Effective Data Rate	38.4 Kbps
Hardware Encoding	Manchester encoded [2:1]
Antenna Type	Omni-directional whip
Transmission Power Range	-20dBm to 5dBm
Max. Packets Rate (100% Duty Cycle)	42.93 Packets/Sec



a. Context Block Diagram



b. External Interfaces

Figure 4.1 Crossbow Mica2 868/916MHz Mote (MPR400CB) [14,35]

Since the above mentioned resources seem unfit for computationally expensive or power-intensive operations, explicit energy saving techniques are necessary to extend battery lifetime as long as possible. The Mica2 radio component when transmitting draws 30% more current than the CPU when it is active [24,15]. Low-power radio operation is necessary to carry out long-term monitoring with sensor network deployments. If the radio and CPU are constantly active, battery power will be consumed in less than a week [16 -19].

The Mica2 Mote features several new improvements over the original Mica Mote. These features make the Mica2 better suited to experimental deployments such as 868/916MHz multi-channel transceiver with extended range, wireless remote reprogramming, wide range of sensor boards and data acquisition add-on boards, and supported by MoteWorks™ platform [23,25] for WSN applications. MoteWorks™ is based on the open-source TinyOS operating system [25] and provides reliable, ad-hoc mesh networking, cross development tools, server middleware for enterprise network integration and client user interface for analysis and configuration. MoteWorks™ [23] enables the development of custom sensor applications and is specifically optimised for low-power and battery-operated networks. MoteWorks™ 2.0 provides a complete software development environment for WSN applications. Included is a collection of flexible software packages that enables both quick-and-easy out-of-the-box deployment of sensor systems for monitoring and alerting, to powerful tools to empower custom development of pervasive sensory networks [9]. The ATmega128L runs MoteWorks™ 2.0 platform from its internal flash memory [23,25].

A single Mica2 mote (MPR400CB) can be configured to run sensing application and the radio communications stacks simultaneously. The Mica2 51-pin expansion connector supports Analog Inputs, Digital I/O, I2C, SPI and UART interfaces. These interfaces make it easy to connect to a wide variety of external peripherals [9,14]. Any Mica2 Mote can function as a base station when it is connected to a standard PC by attaching an interface or gateway board (MIB520) [9]. The base station serves as the traffic sink. A mote interface board allows the aggregation of sensor network data onto a PC or other computer platform and allows for motes programming. There are different modules of serial or USB interface boards (MIB520CA). In addition to supporting Joint Test Action Group (JTAG) code debugging, MIB520-USB gateway supports USB connectivity interface for the Mica2 Motes for communication, data transfer and in-system

programming [9]. Finally, Mica2 Motes can be integrated with a sensor board (MTS) or data acquisition (DAQ) board (MDA) that support a variety of sensor modalities [9]. MTS400 series supports environmental monitoring (e.g., Ambient light, relative humidity, temperature, 2-axis accelerometer, and barometric pressure) for Mica2 with built-in sensors. MTS420 series also supports a Global Positioning system (GPS). The MTS/MDA boards are not used here as data packets are generated by the application.

4.3.2 Underlying Layers (The physical and MAC layers)

TinyOS operating system [24,25] provides a variety of tools, including a programming environment and a complete network stack on wireless sensor node platform. This stack contains a basic radio driver: physical and link layer protocols, and an adjustable energy efficient MAC layer, i.e., B-MAC with low-power listening (LPL) scheme, the default TinyOS MAC protocol developed at the UC Berkeley [21]. TinyOS CC1000 [32] has 128bytes maximum MAC frame size and employs Frequency Shift Keying (FSK) modulation Scheme.

At the Physical Layer, Mica2 mote uses a low powered radio the Chipcon CC1000 RF transceiver [20] which is a single chip, very low-power, Multichannel radio frequency transceiver supporting 23 different power levels and operates in frequency range 300 to 1000MHz. The Mica2 (MPR400CB) features a digitally programmable/tuneable output radio with power levels adjustable from -20dBm to +5dBm centred at the 868/916MHz setting within two frequency regions: (868-870MHz) and (902-928MHz). However, CC1000 power levels are not distributed evenly across this range and the default output power is 1mW (0 dBm) at level 14. The CC1000 radio uses Frequency Shift Keying (FSK) modulation with an effective data rate or throughput of 38.4Kbps (76.8KBaud). The CC1000 radio has an integrated bit-synchroniser and uses a hardware-based

Manchester encoding scheme to encode the transmitted data. It also uses the linear received signal strength indicator (RSSI) to measure the strength of the received signal.

Due to the highly dynamic nature of WSNs, the inherent advantages of contention-based MAC protocols, e.g., B-MAC [21], makes them the preferred choice, and they have been widely adopted in WSNs. B-MAC was preferred for the MAC layer for the implementation of the proposed routing scheme. Although B-MAC protocol is not as energy-efficient as schedule-based protocols, it has several advantages as well as most CSMA/CA. First, B-MAC scales more easily across changes in sensor node density or traffic load. Second, it is more flexible as topologies change, because there is no requirement to form communication clusters as in cluster-based routing protocols. Third, it is asynchronous and does not require fine-grained time-synchronisation. Instead, each packet is transmitted with a long enough preamble so that the receiver is guaranteed to wake up during the preamble transmission time. It also employs an adaptive preamble sampling scheme to reduce the duty cycle and minimise idle listening without overhearing avoidance. Before a sender sends out a packet to a receiver, it will first send a preamble long enough for all its neighbours to wake up, detect activity on the channel, receive the preamble, and then receive the packet.

In addition to the receiver, all the other neighbours of the sender will receive the packet, even though the packet is not addressed to them, i.e., overhearing. In this situation, the helpful information used (e.g., link quality estimations and node *id*) for routing decisions in the proposed scheme is being embedded in the packet header. When a sensor node receives a packet not addressed to itself, it can retrieve this helpful information from the packet header before dropping the packet. Finally, B-MAC has an awareness of the protocols that run above it and offers control to the upper layer protocols, allowing the routing and application layers to change parameters like the low-power listening duration

or the number and type of retransmissions used. Thus, B-MAC allows each node to overhear packets transmitted by its neighbours; this allows routing layer to employ snooping for the sake of link quality estimation, and in-network data aggregation.

B-MAC also provides an interface by which the application can adjust the sleep schedule to adapt to changing traffic loads which is an important MAC feature for time-sensitive data aggregation provided by the proposed routing scheme. B-MAC does not perform link-level retransmission or hidden node avoidance using RTS/CTS schemes as it has been assumed that such schemes will be implemented at higher layers if necessary. On Mica2 sensor nodes with CC1000 radios, B-MAC supports synchronous acknowledgments that require only a few extra bit times on the end of each packet to transmit. This depends on the ability of the sender and receiver to quickly switch roles at the end of a packet transmission and remain synchronized before any additional sender can sense an idle channel and begin transmitting. However, these acknowledgements can be disabled and implicit acknowledgements are used as stated in [31]. Moreover, B-MAC uses the energy detect indicator as a carrier sense mechanism [22] which is common to many existing radios. It is based on RSSI readings obtained from the radio front end. B-MAC is a packet-collision avoidance scheme and integrates a power management scheme within the MAC protocol that utilizes low power listening and an extended preamble to achieve low power communication. B-MAC was originally developed for bit streaming radios like Mica2's Chipcon CC1000 bit-level radio, which provides low-level access to the individual bits received by the radio [20].

Hence, B-MAC can generate long preambles with CC1000 radio but the recommended preamble length in B-MAC is 100ms [21], which is used in the deployed WSN experiment. Even though the official version of B-MAC suffers from the long preamble that dominates the energy usage, the modified version of B-MAC, provided by TinyOS

[25], has been shown to outperform other MAC protocols, and has been carefully tuned for the CC1000 radio used on Mica2 motes. It has been claimed by the authors of B-MAC that, B-MAC performs well by surpassing existing MAC protocols in terms of throughput that consistently delivers 98.5% of packets, latency, and for most cases energy consumption [21].

4.4 TinyOS-Based Programming Environment

The indoor implementation is carried out using the low-power Mica2 (MPR400CB) wireless sensor network platform running the component-based operating system TinyOS [24,25] which is written in an event-driven language called Network Embedded Systems C-like language (nesC) [11-13]. Since TinyOS-1.x version has several differences from TinyOS-2.x, TinyOS-2.x version is not fully backward compatible with version TinyOS-1.x. [24,27]. Hence, the recent official stable release TinyOS-2.1.0 [25] that supports different wireless networked embedded platforms including Mica2 and TelosB is used for all indoor experiments of this chapter as well as for all outdoor experiments of the next chapter.

4.4.1 Component-Based Programming

This section introduces the Operating System, specifically, TinyOS-2.x [10,24,25], used for the experimental work of this thesis as a firmware of sensor nodes and the base station. TinyOS is the de-facto operating system and programming environments for sensor motes. Typically, TinyOS is an open source component-based real-time operating system specifically designed for embedded WSNs, which was initially released in year 2000 under Berkeley Software Distribution (BSD) licenses [25,27]. TinyOS is a first-in-first-out (FIFO) scheduler, in which the interrupts are handled immediately while background tasks are rescheduled, that is put on a queue and executed when there is no other task being executed [27].

TinyOS is implemented using the event-driven programming language as a set of cooperating tasks and processes. TinyOS-2.x is supported by the component-based programming model of nesC-1.2.8. The nesC supports a programming model that integrates reactivity to environment, concurrency and communication [12,13]. TinyOS applications are a collection of *components* wired or linked together to form an executable module. TinyOS defines a number of concepts that are expressed in nesC. First, nesC applications are built out of *components* with well defined bidirectional *interfaces* (*command* and *events*). Second, nesC defines a *concurrency model*, based on *tasks* and *hardware event handlers* and detects data races at compile time.

Figure 4.2 shows the basic building blocks of nesC application (*Interfaces components and Concurrency Modules*). *Interfaces Components* can be classified as *provides* and *uses interfaces* components [27]. While a *provide interface* is a set of methods that calls upper layers components, *uses interface* is a set of methods that calls the lower layer components. *Interface component* defines or declares a set of functions called *commands* provided by the interface provider, and another set of functions called *events* used by the interface user. *Concurrency Module* (Also known as *Implementation Component*) has two internal threads: *tasks* and *hardware event handlers* which are implemented by the *concurrency control module*.

```
//Example of nesC Application Methods.
// Interfaces Components: VoltageC
  provides
      interface StdControl;
      interface Energy;
   uses
      interface Voltage;
```

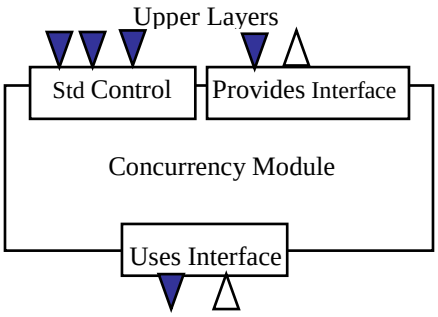


Figure 4.2 Basic Blocks of nesC Application

In nesC, *implementation components* can be either *modules or configurations*. *While a Module* is a nesC component consisting of application code written in a C-like syntax, a *Configuration* (also known as *Wiring*) is a component that wires or assembles other components together. Every application has a single top-level configuration that specifies the set of components in the application and how they invoke another by connecting interfaces of existing components [24,25,27]. *TinyOS concurrency module* executes only one program consisting of a set of components threads (*tasks* and *hardware event handlers*). While a *task* doesn't preempt another *task* as they are scheduled to be executed and buffered into a single queue, *hardware event handlers* are executed in response to a hardware interrupt as they may preempt the execution of a *task* and other *hardware event handler*. Commands and events that are executed as part of a hardware event handler must be declared as *async*. To build a TinyOS application some files are required such as the *".h" header file* which contains the definition of constants and structures; the *".nc" configuration* and *implementation files* where the *interfaces* are liked; the *".nc" module file* where the main code is written; and the *"make"* file which contains the compilation rules to install and download the code into the destined wireless platform, e.g., Mica2 or TelosB motes.

4.4.2 Protocol Implementation

The experimental implementations in this thesis use various APIs and libraries provided by TinyOS-2.x as well as the implemented TinyOS modules of the proposed routing scheme. TinyOS can be installed in different operating systems environments like Windows, Mac or Linux. However, Linux Ubuntu is chosen here to install TinyOS on. The entire packages of TinyOS can be installed through *Synaptic Package Manager*, a graphical user interface (GUI) to easily install, remove, configure or upgrade software packages based on the *apt-get* command line tool. But it is also possible to install them

through the shell by using the *apt-get* commands. By default, most of the packages are included in the repositories, but in the case of TinyOS it is necessary to add a third party repository in order to install the package through Synaptic or commands. TinyOS package comes with libraries of *nesC* and *Java* languages in addition to Graph Visualization (*Graphviz*) software. *Graphviz* is a package of open source tools for drawing graphs based on scripts of *nesC* which is the programming language used to build applications in TinyOS. Once coding is completed and debugged, there are many commands that are used to build and install TinyOS applications into the motes such as "*motelist*" command which lists which USB ports have devices attached to it; "*make platform"* command compiles the TinyOS program but does not download it into the mote, it just create a TinyOS image as it is only used to check possible errors before starting installing on the sensor mote; and "*make platform install,ID bsl,/dev/ttyUSBx"* command compiles and downloads the program into the mote attached into USB port numbered with *x*, after few seconds the mote will start executing the code. "*reinstall"* is used instead of "*install"* of make command to install and download the same code repeatedly into all motes used in the experimental testbeds.

The TinyOS modules of the proposed routing scheme are shown in Figure 4.3. The implemented modules are connected by bidirectional interfaces (commands and events). Commands are sent from the Routing Engine module, and events are sent in the opposite direction towards the Routing Engine module. The Routing Engine module is designed to collect sensor readings and relay them to a single base station. It generates dummy data packets at selected source sensor nodes. In order to track those generated data packets, the source sensor node's *id* is inserted into each generated data packet.

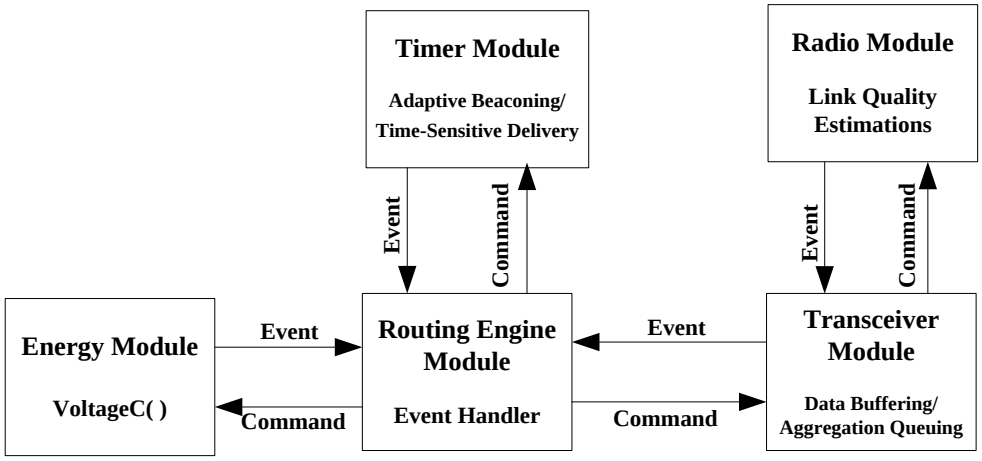


Figure 4.3 The Implemented TinyOS Modules of the RLBR Scheme

When a data packet is either generated at a source sensor node or received at a relaying sensor node, the Timer module sends an event to the event handler in the Routing Engine module. The event handler generates a control packet and fills it with the routing function parameters before sending a command along with this packet to the Transceiver module. The control packet provides the Transceiver module with all necessary information to relay the data packets towards the base station. The Transceiver module buffers the control packet and abstracts all parameters of the routing function from the application. The Transceiver module also handles the link quality estimations with the other routing parameters of the parent selection by sending control packets to the radio module (i.e., Mica2's CC1000 radio module for indoor experiment). The radio module sends back an event to the Transceiver module requesting the data packet, and the data packet is returned to the radio module to be relayed and sent over the wireless medium toward the base station.

When a data packet is received by the radio module, it is buffered in the Transceiver module towards the Routing Engine module. The routing probability value of the routing scheme is calculated based on the locally heard information and the routing parameters embedded into the relayed packet. This value is checked against a random number

generated by a probabilistic random function generator in the Transceiver module to determine whether or not the packet will be relayed. The routing probability value is also a packet variable and depends on the importance of the packet data (e.g., bounded packet delivery deadline). This value can be adjusted at the source sensor node at which the data packet was generated to reflect the best performance in terms of reliable real-time packet delivery. Finally, the data packet is returned to the Routing Engine module and sent to the radio module through the Transceiver module to be relayed and sent over the wireless medium toward the base station.

Since the mote system is powered by batteries, the battery voltage level can be regularly measured and computed by the TinyOS's VoltageC components of the energy module as a percentage of the full capacity. Battery voltage readings obtained by the TinyOS's VoltageC components are used to calculate the residual battery capacity. This is fed into the implemented Routing Engine modules of the routing scheme in order to be used in the routing cost function in favour of the most energy efficient route.

Sensor motes that generate a data packet must store their own routing function parameters (e.g., next hop link quality probability, residual energy, node *id*, hop count, packet sequence number, and relaying load profile) as a field within the packet before transmitting it. Each sensor node maintains a backup table of a set of neighbouring nodes using the averaging window filter. The packet sequence number is used for network statistics of packet transmissions in order to avoid packet duplication and unnecessary overloaded packet relaying.

4.5 Experimental Evaluation

To develop an understanding of routing performance, this section describes in details the experimental setup settings and testing scenarios used throughout the indoor experiments

performed on Mica2 motes. All evaluations performed in TinyOS-2.1.0. The testbed network represents a realistic setting for examining and evaluating the performance of the proposed routing scheme principally against the official TinyOS routing layer implementation of MintRoute [33,52] as well as with other existing widely-used routing protocols such as Directed diffusion [47], DTRP [130,131], Gossiping [125] and TinyAODV [134] on Mica2 platform [14,35]. All evaluations are performed using the stable version 2.1.0 of the standard TinyOS [25]. The standard TinyOS-2.1.0 CSMA B-MAC layer [20] for CC1000 radio [21] is used as an underlying low-power listening (LPL) MAC protocol on Mica2 motes over indoor low-interference channels of 868/916MHz.

In the experimental evaluation, all sensor nodes generate traffic of fixed size packets (unless specified) at different transmitting rates under particular testing scenarios. The experimental approach considers a many-to-one real-time event-driven sensor network where sensing nodes deliver their sensing measurements to a single base station under a time constraint and with the overall target of reliable communications and minimised energy consumption of the relaying sensor nodes. All sensor nodes are homogeneous and commence transmitting with the same residual power capacity.

4.5.1 Performance Metrics and Observed Entities

This section describes in details the performance parameters used to evaluate the operation of the sensor network by means of the proposed routing scheme. The performance metrics addressed in this section are considered throughout the thesis for all real testbed indoor and outdoor experiments of chapters four and five respectively as well as for all large-scale-computer simulations of chapters six and seven later. Since Link quality Indicator (LQI) metric is exclusively provided by IEEE802.15.4–compliant

radio chips, e.g., Chipcon's CC2420 radio transceivers, LQI is addressed later in the next chapter where outdoor experiments are conducted using TelosB's CC2420 radio.

In the real WSN testbed, these performance metrics are observed by the base station, relayed to the attached laptop, and recorded in log files for intensive performance analysis using Matlab scripts. The log files record the observed metrics such as Received Signal Strength Indicator (RSSI), Link Quality Indicator (LQI) provided by IEEE802.15.4-compliant radios, and radio packet record that contains packet sequence number, timestamps, node level, node *id*, and Cyclic Redundancy Check (CRC). These metrics will be used to evaluate the routing efficiency in the deployed scenarios and also how the sensor network behaviour is characterized in terms of: *packet delivery performance* to assess the significance of wireless link reliability on packet loss probability; *average end-to-end delay* to evaluate the multihop load-aware data aggregation and hop count effect on data delivery time; and *average dissipated energy* to estimate the energy depletion rate of a sensor node to achieve reliable multihop data collection.

- *Received Signal Strength Indicator* (RSSI): Received Signal Strength Indicator (RSSI) is a generic signal quality metric of the radio receiver. It is sampled by the in-system analog-to-digital converter (ADC) of the RF transceiver chip, e.g., Chipcon's CC1000 radio [20]. RSSI measurement is a radio vendor dependent that provides some level of classification of the received signal. RSSI represents the amount of signal energy received by the sensor node. It can be measured by most radio transceivers. RSSI readings have a range from -100 dBm to 0 dBm and the maximum error (accuracy) is 6 dBm. It is calculated over 8 symbol periods [20].

In indoor experiments, RSSI measurements provided by Mica2's CC1000 radio are selected to estimate link reliability. RSSI represents the amount of energy received by the sensor node's radio. It is measured on Mica2's ADC channel-0 *($Count_{ADC\_0}$)* and is available to the software. TinyOS-2.x provides this measurement automatically, while in earlier version it must be enabled by the user. The conversion from ADC's channel-0 counts to RSSI in *dBm* can be calculated by Equations 4.1 [35], where V_{batt} is Mica2 battery voltage level and *FS* is Mica2 ADC full scale.

$$RSSI\,(dBm) = \frac{-50.0 \times V_{Batt} \times Count_{ADC\_0}}{FS} - 45.5 \qquad (4.1)$$

Although the RSSI samples provide good predictive values and effectively estimate the intermediate link better than Chip Correlation Indicator (CCI) readings [65], the power level of the RSSI metric can not reflect the quality of the link as accurately as the Link quality Indicator (LQI) provided by the newer version of Chipcon's CC2420 transceiver [78] for IEEE 802.15.4 and ZigBee [87].

- *Packet Delivery Performance*: The average successful *packet reception ratio* (*PRR*) is one of the basic metrics used for evaluating packet delivery performance and to measure link quality. *PRR* is also known as *packet delivery fraction* which is the percentage of the number of successfully received data packets that correctly passes the Cyclic Redundancy Check (*CRC*) at the base station divided by the total number of data packets originally sent (considered) by the source sensor nodes as expressed in Equation 4.2. To measure the *PRR*, the sequence number field of each packet is used to count how many packets have been received by the base station, and CRC field is computed over the entire packet to determine successful packet reception by the base station. The *PRR* metric also indicates the successful transmissions ratio and its complement is *packet loss ratio* (*PLR*) as in Equation 4.3.

$$PRR = \frac{Successfully\ received\ packets}{Total\ number\ of\ sent\ packets} \times 100 \tag{4.2}$$

$$PLR = \frac{Sent\ packets - Successfully\ received\ packets}{Total\ number\ of\ sent\ packets} \times 100 \tag{4.3}$$

Moreover, *Packet error ratio* (*PER*) (also known as *block error ratio*) corresponds to the percentage of the number of erroneously or faulty received data packets that failed the CRC check at the base station divided by the total number of all received data packets (successful and erroneous) as in Equation 4.4. A data packet is considered to be erroneous if either failed to pass CRC check due at least one erroneous bit or its length/type is incorrect. Prior to the estimation the *PER*, the number of data packets to be used as a basis for the *PER* calculation has to be defined. Since faulty data packets occur randomly, measured *PER* values are difficult to reproduce on rule basis according to a recognised distribution curve and the number of data packets tested directly could influence *PER* measurement accuracy. Hence, packet error probability in the network needs to be calculated as well as the distribution function of the discrete random variable which describes the number of errors in a packet is also needed to be determined. Accordingly, S sent data packets are examined for errors by assuming that errors occur with *probability* P. The E erroneous packets are identified with P_E error probability $P_E = \binom{S}{E}(P)^E(1-P)^{S-E}$.

$$PER = \frac{Erroneously\ received\ packets}{Total\ number\ of\ received\ packets} \times 100 \tag{4.4}$$

In low-power WSNs, the *PER* may be affected by either channel's signal-to-noise ratio, signal strength attenuation, interference, distortion, bit synchronization problems, multipath fading, or a combination of them. Packet delivery measurements are considered using an average filter to better reflect packet delivery performance on dynamic, asymmetric and time-varying wireless links.

Other metric related to packet delivery performance, the a*verage End-to-End Delay*, which measures the average one-way latency observed between transmitting a packet from the source sensor node and receiving it at the base station including propagation time. To measure the average *end-to-end delay*, send and receive timestamps at the MAC layer are used as described in [28]. All wireless sensor motes are positioned in predetermined locations and when the data transmission begins, the source sensor node sends packets with its send timestamp and the base station timestamps the received packets with its received time. Clock skew is the most significant source of error as the internal clocks of the sensor mote can drift at a linear and predictable rate. Therefore, it is necessary to compensate for this skew. In order to correlate individual measurements, the clocks of the sensor nodes need to be synchronized. Skew compensation with least square linear regression [29] is used offline to correct *end-to-end* delay and to consider clock drifts.

In the view of data delivery rate, an additional network performance metric related to the *end-to-end* delay is the average network *throughput* (known also as *packet reception rate or delivery rate*) which is different from the *PRR*. It is the average rate of successful data packet delivery in *bits/sec* (or *data packets/sec*) passes through multihop sensor nodes over a communication channel.

- *Average Dissipated Energy* (E_{avg}) measures the ratio of total dissipated energy per sensor node in the network to the number of distinct events received by the base station. It computes the average work done by a participating sensor node in delivering data of interest to the base station. This metric also indicates the overall lifetime of sensor nodes.

During each experimental run, the rate at which the data packets transferred is tracked, and the routing scheme's energy expenditure per sensor mote is calculated based on the amount of energy required getting the data packets to the base station. Equation 4.5 is used to calculate the *Average Dissipated Energy* (E_{avg}) of each run for a given time.

$$E_{avg} = V_{batt} \times I_{drawn} \times Cycle\ Time \times Cycles\ \ Count \tag{4.5}$$

Where, the battery voltage level (V_{batt}) of the sensor nodes is measured based on monitoring the reference voltage (V_{ref}) of the Analog-to-digital (ADC) of the mote system. As the ADC reference is the mote batteries, consequently as the V_{batt} drops the V_{ref} tend to decrease in value. The ADC output count $(Count_{ADC})$ is converted to volts, and V_{batt} is computed using Equation 4.6, where *FS* is the ADC full scale.

$$V_{batt}(V) = \frac{V_{ref} \times FS}{Count_{ADC}} \tag{4.6}$$

The drawn current (I_{drawn}) is the current consumed by the mote system, which is taken from the measured current consumption model for motes system stated in [42]. The time spent per *CPU cycle* depends on the type of the mote system, i.e., (1/7.3828)μs and (1/8)μs for Mica2 [14] and TelosB [86] respectively. The *average dissipated energy profile* of mote system is estimated based on the number of *CPU cycles* spent during mote's tasks. Prior to all experiments as stated earlier in the preliminary analysis of the previous chapter in Section 3.7, the *CPU cycle profile* due to the routing computation overhead is preliminary estimated using *Avrora* emulator [111]. Finally, the residual battery capacity is calculated by converting the measured battery voltage level (V_{batt}) to energy. This is fed into the implemented TinyOS modules of the routing scheme in order to be used in the routing cost function in favour of the most energy efficient route.

In addition, the following observed entities are also recorded and used for generating statistics and estimates:

- *Packet Sequence Number* (*SeqNum*): An application-level sequence number field embedded in the sensor data packet header and initialised at the source sensor node, which will be increased by one every time the packet traverses an intermediate sensor node. The *Packet Sequence Number* field is used by the base station to detect how many packets have been passed through the base station and to detect a packet loss. Each sensor node uses the *Packet Sequence Number* of each packet transmission to eliminate packet duplications.

- *Cyclic Redundancy Check (CRC) field*: TinyOS has a CRC field in its radio packet to indicate whether the packet received passes the CRC checking. Chipcon radio chip (CC1000 or CC2420) has an automatic CRC checking capability and the CRC scheme used in is CRC-16 [87].

4.5.2 Experimental Setup and Testing Scenarios

In order to evaluate the suitability of the proposed routing scheme for indoor WSN, a set of indoor experiments are run and repeated 7 times on the testbed network for an arbitrary topology as shown in Figure 4.4. This random topology introduces overloaded relays and routing bottlenecks to the test the suitability of the routing scheme for such intended critical network connectivity situations. This medium-scale indoor testbed consists of a single base station and 20 Mica2 motes deployed inside a roofed showground building. Mica2 motes are labelled with numbers and placed in predetermined locations on the floor to limit the radios transmission range and to stimulate longer routes of 7 to 9 hops distance towards the routing tree root (i.e., the base station) at the perimeter of the deployed testbed where the traffic is recorded.

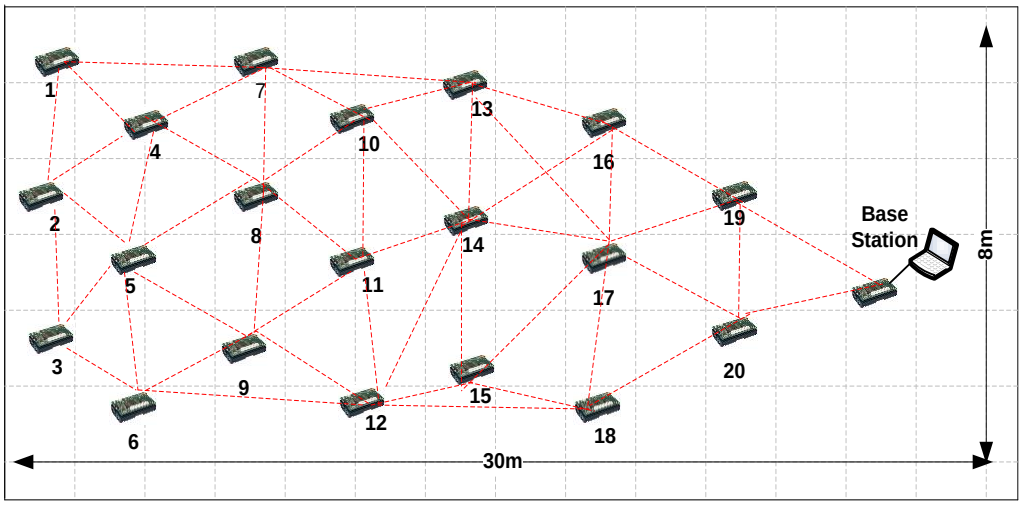


Figure 4.4 Indoor Deployment Topology

The RLBR scheme sets up a spanning tree towards the base station and is configured to generate and broadcast data packets at a given transmitting rate per source sensor node. While the network is operating, the number of packets received by the base station is recorded for each run and the average of these runs is taken. The base station acts as a bridging device between the sensor network and the laptop; relaying the data packets from the sensor nodes to the laptop and the route-setup packets from the laptop to the sensor nodes. Also the base station acts as a logging device for all required metrics and measurements such as RSSI, CRC, time-stamp, packet sequence number. This information is then appended into the received packet. Then, the packet logger/parser program in the laptop processes these received packets, and save them to a log file for analysis using Matlab scripts.

Since the Mica2 CC1000 radio library for the default TinyOS-1.x release [24,32] that comes with Mica2 motes does not support the time-stamping interface as described in [24] and [28], TinyOS-2.1.0 [25] is installed and used instead for all experiments. If the local clocks on sensor nodes were precise and, hence, the offset of the local times were constant, a single synchronisation point would be sufficient to synchronise two sensor nodes. However, the frequency differences of the crystals used in Mica2 motes introduce

drifts up to 40μs per second [24]. This would mandate continuous re-synchronisation with a period of less than one second to keep the error in the microsecond range, which is a significant overhead in terms of bandwidth and energy consumption. Therefore, estimation of the drift of the receiver clock with respect to the sender clock is considered.

As mentioned above, the radios transmission range of the motes is limited by placing them directly on the floor. Then, the packet delivery rates are recorded. In the indoor environment, where space is more limited, the transmitting power of the sensor nodes is set to be at the lowest output power level of -20dBm (10μW) rather than the default power setting of 0dBm. To allow a multihop communication radius of 3 to 4 meters, variable in-between spaces are allowed to keep adjacent sensor nodes within the transmission range of each other and to provide a reliable delivery performance over shorter multihop distances and to minimise the possibility of long one-hop direct opportunistic reception. This will allow examination of the routing reliability, as network reliability drops considerably over more number of hops. However, as encountered in preliminary testings and also shown in [56,120], it is still likely that some reliable long distance links will form.

Due to the jitter in the real world network, transmission start times vary with a mean of few milliseconds. Furthermore, obtaining reliable signal strength measurements for link state indicator can take up to 7ms as this is not fixed in the CC1000 radios [20]. Therefore, the times at which the signal strength is measured need to be chosen carefully at the receiver. RSSI measurements are taken in the middle of packet transmission periods so substantial jitter can be tolerated. Mica2 motes use CSMA-based MAC protocol, i.e., TinyOS B-MAC [21] that performs Clear Channel Assessment (CCA) and then start transmitting. The automatic ACK feature as well as the retransmissions of the

automatic repeat request (ARQ) is disabled, while the link layer functionality is provided using Implicit Acknowledgement as stated in [31] to avoid additional MAC layer overheads. In Mica2 CC1000 radio implementation [20,32]; the data path does not implement software Manchester encoding but it is provided by the CC1000 hardware. The CC1000 hardware takes care of bit-synchronization and clocking as the data path interfaces to the radio at the byte level. The bytes are not necessarily aligned with the data packet when they are coming off the radio. The data path determines and executes the necessary bit shifts on the incoming stream. The CRC computations are running on the data packet received by the base station in order to distinguish between *successfully* and *erroneous* received packets and to gauge packet delivery performance in terms of *PRR* and *PER* respectively.

During each experimental run, the routing scheme's energy expenditure per sensor mote is calculated. The *average dissipated energy* (E_{avg}) required to get the data packets to the base station is estimated by converting the measured battery voltage to energy by Equation 4.5 ($E_{avg} = V_{batt} \times I_{drawn} \times Cycle\ Time \times Cycles\ Count$) where, V_{batt} is the battery voltage level, I_{drawn} is the current drawn by the Mica2 mote system, *Cycle Time* is the time spent per Mica2's *CPU cycle* which is (1/7.3828)μs, and *Cycles Count* is the number of *CPU cycles* spent during Mica2 mote's tasks. I_{drawn} *and Cycles Count* are calculated based on the preliminary analysis stated earlier Section 3.7.

To measure V_{batt}, the voltage reference of the Analog-to-Digital (ADC) of Mica2 mote system is regularly monitored. As the reference of Analog-to-Digital (ADC) is the battery voltage (V_{batt}), the voltage reference of the ADC (V_{ref}) drops according to V_{batt}. The TinyOS's VoltageC components are used by the implemented TinyOS modules of the routing scheme for converting the ADC's output count ($Count_{ADC}$) to volts. In other words, since the ADC uses the battery voltage as a full scale (*FS*) reference, the ADC

full scale voltage value changes as the battery voltage changes. Moreover, the battery life decreases as the mote operates. Thus, V_{batt} can be regularly computed and measured from the ADC's output by the TinyOS's VoltageC components as a percentage of the initial value of the full capacity. The V_{batt} readings can be calibrated by attaching a Precision Micro-power Shunt Voltage Reference (LM4041) with reverse breakdown voltage of 1.223V to ensure that the ADC has an accuracy of better than ±0.1%. As the reading logic of the ADC output is uneven, it takes a short time to settle and then the fixed reading from the ADC's output count ($Count_{ADC}$) is measured. Then $Count_{ADC}$ is converted to volts by Equation 4.6 where, $V_{batt}(V) = (V_{ref} \times FS) / Count_{ADC}$. V_{batt} readings obtained by the TinyOS's VoltageC components are not only used to estimate the amount of energy E_{avg} as in equation 4.5, but also used to calculate the residual battery capacity. This is fed into the implemented TinyOS routing engine module of the RLBR scheme in order to be used in the routing cost function in favour of the most energy efficient route.

Finally, for the initial set of all experiments, all sensor nodes begin with equal battery power levels using fresh AA batteries with full capacity of roughly 3Volts. At the end of each set of runs, the V_{batt} of Mica2's batteries is tested per mote to ensure batteries sufficient energy to power up the motes. Each Mica2 mote with its batteries is connected to a USB port of a laptop using a MIB520 USB gateway. Then, TinyOS's VoltageC components are used to regularly generate the current readings of V_{batt} as a percentage of the initial full capacity. As the mote sends messages the battery life decreases. Those batteries with readings less than half of their full capacities are replaced.

4.5.3 Results and Empirical Observations

Although the network testbed is positioned in indoor environment with very limited ambient noise, it has a number of link reliability challenges represented in: unreliable wireless transmissions that limit the number of traversed packets that can be in flight concurrently from source to destination; MAC protocol contention problems due hidden sensor nodes; properties of the physical layer that may constrain the throughput achievable over a multihop route. Observations and results obtained from the experimental testing are presented in this section.

❖ *Indoor RSSI Measurements*

This subsection discusses the general effect of indoor RSSI measurements on network reliability and link asymmetry. The RSSI is measured indoor within short distances and different orientations of Mote's antenna, then the averaged results are recorded. The RSSI readings are measured at the receiver sensor node. Figure 4.5 shows the overall trend of RSSI measurements as a function of transmission distance and mote/antenna orientation at the lowest transmission power. In the indoor environment, different random deployment topologies have an unobservable effect on the overall link reliability of the sensor nodes. This confirms the suggestions stated in [56,120] that using RSSI independently may be an inadequate link quality metric for predicting the link reliability and connectivity of low-power transmissions.

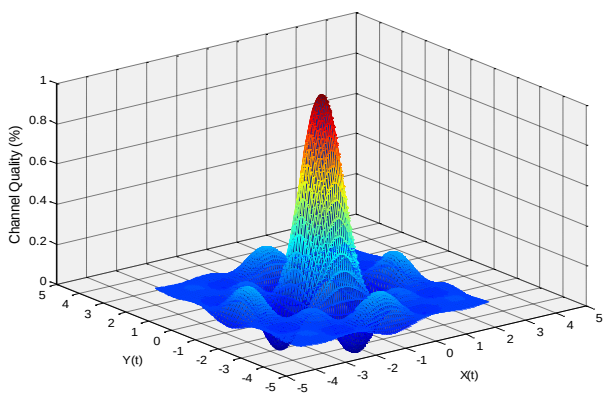


Figure 4.5 The Effect of Short Distances and Motes Orientations on Channel Quality

For example, it has been observed that the low RF output power does not indicate poor RSSI. Besides that, using RSSI independently in parent selection process of the routing scheme results in a routing tree with an undesirable large number of hops and extra overheads as more messages being forwarded over good links. Therefore, the link quality need to be computed based on bit or packet error estimations in addition to the strength of the received signal. Hence, link quality values should be combined with the packet error rate as in the newer versions of Chipcon radios such as CC2420 radio chip [78] on TelosB motes [42,86] that produces reliable correlation value for the overall link quality, LQI [87]. This link metric is considered in the outdoor experiment of the next chapter using TelosB motes.

Increasing the distances between the sensor nodes, makes the RSSI values seem to oscillate in descending way. The RSSI values measured by Mica2's CC1000 radio have a trend to fluctuate as shown in Figure 4.6 where the values presented are average values from the packets that are received with fixed packet size. Within short distances of 1 to 3 meters, the RSSI is generally stronger. For longer distance longer than 4m, at 10m, the link quality has stronger RSSI readings.

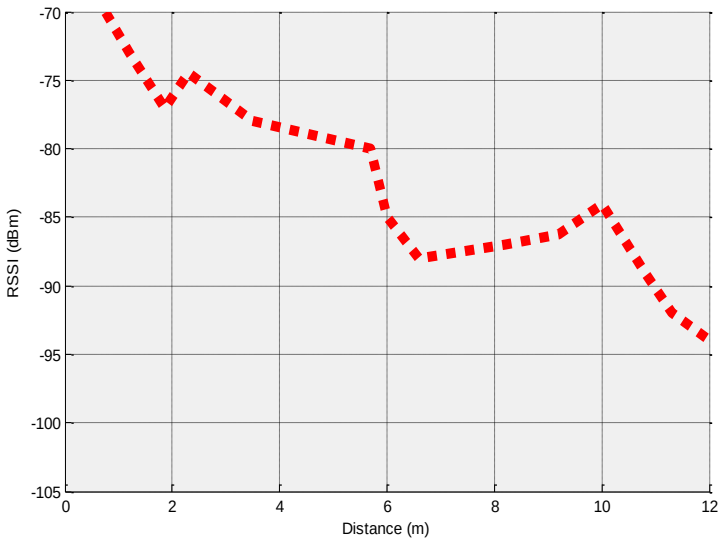


Figure 4.6 Indoor Measurements of CC1000's RSSI vs. Distance

However, the RSSI readings follow an exponential decay while the successful packet reception ratio is still high. After approximately 12m of distance with low RF power and with the motes placed on the ground, the signal is noisier and its strength deteriorates to the minimum receive sensitivity of the CC100 transceiver which ranges between -98dBm and -110dBm based on the data rate, format and frequency [14,20]. This receive sensitivity can be interfered by another oscillator from adjacent sensor motes. A distance of at least 65cm should be maintained between the adjacent Mica2 nodes to avoid local oscillator interference (e.g., RF coupling or crosstalk). However, at low transmission power levels, motes are still able to communicate with each other.

In addition, the signal strength values also decrease as the distance between sensor nodes increase with various packet sizes/lengths. Due to signals interference and noise, Figure 4.7 shows that within smaller spacing, the RSSI of short length packets of 25bytes are generally stronger than with the longer length packets and a lower packet loss. However, for distances longer than 12 meters, packets of maximum length of 128bytes tend to give stronger RSSI readings as the receiver sensitivity depends on the FSK separation and the assigned RF frequency.

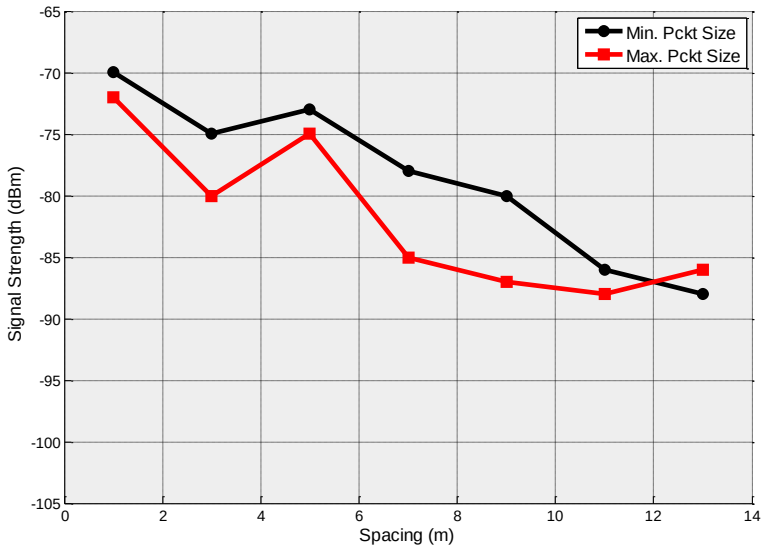


Figure 4.7 Indoor Measurements of CC1000's RSSI vs. Packet Length

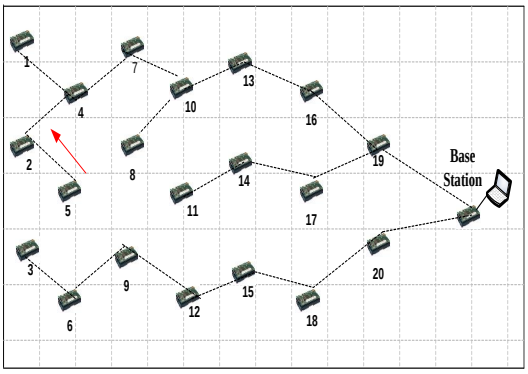
❖ *Routing Hole Problem*

During the parent selection process, MintRoute uses link quality estimations with the surrounding neighbours together with cumulated route quality estimations to the base station. Unlike RLBR, MintRoute does not include the hop count metric in the route updates. This can lead to undesirable results in MintRoute, when a sensor node has optimal routes with two or more neighbours with the same best link quality. MintRoute will then arbitrarily choose one of them as its new parent node using its default Minimum Transmissions (MT) metric, which results in optimal route that could be in some direction far away from the base station and in the worst case in the opposite direction of where the base station is located. This results in an undesirable routing hole problem. However, next packet transmissions may probably reduce the already perceived link quality in MintRoute, which makes the current selected parent less attractive and minimises the possibility of routing hole problem. The natural occurrence of suboptimal routes is taken into account by the RLBR scheme when performing parent selection by adopting the tree-level number in terms of the least number of hops and used as a tie breaker; this advantage does not apply for MintRoute as the hop count is completely ignored in its route update process.

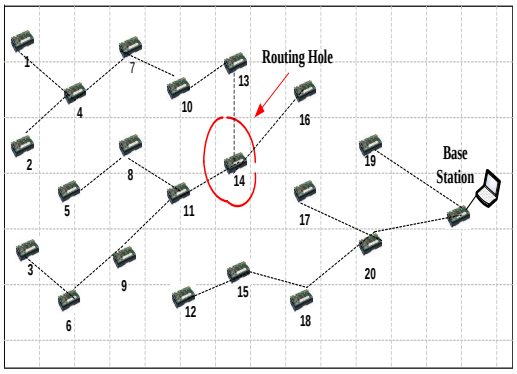
In addition, MintRoute improperly assumes that links are stable with independent packet losses and uses this assumption to derive link quality estimations inaccurately based on long-term link estimations. Furthermore, the parent selection process in MintRoute is merely based on link quality. When the link quality degrades, neighbouring sensor nodes will choose other sensor nodes with a better link quality. For example, creating routing holes in MintRoute is straightforward due to purely relying on the best link quality. When a sensor node has the base station as one of its neighbours, the sensor node will not automatically choose it as its parent. Instead, it will choose the neighbour with the best link quality. A sensor node to be selected as a next hop or parent, it must

have both a high send and receive quality with its neighbours. To obtain a higher send link quality value, the high value must be included in a route update message sent by the relay node that caused a routing hole. To obtain a higher receive value, the relay node will have to keep sending packets to prevent the reduction of the receive value by the node. The number of packets that might be lost also lowers the receive quality.

Figure 4.8 (a) shows an example of how routing in MintRoute picks sensor node 2 as a parent for node 5 instead of node 8 and constructs the optimal route from sensor node 5 to through sensor node 2 even though node 2 is in the opposite direction of where the base station is located. Again in Figure 4.8 (b), after a given time the status of the routing tree changes and sensor node 14 is selected at this time as a parent for node 16 instead of selecting node 19 and constructs the optimal route through sensor node 14 even though node 14 is in the opposite direction of where the base station is located. In addition, sensor nodes 11, 13 also select node 14 as their parent with best ink quality.



(a). Status of fully-connected routing tree



(b). Status of routing hole problem

Figure 4.8 Pure Reliability-Oriented Routing in MintRoute Protocol

MintRoute using optimal routes that purely based on link quality estimations using its MT metric. This leads MintRoute to cause a routing hole to the downstream child nodes at sensor node 14. As a result, MintRoute is deemed to be unreliable in packet transmissions as it greatly relies on link estimates for parent selection. Sine lower traffic load scenarios do not provide enough packets for passive link quality estimations used

by MintRoute, the resulted link estimations will be inaccurate than in higher traffic scenarios. This also makes MintRoute unaware of the link asymmetry problem.

❖ *End-to-End Packet Delivery Performance*

In multihop indoor WSN, the achieved packet delivery performance may be inferior than it should be for several reasons at different layers. At the MAC layer, specifically the B-MAC used on Mica2 motes, CSMA-based MAC protocol backoff waiting times at each wireless sensor node could cause a packet to be lost before it has been transmitted if a sensor node senses a busy wireless channel for a maximum number of times. In this situation, the sensor node will simply discard the packet and move on to the next packet. Besides that, packet loss due to link failures or collisions leads to a high rate of link layer retransmissions; thereby resulting in a low packet reception ratio (*PRR*) and inversely a high packet loss ratio (*PLR*). As a consequence of packet retransmissions, a considerable amount of the energy is spent for repairing lost transmissions.

At the physical layer, indoor environment surroundings and the orientation of Mica2 motes and their antennas have a negative effect on packet delivery performance. In addition, high signal strength is a necessary but not a sufficient condition for good *PRR*, especially when a higher transmission power is used, conceivably due to the effect of Multipath Rayleigh Fading Channel (MRFC) [38]. Furthermore, there is a number of factors that cause a packet to be corrupted and thereby a packet is to be considered lost or not received at all at the destined recipient. A packet may be lost due to errors in the wireless transmission, signal degradation caused by multipath fading, packet drop due to channel congestion, faulty mote hardware, and packet collision due to the hidden node problem [37]. In addition to this, packet loss probability is also affected by signal-to-noise ratio (SNR) and distance between the transmitter and receiver when a lower transmission power is used. As a result, determining the source of the packet loss is

complicated and unclear in terms of the hardware. In addition, this indoor experimental testbed indicates that low-power radio connectivity is likely to be non-uniform, even in ideal settings.

At the link layer, a packet loss due to link failure is the most common in WSN channels. When data aggregation is enabled by the routing scheme, a single link failure might result in a sub-trees of aggregated values being lost. If the failure is close to the base station, the influence on the resulting aggregate can be negatively significant. Figure 4.9 shows the impact of disconnectivity and link failures in terms of packet retransmissions on packet reception ratio at the base station for the RLBR and MintRoute with disabled link layer acknowledgements. Although link failure rate is very low, a small percentage of sent packets are lost due to packet collisions.

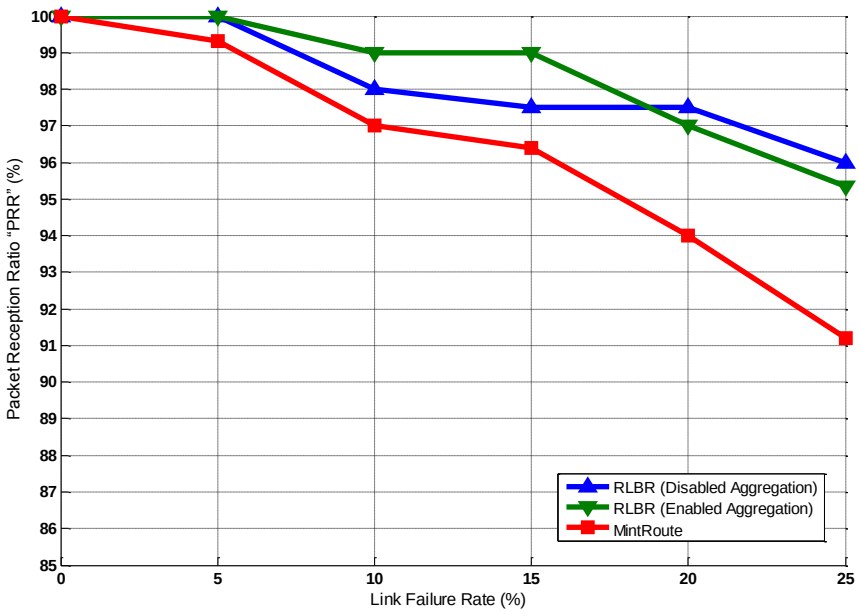


Figure 4.9 The Effect of Link Failure Rate on Packet Reception

Overall, RLBR outperforms MintRoute owing to its lighter traffic as a result of data aggregation, which leads to fewer packet collisions. But when the link failure rate starts to increase above about 20%, the *PRR* in RLBR with enabled aggregation is lower than in RLBR with disabled aggregation as each encapsulating data packet contains more

aggregated packets being lost. On average, when data aggregation is disabled, most sent packets are successfully delivered by greater than 95% and the packet loss is lower even tough the link failure rate increases. Since CC1000 transceiver has a bit-level interface, the average end-to-end delivery is evaluated in terms of *throughput or bit transfer rate* (Kbits/sec) between the transmitter and the receiver. The transmission rate at the source sensor node is programmed prior to the experiment and the average of multiple runs with different sending rates was considered. Figure 4.10 demonstrates how the RLBR outperforms MintRoute as the bit transfer rate changes through few hops from the source sensor node (T_x) to the base station (R_x).

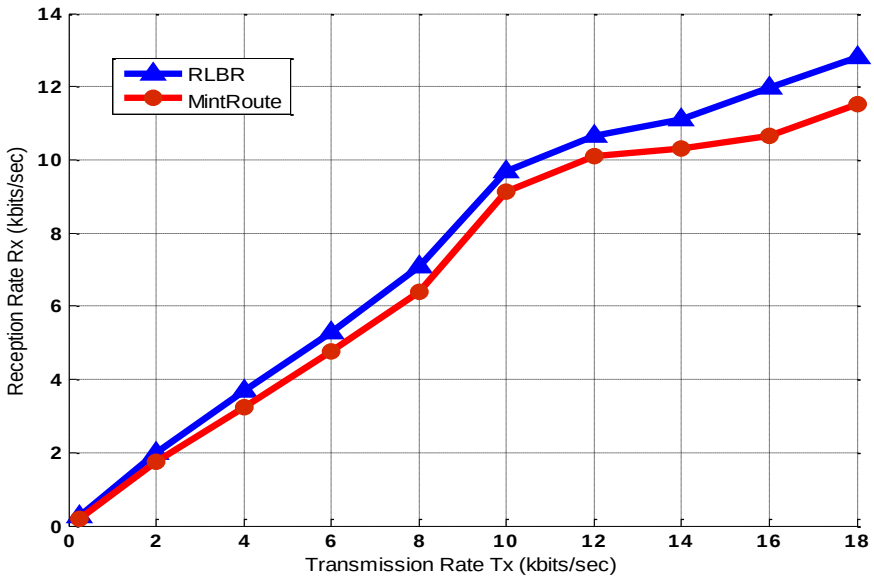


Figure 4.10 Average Throughput (Kbits/sec)

In terms of hop count, the entire network reliability drops noticeably as the network size grows. Figure 4.11 shows the multihop routing overhead in terms of how the reception rate for both protocols decreases as the number of hops increases for a constant transmission rate of 7Kbits/sec. MintRoute performs poorly in the small-scale deployed testbed topology due to the limitations of its route searching and maintenance compared to RLBR. This leads to a prediction that MintRoute also cause a lower end-to-end transfer rate in large-scale WSN with large number of hops between the source sensor node and the base station. Besides that, for example, CC100 radio's hardware-based

Manchester coding has much more overhead and also has a negative effect on packet delivery performance in multihop settings. This leads to per relay node transmission and reception overhead as stated in [54] and shown in Figure 4.12.

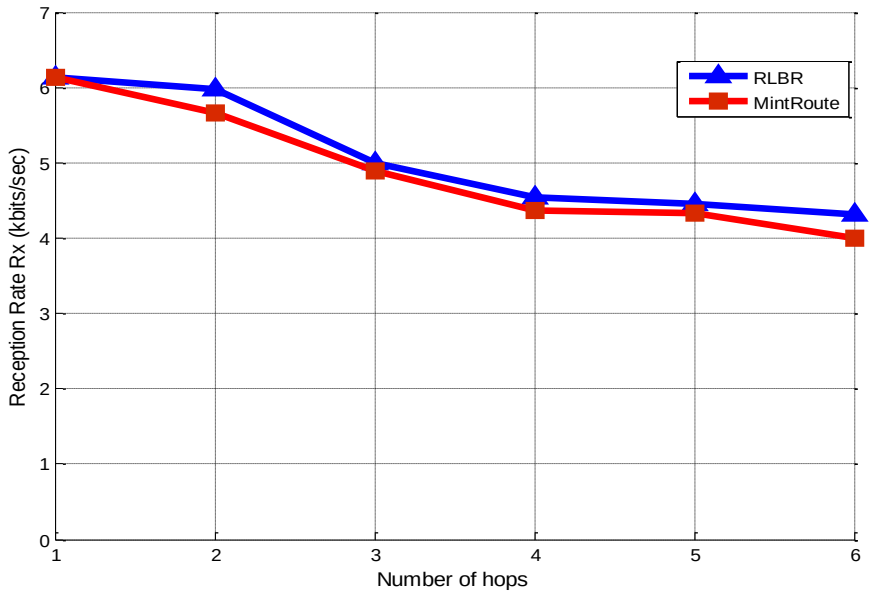


Figure 4.11 Throughput vs. Hop Count

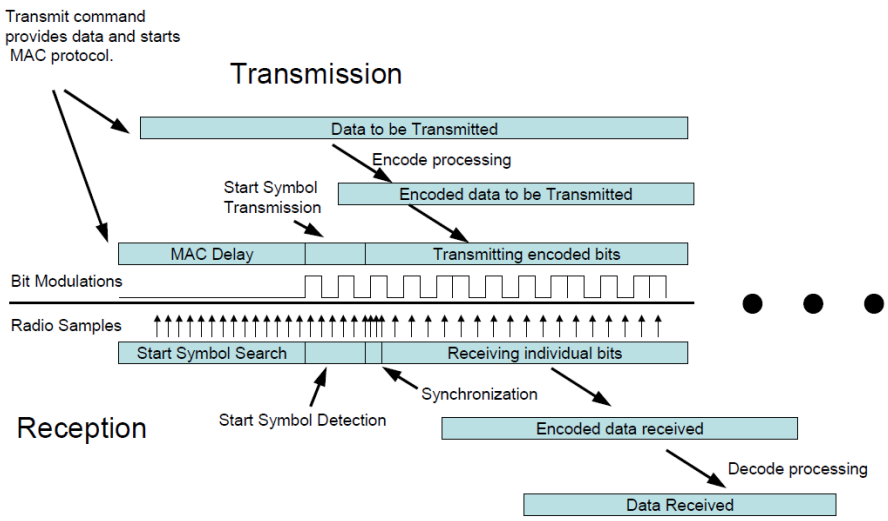


Figure 4.12 Two-Way Per-Hop Communication Overhead [54]

The packet transmission rate of the source sensor nodes is programmed prior to the experiment and the average of multiple runs with different number of source sensor nodes is considered. This empirical study shows that packet delivery performance varies obviously depending on how and where all sensor nodes are deployed, and on the traffic workload in terms of the number of source sensor nodes at which the data packet are

initiated. The traffic load is intentionally increased by arbitrarily selecting and adding more source sensor nodes at the leaves in lieu of increasing the rate at which packets are sent per second per source sensor node. This will allow evaluating reliability of the entire network with different loads of traffic, and the packet delivery performance will be more representative of the WSN as a whole. In each run of the experiments, a number of source sensor nodes is arbitrarily added and programmed to generate data packets at a constant rate of one packet per second. As shown in Figure 4.13, in all traffic loads and topology scenarios of different numbers of source sensor nodes, RLBR achieves higher packet reception ratios as a result of the per-hop load balancing. This follows the fact that RLBR considers overhearing-based packet aggregation to overcome high traffic loads, and adaptive beaconing to cope with topology changes. At low traffic loads with fewer source sensor nodes, passive link quality estimations used by MintRoute are inaccurate due the unawareness of the link asymmetry problem. MintRoute improperly assumes that intermediate links has stable with independent packet losses as fewer packets being sent.

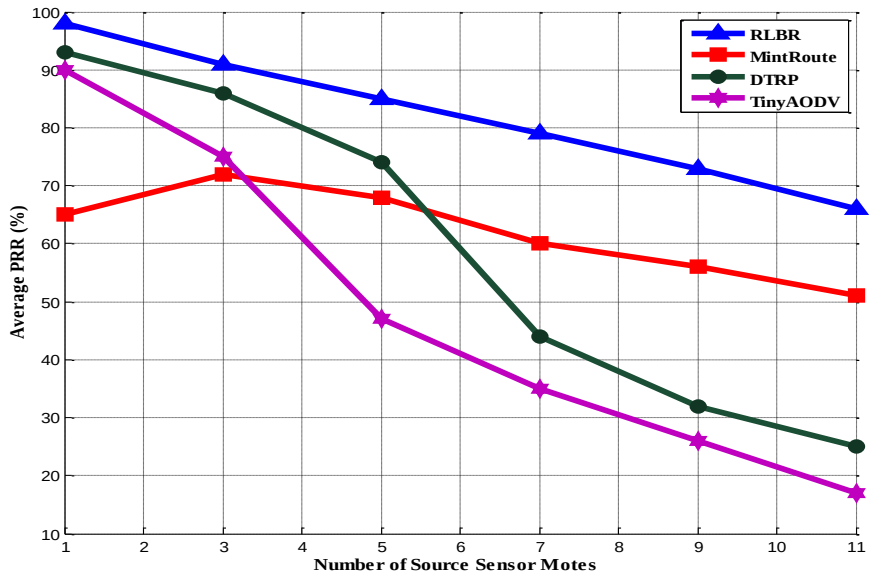


Figure 4.13 Load-Balancing: Average *PRR* vs. Traffic Loads

On the other hand, DTRP and TinyAODV protocols achieve moderately better packet reception ratios than MintRoute at low traffic loads as DTRP and TinyAODV do not require control packets (beacons) to resolve the next hop node towards the base station. However, later at high traffic loads with more added source sensor nodes, MintRoute outperforms DTRP and TinyAODV as it incorporates the snooping capability at the MAC layer [33,127] in its route updates to overcome traffic bottlenecks. At higher traffic loads, all these protocols negatively disseminate the injected data packets far away from the base station as hop count is ignored in their routing scheme. This will lead to unnecessary workload and additional energy dissipation.

❖ *Average Dissipated Energy*

The total network-wide energy expenditure is due to: the *parent selection process*; packet transmission; packet overhearing/receiving; failed packet reception, and updating the routing table. In B-MAC, overhearing a packet consumes the same energy as receiving a packet as B-MAC requires sensor nodes to receive the whole packet before discarding failed ones. Failed packet reception that may result from packet collision or link failure requires packet retransmission to be successfully received at the destined recipient. This requires more energy consumption. Figure 4.14 shows the total dissipated energy required for retransmissions due to packet loss or link failures. Since the RLBR scheme has the feature of employing the data aggregation for less communication overhead, packet transmission is less than that in MintRoute. Reducing the number of packets sent results in less energy consumed for packet receiving, overhearing, and failed packet retransmission. In addition, the total dissipated energy for packet transmission is still much lower in RLBR than in MintRoute even though the RLBR scheme requires only 0.48% of computation overhead for parent selection process as stated earlier Section 3.7. On average, RLBR has higher node energy efficiency,

consuming 35% less energy than MintRoute. MintRoute keeps transmitting route message at constant rates [33] and doesn't adjust to accommodate topological changes. In terms of energy, the non-adaptive beaconing followed by MintRoute consumes additional energy and is not energy efficient even on a fixed indoor testbed that doesn't experience high rate of link failures.

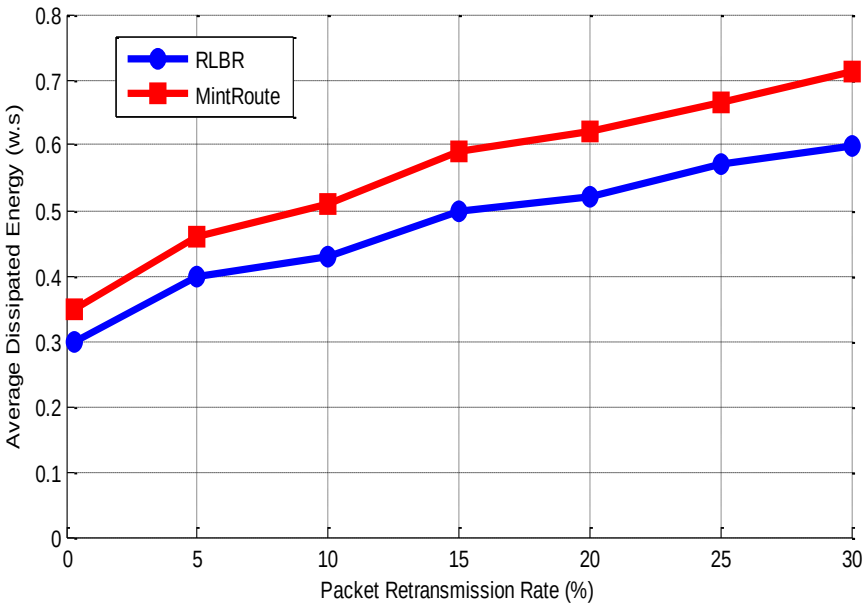


Figure 4.14 Average Dissipated Energy due to Link Failures

To examine how the routing scheme copes with topology changes in the indoor environment, the transmission power of sensor nodes is kept to the lowest level in order to keep the power consumption per sensor node minimised as possible but the nodes spacing is changed arbitrarily to maintain reliable multihop connectivity within a limited indoor space. It can be observed from Figure 4.15 that as in-between spacing between nodes increases the average dissipated power by the sensor nodes for transmission and receiving during their operation instantaneously increases faster in MintRoute than in RLBR. This owing to the fact that MintRoute uses only reliability metric to selects suboptimal routes which can be in a direction faraway from the base station when the link quality.

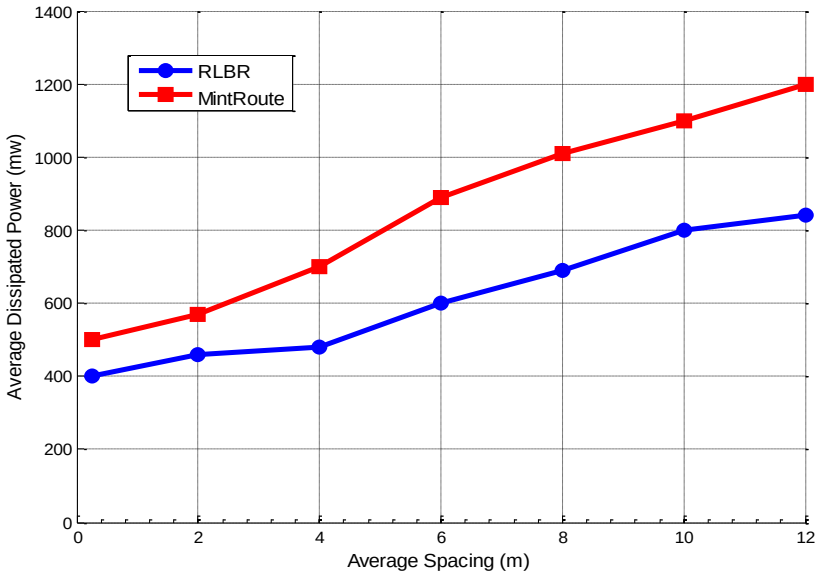


Figure 4.15 Average Dissipated Power due to Topological Changes

In terms of energy dissipation cost, Figure 4.16 demonstrates how the proposed scheme consumes of energy under different traffic loads and topology scenarios of different numbers of source sensor nodes. Since the exchange rate of route messages is fixed in MintRoute, DTRP and TinyAODV, their energy dissipation can be minimised as at low traffic rates. However, these protocols expend more energy at high traffic rates and spend a longer time to cope with the topological changes using their fixed non-adaptive beaconing. During this time, most forwarded packets are routed through optimal paths based on link quality as in MintRoute or based on hop count as in DTRP and TinyAODV; this leads to additional energy consumption and thus offsets the benefit of energy balancing. Hence, the RLBR scheme, using its adaptive beaconing, considers the acceleration of route message exchange rate for reactively propagating the topological changes. Although the other protocols may occasionally balance the traffic load with parent switching based on their default metrics (e.g., MT metric in MintRoute and hop-count in DTRP and TinyAODV), these protocols do not apply a metric that considers energy or workload balancing in their routing schemes.

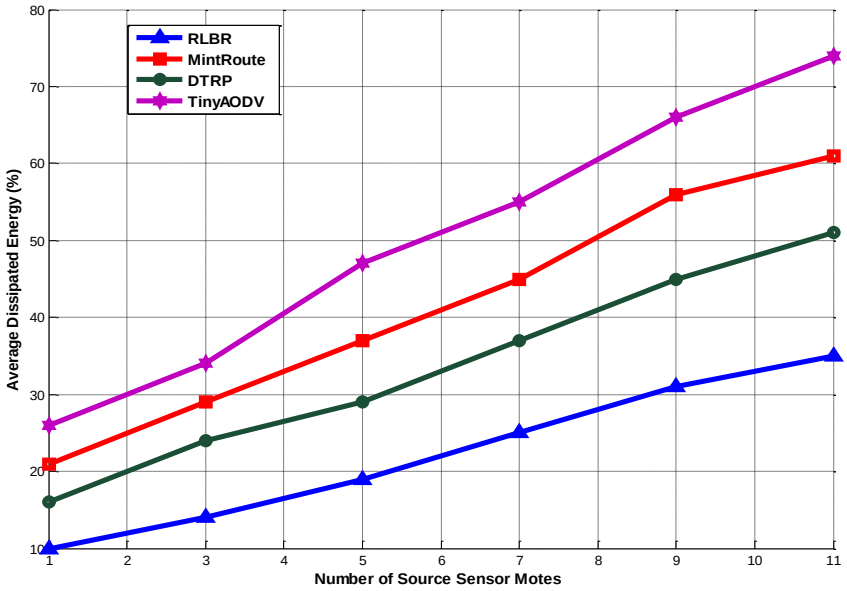


Figure 4.16 The Effect of Traffic Load on Energy Balancing

4.6 Discussion and Conclusion

Since the wireless links in low-power WSNs are unstable and unpredictable, and the loss of packets happens frequently in communications, the link quality metric is mainly used by most reliability-oriented routing protocols to select the optimal link. If the reliability of communication is purely deemed as a routing cost metric, a number of nodes will be exhausted quickly. Consequently, this number of inoperative sensor nodes is extremely essential to the lifetime of the entire network; if these relaying sensor nodes fail to relay packets, the network's functionality will be ruined. In other words, if only the link reliability metrics are considered in WSNs, it may create a long hops route, and the high quality paths will be frequently used. This may lead to shorter lifetime of the high quality routes and longer delivery delays; thereby the entire network's lifetime will be significantly minimized and the network performance will deteriorate.

In indoor environments, multipath fading is more severe. There is a noticeable variance in the corresponding *end-to-end packet delivery performance* because of a fluctuation in the signal strength and link quality below the sensitivity threshold of the RF transceiver

due to interference, fading, or shadowing state. It is also due to the fact that the channel is sampled at different times for forward and reverse link estimations. In the most cases, the *packet delivery performance* for the reverse link is different from its counterpart for the forward link as a consequence of the time-varying nature of the wireless communication channel. Although the indoor experiment is performed with stationary motes, the RSSI values have a tendency to fluctuate, and do not imply a steady link with various packet sizes where RSSI is not good enough indicator of link quality. Even with high RSSI values, there might be severe interference. RSSI could yield a routing tree with additional number of hops and extra messages being sent and overheard at the same time as lower transmission power does not mean poor link quality. In a multihop sensor network, the *packet delivery performance* drops heavily as the number of hops increases for a constant transmission rate. This is due to the packet process overhead per relay (e.g., encoding and/or decoding) and wireless signal propagation delay. While source sensor nodes faraway from the base station are likely to have a lower *end-to-end packet delivery performance* for their generated data packets, sensor nodes that are closer have faster energy dissipation for packet relaying. It has been also observed that the average power dissipated by the sensor nodes during their operation increases as the inter-nodes spacing increases. Losing packets before reaching the base station not only wastes energy and network resources, but also degrades the quality of network functionality.

The experience with the experimental work done so far has revealed several underlying issues that stem from the properties of the reliability-oriented and cost-based routing protocols. The RLBR scheme is tested and investigated in real indoor environment, and the detailed experimental measurements are captured for performance analysis against the most-widely used routing protocols combined with the resource constraints of the mote platform. RLBR achieves over 35% energy savings over the standard network layer currently provided by TinyOS MintRoute and other most widely-used routing

protocols for WSNs. It also reaches a better connectivity rates and communication reliability in terms of *end-to-end packets delivery performance*. The experimental results show that RLBR scheme performs well as it has a lower control overhead, fair average delay, high success ratio of packet delivery and moderate energy consumption.

Finally, using CC1000 RF chip's RSSI independently may not be an adequate indicator of the link quality for reliable connectivity; even with high RSSI there might be severe interference [59,65]. Since the CC1000 radio does not do some of its processing itself and requires the CPU to do so [20], more communication overhead may be introduced and added to the routing scheme. Therefore, for better understanding of low-power wireless link reliability, a newer form of Channel State Information (CSI), namely, link quality indicator (LQI) [87], is used with RSSI for improved link quality estimations in the outdoor experiments of the next chapter. While LQI measurement is not supported by Mica2's CC1000 radio, it is supported by IEEE802.15.4-compliant RF transceivers such as TelosB's Radio CC2420 that provides more reliable RSSI/LQI/bit error patterns of the received packets.

Chapter 5
Outdoor Field Experiments

5.1 Background

Reliable delivery of data can be ensured through the careful selection of error free links, quick recovery from packet losses, and avoidance of overloaded relay sensor nodes. Since link failures and packet losses are unavoidable, sensor networks may tolerate a certain level of reliability without significantly affecting packets delivery performance and data aggregation accuracy in favour of efficient energy consumption. An effective hybrid approach trades off between energy, reliability, cost, and agility while improving packet delivery, maintaining low packet error ratio, minimizing unnecessary control packets transmissions, and adaptively reducing control traffic for high success reception ratios of representative data packets. Based on this approach, the Reliable Load-Balancing Routing (RLBR) scheme, proposed in Chapter 3, can achieve moderate energy consumption and high packet delivery ratio even with a high link failure rates. In this chapter, the performance of RLBR scheme is experimentally and rigorously investigated for performance analysis using an outdoor interference-prone sensor field testbed of 2.4GHz TelosB motes [42,86], low-power wireless sensor platform from Crossbow [9]. Currently, TelosB modules represent the state-of-the-art IEEE802.15.4 [87] compliant wireless platform for low power sensor networks. TelosB platform is based on the TinyOS development environment [25]. This chapter presents an empirical study on how to improve energy efficiency for reliable multihop communication by developing a cross-layer lifetime-oriented routing scheme and integrating useful information from different layers. The proposed approach aims to redistribute the relaying workload and the energy usage among relay sensor nodes to achieve balanced energy dissipation, thereby maximising the functional network lifetime.

The RLBR scheme is benchmarked with the TinyOS routing layer implementation of MultihopLQI [43] as well as with other most recent widely-used reliability-oriented routing protocols for WSNs. As MultihopLQI routing layer is well-established, well-tested, highly used collection tree protocol that is part of the TinyOS-2.x distribution, and has been used in recent WSNs deployments [51,76]. Therefore, the benchmarking with such protocol is considered a reasonable evaluation. The RLBR scheme is shown to be more robust and energy efficient than the MultihopLQI collection layer of TinyOS2.x and other benchmark routing protocols for WSNs. The detailed experimental results show that RLBR scheme maintains higher than 95% connectivity in interference-prone medium while achieving an average of over 35% energy savings.

5.2 Related Work

A number of reactive cost-based reliability-oriented routing protocols have been developed for mote-dominated WSNs using TinyOS [25]. MultihopLQI [43] is a reactive collection protocol that does not employ routing tables or blacklisting but it maintains a state for one parent at a certain time. MultihopLQI uses LQI, to obtain the reliability cost of a given route. MultihopLQI operates connectivity discovery and route maintenance through a sink initiated reactive flooding and performs route selection based on CSI and link-level acknowledgments. Furthermore, CTP [66] is the recent version of MintRoute [33] that uses ETX metric [46] and has a number of special features such as congestion-based packet rescheduling. However, it does not employ load balancing. MultihopLQI and CTP experience the asymmetric link problem as child sensor nodes might not get their packets acknowledged from their current parents although maximum number of successive transmission failure is reached. The RLBR scheme solves the asymmetric link problem by using active bidirectional monitoring of link status and switching to a new valid parent when exceeding a threshold of maximum

successive transmission failures, and puts the old invalid parent into blacklist to avoid switch oscillation.

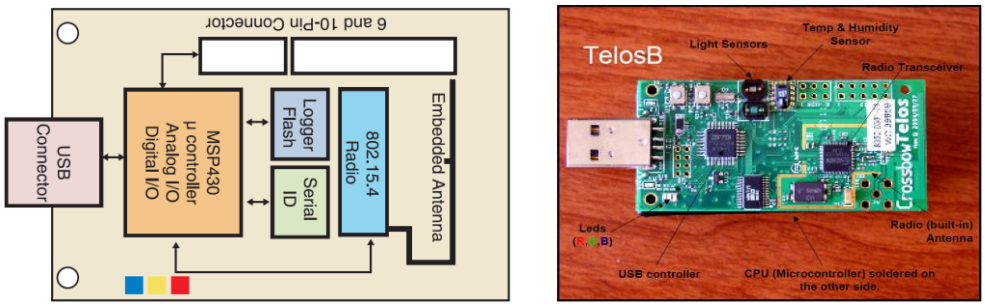
In terms of energy efficiency, MultihopLQI performs better than CTP and MintRoute and it has been also deployed in many recent sensor field experiments [51,76]. Hence, in addition to other most widely-used collection protocols, MultihopLQI is employed as the main benchmark protocol for the outdoor sensor network experiment conducted in this chapter on TelosB motes.

5.3 Implementation Platform: *TelosB Motes*

To extend the performance evaluation of the indoor experiments, this section investigates the implementation challenges in an outdoor environment using Crossbow's TelosB low-power wireless platform [42,86].

5.3.1 Platform Details

Crossbow's TelosB mote (TPR2420CA) [42,86] is an open source radio platform fully compatible with the TinyOS [25] and designed to enable low-power WSNs experimentations. As shown in Figure 5.1, TelosB combines a low-power 8MHz MSP340 microcontroller with 10kbytes RAM; an optional sensor suite including integrated light, temperature and humidity sensor; Universal Serial Bus (USB) programming capability; and CC2420 Radio Frequency (RF) chip [78] compatible with IEEE 802.15.4 [87] standard. The CC2420 provides the data link and offers up to 250kbps data rate. TelosB sensor nodes are equipped with onboard PCB integrated antenna. TelosB operates within the 2.4GHz ISM band and employs the modulation format of the Offset-Quadrature-Phase Shift Keying (O-Q-PSK). When TelosB is attached to the USB port no battery pack is needed. Table 5.1 summarises the specifications of the TelosB mote [86].



a. Internal Components

b. External View

Figure 5.1 Crossbow TelosB 2.4GHz Mote (TPR2420CA) [86]

TABLE 5.1 CROSSBOW TELOSB MOTE (TBR2420CA) SPECIFICATIONS [42,86]

Component	Feature
Processor	8 MHz TI MSP430 MCU with 10kbytes RAM
In-System Program Memory	48 Kbytes
In-System Data memory	16 Kbytes
External Serial Flash Measurements Memory	1 Mbyte
Radio Chip Transceiver	Chipcon CC2420 Radio with Receive Sensitivity of -94dBm
Operating Frequency Band	2.4GHz (2400MHz to 2483.5MHz)
Modulation Format	DSSS with O-QPSK
Antenna	Integrated Onboard, Inverted-F Antenna
Transmission Power Range	-24dBm to 0dBm
Effective Data Rate	250 Kbps

5.3.2 Underlying Layers

At the physical layer, TelosB mote uses a low powered radio “Chipcon CC2420 2.4GHz RF transceiver” which is a single-chip IEEE 802.15.4 RF transceiver with programmable output transmission power and MAC support [78]. IEEE802.15.4 MAC and physical layer protocol standard for short-distance, low-power, and low-data-rate wireless personal area network (LR-WPAN) [87]. Furthermore, CC2420 radio chip supports functions such as packet handling, data buffering, burst transmissions, address recognition, clear channel assessment, link-quality indication and timing information

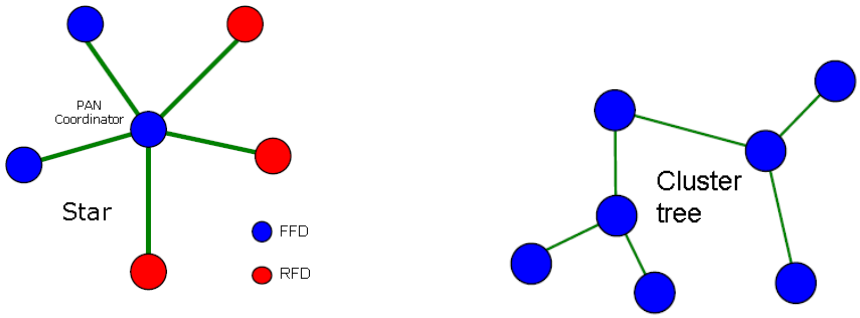
[78]. Unlike the CC1000 radio chip used on the Mica2 motes in the indoor experiment of the previous chapter, CC2420 functions reduce the load on the host processor and enable the CC2420 to interface with low-cost microcontrollers. In addition to the power consumed for the physical layer processes, the power consumption of CC2420 also processes a part of the MAC layer functions including a First-in-First-out (FIFO) digital interface for data manipulation, CRC and encryption [78]. This will introduce less communication overheads to the upper layers.

The physical layer offers bit rates of 20kbps (a single channel in the frequency range 868–868.6MHz), 40kbps (ten channels in the range between 905 and 928MHz) and 250kbps (16 channels in the 2.4GHz ISM band between 2.4 and 2.485GHz with 5-MHz spacing between the centre frequencies). There are a total of 27 channels available: 16 channels in the 2.4GHz ISM band, 10 channels in the 915MHz, and one channel in the 868MHz band. However, the MAC protocol uses only one of these channels at a time; it is not a multichannel protocol. The MAC layer combines both schedule-based as well as contention-based schemes [87].

The physical layer has many features in addition to transmitting and receiving packets across the physical medium, which include: activating and/or deactivating the radio transceiver to be either in awake or sleep mode based on the MAC sub-layer to allow support for energy efficiency; energy detection (ED) to estimate the received signal power within the bandwidth of the current channel over a time of eight symbols epochs; link quality indication (LQI) to measure the strength or the quality of the received signal using receiver ED; signal to noise ratio (SNR) estimation to be used by the routing layer; channel selection, clear channel assessment (CCA) to determines whether the channel is busy or idle by gauging the modulation and spreading characteristics of the IEEE802.15.4 and comparing the energy in the channel with the ED threshold according

to the carrier sense with energy detection mode; and channel frequency selection to tune the transceiver into a specific channel requested by application layer.

At the MAC layer, TinyOS IEEE802.15.4 has 128 byte maximum MAC frame data size and employs DSSS with O-Q-PSK modulation Scheme. Packet fields contains preamble of 32 bits used for synchronization, 8 bits start of frame delimiter (SFD), 8 bits PHY Header, and 0 to 1016 bits physical service data unit (PSDU) field. The IEEE802.15.4 protocol is asymmetric in that different types of sensor nodes with different roles are used. The standard distinguishes on the MAC layer two types of sensor nodes: a Full Function Device (FFD) that can operate as a coordinator or a device; and a Reduced Function Device (RFD) that can operate only as a device. A device must be associated to a coordinator node, which must be a FFD, and communicates only with this, this way forming a *star* network topology. Multiple coordinators can operate in a point-to-point fashion to form a *cluster-tree* or *mesh* network topology as shown in Figure 5.2. IEEE802.15.4 is a fully handshaking protocol for transfer reliability and also offers security and application services for upper layers.



a. Star Topology

b. Point-to-Point Topology

Figure 5.2 IEEE802.15.4 MAC Topologies

Due to the highly dynamic nature of WSNs, the inherent advantages of contention-based IEEE802.15.4 MAC protocol [87] make it the preferred choice as an underlying layer providing MAC services to the proposed scheme at the routing layer. In addition,

neighbours of the sender other than the attended receiver sensor nodes possibly will receive sent packets, even the packets that are not destined to them, i.e., overhearing. In this situation, the routing information used for routing decisions in the proposed scheme can be imbedded in the packet header. When a sensor node receives a packet not addressed to itself, it can retrieve this helpful information from the packet header before dropping the packet. Hence, IEEE802.15.4 MAC protocol is chosen to be aware of the protocols that run above it and offers control to the protocols that sit on top of it, allowing the routing and application layers to change parameters like the number of retransmissions. Thus, IEEE802.15.4 MAC protocol allows each sensor node to overhear packets transmitted by its neighbours; this allows the upper routing layer to employ snooping for the sake of link quality estimation, and in-network processing and data aggregation.

5.4 Protocol Implementation

The outdoor implementation was carried out using the low-power Crossbow's TelosB mote (TPR2420CA) [42,86], wireless sensor network platform running the component-based operating system TinyOS [25] which is written in an event-driven language called network embedded systems C-like language (nesC) [11-13].

An overview of TinyOS components is addressed in Section 4.4.1. The TinyOS modules of the proposed routing scheme in the outdoor experiments are identical to those explained in Section 4.4.2 and shown in Figure 4.3 of the indoor experiments, with the exception of using the CC2420 radio module instead of the CC1000 radio module. TinyOS-2.x version is not fully backward compatible with version TinyOS-1.x. [27]. Therefore, the recent official stable release TinyOS-2.1.0 [25] that supports different wireless platforms including TelosB motes is used for all experiments.

5.5 Evaluation

In addition to the indoor testbed experiments conducted in the previous chapter, testing and debugging routing protocols on an interference-prone outdoor testbeds is also vital for experimental validation of the routing protocol's efficiency. This section describes in details the experimental setup settings and testing scenarios used throughout the outdoor experiments performed on TelosB2 motes. The testbed network represents a realistic setting for experimentally investigating and evaluating the performance of the proposed routing scheme principally against the TinyOS-2.x routing layer implementation of MultihopLQI [43] as well as with recent collection tree protocols such as CTP [66] and Arbutus [113] on TelosB platform [42,86] running the TinyOS. All evaluations were performed using the stable version 2.1.0 of the standard TinyOS [25].

5.5.1 Performance Metrics

In addition to the performance metrics described earlier in Section 4.5.1, Link Quality Indicator (LQI) metric, which is exclusively provided by IEEE802.15.4–compliant radio chips, e.g., Chipcon's CC2420 radio transceivers, is addressed in this section. LQI is used as link quality metric to evaluate the routing efficiency in the deployed scenarios of the outdoor testbed experiments. To estimate the load balancing performance at the routing layer, a new metric is introduced in this thesis known as *routing overhead metric* ($1/\eta$). It also gives an overall estimation of the energy consumed by the relay nodes for delivering a data packet towards the base station. This metric was already introduced in Section 3.5.1 but it will also be addressed here for completeness.

- *Link Quality Indicator* (LQI): LQI is a hardware-based link reliability metric introduced by IEEE802.15.4 standard specification [87], which measures the error in the incoming modulation of successfully received packets which pass the CRC check sums. LQI can be measured by IEEE802.15.4-compliant radio chips such as

Chipcon's CC2420 RF transceiver [78] on TelosB motes used here in the outdoor experiments. LQI is actually Chip Correlation Indicator (CCI) and its values are related to the chip error rate. Every received packet must be stamped with LQI value as stated by IEEE802.15.4 standard. This value indicates the quality of the link at the time of packet reception. Instead of using each beacon's LQI individually as in most link-reliability routing protocols, e.g., MultihopLQI [43], the proposed routing scheme takes the advantage of averaging LQI measured values in each time frame t to calculate the *averaged time-varying link quality*. This can be calculated using equation 5.1 to better reflect LQI measurements as this controls the effect of the previously estimated on the new estimated LQI values for short-term link estimations. Where α is the history control factor ranges from 0 to 1 in time frame t.

$$LQI_t(\alpha) = \alpha \times LQI_{t-1} + (1+\alpha) \times LQI_t \tag{5.1}$$

According to the IEEE802.15.4 standard, the LQI value must be an integer that is calculated over 8 bits following the start frame delimiter (SFD) and uniformly distributed and bounded within the interval [0, 0xFF]; with 0 being the lowest LQI level L_{low} and 0xFF, i.e., 255 being the highest LQI level L_{high}. Link quality value of a received packet is measured using the receiving signal power of the received packet in the form of a combination of received signal strength and signal-to-noise ratio. The IEEE802.15.4 standard sets the nominal transmit power of the CC2420 RF transceiver to 0dBm and the receiver sensitivity to −94dBm. CC2420 RF transceiver provides LQI values ranges from 50 to 110dBm and correspond to minimum and maximum quality packets respectively. The relationship between the integer values of the time-varying link quality (L_i) at level i and the power of the received packets (P_{rx}) in dBm can be calculated by $L_i = 2.712766P_{rx} + 255$. For example, if the power of the received packet is set at the default value at which P_{rx} = 0dBm, L_i results in an integer value of L_{high} = 255; on the other hand, if the power of the received frame is

set at the receive sensitivity of P_{rx} = -94dBm of the CC2420 RF transceiver, L_{low} = 0. The value of L_i for all received signal power levels in the range between the receiver sensitivity of −94dBm and the nominal transmit power of 0dBm is bounded within the interval [0, 255]. To validate LQI values, the resulted decimal fraction of L_i needs to be curved upward or downward in relation to the value of $(L_i\text{-}L_{i\text{-}1})/2$ to keep integer values of LQI. i.e., LQI = L_i if LQI > $(L_i\text{-}L_{i\text{-}1})/2$; otherwise LQI = $L_{i\text{-}1}$.

- *Routing Overhead Metric* ($1/\eta$): Aggregating data packets at each sensor node of the selected route introduces extra processing overhead which increases energy consumption. In addition, to achieve high success reception ratios of data packets, a transmission of control traffic is needed, which again demands extra energy. Parent selection process also consumes energy. Considering all these factors, the *data packet delivery efficiency metric* (η) is introduced as a measure of the effectiveness of the proposed approach in minimising packet transmissions throughout the network. *Data packet delivery efficiency* (η) accounts for the ratio of the total number of data packets received at the base station to the total number of all control and data packets sent throughout the network. This efficiency metric (η) is used as a benefit metric to gauge end-to-end packet delivery performance of the routing scheme in terms of route message transmission weight. Conversely, the reciprocal of (η), namely, *data packet delivery cost* ($1/\eta$) is used as a *routing overhead metric* to give an overall estimation of the average amount of energy consumed by the relay sensor nodes for delivering a data packet towards the base station. The *routing overhead metric* ($1/\eta$) is expressed in equation 5.2.

$$\frac{1}{\eta} = \frac{Number\ of\ sent\ data\ \&\ control\ packets}{Number\ of\ received\ data\ packets} \tag{5.2}$$

5.5.2 Experimental Settings and Testing Scenarios

The proposed scheme is experimentally tested and evaluated on TinyOS-based outdoor testbed comprising of 30 arbitrarily organised Crossbow's TelosB (TPR2420CA) wireless sensor motes [86], and low power listening (LPL) link layer provided by Chipcon's CC2420 radios [78], specifically, the standard TinyOS-2.1.0 CSMA IEEE802.15.4-compliant MAC layer [87] is used on TelosB motes in an outdoor noisy environment with interference-prone channels of 2.4GHz. Figure 5.3 shows the outdoor experimental testbed network. TelosB motes are randomly deployed and placed in different locations on the ground within a small outdoor venue, i.e., a car parking with an area of approximately 100x40m^2.

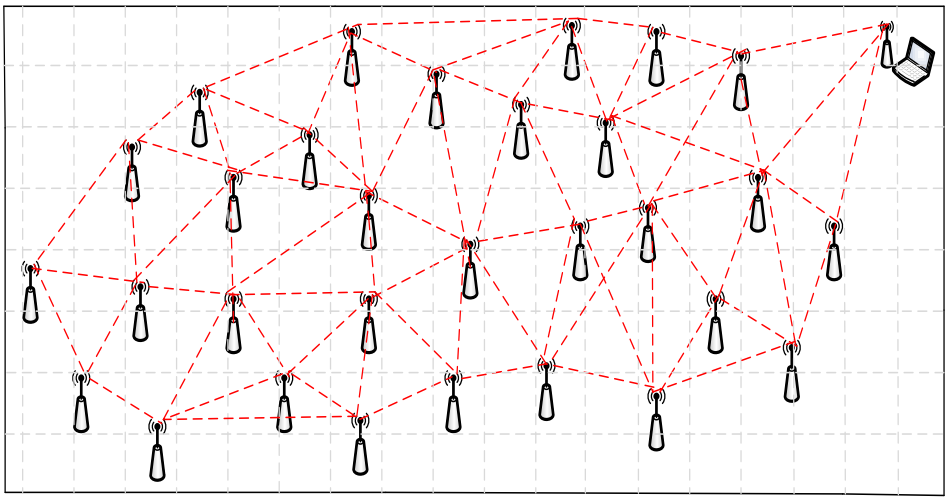


Figure 5.3 Outdoor Deployment Topology

In a tree topology, longer routes of 11 to 13 hops distance are stimulated by picking a routing tree root (i.e., the base station) at the perimeter or the corner of the deployed testbed. Since packet delivery performance deteriorates considerably with more number of hops, longer routes will allow to thoroughly examining the reliability performance of the routing scheme. All sensor nodes generate traffic of fixed size packets at a given transmitting rate under specific testing scenarios. The experimental approach considers a many-to-one real-time event-driven sensor network where sensing nodes deliver their sensing measurements to a single base station under a time constraint and with the

overall target of reliable communications and minimized energy consumption of the relaying sensor nodes. In addition, all motes are homogeneous and commence transmitting with the same residual power capacity using fresh AA batteries. The only exception is the base station which is powered via USB port on a laptop running Linux. The base station acts as a bridging device that has IEEE802.15.4 coordinator functionality. It also acts as a logging device for all metrics and measurements required for evaluating the routing efficiency including RSSI, LQI, CRC, time-stamp, packet sequence number, and appending them to each received packet. Then, the packet logger program in the laptop parses and processes these received packets, and save them to a log file. In other words, the base station relays control packets from the laptop to deployed sensor nodes. These control packets contain adjustment parameters (e.g., transmission rates of originated packets). The base station also relays the collected data packets initiated by sources sensor nodes to the laptop where to be saved in metrics log file.

The sensor motes are broadcasting generated sensing packets to the base station in multihop fashion, and configured to generate packets at a given transmitting rate. Transmission power output of the motes' radios is initially set at its default level of 0dBm and then decreased gradually by steps of 5dBm to the lowest power level of -20dBm (10μW) to allow multihop communication radios of more or less 3 to 4 meters. To maintain an acceptable packet loss, the transmission power is increased as needed to keep RSSI readings of the CC2420 radio above the minimum received sensitivity threshold of -90 dBm as specified in [78]. The base station's transmission power is kept fixed at the nominal transmission power output of 0dBm as it is constantly powered via its USB port from the laptop.

TelosB motes radios are assigned non-interfering channels as TelosB's radio, CC2420, operates in 2.4 GHz ISM band and follows the standard IEEE802.15.4. Unlike CC100 radio used in indoor experiments on Mica2 motes, CC2420 radio is likely to create a lot of interference with IEEE802.11 devices; thereby channel selection is needed. In the outdoor experiments, TelosB motes' radios are configured to use channel 26 of 2.480GHz to avoid overlapping and destructive interference with other operating frequency bands, e.g., IEEE802.11 channels. Since all sensor motes can transmit at any time, the interference needs to be tolerated by using channel 26 in order to avoid physical layer overhead. In addition, MAC layer's automatic acknowledgements (ACK) and automatic repeat request (ARQ), e.g., retransmissions, are disabled to avoid MAC layer overhead and to merely focusing on the delivery performance of the routing layer.

During each experimental run, the routing scheme's energy expenditure per TelosB mote is estimated. As it has been demonstrated earlier in indoor experiments of the previous chapter, the *average amount of dissipated energy* (E_{avg}) required to transmit the data packets to the base station is estimated by converting the measured battery voltage to energy by recalling Equation 4.5 ($E_{avg} = V_{batt} \times I_{drawn} \times Cycle\,Time \times Cycles\ Count$) stated earlier in Section 4.5.1 where, V_{batt} is the battery voltage level, I_{Drawn} is the current drawn by the TelosB mote system, *Cycle Time* is the time spent per TelosB's *CPU cycle* which is (1/8)μs, and *Cycles Count* is the number of *CPU cycles* spent during TelosB mote's tasks. I_{drawn} *and Cycles Count* are calculated based on the preliminary analysis stated earlier in the Chapter 3 in Section 3.7.

Similar to the experimental setup to measure V_{batt} in indoor experiments of the previous chapter, the voltage reference of the Analog-to-Digital (ADC) of TelosB mote system is regularly monitored. As the reference of Analog-to-Digital (ADC) is the battery voltage (V_{batt}), the voltage reference of the ADC (V_{ref}) drops according to

V_{batt}. The TinyOS's VoltageC components are used by the implemented TinyOS modules of the routing scheme for converting the ADC's output count $(Count_{ADC})$ to volts. In other words, since the ADC uses the battery voltage as a full scale (*FS*) reference, the ADC full scale voltage value changes as the battery voltage changes. Moreover, the battery life decreases as the mote operates. Thus, V_{batt} can be regularly computed and measured from the ADC's output by the TinyOS's VoltageC components as a percentage of the initial value of the full capacity. As the reading logic of the ADC output is uneven, it takes a short time to settle and then the fixed reading from the ADC's output count $(Count_{ADC})$ is measured. Then $Count_{ADC}$ is converted to volts by recalling equation 4.6 stated earlier in chapter four Section 4.5.1 where, $V_{batt}(V) = (V_{ref} \times FS) / Count_{ADC}$. V_{batt} readings obtained by the TinyOS's VoltageC components are not only used to estimate the *average dissipated energy* E_{avg} stated earlier in equation 4.5 of Section 4.5.1, but also used to calculate the residual battery capacity converting them to energy values. This is fed into the implemented TinyOS modules of the routing scheme in order to be used in the routing cost function in favour of the most energy efficient route.

Before starting all experiments, all sensor nodes begin with equal battery power levels using fresh AA batteries. At the end of long run experiments of 7 hours, the V_{batt} of TelosB's batteries is tested per mote to ensure batteries sufficient energy to power up the motes. Each TelosB2 mote with its batteries is connected directly to a USB port of a laptop. Then, TinyOS's VoltageC components are used to regularly generate the current readings of V_{batt} as a percentage of the full capacity. As the mote sends messages the battery life decreases. Those batteries with readings less than half of their full capacities are replaced with new fresh batteries.

Finally, since the TelosB mote that acts as the base station has a fixed source of energy as it connected directly to the laptop using its USB port, it can transmit further and other motes that receive its packets will consider them selves as one hop away from the base station but their packets can not reach the base station. This leads to place only few TelosB motes one hop from the base station in order to construct a multihop routing tree and to allow relaying forwarded packets from the source senor nodes towards the base station in multihop fashion.

5.5.3 Results and Empirical Observations

❖ *Outdoor RSSI/LQI Patterns*

Using TelosB motes, a variance in RSSI levels among receivers in addition to the correlation between measured LQI values of a received packet provided by CC2024 radio chip [78] provides bonus information that is not previously available as on Mica2 motes' CC1000 radios [20] used in the earlier indoor experiments of the previous chapter. The correlation values of LQI resulted from CC2420 is the average link correlation value over the first eight symbols following the Start Frame Delimiter (SFD) in a received packet with the highest correlation value. Better Signal-to-Noise Ratio (*SNR*) gives a higher correlation of LQI than in low *SNR* regime, but this is not the case in RSSI values which only measures the energy in the channel. In other words, high LQI correlation values give a better indication of the SNR of the incoming packets and low probability of packet error ratio (*PER*). Bandwidth limitations in the transmitter and receiver chains limit the correlation values to a maximum of about 110dBm as stated in the CC2420 data sheet [78] even for a good link. This is a result of the soft decision at the chip level. Therefore, it is not possible to directly link the correlation value to the chip error rate. However, the correlation value gives a good indication of the link quality and relatively independent of the RSSI level, but it gives poor estimations in the presence of a strong multipath fading of interference-prone channels, where the RSSI

level still indicates a high quality link. However, LQI values provided by TelosB's radio CC2420 sightly fluctuate over the time independently of the RSSI level, i.e., at a receiver sensor node; it was observed that with different signal strengths the link quality values vary slightly. For example, with a high measured signal strength at the receiver mote's radio, the link quality is approximately 108dBm, but with a weak measured signal strength, the link quality is approximately 105dBm even though the signal is about to be lost at the receiver.

❖ *Network Connectivity and Link Dynamics*

Dynamic conditions of the communication channel needs a periodic update of the link quality information. TinyOS-2.x MultihopLQI merely uses link quality information at the physical layer of each received beacon individually. The link quality information is hardware-based and provided by the radio circuitry of the IEEE802.15.4-compliant radio transceiver, e.g., TelosB's CC2420 radio. This pure reliance on one form of channel state information (CSI), i.e., LQI metric, leads MultihopLQI to inappropriately react with the asymmetric links which is a typical feature of low-power WSNs. The RLBR scheme solves the asymmetric link problem by taking the average of the link quality values to provide better packet delivery ratio estimations based on an averaging filter. It also uses bidirectional link estimations based on required retransmissions for active bidirectional monitoring of link status. This allows the RLBR to properly switch to alternate parents when exceeding a threshold of maximum transmission failures. As overall, MultihopLQI performs improperly in the deployed topology due to the limitation of purely depending on the LQI metric to reflect the quality of established link. LQI fluctuates over the time where MultihopLQI uses LQI values of individual beacons instead of averaged values. This is the main reason for its inferior performance to react with the asymmetric links in the process of finding the potential parent and the route searching phase.

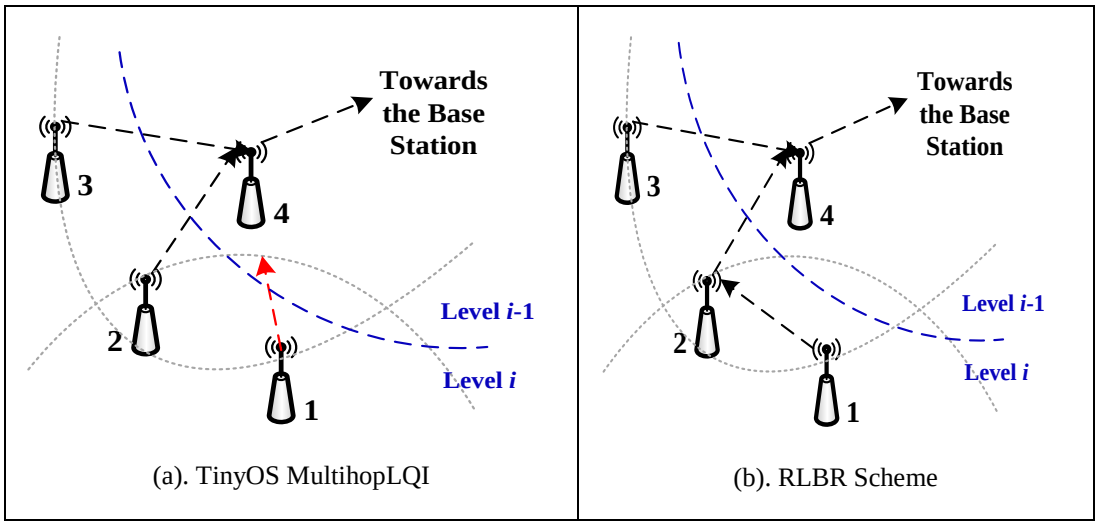


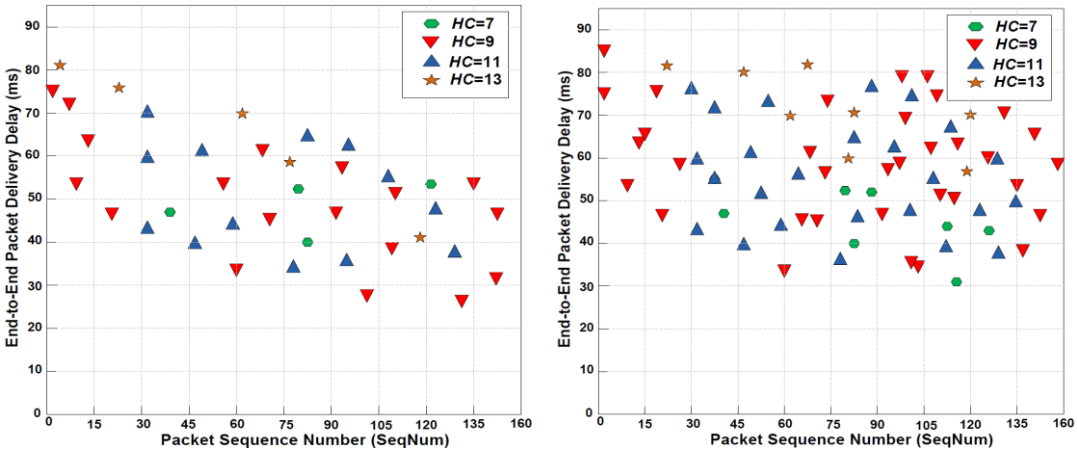
Figure 5.4 Asymmetric Link Problem

Figures 5.4 (a) and (b) show the immediate communication ranges of nodes 1 and 4 in the dotted lines for two situations of MultihopLQI and the proposed scheme respectively. As illustrated in figure 5.4 (a), with MultihopLQI protocol, sensor node 1 chooses sensor node 4 as its parent, but node 4 never receives acknowledgement packets back from node 1. Therefore, sensor node 1 loses its current parent, node 4, even it still receives route messages beaconing from node 4. This is caused by the asymmetric links between nodes 1 and 4 makes node 4 unreachable for node 1's packets. As a result of routing loops prevention in MultihopLQI, node 1 can not choose node 2 as its new parent because both are at the same level *i* of the routing tree. The proposed scheme solves the problem of asymmetric links by averaging link quality values. Based on averaged link quality values, sensor node 1 can switch to other reachable neighbour nodes e.g., node 2, to be its new valid parent after maximum transmission failures due to link asymmetry and transmission range. In addition, the proposed scheme aims to increase the participation in the parent selection by allowing a sensor node to switch to an alternate neighbouring node and pick it as its new parent at the same level. Choosing a parent node from the same level gives the routing scheme more flexibility and unrestricted membership of parent candidates in the parent selection process. After the maximum number of failed transmissions is reached due to link asymmetry or varying

transmission range, node 1 selects node 2 as its new valid parent at the same level *i*-1 as shown in Figure 5.4 (b). Routing loops at the same level are avoided using nodes' *id*s as tiebreaker in addition to tree level number as illustrated in Section 3.4.4.

❖ *Route Configuration Overhead*

The packet transmission rate of the source sensor node is programmed prior to the experiment and the average of multiple runs with different number of source sensor nodes is calculated. Figure 5.5 (a) and (b) show how the proposed scheme and MultihopLQI respectively build and maintain a multihop route in the deployed topology in terms of average end-to-end delivery delay and average hop count (*HC*) via presenting a snapshot of transmitted packets' sequence numbers. During the beginning of the transmission time, the routing scheme has a slightly higher delivery delay due to the uneven distribution of source sensor nodes that are located away from the base station. The random topology of the deployed network shown earlier in Figure 5.3 causes a route configuration overhead as packets are traversing longer routes with more number of hops towards the base station, e.g., *HC*=9 to 11 as an average.



(a). RLBR Scheme (b). TinyOS MultihopLQI

Figure 5.5 Route Configuration Overhead

In the RLBR scheme, longer routes are caused by the chance of choosing parent nodes from the same tree level to increase the flexibility in the parent selection process. However, compared to MultihopLQI, the delivery performance of the proposed scheme

immediately improves as the end-to-end packet delivery delay decreases considerably once the route has been constructed even with a higher number of hops. As the network stabilise, the proposed scheme adaptively constructs a reliable route as it requires fewer retransmissions of route messages. On the other side, MultihopLQI suffers from a poor packet delivery performance during the whole run as it has a higher number of retransmissions to cope with the uneven distribution of the source sensor nodes. This causes MultihopLQI to perform frequent route repairs with longer delays; thereby increasing energy consumption for retransmissions.

❖ *Recovery from Link Failures*

The RLBR scheme provides a faster recovery from broken links due to the hybrid approach utilising backup neighbouring routing tables. This can be seen in Figure 5.6 when a link is broken at 100ms of the transmission time. Once an alternative reliable energy-efficient route is established using consecutive repair phases, the average end-to-end delivery rate increases considerably, thereby the average throughput is improved. This alternative route requires only smaller number of retransmissions to successfully deliver a data packet at an average delivery rate of 99.6% after 40ms from the time at which the route was broken. On the other hand, MultihopLQI provides an average delivery rate less than 78% after the same period of time. As the time passes, the RLBR achieves a higher delivery rate. Conversely, MultihopLQI begins with a higher delivery rate and initially achieves a lower average end-to-end delivery delay. This is because the route configuration start-up time required by the RLBR for updating routing tables and parent selection process takes some time. As MultihopLQI maintains only a state for one parent node at a time, neither routing tables nor blacklisting are used. However, this results in the additional energy cost associated with the significantly increased packets retransmissions required to successfully deliver a data packet.

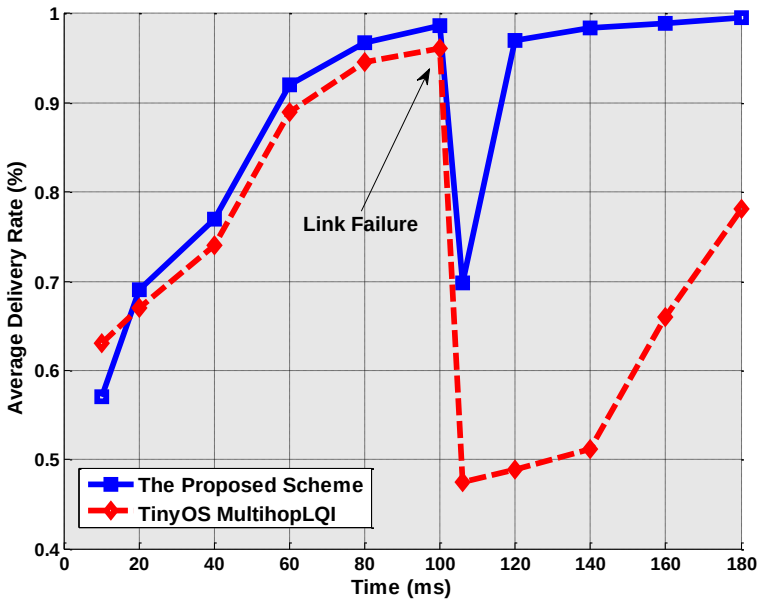


Figure 5.6 Average Delivery Rate vs. Link Failures

In the view of the cost of beaconing route messages, e.g., control packets, over a few hours, the beaconing rate is adaptive on per sensor node basis. It starts with a slightly high rate in the proposed scheme at the beginning due to the rapid establishment of the routing tree then begins to decrease and becomes stable at a lower rate. Figure 5.7 illustrates, on an hourly basis, the average number of route messages (control packets) that were transmitted per sensor node in order to build and maintain the routing tree. The message beaconing pattern in the proposed scheme is slightly increased at the fourth hour due to an intentional link failure. This failure was introduced to demonstrate the rapid reconstruction of an alternative, but longer, route. Once again it adaptively embarks on an uneven rate pattern in order to become stable eventually. By comparison, MultihopLQI avoids routing tables by only maintaining a state for the best parent sensor node at a given time. It keeps transmitting control beacons at a constant rate of 30 beacons per seconds, considerably higher than the proposed scheme. The average mount of route messages sent per sensor node in MultihopLQI linearly increases over long run in terms of seven hours.

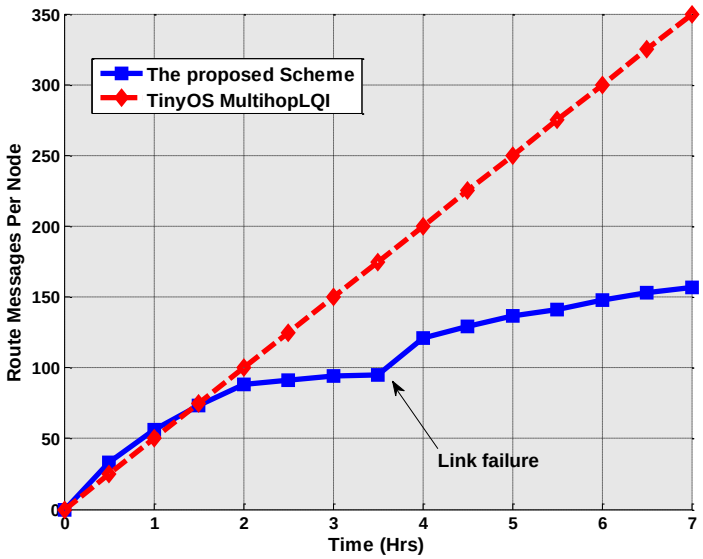
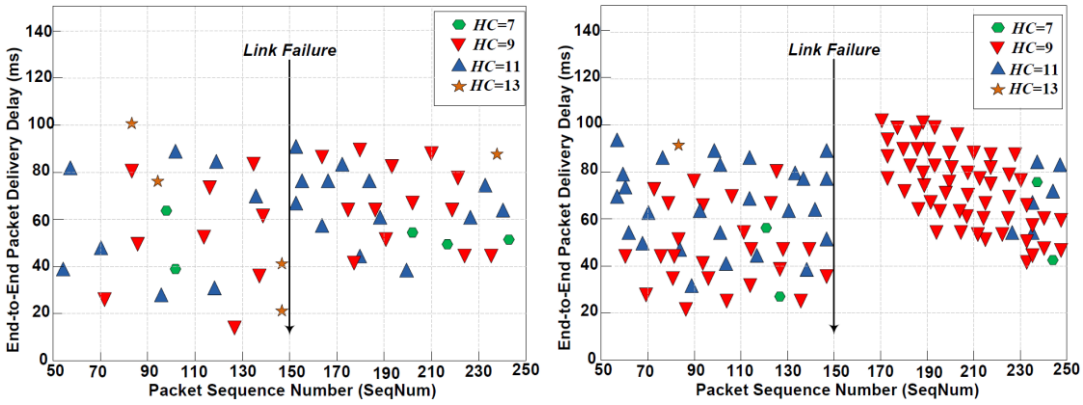


Figure 5.7 Average Number of Route Messages per Sensor Node

In order to jointly evaluate the reliability and delivery performance of the routing scheme, a number of intermediate wireless sensor nodes were switched-off or removed to create broken routes between source sensor nodes and the base station. Figure 5.8 (a) and (b) illustrates the end-to-end delivery performance of the proposed scheme and MultihopLQI respectively in terms of end-to-end delay and hop count (*HC*) when a route is broken after a packet with a sequence number (*SeqNum*) of 150. The proposed scheme reacts efficiently and responds swiftly to recover from a broken link along the preselected path. It maintains an alternative, reliable and energy-efficient route to recover. This route reconfiguration time is 66.40ms. This newly constructed route is used temporarily as a backup route to deliver source-originated data packets in a timely manner towards the base station. However, the alternative route may require additional hops, leading to an increase in the average end-to-end packet delivery delay. In this case it is slightly increased to 81.32ms. In contrast, MultihopLQI is incapable of rapidly recovering from broken routes if a wireless mote on a preselected route is removed. Even though MultihopLQI results in a shorter average end-to-end delay for packet delivery of about 78.43ms, recovering from the broken route takes a much longer time of around

98.52ms. Overall, MultihopLQI lacks stability, frequently restructuring its routing tree in response to changes in its LQI, hardware-based, reliability metric. Although MultihopLQI did recover from the link failure, its delivery ratio was noticeably reduced after a shorter time. This leads to a lower average packet delivery rate for MultihopLQI as compared to the proposed scheme, validating the aforementioned results.



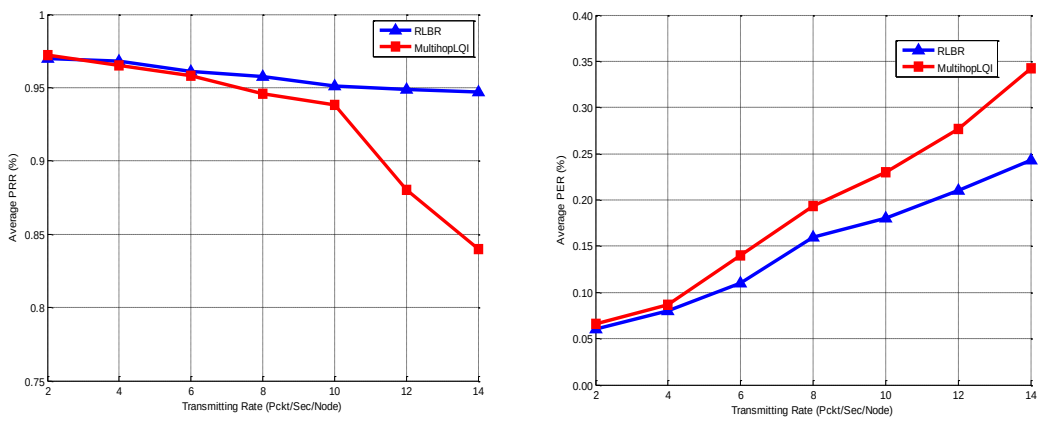
(a). RLBR Scheme (b). TinyOS MultihopLQI

Figure 5.8 Responsiveness to Route Recovery

❖ *Packet Delivery Performance*

In the view of the *per-hop packet delivery performance* of the routing scheme, all nodes were selected and programmed to generate data packets at transmission rates of 2 to 14 packets per second per source node in step of 2. In addition, since all nodes can either send their packets and/or forward other nodes' packets, the *per-hop packet delivery performance* can be gauged in terms of the *average packet reception ratio* (*PRR*) which is the percentage of successfully received packets at the base station divided by sent packets within the network. Averaged values of packet delivery measurements are considered using an average filter to better reflect packet delivery performance on dynamic, asymmetric, and time-varying lossy links. With the view of considering different routing scenarios with various packet forwarding loads, *PRR* was estimated by running the experiment repetitively for 7 times each for 10 minutes with various packets

transmitting rates. It can be seen from Figure 5.9 (a) that the proposed scheme has an improved *PRR* as the transfer weight rises by increasing the packet transmission rate of the source sensor nodes. While MultihopLQI starts to achieve a lower *PRR* at high packet transmission rates greater than 10Pckt/Sec/Node due to selecting optimal routes at all times based only on link quality, the proposed scheme maintains a higher *PRR* by reorganising the routing tree with suboptimal links to redistribute the amount of forwarded packets over sensor nodes with lower relaying loads. Furthermore, the advantage of redistributing packet relay weights is also validated in terms of the *average packet error ratio* (*PER*) which is calculated as the percentage of the sum of the erroneous received packets that failed to pass the CRC check sums at the base station to the total number of all received packets, i.e., successful plus erroneous. A data packet is considered to be faulty if either fails to pass the CRC check due at least one faulty bit or its length/type of the packet's type field is incorrect. Figure 5.9 (b) illustrates how the average *PER* changes at the base station as the packet transmitting rate rises at the source nodes.



(a). Average *PRR*

(b). Average *PER*

Figure 5.9 Packet Delivery Performance over Lossy Links

Unlike MultihopLQI, the RLBR doesn't use individual LQI measurements to estimate link connectivity for selecting reliable routes but takes advantage of averaged values of time-varying LQI and success reception to avoid lossy links and to cope with dynamic

link asymmetry problem. This could lead to minimising packet retransmissions due to link failures. Consequently, the packet delivery using the RLBR results in a lower average *PER* compared to MultihopLQI in conditions of poor connectivity and link asymmetry. However, at low transmission rates MultihopLQI also achieves low average *PER* as the variation in packets relaying weight still small at low transmitting rates.

Packet loss ratio (*PLR*) in WSNs typically depends on a composite set of parameters, including the location of deployed sensor nodes in relation to the base station, spacing between adjacent sensor nodes, gain of mote's antenna, the environmental conditions that affect the quality of the channel, and the number of hops that the packets traversing to be successfully delivered to the base station. Figure 5.10 shows how selected parent nodes along the routing path towards the base station can have different packet loss readings based on their distances and locations in relation to the next hop node. To keep the results independent of the transmitting power, the results are averaged for many runs with the lowest transmitting power output of -20dBm and the network is deployed in a grid topology with various one-hop spacing distances, i.e., 1, 2, 3, 4, 5 meters.

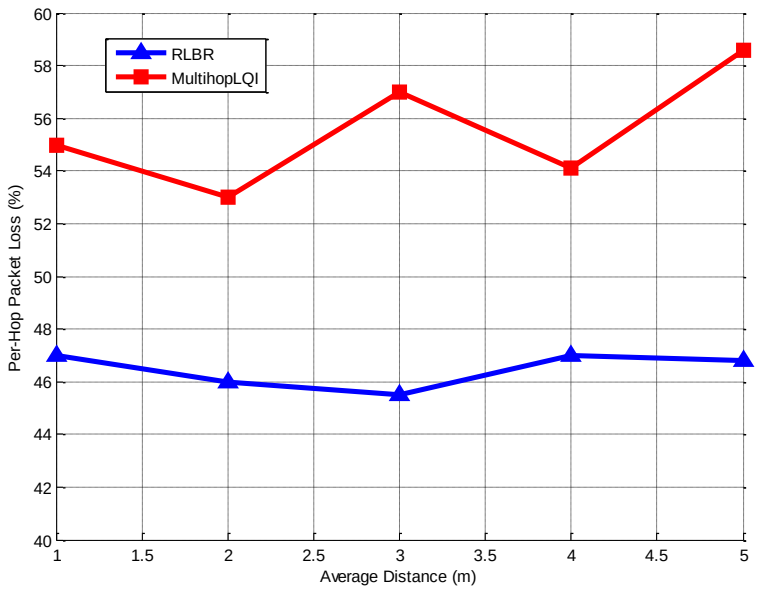


Figure 5.10 Per-Hop Packet Loss in Grid Topology

Using the lowest RF transmission power is to create poor channels with higher loss ratios and to stimulate a poor connectivity environment and instability in the routing topology. The packet loss may also occur due to the dynamic surrounding environment of different existing objects. Averaged values of packet delivery measurements are considered to better reflect packet delivery performance on dynamic, asymmetric, and time-varying lossy links. Since the RLBR uses combined active bidirectional software and hardware link estimators, it can choose the most reliable parent sensor node with the lowest packet loss ratio. However, the effective parent selection approach used by the proposed routing scheme could lead to an additional computation overhead. Conversely, MultihopLQI depends only on individual LQI values that are provided by the physical layer of the RF transceiver. This could yield parent sensor nodes with a higher packet loss of more than half of the transmitted packets towards the base station due to asymmetric links problem and poor connectivity. This higher per-hop packet loss causes MultihopLQI to increase packet retransmissions to deliver the packet successfully towards the base station; thereby resulting in a larger amount of energy expenditure.

- *Balanced Energy Depletion*

The average dissipated energy is calculated in terms of the traffic weight by running the experiment for 7 hours with a fixed packet transmission rate of one packet per second per sensor node. Updating routing tables and the parent selection process in the RLBR scheme requires a slight computation overhead; this could cause minor additional computational energy dissipation. On the other side, MultihopLQI maintains only a state for one parent node at a time and neither routing tables nor black-listings are used with less storage of routing information and small amount of energy required for computations. However radio communications are the major energy consumer since MultihopLQI consumes more energy for packets retransmissions due to packet collisions and instability in its routing topology based on LQI as reliability metric. MultihopLQI

does not consider the available energy at each relaying sensor node. This makes the energy dissipation cost of MultihopLQI protocol as a function of the packets transmissions and results in unbalanced energy usage on the relaying sensor nodes as packets transmission consumes much higher energy than computations. In MultihopLQI, sensor nodes broadcast control packets at constant rate and its beaconing rate doesn't adjust with topological dynamics in favour of energy efficiency.

In terms of energy, non-adaptive high rate beaconing expends more energy for unnecessary transmissions in conditions requiring infrequent topological changes. In addition, most relayed packets are routed through optimal routes based mainly on link quality. As a result, the selected route will be used frequently and the sensor nodes along this route will be exhausted quickly. This leads to an imbalance in the energy utilisation throughout the entire network. Compared to MultihopLQI, the RLBR makes trade-offs between routes based on link reliability and energy efficiency in favour of a more even distribution of the forwarded packets among the relaying sensor nodes. In addition, the RLBR broadcasts fewer route messages over the life of the network. As a result, it consumes only about 35% of the energy required for the transmissions of route messages compared to MultihopLQI.

To effectively estimate the average amount of energy consumed by relay sensor nodes for delivering a data packet towards the base station, the *average packet delivery cost* ($1/\eta$) is used as a *routing overhead metric*. As stated earlier in equation 5.2 of Section 5.5.1, this cost metric ($1/\eta$) accounts for the ratio of the total number of all control and data packets sent throughout the network to the total number of data packets received at the base station. Figure 5.11 demonstrates how the packet delivery cost ($1/\eta$) for the RLBR scheme and MultihopLQI changes over the long run and gives an estimation of the average energy cost incurred for packet transmission throughout the network. The

RLBR transmits a smaller amount of route messages or control packets than MultihopLQI. The decrease in route messages transmissions of the RLBR is a result of avoiding unneeded route message transmissions using data aggregation, adaptive beaconing, and reliable and efficient route selection. This results in lower beaconing rates, lower control cost and routing overhead while network topology stabilizing; thereby achieving a much lower energy consumption in the RLBR scheme.

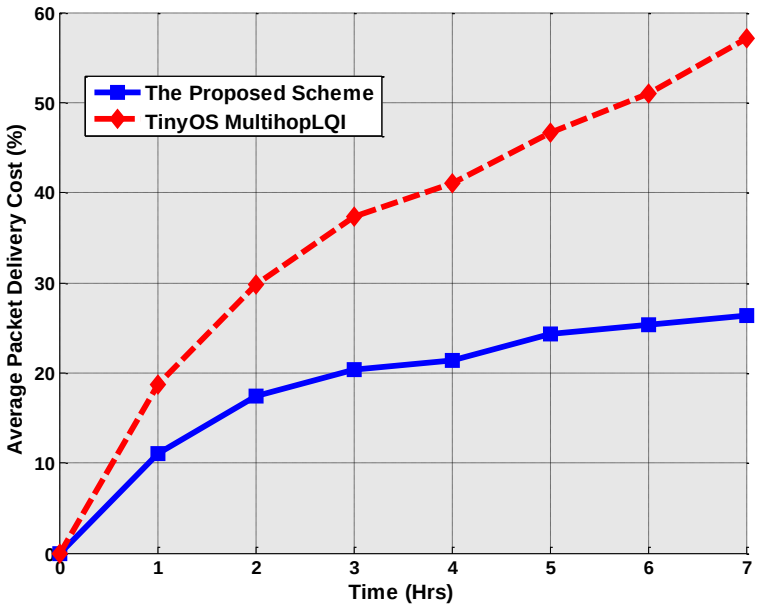


Figure 5.11 Average Packet Delivery Cost ($1/\eta$)

5.6 Discussion and Conclusion

In this work, the RLBR scheme was tested based on a per-hop load balancing mechanism of the routing layer. It leverages recent advancements over the standard network layer components provided by the TinyOS2.x. The RLBR allows for adapting the amount of traffic to the fluctuations in network connectivity and energy expenditure. From a reliability viewpoint, it creates a routing tree using the estimated numbers of transmissions and retransmission to the base station and link quality estimations based on sequence numbers (*SeqNum*) of successfully received packets. The RLBR performs well with a high success rate of packet delivery and moderate energy consumption.

As observed throughout the conducted experiments so far, packet error results in an unsuccessful CRC and it cannot be distinguished if it is either due to MAC layer collisions or physical layer packet errors. At the MAC layer, a considerable amount of the energy is spent for repairing lost transmissions due to poor connectivity resulted from the surrounding environmental conditions and the moving objects, e.g., cars and people. Therefore, MAC layer is vital for the upper routing layer, which can reduce the benefit of a smart routing scheme. The *PER* statistics is also affected by various factors such as signal-to-noise ratio of the transmission channel, signal strength, bit synchronization problems, multipath fading. In addition, the RLBR scheme also aims to balance the weight of relayed packets to evenly distribute the energy utilisation. It consumes less energy while reducing topology repair latency and supports various aggregation weights by redistributing packet relaying loads.

To this point, the results obtained experimentally in chapters 4 and 5 have been obtained using a real world sensor network which is important and effective. While the results revealed so far have highlighted the substantial performance gains of the RLBR scheme, the next chapter aims to extend the prior testbed experiments using simulations for larger networks and to evaluate the impact of traffic relaying on an individual node lifetime.

Chapters 6
Maximum Network Lifetime Routing

6.1 Background

Organizing wireless sensor networks (WSNs) using energy efficient routing algorithms enables the efficient utilization of the limited energy resources of the deployed sensor nodes. However, the problem of uneven energy consumption will occur due to imbalanced workload. This is tightly bound to the role of an individual node (e.g., relay or parent node) and to the location of a particular node in the network (e.g., along the routing path and closer to the base station). When the network is deployed randomly using homogeneous sensor nodes where all nodes have the same level of built-in resources, it is important to ensure that energy dissipation of these identical nodes is balanced and distributed evenly with the purpose of maximising network longevity. An imbalanced workload on gateway nodes may lead to a hot-spot or routing-hole problems [104,105]. In this chapter, the RLBR scheme is extended to reduce the impact of traffic congestion on an individual relaying node lifetime without deteriorating the end-to-end reliability performance. RLBR is tested using simulations to validate the RLBR on larger-scale WSN. The simulations show a balanced energy usage and a significant lifetime gain per relaying node; thereby avoiding an early termination of the entire network.

6.2 Routing Hole Problem

The routing-hole problem can be considered as a natural result of the tree-based routing schemes that are widely used in WSNs, where all nodes construct a *many-to-one* multihop routing tree to a centralized root, (e.g., gateway). For example, relaying nodes with the best link state on the routing path and closer to the base station have a heavier

workload and experience a faster energy depletion rate than their peers. This shortens the lifetime of these nodes and leads to *network partitioning* [61,104,105] as reliability-oriented routing protocols typically use reliability metrics to avoid unreliable links and thus directly make the energy problem worse. It is the aim of load balancing schemes to avoid the formation of hot spots, or at least reduce the significance of the problem and avoid ruining the energy conservation.

The availability of multiple routes to the base station depends on the topology of the network and its surroundings and is constrained by the radio characteristics. In the best possible load balancing scenario, all sensor nodes can reach the base station directly in one hop and only send what they generate. At the opposite end of the load balancing spectrum, one particular relay or a small number thereof may be the only way for sensor nodes to reach the base station, thus forming a *topological bottleneck*, resulting in early network partitioning. An extreme case is a line network where only one nearest neighbour routing choice is possible. An illustration of these situations is shown in Figure 6.1, which explains how the closer a node is to the base station, the higher its workload. Each relay or parent node is a topological bottleneck with respect to the upstream or children nodes. Regardless of the routing strategy, the mainstream nodes closer to the base station have to forward more packets than the ones towards the periphery of the network.

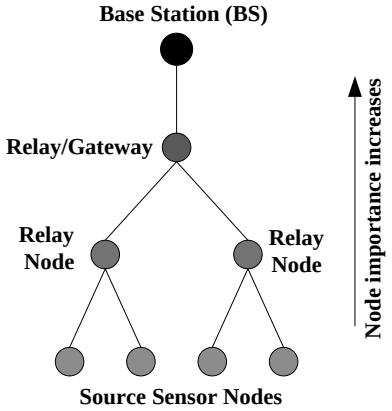


Figure 6.1 Many-to-One Nearest Neighbour Routing

6.3 Related Work

The reliability-oriented routing protocols tend to overexploit one-hop sensor nodes with the best channel towards the base station; consequently the network life time is considerably reduced by frequently using such optimal links [120,121]. On the other hand, load balanced routing can extend the network lifetime with respect to link reliability by using suboptimal links towards the base station [56,104,105,113]. In addition, many energy-efficient strategies have been proposed to collect and route the data packets towards the base station, trying to maximize the lifetime of sensor nodes while maintaining system performance and operational fidelity [49,71,93,113,119,123].

6.3.1 Definitions of Network Lifetime

In the literature, the meaning of network lifetime has many definitions depending on the sensor network's application and/or deployment topology as the network lifetime has a great significance in the design of WSNs [61]. In [135], network lifetime is generally defined as the time after which a certain fraction of sensor nodes run out of their batteries, resulting in a routing hole or hot spot within the network. However, network lifetime depends typically on other factors such as the region of the observation, the source sensing behaviour within that region, base station location, deployed topology, number of nodes, radio path loss characteristics, efficiency of node's hardware circuitry and the energy available on a sensor node. Network lifetime is defined in [136] as the time span from the deployment to the time when the network is considered non-functional. However, the time at which a network should be considered non-functional is application-specific. For example, network lifetime can be defined either as the time when the first sensor node dies [139], a percentage of sensors die [140], the network partitions [141], or the loss of coverage [137] occurs although the remaining nodes can accomplish the assigned task. Therefore, another definition based on the ratio of dead nodes to the total number of nodes in the network is often used as defined in

[138,142,143]. Other definitions are also proposed based on the sensing area coverage [137] or network connectivity [136].

Since determining the network lifetime is application dependent and depends on the importance and the location of deployed sensor nodes, the upper bounds of network lifetime have been studied and derived in the literature based on different characteristics such as the spatial behaviour of the source sensor nodes, sensing coverage and network connectivity. In view of the spatial behaviour of the data source, a simplified version is initially considered in [144] where the data source is a specific point, and the source is connected to the base station with a straight line consisting of relaying sensors. The optimum length of a hop is derived and consequently the number of hops in the path to minimize the total energy consumed for the data delivery. Then, the assumption of a source concentrated on a point is removed and assumed that the source sensor nodes are distributed over an area. This work has been extended in [145] to networks whose sensor nodes may perform different tasks of sensing, relaying and aggregating. Network lifetime based on the sensing area coverage was studied in [137]. It is assumed that the sensor nodes have a circular sensing region and are distributed over a squared area. Although the upper bound is derived for an estimated situation when the area goes to infinity, it has been shown through simulation that the derived bound is also reasonable for networks over a finite area [137]. Network longevity also has been studied in [135] based on network connectivity to find the probability distribution function (PDF) for maximising network lifetime where the network is divided into domains and each domain contains a randomly deployed and deterministically assigned number of sensor nodes. The lifetime is defined as the time when a routing hole occurs in the routing scheme and the data of interest does not reach the base station [135].

6.3.2 Benchmark Protocols

TinyOS-2.x implementation of MultihopLQI [43] is selected as the reliability-oriented routing protocol benchmark. As experienced earlier in the outdoor testbed of the previous chapter, MultihopLQI always picks the most reliable links and does not explicitly pursue energy or load balancing in its routing scheme. This could yield a routing tree with a large number of hops and extra messages being sent and overheard. However, occasional load balancing is obtained in the presence of fading, which modifies link quality values and occasionally influences parent selection decision. Similar to MultihopLQI, the RLBR scheme uses Link Quality Indicator (LQI) values which are computed based on bit or packet error rates (also known as correlation values) [78], but it also uses suboptimal links and considers the residual energy capacity which is estimated during network operation. The major difference between the RLBR scheme and MultihopLQI lies in the parent selection process. A parent sensor node in the RLBR scheme is selected when it offers the most energy efficient route to the base station and link costs based on Channel State Information (CSI) are used to obtain route costs. In the RLBR scheme, each sensor node evaluates the parent announcements it has received to select the parent with highest lifetime metrics by selecting a next hop with the highset residual energy and the most reliable link. If a sensor node receives announcements with similar residual energy and link quality, it uses a tiebreaker by picking a potential parent with fewer hops toward the base station while considering the relaying workload.

On the other side, numerous energy-aware routing protocols have been proposed in the literature for WSNs [5,34,42,49,73,88,121,126,141,142]. One of the most popular of such protocols is Energy-Aware Routing (EAR) which is proposed in [49]. EAR is a reactive routing protocol. It is a destination-initiated protocol where the consumer of data initiates the route request and maintains the route subsequently based on geographical information. Multiple paths are maintained from source to destination. However,

diffusion sends data along all the routes at regular intervals, while EAR uses only one path at all times. Therefore, the potential problem in existing energy aware routing protocols are that they find the lowest energy cost route and use that route for all communications. However, that is not the optimal solution for maximising network lifetime as using a low energy cost route frequently leads to energy depletion of the sensor nodes along such route and may lead to network partitioning. To counteract this problem, the RLBR scheme uses suboptimal energy efficient routes. This ensures energy balancing and that the lowest energy cost routes will not get depleted and the network degrades gracefully as a whole rather than getting partitioned. To achieve energy balancing, multiple routes are determined between source and destinations, and each path is assigned a probability of being chosen, depending on the link reliability, workload and energy metrics. Every time data is to be sent from the source to destination, one of the routes is chosen depending on the probabilities. This means that none of the routes is used all the time, preventing uneven energy dissipation.

6.4 Network Energy Dissipation Model

6.4.1 Model Description and Assumptions

A large-scale WSN is considered to be event-driven tree-like network deployed randomly with stationary homogeneous sensor nodes and a single perimeter base station. Initially, the base station periodically retransmits route advertisements (e.g., periodic beaconing) so that the routing tree is continuously maintained. Then, the beaconing rate is adapted to topology dynamics (details are explained in Section 3.4.3). Source nodes control the rate of the generation rate of data packets and all the data packets are of the same size. Nodes estimate their link quality with their neighbour peers. Since the predominant traffic pattern in the network is a *many-to-one*, relaying nodes perform data aggregation along the routing tree. Aggregation points occur close to the event source as

early as possible to maximize the aggregation benefit. As shown in Figure 6.2, each node communicates immediately with other neighbour nodes and the base station if they are within their radio transmission range using a CSMA-based MAC protocol, specifically, IEEE802.15.4. Network lifetime is considered as the time until network partitioning occurs and the network can no longer perform its assigned tasks (e.g., the base station no longer receives data packets of interest from the source nodes). However, the metric used to determine the network lifetime is also application dependent. Thus, the probability of reaching a maximum lifetime for a determined number of randomly deployed nodes is derived based on an energy dissipation model (explained in Section 6.4.3). The total energy dissipation per relaying node for transmitting and/or receiving a packet of *n* bits over one hop wireless link is estimated during the average interval between the first and last data packet arrivals at the base station. Residual energy level is estimated after transmitting or receiving one unit of data.

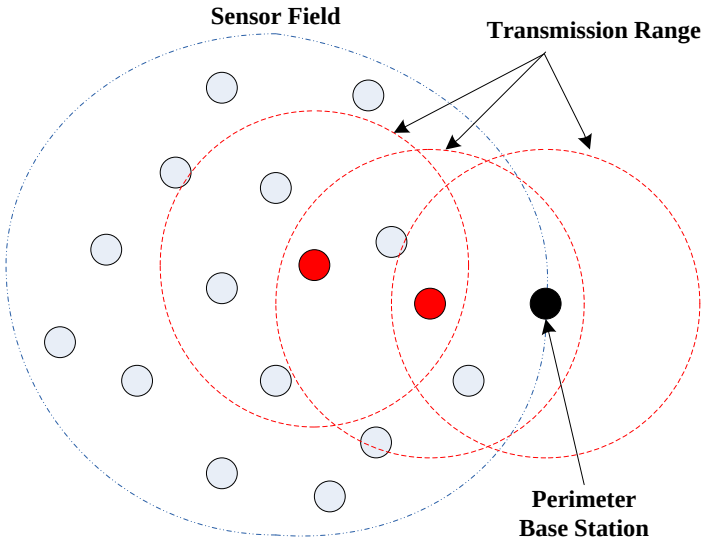
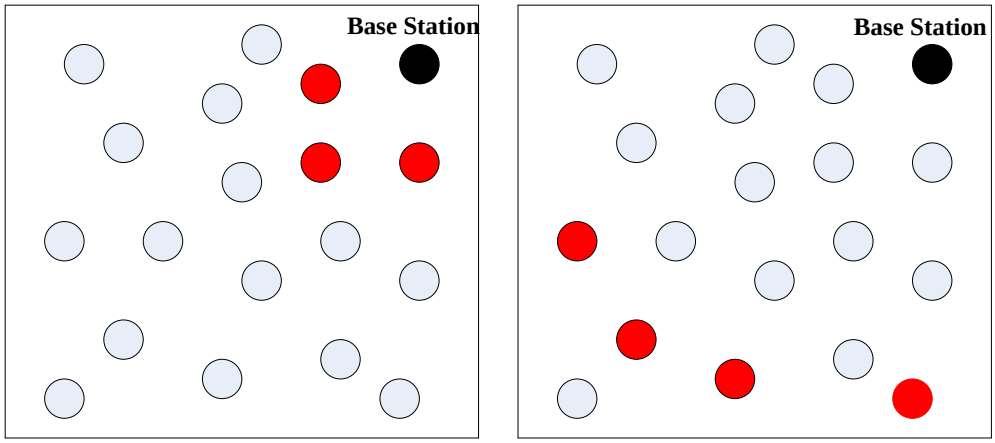


Figure 6.2 Homogeneous Sensor Nodes with Fixed Transmission Powers

6.4.2 Importance of Deployed Sensor Nodes

Avoiding the hotspot problem in WSNs depends strongly on the location of sensor nodes and thus it is a deployment problem. Figure 6.3 shows how the location and the

importance of different sensors could affect network lifetime. In the figure, the red sensor nodes represent the sensors that have run out of energy and the white ones denote the ones that are still alive. In both scenarios shown in Figures 6.3 (a) and (b), the sensor network is not fully functional since in both cases a number of data packets of interest from some sensors can not reach the base station. For example, in Figure 6.3 (a), although there are only few dead relay sensor nodes, the base station cannot receive data from most of the downstream sensor nodes. In Figure 6.3(b), there are a small number of dead sensor nodes, but the base station can still receive data from most of the sensors within the network. Consequently, the damage to the sensor network by failed sensor nodes is not only related to the number of failed sensor nodes but also related to the location and the importance of such nodes.



(a). Dead Nodes Close to the Base Station **(b). Dead Nodes Distant from the Base Station**

Figure 6.3 Importance of Deployed Sensor Nodes

To this end, nodes in the sensor network have different importance. Each node gauges the importance of its upstream parent node. Based on this analysis, the closer the node to the base station, the more important and the more critical it is. Since some critical relay nodes malfunction due to power failure or physical damage can cause significant topological changes and may require network reorganisation It is very important to minimise energy consumption of each individual node in order to maximise lifetime. As

a result, lifetime analysis at the sensor node level is performed and discussed using large-scale simulations.

6.4.3 Energy Consumption per Relaying Sensor Node

Characteristically, a wireless sensor node has different integrated circuitry components including a sensing system, analog-to-digital conversion (ADC), digital signal processing (DSP) and a radio transceiver. The sensing system application dependent and the communication components are the major energy consumer. Energy efficiency of WSN has been generally based on a simple first order radio model of a sensor node [34,139]. In Figure 6.4, a realistic energy dissipation model is presented by separating the energy expenditure of each communication component, and considering the radio environment. This provides a clear understanding of the energy consumption of the communication components that are limiting the routing performance.

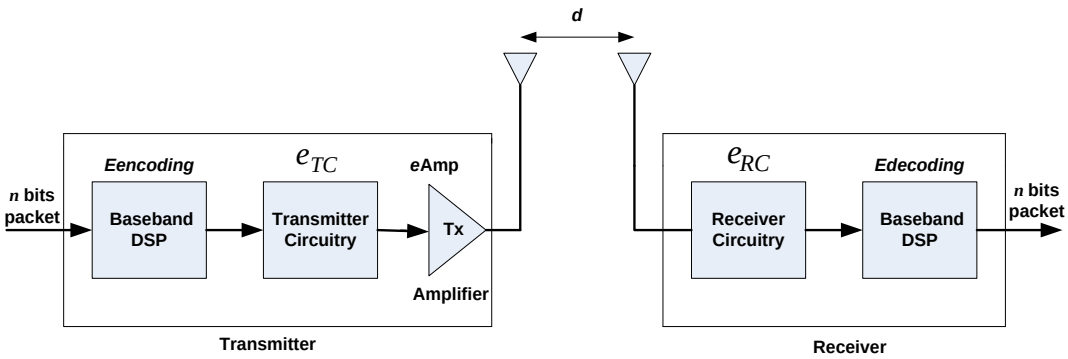


Figure 6.4 Components of Communication Model [34,139]

Firstly, the total energy dissipation per node for transmitting a packet of *n* bits over one hop wireless link can be expressed as in Equations 6.1 to 6.3 respectively [139].

$$E_{Total\_TX} = (E_{TX}(n,d) + P_T \times T_{st} + E_{enc}) \tag{6.1}$$

But, $E_{TX}(n,d) = n(e_{TC} + e_{Amp} \times d^{\alpha})$

$$= n \times e_{TC} + n \times e_{Amp} \times d^{\alpha} \tag{6.2}$$

$$= h + c \times d^{\alpha} \tag{6.3}$$

Where,

P_T = power consumed by the transmitter circuitry during startup time.

T_{st} = transceiver startup time (MAC protocol dependent).

E_{enc} = energy dissipated to encode transmitted data packets.

e_{TC} = energy dissipated per bit by the transmitter circuitry.

e_{Amp} = energy used to run the transmitter amplifier.

n = packet length in bits.

d = distance between transmitter and the intended receiver sensor node.

α = pathloss exponent.

c = pathloss coefficient.

h = overhead energy for a packet transmission.

$e_{Amp} \times d^{\alpha}$ = energy dissipated per bit transmission over distance d.

Secondly, the total energy dissipation per node for receiving a packet of n bits over one hop wireless link can be expressed as in Equations 6.4 and 6.5 respectively [139].

$$E_{Total\_RX} = (E_{RX}(n) + P_R \times T_{st} + E_{dec}) \tag{6.4}$$

But, $$E_{RX}(n) = n \times e_{RC} \tag{6.5}$$

Where,

P_R = power consumed by the receiver circuitry during startup time.

E_{dec} = energy dissipated to decode received data packets.

e_{RC} = energy dissipated by the receiver circuitry per bit.

The effect of the transceiver startup time (T_{st}) depends mainly on the type of the underlying MAC protocol used. To minimize power consumption it is desired to have the transceiver in a sleep mode as much as possible however constantly turning on and

off the transceiver also consumes energy to bring it to readiness for transmission or reception. e_{TC}, e_{Amp}, and e_{RC} are hardware dependent parameters. The path loss exponent α depends on the surrounding terrain and is determined by empirical measurements. The typical value of α for WSNs varies from 2 for free space propagation model to 4 for multipath fading or shadowing channel models [146].

Typically, there are two possible transmission power scenarios: variable (also known as adjustable or dynamic) and fixed (also known as constant) transmission power. In the variable transmission mode, the radio transceiver is able to adjust its output signal power level depending on the distance to the intended receiver and the power consumed for transmission can be minimized as needed. In constant transmission mode, the radio transceiver transmits at the same fixed output power level for all transmissions irrespective of the distance between the transmitter and the intended receiver. In this chapter, fixed transmission power is considered because the majority of the available commercial radio transceivers (e.g., CC100 [20] and CC2420 [78]) do not have the capability for dynamic power adjustments even though the adjustable transmission power could benefit the network lifetime as illustrated in the next chapter. Using the constant transmission mode, the energy dissipated per bit transmission of an individual sensor node is fixed to a value of ($e_{Amp} d^{\alpha}$) at the transmitter node from equation 6.2. This energy is consumed for amplifying the signal to achieve adequate signal-to-noise ratio (SNR) over a distance *d*. As stated in [34,139,146], it is considered that the energy consumed for transmitting is proportional to the distance between transmitter and the intended receiver. The energy dissipation per relay sensor node in a multihop network is merely modeled to the actual relaying and communication process (i.e., transmitting and receiving). The energy spent for encoding, decoding and transceiver startup is normalized and not considered in the simulations of this chapter.

The total energy dissipated by a parent node to relay a packet of *n*-bits from source node to the base station can be combined from Equations 6.2 and 6.5 to form Equation 6.6.

$$E_{relaying} = n(e_{TC} + e_{RC} + e_{Amp} \times d^{\alpha}) \tag{6.6}$$

The current residual energy level of a node after relaying one packet of *n*-bits can be calculated by deducting the initial or the previous energy value from the value of the energy dissipated by a node for relaying by Equation 6.7.

$$E_{residual} = E_{initial} - E_{relaying} \tag{6.7}$$

From Equation 6.8, the energy consumption of relay node is used to measure the average energy dissipated by this node in order to relay (transmit and receive) *N* data packets of *n*-bits from the source sensor node to the base station over a WSN of *M* sensor nodes. This metric is used to indicate the energy efficiency level of the WSN and to give an indication of the network status in terms of energy consumption.

$$Average\ Dissipated\ Energy = \frac{\sum_{i=1}^{M} E_{relaying}}{M \times N} \tag{6.8}$$

Where,

M = the total number of operational sensor nodes in the network.

N = the amount of data packets received by the base station.

6.5 Performance Evaluation

This section presents the simulation-based methodology to evaluate the operation of a WSN by using the RLBR scheme and to benchmark it with other routing protocols such as TinyOS MultihopLQI [43] and EAR [49]. To keep the performance evaluation reasonable, the impact of network routing is considered on energy efficiency together with load balancing and the entire network lifetime.

6.5.1 Evaluation Metrics

Key metrics are used for evaluating the performance of the RLBR scheme against the chosen benchmarks using rigorous simulations. These includes: *operational network lifetime*; *average dissipated energy*; *packet delivery ratio*; and *average end-to-end delay*. While the operational network lifetime and average dissipated energy metrics are used to evaluate the benefit of network lifetime maximization, the packet delivery ratio and the average end-to-end data packet delay metrics are used to measure the best-effort traffic of the routing scheme. Lifetime and energy metrics evaluate the energy efficiency of the routing scheme. End-to-end delay is one of the most important metrics when analysing the performance of real-time QoS-oriented routing schemes. However, all these performance metrics are interrelated and not completely independent. For example, low workload impacts both the packet delivery ratio and delay, as this may lead to lower routing congestion and less multiple-access interference.

- *Operational network lifetime* can be obtained by calculating the average time spent between the commencement of the simulation and the last data packet received at the base station.
- *Average dissipated energy* is defined as the average energy consumed by the node for transmitting or relaying data packets from the source node to the base station. This metric is used to indicate the energy efficiency level of the deployed WSN.
- *Packet delivery ratio* (PDR) is the ratio of the number of successfully delivered data packets at the base station to the generated and injected data packets by sensor source nodes as in Equation 6.9. This metric also indicates the successful transmission rate. From the data delivery point of view, the higher the packet delivery ratio, the lower the packet loss, the more efficient the routing protocol.

$$PDR = \frac{Number\,of\ Successfully\,Delivered\ Data\,Packets}{Total\,Number\,of\ Transmitted\,Data\,Packets} \times 100 \qquad (6.9)$$

- *Average end-to-end packet transfer delay* is defined as the averaged amount of time spent to relay data packets successfully across the network from the source sensor node to the base station. This metric evaluates the additional overhead required by the routing scheme to maintain real-time data routing toward the base station. Average end-to-end delay includes all possible delays caused by packets buffering and queuing during route discovery latency, retransmission delays at the MAC layer, propagation and transfer times.

6.5.2 Simulations Settings and Parameters

Large-scale simulations are implemented in NS-2.33 network simulator [7] to evaluate the RLBR scheme. The simulated network is composed of 100 static sensor nodes deployed randomly in a square sensor field of maximum size of 100x100 meters square with a single stationary base station deployed at the corner. To simulate different workloads, a varied number of sensor nodes of 30, 50 and 70 are randomly selected as *sources* in different parts of the deployment area. While the energy dissipation for computations in the sensor node is ignored in the simulations, the energy consumed for communications is estimated by implementing the network energy model from Section 6.4. The network parameters including radio transmission range, transmission rate, sensitivity and output transmission power are implemented in a NS-2 radio model and configured according to the radio parameters specified in the data sheets of Chipcon 2.4GHz CC2420 RF transceiver [78] and TelosB data sheet [86]. IEEE 802.15.4 [87] is used as the underlying MAC and physical layer protocol with bandwidth of 250Kbps. The wireless medium is simulated by using the multipath shadowing propagation model stated in [146] as it characterises realistic propagation behaviour. The pathloss exponent is set to the value of 2, and the shadowing deviation is set to the value of 4 to represent the typical characteristics of an outdoor environment.

A subset of the sensor nodes are selected as *sources*. To increase the network depth of the routing tree, the base station is deployed on the periphery of the sensor field and the leaf source nodes are located distant from the base station for a *many-to-one* topology. It is assumed that the experiments follow an event-driven model and the source nodes detect similar stimulus. Therefore, their sensed event data can be aggregated along the routing path as early as possible. Standard data aggregation functions are used, i.e., max, min, and average. Each source node generates data packets of the same size which does not exceed 128 bytes. Source nodes send continuous bit rate (CBR) traffic to the base station through the discovered routes within the simulated network. Data packets are injected from all sources into the network at a constant rate of 1 packet per second. However, to reflect different traffic workloads, various numbers of sources are set. This will represent a packet delivery performance of the network as a whole rather than evaluating the routing performance of a subset of relay sensor nodes connecting a single source sensor node to the base station. The rates at which the data packets are transferred to the base station and the amount of energy required to transmit the data packets relayed towards the base station are monitored and measured. Sensor nodes estimate link quality by observing packet success and loss events.

At the beginning of each simulation run, each sensor node is assigned with the same initial energy level of 10 Joles. The base station has its persistent energy supply as it is usually the case in real WSN applications. The base station's *id* is known at compile-time. Sensor nodes use up their available energy during the simulation period and the remaining energy level is measured after the simulation runs for a period of time. Once a sensor node runs out of energy, it is considered inoperative and can no longer transmit or receive any data or control packets. To minimize the variations on routing performance from the MAC layer, no energy conservation strategy is introduced in the underlying

MAC layer. By this, the most conservative measurements are given on the energy conservation routing strategy for the RLBR scheme over the benchmark schemes.

6.5.3 Simulation Results

❖ *Operational Network Lifetime*

Using simulations of a larger network featuring 100 nodes with a range of source nodes between 30 and 70 in number, the RLBR scheme balances the energy consumption and keeps updating energy efficient routes. Overall, Figure 6.5 shows that the network lifetime declines as the number of deployed source nodes increases, due to the high volume of control and data packets that are retransmitted throughout the sensor network.

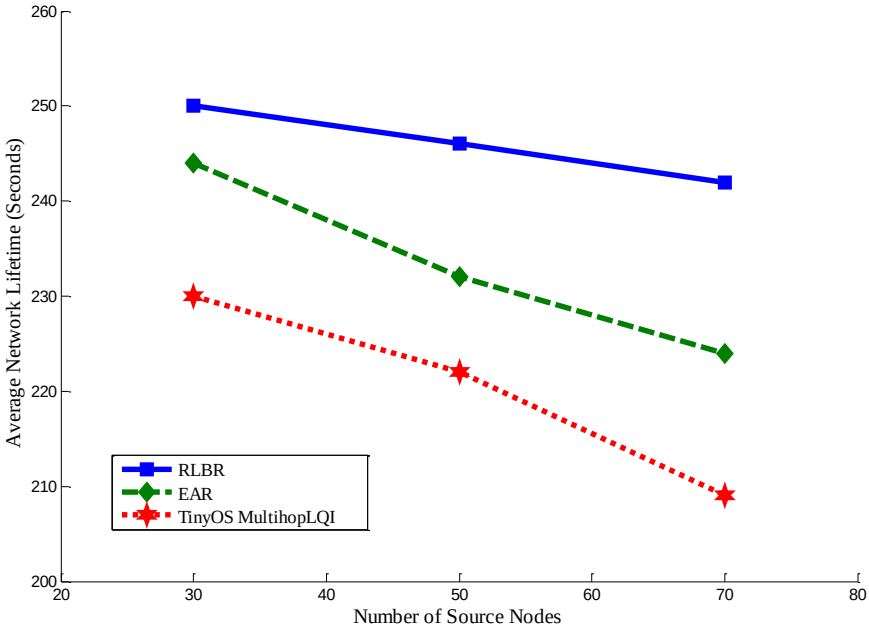


Figure 6.5 The Average Network Lifetime (Seconds)

From network lifetime point of view, the RLBR results in a slower and a more graceful linear degradation of the network lifetime as a result of employing a transmission probability mechanism to distribute the traffic load over available routes. The EAR protocol also performs well compared to MultihopLQI protocol. However, the EAR may not always be efficient as it uses a single selected route at all times which leads to network partitioning afterwards. Although the MultihopLQI protocol has an occasional

ability to balance the traffic load based on link quality estimates, the large numbers of redundant packet copies that are retransmitted between different sensor nodes depletes the available energy more rapidly. From node lifetime point of view, Figure 6.6 shows the number of exhausted nodes as the simulation time passes. Since the lifetime of individual nodes has been maximised using the RLBR scheme, it can be observed that the RLBR scheme performs better than other routing schemes and keeps more nodes alive.

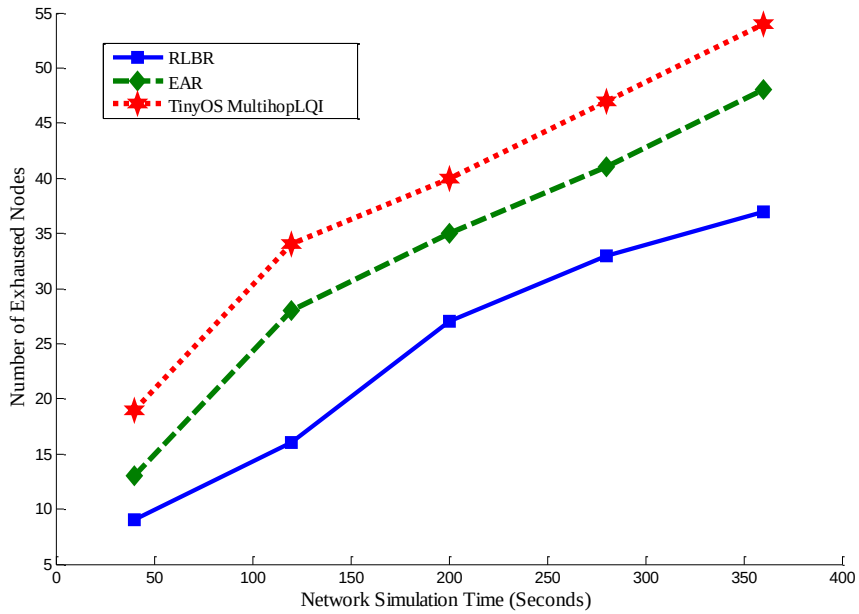


Figure 6.6 The Number of Exhausted Nodes during Simulation Time

Other routing schemes occasionally balance the traffic load using different routes as a direct effect of their route cost metrics. For example, MultihopLQI protocol balances the traffic load using LQI values in the route selection, resulting in occasional balanced energy depletion by fewer relay nodes. The traffic workload through other nodes can be higher which significantly increases their residual energy dissipation in rerouting upstream data packets. Therefore, more heavily-loaded relaying sensor nodes along the routing path will drain out their batteries in a shorter period of time and the total number of inoperative nodes is quite high. Sensor nodes with lighter traffic load can survive longer. However, the number of lightly loaded nodes is marginal as the traffic increases

by adding more source nodes. Since the RLBR scheme increases the cost of using the outgoing links from the nodes that have their residual energy decreased to the threshold, it conveys data packets through nodes with higher residual energy levels. Thus, the least number of nodes are exhausted and the number of inoperative nodes rises gradually during the same period of time.

❖ *Average Dissipated Energy*

Figure 6.7 illustrates the relationship between the average dissipated energy during network operation and the number of source nodes at which data traffic is generated. As an overall trend it can be seen that the averaged dissipated energy by the nodes in all routing schemes has an increasing trend as the number of source nodes becomes higher. However, the RLBR scheme can cause lower energy consumption than other schemes. Compared with the other protocols, the RLBR scheme performs favourably with energy consumption increasing linearly with the number of source nodes. This demonstrates that the RLBR scheme is capable of supporting larger WSN than the other protocols.

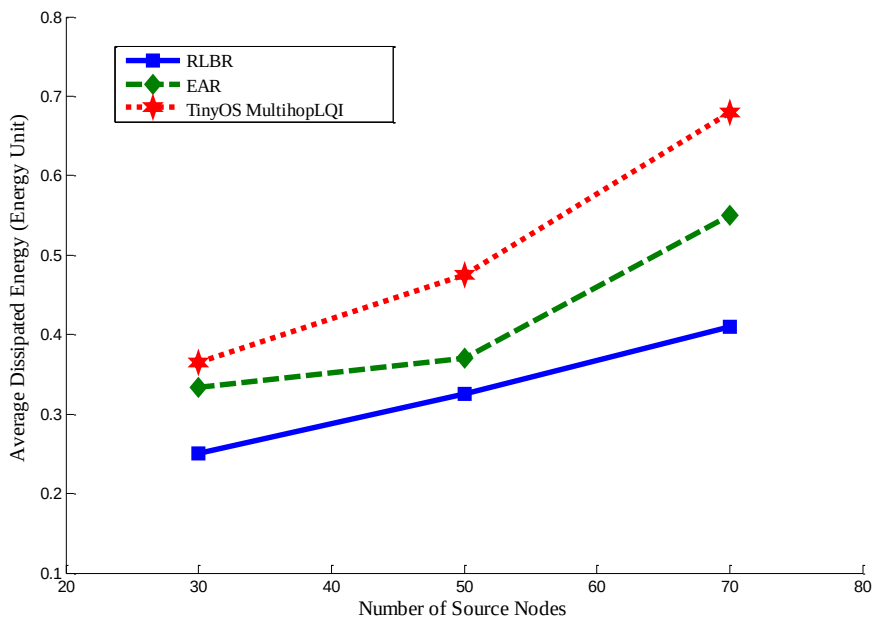


Figure 6.7 The Average Dissipated Energy

In contrast, the other protocols dissipate more energy for the same number of nodes and the energy dissipation increases considerably as the number of generating nodes grows. These protocols occasionally distribute the workload as a result of their metrics. For example, MultihopLQI protocol uses its default link quality information to occasionally distribute the load over optimal routes which is an implicit outcome of the LQI metric. Although this mechanism can avoid network partitioning occasionally at an early stage, it may not always be as efficient as the RLBR load balancing scheme.

Figure 6.8 shows the change in the node's average residual energy level after a period of data transmission. It is obvious that the change in the number of source nodes has an impact on the individual node's residual energy level. As an overall trend, the average remaining energy level decreases with higher number of source nodes. The RLBR scheme can reduce the redundant data copies in the network using data aggregation which results in a lower traffic load handled by each individual relaying node. This makes the average remaining energy level to degrade much slower than the other protocols. As a result, the load balancing mechanism of the RLBR keeps a balanced network workload towards the base station and maintains balanced energy dissipation.

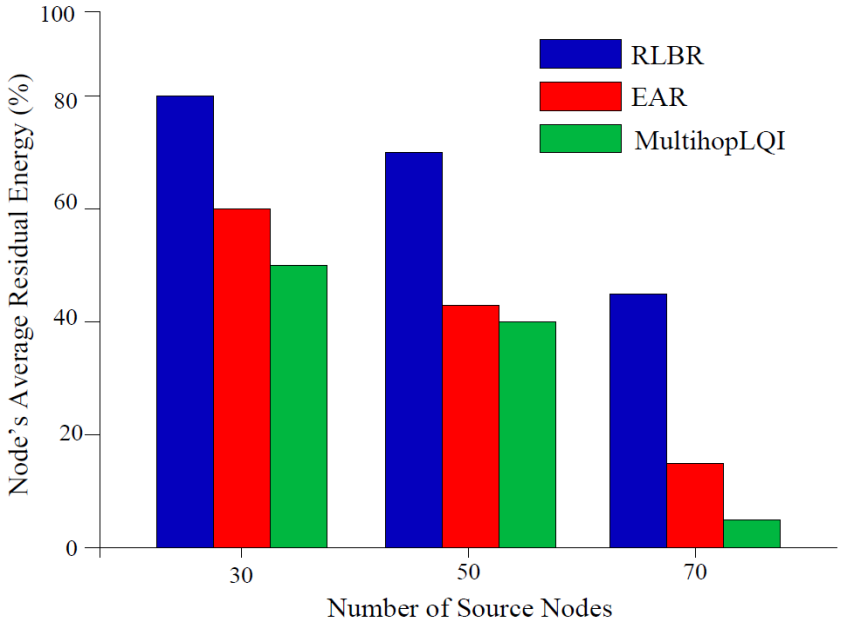


Figure 6.8 Sensor Node's Average Residual Energy

❖ *Packet delivery ratio*

Despite the random selection of source sensor nodes, the RLBR scheme outperforms the other protocols and delivers a higher percentage of data packets with different load scenarios as shown in Figure 6.9. This is due to the implementation of data packet aggregation. The consistent packet delivery ratios for the RLBR scheme in the random network show its scalability and reliability. However, simulations results do not accurately reflect the experimental observations that some packets are skipping over the intended sensor node [120,121]. As a result of the simulated wireless links that are mainly based on a connectivity matrix and cannot precisely consider the signal attenuation in the real world channel [146], the simulation results show that the packet delivery ratios are much higher than the experimental results as experienced in the previous empirical work of the two preceding chapters.

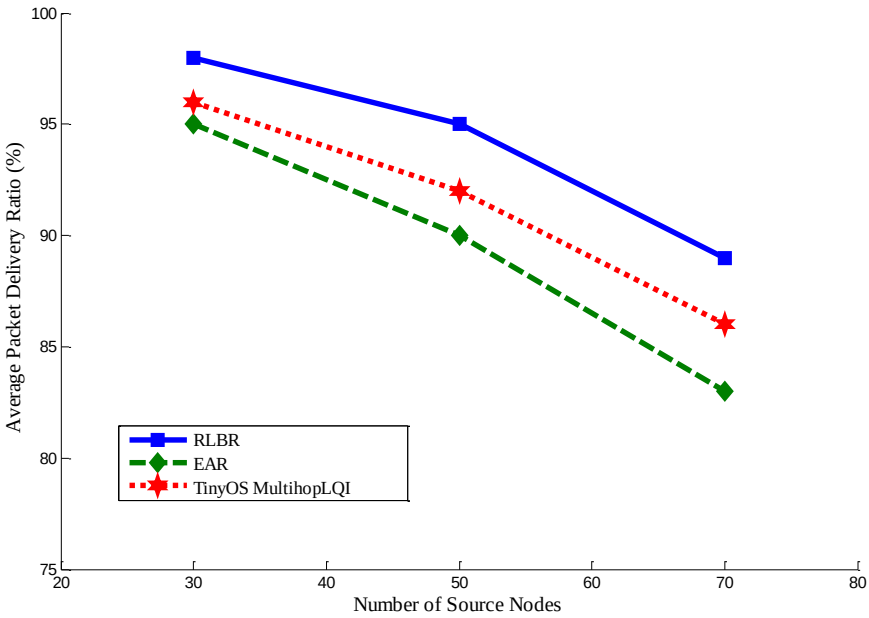


Figure 6.9 The Packet Delivery Ratio

❖ *Average end-to-end data transfer delay*

As shown in Figure 6.10, the average end-to-end delays for all protocols increase as the number of sources grows. The RLBR scheme causes a shorter delay as a result of

reducing the probability of packet retransmissions. In addition, it benefits from locally obtained routing information and the lower routing overhead caused by packet rerouting. On the other side, EAR and MultihopLQI cause longer delays as EAR requires geographical information for its routing decisions and MultihopLQI requires a longer time for route discovery as it keeps one parent at a time.

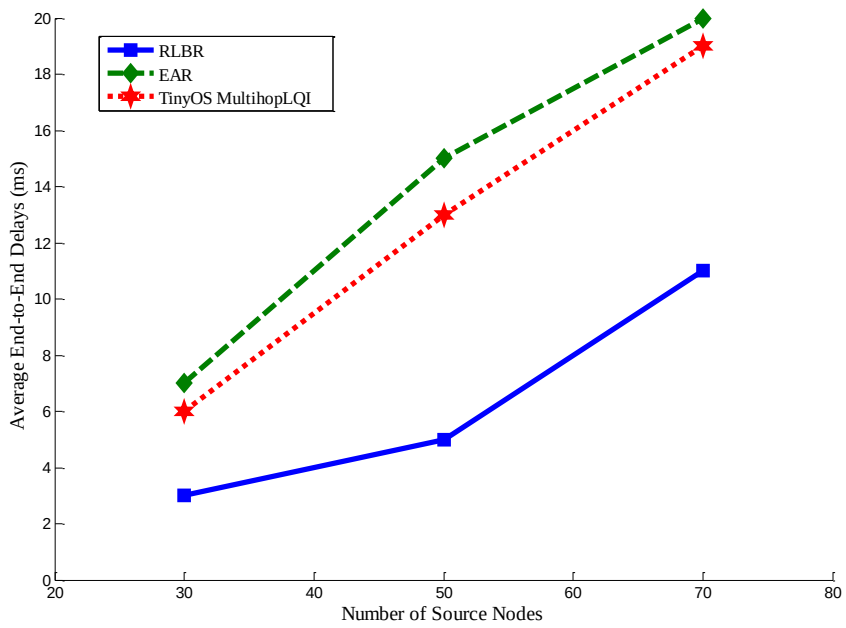


Figure 6.10 The End-to-End Packet Transfer Delay

6.6 Conclusion

Through large-scale simulations in NS-2.33, the feasibility of the load balancing scheme is revealed by demonstrating the improved network lifetime in various traffic scenarios with different numbers of source nodes. The simulations show a balanced energy usage and a significant lifetime gain per relaying node. The RLBR scheme reduces the impact of traffic congestion on a node lifetime without a deterioration of the end-to-end reliability performance.

Chapter 7
Per Link Transmission Power Control

7.1 Background and Motivation

Communication is the major energy consumer compared to computation and sensing operations performed by a battery-powered wireless sensor node [83]. The reduction of communication power consumption in WSNs can be achieved using adaptive transmission power adjustment schemes [3,61,115]. Although transmission reliability can be enhanced further by transmitting route discovery messages and data packets at unnecessarily high transmission power outputs, this may introduce excessive interference, collisions and wastes energy. The lifetimes of sensor nodes equipped with adaptive power control radio transceivers can be maximised if the intended recipient can successfully receive the transmission at a lower power. However, the surrounding environments together with energy restriction of the wireless sensor nodes make a reliable WSN routing a challenging task. Given a limited energy supply, routing reliability and energy efficiency are the most important issues in WSNs [120,157].

In this chapter, a new per link variable transmission power control scheme is proposed. It is a topology-control scheme that aims to dynamically change the transmission power output to the lowest possible transmission power level for the reduction of power consumption while maintaining reliable network connectivity and coverage. The design of the scheme is guided by the empirical observations obtained from the testbed experiments conducted in the preceding chapters, i.e., Chapters 4 and 5, and appears in the existing work [120,157]. The proposed scheme uses a neighbourhood-based power control algorithm which increases the transmission power to the optimal level to reach each neighbour. It assumes that sensor nodes can use different transmission power levels for different neighbours based on link reliability status. It creates a predictive

transmission power model at each sensor node for its neighbours and uses a feedback-based closed loop algorithm between the transmitter node and its neighbours to characterize the interrelation between the transmission power level and measured link quality. This will allow for adapting control of the transmission power rather than setting the transmission power output level during the network run time. The transmission power level is derived from the realistic cross-layer characteristics by considering network connectivity in terms of a link quality (e.g., link quality indicator (LQI) [87]) constraint given by the highest acceptable threshold of packet delivery performance (e.g., packet reception ratio (PRR)) between a directional pair of nodes. Avoiding transmitting at unnecessarily high transmission power outputs will significantly reduce the power consumed for transmissions. The work in this chapter is subject to the specific routing protocols that use similar routing reliability metrics with the RLBR scheme proposed in Chapter 3; however, the approach can be also extended to any routing protocol that proactively maintains a routing table.

7.2 Design Issues

In large-scale WSNs, sensor nodes may be densely located in close proximity to one another. This unique characteristic of WSNs may allow for a large portion of the nodes to reduce their radio output power and still communicate effectively with neighbouring peers while avoiding unnecessary further overhearing or opportunistic reception [3]. In the majority of WSN applications that involve large-scale deployments, the sensor nodes are scattered randomly and may be over-deployed with redundancy. As a result, adaptive transmission power schemes can reduce unnecessary energy dissipation for communications while assuring the required level of network connectivity [158,159,160,161]. Power transmission adjustment schemes are quite a complex and challenging cross-layer issue as they can be integrated with the MAC or routing

protocols. Increasing radio transmission power has a number of interrelated performance consequences. While some of these consequences are beneficial, others are destructive from a reliability and energy efficiency point of view [150,158,159,160,161]. For instance, increasing radio transmission power can expand the communication range of the transmitter which may improve connectivity and link quality in the form of availability of end-to-end paths and the absence of other interfering traffic [162,163]. On the other side, increasing radio transmission power can cause an increase in the energy waste throughout the network due to increased overhearing and also can introduce additional interference and collisions due to the increased number of communicating neighbouring motes within the increased transmission range [3,150,164,165,166].

7.3 Related Work

In the literature, there are several approaches of topology control using transmission power control for WSNs [3,150,158,159,160,161,162,163,164,165,167,168,169,170, 171,172]. These approached can be classified based on the reliability metric(s) used to derive the optimal transmit power. The optimal transmit power can be derived either based on quality of Service (QoS) constraint [161], network connectivity [150,162,167, 168,171,172], link quality and signal strength [3,164,165], or successful packet reception [163,169]. Based on QoS, the maximum tolerable bit error rate (BER) is estimated at the end of a multihop route with an average number of hops [161]. Based on network connectivity, each node needs to determine its transmission power to the recipient nodes which acknowledged the beacon message it has sent [150,162,167,168,171,172]. Authors in [168] introduce a clustering-based (known as common-power “COMPOW”) transmission power control algorithm for power control on ad hoc networks. COMPOW finds the smallest common power at which the network is connected while reducing the energy consumption. Authors in [171] propose a directional information-based (also

known as cone-based) power control algorithm which increases the transmission power to the maximum level to reach a neighbour node in every direction. Based on link quality or signal strength, each node maintains a list of neighbours and ranks them in order of their link quality values (i.e., RSSI and/or LQI) and then adjusts the radio transmission power accordingly for set of neighbours [3,164] or for each neighbour node individually [165,170]. Based on successful packet reception, each node adjusts its radio transmission power to its neighbours whose PRR values are higher than the threshold while it filters out the sensor nodes that have PRR values lower than the threshold [163,168,169]. Although the aforementioned schemes claim possible improvements in energy savings and throughputs, they either only consider a conventional graph-theoretic approach [161], ignore the realistic restrictions of various traffic workloads [157], or do not take into account the minimisation of interference and overhearing [166,170].

7.4 Design and Implementation

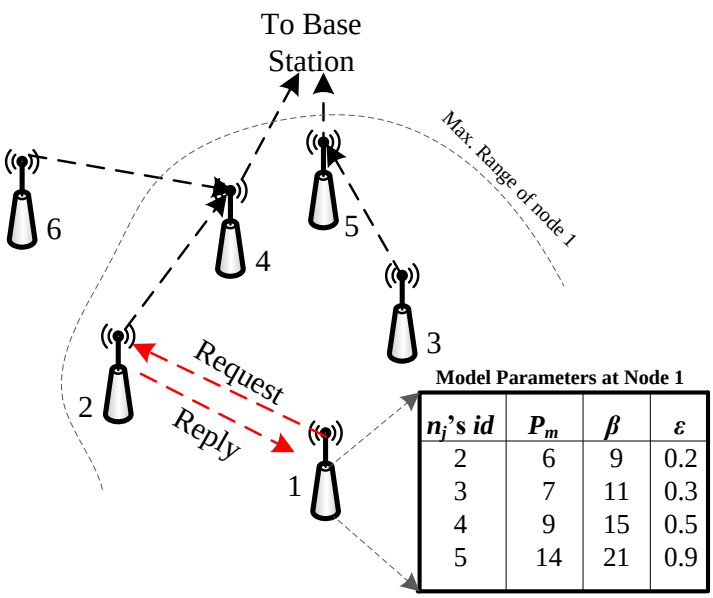
7.4.1 Energy Model

Sensor nodes in general perform different tasks and processes. However, the process of receiving and transmitting data is the major energy consumer [61]. In the proposed model, it is assumed that the energy consumption of sensor motes is merely due to data transmission and reception while neglecting the amount of energy consumed by other processes. A simple radio model is used for the RF transceiver energy consumption as discussed earlier in Section 6.4.3. The energy $E_{TX}(n,d)$ required for a sensor node to transmit a packet of *n* bits over distance *d* is expressed in equation 6.2 "$E_{TX}(n,d) = n(e_{TC} + e_{Amp} \times d^{\alpha})$". Where $e_{Amp} d^{\alpha}$ accounts for the radiated energy required to transmit a bit of data over a distance of *d* between the sender and destined recipient, this energy is consumed for amplifying the signal to achieve adequate signal-to-noise ratio (SNR) over a distance *d*; e_{TC} is the energy dissipated per bit by the transmitter

circuitry; the parameter α is the power index for the channel path loss of the antenna. In general, α varies from 2 for free space propagation model to 4 for multipath fading or shadowing channel models [146].

7.4.2 Per link Transmission Power Model

Figure 7.1 presents an overview of the transmission power predictive model where the full transmission power at each sensor node is divided up into a number of levels. Each sensor node maintains a neighbourhood routing table that includes N neighbour sensor nodes with their *id*s, minimum transmission power level P_m and feedback closed loop control configuration parameters (i.e., β and ε) between each pair of sensor nodes (explained in the next paragraphs). P_m is the minimum transmission power level that ensures a reliable link quality between two communicated of sensor nodes. This per link transmission power model makes each pair of neighbouring nodes use the same transmission power level when communicating to each other. This is needed to keep track of the optimal transmission power level by exchanging configuration parameters.



n_j's *id*	P_m	β	ε
2	6	9	0.2
3	7	11	0.3
4	9	15	0.5
5	14	21	0.9

Figure 7.1 Per Link Transmission Power Predictive Model

These parameters include the link quality samples and other elements (explained in the next paragraphs) used by the closed loop *controller* algorithm (explained in Section 7.4.3) to gradually adjust the transmission power to the minimum predicted level. This *controller* algorithm maps the relationship between the transmission power levels and the corresponding link qualities. However, this relationship appears not to be completely correlated [120,157]. Since the packet reception ratio (PRR) estimations have a strong correlation with the hardware-based link quality values (i.e., LQI), PRR is used as a threshold to make the decision on the selection of the desired link quality lq_d value at the receiver sensor node n_j. The empirical results stated in [59] indicate that when the RSSI is above the sensitivity threshold (e.g., about -87dBm), PRR is at least 85%. Around this sensitivity threshold, however, the PRR is not correlated possibly due to variations in local phenomena such as noise. However, the average LQI computed over long period of time has a better correlation with PRR estimations [120,157].

To characterise the correlation between the transmission power levels and the corresponding link qualities, a generalised linear regression model [173] is used over measured samples of link quality values at different transmission power levels. This model generalizes a linear regression of the variance of lq_j^i samples to be a function of the predicted transmission power level. lq_j^i refers to the wireless link quality of the radio channel between a pair of sensor nodes. It is measured at a neighbouring sensor node n_j (j is used as n_j's id) with the ith transmission power level P_i. To acquire the link quality lq_j^i samples, each sensor node transmits a number of routes beacons at a range of k different transmission power levels P_i. k can be increased to improve the accuracy of the scheme according to the application. Using request/reply route searching beacons, each recipient sensor node n_j of these beacons caches the measured lq_j^i values of each received beacon and return those values in the acknowledgment message. This will reflect on the

bidirectional dynamic changes of link quality values over time to accurately predict the required P_m for an acceptable packet delivery performance (e.g., 85% < PRR < 95%). Basically, the model is represented in equation 7.1a. Where the elements of V_P are the dependent *regressands* (transmission power levels); the elements of M_{lq} are the observed *regressors* (*lq* samples); the elements of β are the *regression coefficients* used for statistical estimation and inference; and the elements of ε are the *error coefficients* (also known as *disturbance* or *noise* term) which influence the accuracy of the system. As expressed in equation 7.1b and 7.1c, the model uses a vector V_P that contains the values of k transmission power levels ($V_P = \{P_1, P_2, ..., P_k\}$) and a matrix M_{lq} that contains N vectors m_j of the corresponding lq_j^i samples ($M_{lq} = \{m_1, m_2, \ldots, m_N\}$). Each m_j vector is assigned for each neighbouring senor node n_j that acknowledged the receipt of the k beacons transmitted by the transmitter sensor node ($m_j = \{ lq_j^1, lq_j^2, \ldots, lq_j^k \}$).

$$V_P = M_{lq}^T \times \beta + \varepsilon \tag{7.1a}$$

$$\begin{bmatrix} P_1 \\ P_2 \\ \vdots \\ P_k \end{bmatrix} = \begin{bmatrix} m_1 \\ m_2 \\ \vdots \\ m_N \end{bmatrix}^T \times \begin{bmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_N \end{bmatrix} + \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_k \end{bmatrix} \tag{7.1b}$$

$$\begin{bmatrix} P_1 \\ P_2 \\ \vdots \\ P_k \end{bmatrix} = \begin{bmatrix} lq_1^1 & lq_2^1 & \ldots & lq_N^1 \\ lq_1^2 & lq_2^2 & \ldots & lq_N^2 \\ \vdots & \vdots & \ldots & \vdots \\ lq_1^k & lq_2^k & \ldots & lq_N^k \end{bmatrix} \times \begin{bmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_N \end{bmatrix} + \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_k \end{bmatrix} \tag{7.1c}$$

The values of β and ε can be estimated by adopting the ordinary least square regression estimator in equation 7.2 based on the samples in the vectors. This estimator the simplest the most used estimator to analyse experimentally observed data. It also requires a low computational overhead and can be easily implemented in sensor node system with insignificant energy consumption. This approximation approach minimizes the sum of squared residuals (R^2). The residual of a sample is the deviation between the

the attempted transmission power level P_i at which the lq_j^i is sampled and the predicted minimum transmission power level P_m.

$$\sum\left(P_i - P_m\right)^2 = R^2 \tag{7.2}$$

Accordingly, this leads to a closed form expression for the estimated value of the configuration parameters in β and ε. The values of β and ε elements can be calculated in equations 7.3 and 7.4 respectively. Where i is the number of attempts of different power levels and j is the neighbouring senor node's *id*. Since the configuration parameters of β and ε are functions of time, this predictive model is frequently updated while the environment conditions change over time by using the most recent link quality samples to dynamically adjust the transmission power.

$$\beta = \begin{bmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_N \end{bmatrix} = \frac{1}{k\sum_{i=1}^{k}\left(P_i\right)^2 - \left(\sum_{i=1}^{k} P_i\right)^2} \times \begin{bmatrix} \sum_{i=1}^{k} lq_1^i \sum_{i=1}^{k}\left(P_i\right)^2 - \sum_{i=1}^{k} p_i \sum_{i=1}^{k}\left(lq_1^i \times P_i\right) \\ \sum_{i=1}^{k} lq_2^i \sum_{i=1}^{k}\left(P_i\right)^2 - \sum_{i=1}^{k} p_i \sum_{i=1}^{k}\left(lq_2^i \times P_i\right) \\ \vdots \\ \sum_{i=1}^{k} lq_N^i \sum_{i=1}^{k}\left(P_i\right)^2 - \sum_{i=1}^{k} p_i \sum_{i=1}^{k}\left(lq_N^i \times P_i\right) \end{bmatrix} \tag{7.3}$$

$$\varepsilon = \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_k \end{bmatrix} = \frac{1}{k\sum_{i=1}^{k}\left(P_i\right)^2 - \left(\sum_{i=1}^{k} P_i\right)^2} \times \begin{bmatrix} N\sum_{j=1}^{N}\left(lq_j^1 \times P_1\right) - P_1 \sum_{j=1}^{N} lq_j^1 \\ N\sum_{j=1}^{N}\left(lq_j^2 \times P_1\right) - P_1 \sum_{j=1}^{N} lq_j^2 \\ \vdots \\ N\sum_{j=1}^{N}\left(lq_j^k \times P_1\right) - P_1 \sum_{j=1}^{N} lq_j^k \end{bmatrix} \tag{7.4}$$

7.4.3 Scheme Implementation

The *controller* algorithm uses the *reactive* component of the RLBR routing scheme to build the routing paths by transmitting request/reply route searching beacons. It restricts these beacons to the neighbouring sensor nodes that are iteratively added to the route being discovered during RLBR's searching phase and updated in the routing table. The broadcast nature of wireless communication is used to simultaneously measure link

qualities by all neighbours and to detect possible packets collisions. If a sensor node overhears a message from a neighbouring node on the discovered route, it means that there is a potential for collisions between their packets. By adjusting the transmit power of each sensor node on the route, the potential collision area and energy consumption of each sensor node can be reduced. When a neighbouring sensor node receives a request beacon, it will rebroadcast a reply beacon with a measured link quality lq_j^i which will be used by a power function to calculate the minimum transmission power level P_m required for the desired link quality lq_d. This is stated in Algorithm 7.1 which is implemented here from the prospective of the sender.

Algorithm 7.1 Adjusting Transmission Power Level (P_i)

Initialisation: *Predictive Model*

Input: lq_j^i // *link quality measured at node* n_j

Input: lq_d // *Desired link quality*

Output: P_m // *Minimum Transmission Power Level*

For each sensor node that has a packet to send

  While (Reply Packet Receive with $(lq_j^i - lq_d)$) // *Feedback* $(lq_j^i - lq_d)$ *is reported*

    If $(lq_j^i < lq_d)$ Then // *calculate the minimum transmission-*
    // *power level*

      $$P_i = P_i + \left(\frac{lq_j^i - \varepsilon_i}{\beta_j} \right)$$ // P_i *is increased*

      If ($RSSI_m$ *with PRR > 85%*) Then // *Detection of adequate signal strength*

      $$P_i = P_i - \left(\frac{lq_j^i - \varepsilon_i}{\beta_j} \right)$$ // P_i *is decreased*

      Endif

    Endif

  Endwhile

Endloop

The algorithm has two phases, the initialisation phase and the power adjustment phase. In the initialization phase, each sensor node broadcasts request beacons at different transmission power levels. Since all neighbouring sensor nodes can simultaneously receive request beacons and measure its link qualities lq_j^i (i.e., RSSI/LQI and PRR), each of these nodes selects the minimum transmission power level (P_m) based on the predictive model stated earlier in Section 7.4.2. In the power adjustment phase, the

closed loop *controller* algorithm is run to monitor the dynamic changes of lq_j^i and adjust the transmission power accordingly. Each sensor node runs the predictive model and attempts a numbers of transmissions with different transmission power levels for its neighbours that are cached in the neighbouring table.

When *the sender* node has a packet to transmit to its neighbour n_j, it adjusts its transmission power output to the level P_i currently indicated by the predictive module in its neighbouring table, and then transmits its request beacon packet. When the packet is received at node n_j, the link quality is measured (e.g., RSSI/LQI and PRR) and a reply/feedback packet is sent back to *the sender* node with the same transmission power level P_i. This reply packet contains the closed-loop feedback information which represents the difference between the desired link quality lq_d and measured link quality lq_j^i (i.e., $(lq_j^i - lq_d)$) and the measured signal strength ($RSSI_m$). When *the sender* node receives the reply packet, its predictive module uses the link quality difference as a control input and accordingly estimates the new transmission power level for its neighbour node n_j (the current transmission power level P_i is adjusted to the minimum transmission power level P_m). This new minimum level is immediately updated into the local neighbouring table of *the sender* node. For example, if $(lq_j^i < lq_d)$ in terms of successful reception and LQI (e.g., LQI of PRR threshold < 85%) then P_i is increased, otherwise if the measured lq_j^i in terms of signal strength is unnecessarily high (e.g., $RSSI_m$ of PRR threshold > 95%) then P_i is decreased. However, the empirical results stated in Chapters 4 [120] and 5 [157] indicate that $RSSI_m$ patterns are time-varying and not correlated with the PRR threshold values in different environments (e.g., $RSSI_m$ of 95% PRR is approximately -89dBm in indoor environment and about -91dBm in outdoor environment). Hence, avoiding transmitting with unnecessarily high $RSSI_m$ will

significantly reduce the power consumed for transmissions. Where $RSSI_m$ is the signal strength measured at node n_j. lq_d and $RSSI_m$ are extracted from the link quality lq_j^i information (i.e., $(lq_j^i - lq_d)$) replied with the closed-loop feedback message. Where j is the neighbouring sensor node n_j's *id*; and i is the number of attempted transmission; β_j and ε_i are calculated at *the sender* node using equations 7.3 and 7.4 respectively.

7.5 Performance Evaluation

7.5.1 Simulation Settings and Parameters

The proposed scheme was evaluated using NS-2 network simulator [7] by simulating a network composed of 100 fixed sensor nodes deployed uniformly randomly in a square area of 1000m x 1000m with a single stationary base station deployed at the corner. IEEE 802.15.4 [87] is used as the underlying MAC and physical layer protocol with bandwidth of 250Kbps. The multipath shadowing propagation model [146] is used to characterise the wireless medium. The pathloss exponent (α) is set to the value of 2. The network parameters including radio transmission range, transmission rate, sensitivity and output transmission power are set according to the parameters specified in the data sheets of Chipcon 2.4GHz CC2420 RF transceiver [78] and TelosB data sheet [86]. Each source node generates data packets of the same size which does not exceed 128 bytes. Source nodes generate continuous bit rate (CBR) of traffic to the base station through the discovered routes within the simulated network. Data packets are injected from all sources into the network at a constant rate of 1 packet per second. Since the end-to-end PRR estimations are calculated statistically at the receiver, the average PRR estimations were recorded over long urn simulations of 7 hours. The output transmission power is immediately adjusted to the proper transmission radius after a reply message is received with a low link quality or when a broken link or a failed node is detected by the RLBR

scheme at the upper layer. The performance of the proposed scheme is evaluated by comparing it with other existing transmission power control algorithms, specifically, the common-power (COMPOW) [168] and the directional-based (cone-based) [171].

7.5.2 Results

❖ *Relative Transmission Dissipated Energy*

Figure 7.2 shows the relative dissipated energy for transmissions during simulation time. The energy consumed at the maximum transmission power P_{max} (100%) is taken as a reference. It demonstrates that the proposed scheme consumes the least transmission energy compared to the benchmark power control schemes. This is because a large number of sensor nodes reduce their transmission power levels to the optimal level in order to save energy in the proposed scheme. However, the proposed scheme has additional energy consumption as a result of the request/reply packet transmissions overhead and the non-uniformity of per link transmission power levels.

The proposed scheme accurately adjusts the per link transmission power output to the appropriate level by gathering more sampled link quality data from each neighbour node. The other schemes have higher energy consumption as the transmission power level of each node does not change accurately due to link quality variations compared to the proposed scheme. With long hop neighbours, the other schemes tend to use to the maximum transmission power level even with high link quality. However, the overall energy consumption of these schemes varies as the network throughput varies.

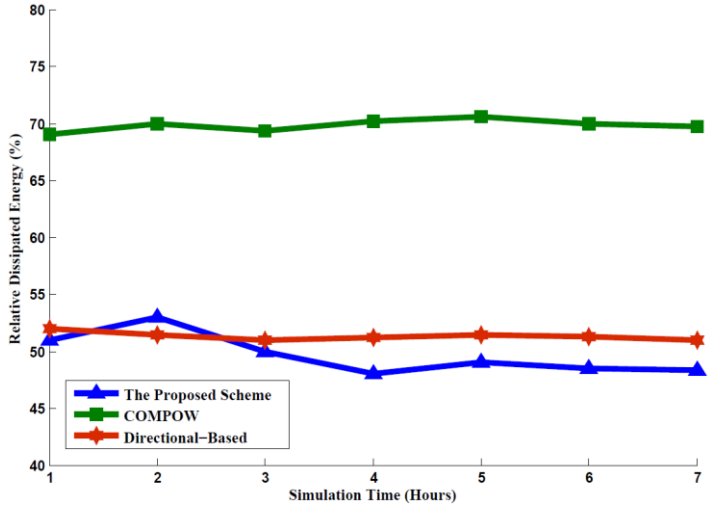


Figure 7.2 The Relative Dissipated Energy for Transmissions

❖ *Network lifetime*

To understand how the network topology evolves, the network lifetime is measured in terms of the number of nodes that still operate over simulation time. The transmission power is immediately adjusted to the proper transmission radius if a parent node is inoperative. Packets are sent with the properly reconfigured transmission power using the predictive power model. From Figure 7.3, it can be seen that the proposed scheme as well as the directional-based scheme performs significantly better than the COMPOW scheme. When both the proposed scheme and directional-based still have around 90 operative nodes, the COMPOW has almost 60 nodes operative.

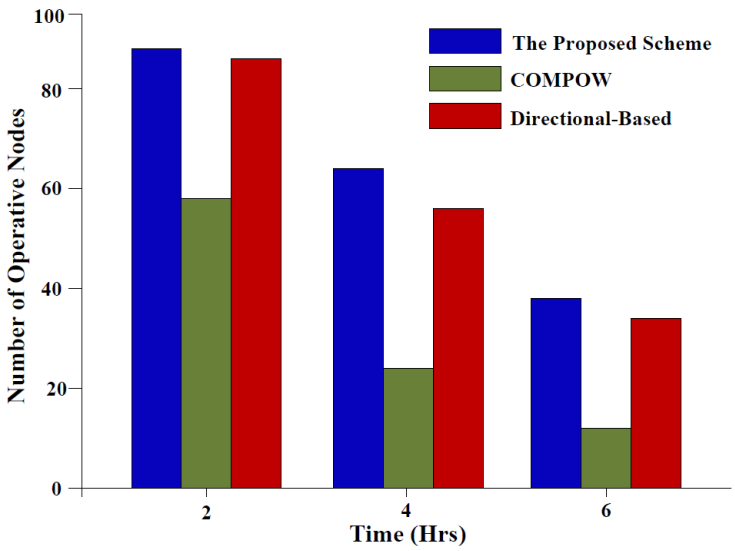


Figure 7.3 The Network Lifetime in Terms of Operative Nodes

❖ *Packet Delivery Performance*

Figure 7.4 shows the average end-to-end PRR during simulation time. The proposed scheme achieves about 98.80% PRR by taking into account the per neighbour link quality. However, there are insignificant packet losses due to unstable link qualities using the random shadowing propagation model [146]. Similar to the proposed scheme, the COMPOW scheme also achieves high PRR. The link reliability maintained using a single transmission power level for all neighbours makes the Directional-Based scheme [171] vulnerable to link changes where the PRR drops considerably.

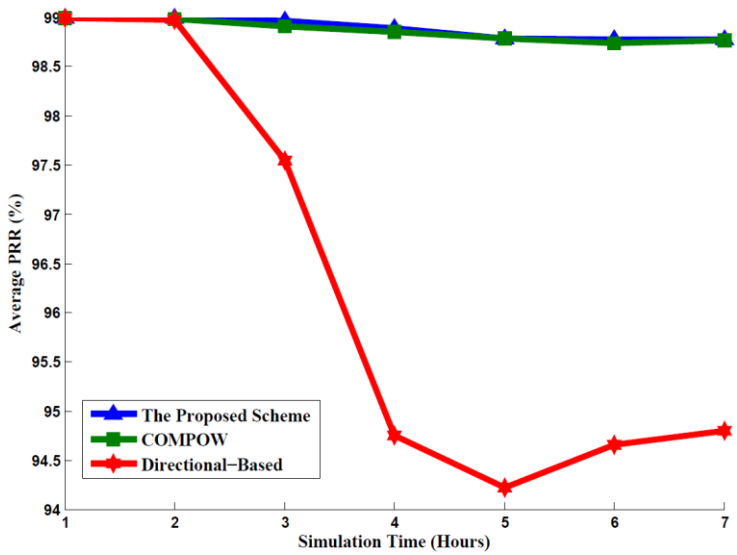


Figure 7.4 The Packet Delivery Performance over Simulation Time

7.6 Conclusion

Studying network lifetime maximisation using transmission power control based on link reliability was the scope of this chapter. A new variable transmission power control scheme was proposed which is a joint power-control and routing scheme by tuning the transmission power output of the sensor mote to find the optimal mote transmission power that minimises route energy dissipation, while preserving network connectivity and coverage. Simulations were conducted to evaluate the performance of the proposed scheme over long runs. The results show that the proposed scheme achieves significant

energy savings and better end-to-end packet delivery reliability compared to the other transmission power control schemes. Finally, this work provides radio transceivers designers with the ability to adaptively control the transmission power rather than setting the transmission power output level during the network run time.

Chapter 8
Conclusion and Future Work

Since Wireless Sensor Networks (WSNs) are the most resource-constrained type of multihop ad hoc networks and fundamentally different from other well planned wireless networks, their protocols should be simple yet reliable with low computation and communication overhead. This can be accomplished using cross-layer protocol architectures that exploit wirelessly obtained routing information to achieve orders of magnitude improvement in reliability and energy efficiency and improvements in network lifetime. The work described in this thesis has demonstrated the advantages of a reliable and load balancing routing scheme by designing and evaluating the proposed solution on different platforms and under different scenarios.

8.1 Summary of Contributions

This research has made three major contributions in the areas of routing reliability and resiliency (via adaptive beaconing and multipath redundancy), balanced network energy usage (with efficiently contained load balancing) and low node energy consumption (with per link transmission power adaptability). These have been achieved without adding protocol complexity or resource consumption in support of the primary objective of network lifetime maximisation.

To achieve the first two contributions, a reliable energy-efficient collection tree routing scheme, RLBR, was proposed based on a per-hop load balancing routing scheme. It is the first to explore the integration between energy-aware and load-aware routing metrics for maximizing network lifetime, and proves that better results can be obtained when the metrics of the routing algorithms are not only restricted to link reliability and energy status of sensor nodes, but also consider load-related metrics of delay-sensitive data

packets that are pending and being aggregated through relay nodes. The RLBR scheme was implemented on real sensor motes as well as using large-scale simulations. Its performance was tested on an indoor testbed and outdoor sensor field deployments. The results show that the RLBR outperforms even sophisticated link estimation based sensor network routing protocols. It leverages recent advancements in the standard network layer components provided by the TinyOS2.x implementation. The RLBR consumes less energy while reducing topology repair latency and supports various aggregation weights by redistributing packet relaying loads. It transmits a smaller number of route messages. The decrease in route message transmissions of the RLBR scheme is a result of using adaptive beaconing. This results in lower beaconing rates and lower control cost while the network topology stabilises; thereby achieving lower energy consumption. The experiments conducted here have highlighted the substantial performance gains of the proposed solution. The RLBR performs well with a high success rate of packet delivery and moderate energy consumption.

To achieve the third contribution, a new pair-wise variable transmission power control scheme was proposed which is a topology-control scheme aims to dynamically change the per link transmission power output to the lowest possible transmission power level for the reduction of power consumption while maintaining reliable network connectivity and coverage. The performance of the scheme was evaluated over long run simulations. The results show that the proposed scheme achieves significant energy savings and better end-to-end packet delivery reliability compared to the existing transmission power control schemes.

8.2 Future Work

In this thesis, all experimental testbeds and simulated WSNs were static. Efficiently making the RLBR scheme suitable for mobile WSN is the major scope of the future

research. When the mobility is low, the proactive approach of the RLBR can be activated to reconfigure the network topology. When the mobility is high, the reactive "on-demand" approach of the RLBR can be activated to reconfigure the network topology while the adaptive beaconing mechanism keeps the control traffic reconfiguration low. A new link estimation mechanism can be proposed to effectively adapt to faster mobility of sensor nodes. The simulation model proposed in this thesis will be extended while comparing the performance of other reactive and proactive protocols with RLBR under different scenarios. In terms QoS aware routing, other metrics can be considered such as data rate and end-to-end delay metrics could be an additional metric during the route discovery and maintenance in the routing scheme.

Further research could also be carried out to address MAC layer issues by building an energy efficient MAC protocol specific for the RLBR scheme. Finally, it will be important to develop secure communication for WSNs. A level of security measures is required without draining the limited energy of sensor nodes. Without these security measures in place, the application of sensor networks will be limited.

References

1. D. Estrin, R. Govindan, J. Heidemann, and S. Kumar, “Next century challenges: scalable coordination in sensor networks,” in *Proc. of ACM/IEEE MobiCom’99*, Seattle, Washington, Aug. 15-20, 1999, pp. 211-216.
2. W. Hienzelmen, “Application-specific protocol architectures for wireless sensor networks,” *PhD Thesis*, Massachusetts Inst. of Technology, June 2000, pp. 71-79.
3. R. Ramanathan and R. Hain, “Topology control of multihop wireless networks using transmit power adjustment,” in *Proc. of IEEE INFOCOM’00*, 2000, pp. 404-413.
4. S. Singh, C. Raghavendra, and J. Stepanek, "Power-aware broadcasting in mobile ad hoc networks,” in *Proc. of the 10th IEEE International Symp. on Personal, Indoor and Mobile Radio Communications* (*PIMRC’99*), September 1999, pp. 22–31.
5. M. Singh and V. Prasanna, "System level energy tradeoffs for collaborative computation in wireless networks," in *Proc. of the IEEE IMPACCT Workshop*, May 2002, pp. 52–61.
6. T. Stathopoulos, “Exploiting Heterogeneity for Routing in Wireless Sensor Networks,” *PhD thesis*, University of California at Los Angeles, October 2006.
7. The Network Simulator - ns-2. [Online]. Available: http://www.isi.edu/nsnam/ns/index.html. [Accessed: Aug. 07, 2010].
8. The Prowler: Matlab-Based Probabilistic Wireless Sensor Network Simulator. [Online]. Available: http://www.isis.vanderbilt.edu/projects/nest/prowler/. [Accessed: Jan. 17, 2009].
9. Crossbow Technology, Inc. [Online]. Available: http://www.xbow.com. [Accessed: Aug. 21, 2011].
10. J. Hill, R. Szewczyk, A. Woo, S. Hollar, D. Culler, and K. Pister, “System architecture directions for network sensors,” in *Proc. of the 9th International*

Conference on Architectural Support for Programming Languages and Operating Systems (*ASPLOS'00*), Cambridge, MA, USA, November 2000, pp. 93–104.

11. nesC, "nesC: A Programming Language for Deeply Networked Systems," UC Berkeley WEBS Project [Online]. Available: http://nescc.sourceforge.net/. [Accessed: Sep. 05, 2009].
12. D. Gay, P. Levis, D. Culler, and E. Brewer, "nesC 1.1 Language Reference Manual," May 2003.
13. D. Gay, P. Levis, R. Behren, M. Welsh, E. Brewer, and D. Culler. "The nesC Language: A Holistic Approach to Networked Embedded Systems", in *Proc. of Programming Language Design and Implementation* (*PLDI'03*), June 2003, pp. 91–102.
14. Crossbow Mica2 wireless measurement system datasheet [Online]. Available: http://www.xbow.com/products/Product_pdf_files/Wireless_pdf/MICA2_Datasheet.pdf
15. Crossbow MICA2 AA Battery Pack Service Life Test [Online]. Available: http://www.xbow.com/Support/Support_pdf_files/MICA2_BatteryLifeTest.pdf. [Accessed: Mar. 14, 2010].
16. V. Shnayder, M. Hempstead, B. Chen, and M. Welsh, "PowerTOSSIM: Efficient Power Simulation for TinyOS Applications," [Online]. Available: http://www.eecs.harvard.edu/shnayder/ptossim/. [Accessed: Aug. 23, 2010].
17. V. Shnayder, M. Hempstead, B. Chen, G. Allen, and M. Welsh, "Simulating the Power Consumption of Large-Scale Sensor Network Applications," *in Proc. Of ACM SenSys'04*, Baltimore, Maryland, USA, November, 2004.
18. P. Levis and N. Lee, "TOSSIM: A Simulator for TinyOS Networks", Version 1.0, September 2003.
19. P. Levis, N. Lee, M. Welsh, and D. Culler, "TOSSIM: Accurate and Scalable Simulation of Entire TinyOS Applications", in *Proc. Of ACM SenSys'03*, 2003

20. Texas Instruments, “Cipcon CC1000 Single Chip Very Low Power RF Transceiver”. [Online]. Available: http://focus.ti.com/lit/ds/symlink/cc1000.pdf. [Accessed: Nov. 21, 2008].

21. J. Polastre, J. Hill, and D. Culler, “Versatile Low-Power Media Access for Wireless Sensor Networks,” in *Proc. of the Second ACM Conference on Embedded Networked Sensor Systems* (*SenSys’04*), Baltimore, MD, USA, November 2004, pp. 95–107.

22. K. Jamieson, B. Hull, A. Miu, H. Balakrishnan, “Understanding the Real-World Performance of Carrier Sense,” in *Proc. of SIGCOMM’05 Workshops*, August 2005.

23. Crossbow Technology, Inc., “MoteWorks™ 2.0 Platform” [Online]. Available: http://www.xbow.com/Products/Product_pdf_files/Wireless_pdf/MoteWorks_OEM_Edition.pdf. [Accessed: Nov. 09, 2008].

24. TinyOS Forum, “Interfaces and components – TinyOS”. [Online]. Available: http://www.tinyos.net/tinyos-2.x/doc/nesdoc/mica2/. [Accessed: Nov. 12, 2010].

25. TinyOS Forum, “TinyOS 2.x Documentation”. [Online]. Available: http://www.tinyos.net/tinyos-2.x/doc/. [Accessed: Feb. 02, 2011].

26. F. Zhao and L. Guibas, “Wireless Sensor Networks: An information Processing Approach,” *Morgan Kaufmann Publishers*, San Francisco, CA, 2004.

27. P. Levis, “TinyOS Programming,” June 28, 2006.

28. M. Maróti, B. Kusy, G. Simon and A. Lédeczi, “The Flooding Time Synchronization Protocol,” in *Proc. of the second International Conference on Embedded Networked Sensor* (*SenSys’04*), Baltimore, MD, USA, November 2004.

29. J. Elson, L. Girod, and D. Estrin, “Fine-Grained Network Time Synchronization using Reference Broadcasts,” *in Proc. of the 5th Symposium on Operating Systems Design and Implementation* (*OSDI’02*), Boston, MA, December 2002.

30. G. Anastasi, A. Falchi, A. Passarella, M. Conti, and E. Gregori, "Performance measurements of motes sensor networks," in *Proc. of the 7th ACM International Symposium Modeling, Analysis and Simulation of Wireless and Mobile Systems*, 2004, p.p. 174-181.
31. Q. Cao, T. He, L. Fang, T. Abdelzaher, J. Stankovic, and S. Son, "Efficiency Centric Communication Model for Wireless Sensor Networks," in Proc. of *IEEE INFOCOM'06*, 2006.
32. TinyOS Community Forum, "MICA2 Radio Stack for TinyOS 1.x" [Online]. Available: http://www.tinyos.net/tinyos-1.x/doc/mica2radio/CC1000.html. [Accessed: Nov. 16, 2010].
33. A. Woo, T. Tong, and D. Culler, "Taming the Underlying Challenges of Reliable Multihop Routing in Sensor Networks," in *Proc. of the 1st International Conference on Embedded Networked Sensor Systems* (*SenSys'03*), Los Angeles, CA, USA, November 2003.
34. W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy efficient communication protocol for wireless micro-sensor networks," in *Proc. of the 33rd Hawaii International Conference on System Sciences* (*HICSS '00*), Maui, Hawaii, January 2000, pp. 3005 – 3014.
35. Crossbow Technologies Inc., "MPR-MIB Users Manual" [Online]. Available: http://www.xbow.com/support/support_pdf_files/mpr-mib_series_users_manual.pdf
36. S. Madden, M. Franklin, J. Hellerstein, and W. Hong, "TAG: a Tiny AGgregation service for ad-hoc sensor networks," *ACM SIGOPS Operation Systems Review*, 36(SI):131–146, 2002.
37. A. S. Tannenbaum, "Computer Networks", Prentice Hall PTR, 4 edition, 2003.

38. L. Zhou, S. Zhou, Y. Yao, “Multipath Rayleigh Fading Channels in the Low SNR Regime,” *in Proc. of the IEEE International Conference* (*ICC’06*), Volume 3, June 2006, p.p. 1404 – 1409.

39. J. Zhao, R. Govindan, “Understanding Packet Delivery Performance in Dense Wireless Sensor Networks”, in *Proc. of ACM SenSys'03*, Los Angeles, California, USA, November 5-7, 2003, pp. 233-248.

40. TinyOS Community Forum, “MultiHopRouter”, [Online]. Available: http://www.tinyos.net/tinyos-1.x/doc. [Accessed: Nov. 11, 2009]. http://www.tinyos.net/tinyos-1.x/doc/adhoc.pdf. [Accessed: Nov. 11, 2009]. http://www.tinyos.net/tinyos-1.x/doc/multihop/multihop_routing.html. [Accessed: Nov. 11, 2009].

41. R. Guy, B. Greenstein, J. Hicks, R. Kapur, N. Ramanathan, T. Schoellhammer, T. Stathopoulos, K. Weeks, K. Chang, L. Girod, and D. Estrin, “Experiences with the extensible sensing system (ESS),” in *CENS Technical Report* 61, UCLA, 2006, pp. 311–321.

42. J. Polastre, R. Szewczyk, and D. Culler, “Telos: enabling ultra-low power wireless research”, in *Proc. of the 4th International Symposium on Information Processing in Sensor Networks* (*IPSN’05*), 2005, pp. 364–369.

43. Tinyos multihoplqi collection protocol, “MultihopLQI”. [Online]. Available: http://www.tinyos.net/tinyos-1.x/tos/lib/MultiHopLQI/. [Accessed: Nov. 15, 2010].

44. A. Woo, D. Culler, “Evaluation of Efficient Link Reliability Estimators for Low-Power Wireless Networks,” *Technical Report*, EECS Department, UC Berkeley, UCB/CSD-03-1270, 2003. [Online]. Available: http://www.eecs.berkeley.edu /Pubs/TechRpts/2003/6239.html. [Accessed: Nov. 11, 2009].

45. Centre for Embedded Networked Sensing (CENS), “Extensible sensing system: An advanced network design for microclimate sensing”, in *CENS Technical Report*. [Online]. Available: http://www.cens.ucla.edu. [Accessed: Dec. 10, 2009].

46. D. Couto, D. Aguayo, J. Bicket, and R. Morris, “A High-Throughput Path Metric for Multi-Hop Wireless Routing”, in *Proc. of the 9th Annual International Conference on Mobile Computing and Networking* (*MobiCom’03*), San Diego, CA, USA, 2003, pp. 124-136.

47. C. Intanagonwiwat, R. Govindan, and D. Estrin, “Directed diffusion: A scalable and robust communication paradigm for sensor networks”, in *Proc. of the ACM/IEEE International Conference on Mobile Computing and Networking* (*MobiCom’00*), Boston, MA, USA, Aug. 2000, pp. 56–67.

48. J. Heidemann, F. Silva, C. Intanagonwiwat, R. Govindan, D. Estrin, and D. Ganesan, “Building efficient wireless sensor networks with low-level naming”, in *Proc. of ACM SOSP’01*, 2001, pp. 72-81.

49. R. Shah and J. Rabaey, “Energy aware routing for low energy ad hoc sensor networks,” in *Proc. of IEEE WCNC’02*, Orlando, FL, March 2002, pp. 350–355.

50. R. Govindan, E. Kohler, D. Estrin, F. Bian, K. Chintalapudi, O. Gnawali, R. Gummadi, S. Rangwala, and T. Stathopoulos, “Tenet: an architecture for tiered embedded networks”, *CENS Technical Report* No. 53, 2005.

51. G. Werner-Allen, K. Lorincz, J. Johnson, J. Lees, and M. Welsh, “Fidelity and Yield in a Volcano Monitoring Sensor Network”, in *Proc. of USENIX Symposium on Operating Systems Design and Implementation*, Seattle, WA, Nov. 2006, pp. 213-223.

52. A. Woo and T. Tong, “Tinyos mintroute collection protocol (MintRoute)”. [Online]. Available: http://www.tinyos.net/tinyos-1.x/tos/lib/MintRoute. [Accessed: Nov. 11, 2009].

53. E. Demaine, A. Lopez-Ortiz, and J. Munro, "Frequency Estimation of Internet Packet Streams with Limited Space", in *Proc. of the 10th Annual European Symposium on Algorithms* (ESA'02), Rome, Italy, September 2002, pp. 33-41.

54. J. Hill and D. Culler, "A wireless embedded sensor architecture for system-level optimization", *Technical report*, U.C. Berkeley, 2001.

55. B. Duncan and D. Malan, "Low-power secure routing for mica2 mote," *Computer Science Group*, Harvard University, 2004.

56. K. Daabaj, M. Dixon and T. Koziniec, "Experimental study of load balancing routing for improving lifetime in sensor networks", in *the 5th International Conference on Wireless Communications, Networking and Mobile Computing* (*WiCOM'09*), Beijing, China, Sept. 2009, 3471-3474.

57. M. Zuniga and B. Krishnamachari, "An analysis of unreliability and asymmetry in low-power wireless links", *ACM Transactions on Sensor Networks*, 3(2):1–30, 2007.

58. D. Puccinelli and M. Haenggi, "DUCHY: Double cost field hybrid link estimation for low-power wireless sensor networks," in *Proc. of HotEmNets'08*, Charlottesville, Virginia, USA, June 2–3, 2008, pp. 423-435.

59. K. Srinivasan and P. Levis, "RSSI is under appreciated", in *the 3rd Workshop on Embedded Networked Sensors* (*EmNets'06*), Cambridge, MA, USA, May 2006, pp. 113-121.

60. O. Gnawali, M. Yarvis, J. Heidemann, and R. Govindan, "Interaction of retransmission, blacklisting, and routing metrics for reliability in sensor network routing", in *Proc. of the IEEE Conference on Sensor and Adhoc Communication and Networks*, Santa Clara, CA, USA, October 2004, pp. 34–43.

61. S. Mahfoudh, P. Minet, "Survey of Energy Efficient Strategies in Wireless Ad Hoc and Sensor Networks," in *Proc. of the 7th* International Conference on Networking (ICN'08), Cancun, April 2008, pp. 1-7.

62. R. Fonseca, O. Gnawali, K. Jamieson, and P. Levis, "Four-bit wireless link estimation", in *Proc. of the Sixth Workshop on Hot Topics in Networks* (*HotNets-VI*), Atlanta, GA, USA, November 2007, pp. 115-121.

63. A. Cerpa, J. Wong, L. Kuang, M. Potkonjak, and D. Estrin, "Statistical Model of Lossy Links in Wireless Sensor Networks", in *Proc. of the 6th ACM international Symposium on Mobile Ad Hoc Networking and Computing*, Urbana-Champaign, IL, USA, May 2005, pp. 221-229.

64. K. Woyach, D. Puccinelli, and M. Haenggi, "Sensorless sensing in wireless networks: implementation and measurements", in *Proc. of the 2nd International Workshop on Wireless Network Measurement* (*WinMee'06*), Boston, MA, USA, April 2006, pp. 23-33.

65. K. Srinivasan, P. Dutta, A. Tavakoli, and P. Levis, "Understanding the causes of packet delivery success and failure in dense wireless sensor networks", *Stanford University Technical Report*, 2006.

66. TinyOS-2 TEP 123, "Collection Tree Protocol (CTP)". [Online]. Available: http://www.tinyos.net/tinyos-2.x/doc/html/tep123.html. [Accessed: Nov. 17, 2010].

67. O. Gnawali, R. Fonseca, K. Jamieson, D. Moss, and P. Levis, "Collection Tree Protocol," in Proc. of *the 7th ACM Conference on Embedded Networked Sensor Systems* (*SenSys'09*), Berkeley, California, USA, 2009, pp. 23-33.

68. O. Gnawali, R. Fonseca, K. Jamieson, D. Moss, and P. Levis, "Collection Tree Protocol," *Technical Report*, Stanford University, 2009.

69. J. Choi, J. Lee, M. Wachs, and P. Levis, "Opening the Sensornet Black Box," *ACM SIGBED Review, Special issue on the workshop on wireless sensor network architecture*, April, 2007.M. Wachs, J. Choi, J. Lee, K. Srinivasan, Z. Chen, M. Jain, and P. Levis, "Visibility: A New Metric for Protocol Design", in *Proc. of the 5th ACM Conference on Embedded Networked Sensor Systems* (*SenSys'07*), 2007, pp.

75-87.C. Toh, “Maximum Battery Life Routing to Support Ubiquitous Mobile Computing in Wireless Ad Hoc Networks,” *IEEE Communications Magazine*, 39: 138–147, 2001.G. Werner-Allen, P. Swieskowski, and M. Welsh, “Motelab: A wireless sensor network testbed,” in *Proc. of the 4th International Conference on Information Processing in Sensor Networks* (*IPSN’05*), 2005, pp. 43-51.

70. J. Chang, L. Tassiulas, “Energy conserving routing in wireless ad-hoc networks”, in *Proc. of IEEE INFOCOM’00*, Tel-Aviv, Israel, March 2000, pp. 22–31.
71. Tutonet Wireless Sensor Network Simulator. [Online]. Available: http://enl.usc.edu/projects/tutornet. [Accessed: Dec. 11, 2010].
72. OMNeT++ Simulation Framework: [Online]. Available: http://www.omnetpp.org. [Accessed: Dec. 11, 2010].
73. W. Song, R. Huang, M. Xu, A. Ma, B. Shirazi, R. LaHusen, “Air-dropped sensor network for real-time high-fidelity volcano monitoring,” in *Proc. Of MobiSys’09*, Krakow, Poland, June 22–25, 2009, pp. 135-145.
74. D. Puccinelli and M. Haenggi, “Reliable data delivery in large-scale low-power sensor networks,” ACM Transactions on Sensor Networks, Vol. V, No. N, June 2008, pp. 141-146.
75. Texas Instruments, “Chipcon CC2420 Data sheet”. [Online]. Available: http://focus.ti.com/lit/ds/symlink/cc2420.pdf. [Accessed: Feb. 11, 2011].
76. N. Baccour, et al, “A Comparative simulation study of link quality estimators in wireless sensor networks,” in *Proc. of 17th Annual Meeting of the IEEE/ACM International Symposium on Modelling, Analysis and Simulation of Computer and Telecommunication Systems* (*MASCOTS’09*), South Kensington Campus, Imperial College London, September 21-23, 2009, pp. 123-133.

77. J. Polastre, R. Szewczyk, and D. Culler, “Telos: enabling ultra-low power wireless research,” in *Proc. of the 4th international symposium on Information processing in sensor networks* (*IPSN’05*), Piscataway, NJ, USA: IEEE Press, 2005, pp. 48-53.

78. D. Lal, A. Manjeshwar, F. Herrmann, E. Uysal-Biyikoglu, and A. Keshavarzian, “Measurement and characterization of link quality metrics in energy constrained wireless sensor networks,” in *Proc. of IEEE Global Telecommunications Conference* (*GLOBECOM’03*), 2003pp. 163-171.

79. A. Cerpa, J. L. Wong, M. Potkonjak, and D. Estrin, “Temporal properties of low power wireless links: modeling and implications on multi-hop routing,” in Proc. of *Proc. of the 6th ACM international symposium on Mobile ad hoc networking and computing* (*MobiHoc’05*), New York, NY, USA: ACM, 2005, pp. 414–425.

80. I. F. Akyildiz, W. Su, Y. Sankasubramaniam, and E. Cayirci, “Wireless Sensor Networks: A Survey”, *Computer Networks*, 38: 393–422, 2002.

81. R. Fonseca, O. Gnawali, K. Jamieson, and P. Levis, “Four bit wireless link estimation,” in Proc. *of the 6th Workshop on Hot Topics in Networks* (*HotNets VI*), 2007, 422-452.

82. M. Senel, K. Chintalapudi, D. Lal, A. Keshavarzian, and E. J. Coyle, “A kalman filter based link quality estimation scheme for wireless sensor networks,” in *Proc. of IEEE Global Telecommunications Conference* (*GLOBECOM’07*), 2007, pp. 212-220.

83. Crossbow Inc., TelosB datasheet, [Online]. Available: http://www.xbow.com/Products/Product_pdf_files/Wireless_pdf/TelosB_Datasheet.pdf

84. IEEE 802.15.4-2006 standard. [Online]. Available: http://standards.ieee.org/getieee802/802.15.html. [Accessed: Nov. 11, 2009].

85. N. Burri, P. von Rickenbach, and R. Wattenhofer, “Dozer: ultra-low power data gathering in sensor networks,” in *Proc. of the 6th international conference on*

Information processing in sensor networks (*IPSN'07*), New York, NY, USA, 2007, pp. 450–459.

86. H. Dubois-Ferrier, R. Meier, L. Fabre, and P. Metrailler, "TinyNode: a comprehensive platform for wireless sensor network applications," in *Proc. of the 5th international conference on Information processing in sensor networks* (*IPSN'06*), 2006, pp. 213-221.
87. K. Scott, N. Bambos, "Routing and channel assignment for low power transmission in PCS", in *Proc. International Conference on Universal Personal Communications*, Cambridge, MA, September 1996, pp. 469–502.
88. S. Singh, M. Woo, and C. S. Raghavendra, "Power-aware routing in mobile ad hoc networks," in *Proc. of the 4th ACM/IEEE International Conference on Mobile Computing and Networking* (*MobiCom'98*), Dallas, TX, October 1998, pp. 62-70.
89. N. Bambos, "Toward power-sensitive network architectures in wireless communications: concepts, issues, and design aspects", *IEEE Personal Communications*, 5: 50–59, 1998.
90. J. Chang and L. Tassiulas, "Routing for maximum system lifetime in wireless ad-hoc networks", in *Proc. of the 39th Annual Allerton Conference on Communication, Control, and Computing*, Allerton, NY, October 1999.
91. B. Krishnamachari, D. Estrin, and S. Wicker, "The impact of data aggregation in wireless sensor networks", in *Proc. of the Workshops of 22nd International Conference on Distributed Computing Systems*, *IEEE Computer Society*, Vienna, Austria, July 2002. 575–578.
92. E. Ayanoglu, I. Chih-Lin, R. Gitlin, and J. Mazo, "Diversity coding for tansparent self-healing and fault-tolerant communication networks", *IEEE Transactions on Communications*, 41(11): 1377–1386, 1993.

93. A. Banerjea, “Simulation study of the capacity effects of disparity routing for fault tolerant real-time channels,” *ACM SIGCOMM Computer Communication Review*, 26: 194–205, 1996.

94. S. Lee and M. Gerla, “Split Multipath Routing with Maximally Disjoint Paths in Ad hoc Networks”, in *Proc. of the IEEE International Conference on Communications* (*ICC’01*), St. Petersburg, VA, June 2001.

95. A. Nasipuri, R. Castaneda, and S. Das, “Performance of Multipath Routing for On-Demand Protocols in Ad Hoc Networks”, *Mobile Networks and Applications* (*MONET’02*), 6(4): 339–349, 2002.

96. A. Nasipuri and S. Das, “On-demand multipath routing for mobile ad-hoc networks”, in *Proc. of the 8th International Conference on Computer Communications and Networks* (*ICCCN’99*), Boston, MA, 1999.

97. P. Papadimitratos, Z. Haas, and E. Sirer, “Path-set selection in mobile ad hoc networks,” in *Proc. of the 3rd ACM International Symposium on Mobile Ad Hoc Networking and Computing* (*MobiHoc’02*), Lausanne, Switzerland, 2002.

98. M. Pearlman, Z. Haas, P. Scholander, and S. Tabrizi, “On the impact of alternate path routing for load balancing in mobile ad hoc networks,” in *Proc. of IEEE/ACM Workshop on Mobile Ad Hoc Networking and Computing* (*MobiHoc’00*), Boston, MA, August 2000, pp. 3-10.

99. J. Zheng and M. Lee, “Will IEEE 802.15.4 Make ubiquitous networking a reality?: a discussion on a potential low power, low bit rate standard,” *IEEE Communications Magazine*, 42(6): 140–146, 2004.

100. T. Cormen, C. Leiserson, and R. Rivest, “Introduction to Algorithms”, McGraw-Hill Book Company, 1997.

101. K. Daabaj, M. Dixon and T. Koziniec, “LBR: Load Balancing Routing in Wireless Sensor Networks,” in: Ao, S-L, (ed.) IAENG Transactions on Engineering

Technologies: Volume 4: Special Edition of the World Congress on Engineering and Computer Science-2009. American Institute of Physics (AIP), New York, USA, July 2010, pp. 71-85.

102. K. Daabaj, M. Dixon and T. Koziniec, "Avoiding Routing Holes in Homogeneous Wireless Sensor Network," in *Proc. of the World Congress of Engineering and Computer Science* (*WCECS'09*), Vol I, Berkeley, CA, USA, October 20-22, 2009, pp. 356-361.

103. J. Al-Karaki and A. KamalRouting, "Techniques in Wireless Sensor Networks: A Survey," IEEE Wireless Communications, 11(6):6–28, 2004.

104. T. Schoellhammer, B. Greenstein, and D. Estrin, "Hyper: a routing protocol to support mobile users of sensor networks," *Technical Report 2013*, CENS, 2006.

105. H. Zhang, A. Arora, Y. Choi, and M. Gouda, "Reliable bursty convergecast in wireless sensor networks," *Computer Communications*, 30(13):2560–2576, Dec. 2007.

106. J. Hui and D. Culler, "Ip is dead, long live ip for wireless sensor networks," in *Proc. 6th ACM Conference of Embedded Networked Sensor Systems* (*SenSys'08*), ACM Press, New York, USA, Nov. 2008, pp. 15–28.

107. M. Yarvis, N. Kushalnagar, H. Singh, A. Rangarajan, Y. Liu, and S. Singh, "Exploiting heterogeneity in sensor networks," in *Proc. IEEE 24th Annu. Joint Conference INFOCOM* (*INFOCOM'05*), Mar. 2005, vol. 2, pp. 878–890.

108. B. Titzer, D. Lee, J. Palsberg, "Avrora: scalable sensor network simulation with precise timing," in *Proc. of the 4th international symposium on Information processing in sensor networks* (*IPSN'05*), Los Angeles, California, 2005.

109. T. He, J. A. Stankovic, C. Lu, and T. Abdelzaher, "SPEED: A stateless protocol for real-time communication in sensor networks," in *Proc. of the 23rd International*

Conference on Distributed Computing Systems (*ICDCS'03*), Providence, Rhode Island, May 2003.

110. D. Puccinelli and M. Haenggi, "Arbutus: Network-layer load balancing for wireless sensor networks," in *Proc. IEEE WCNC*, Apr. 2008, pp. 2063–2068.

111. J. N. Al-Karaki and A. E. Kamal, "Handbook of Sensor Networks: Compact Wireless and Wired Sensing Systems," Boca Raton, FL: CRC, 2005, pp. 116–138.

112. K. Akkaya and M. Younis, "Survey of routing protocols in wireless sensor networks," *Ad Hoc Networks*, vol. 3, no. 3, pp. 325–349, May 2005.

113. C. Gomez, P. Salvatella, O. Alonso, and J. Paradells, "Adapting AODV for IEEE 802.15.4 mesh sensor networks: Theoretical discussion and performance evaluation in a real environment," in *Proc. of WoWMoM'06*, 2006, pp. 159–170.

114. H. Cheng and J. Cao, "A design framework and taxonomy for hybrid routing protocols in mobile ad hoc networks," *Communications Surveys Tutorials*, vol. 10, no. 3, 2008, pp. 62–73.

115. P. Samar, M. Pearlman and Z. Haas, "Independent zone routing: an adaptive hybrid routing framework for ad hoc wireless networks," *IEEE/ACM Transactions on Networks*, 12, (4), August 2004, pp. 595–608.

116. V. Mhatre, C. Rosenberg, D. Kofman, R. Mazumdar, and N. Shroff, "A minimum-cost heterogeneous sensor network with a lifetime constraint," *IEEE Transaction on Mobile Computing*, vol. 4, no. 1, pp. 4–15, Jan./Feb. 2005.

117. K. Daabaj, M. Dixon, T. Koziniec and P. Cole, "reliable routing scheme for indoor sensor networks," in *Proc. of the 21st Annual IEEE International Symposium on Personal, Indoor and Mobile Radio Communications* (*PIMRC'10*), Istanbul, Turkey, Sept. 26–29, 2010, pp. 1614-1619.

118. K. Daabaj, M. Dixon, T. Koziniec and D. Murray, "Reliable data delivery in low energy ad hoc sensor networks," in *Proc. of the 16th Asia-Pacific Conference on*

Communications (*APCC'10*), Auckland, New Zealand, Oct. 31-Nov. 03, 2010, pp. 215-220.

119. K. Daabaj, M. Dixon and T. Koziniec, "Traffic eavesdropping based scheme to deliver time-sensitive data in sensor networks," in *Proc. of the 29th IEEE International Performance, Computing, and communications conference* (*IPCCC'10*), New Mexico, USA, December 9-11, 2010, pp. 302-308.

120. K. Daabaj, "Load-Balanced Routing Scheme for TinyOS-based wireless sensor networks," in *Proc. of the IEEE International Conference of Wireless Information Technology and Systems* (*ICWITS'10*), Honolulu, Hawaii, USA, August 28-September 3, 2010, pp. 1-4.

121. C. Diallo, M. Marot, M. Becker, "Link quality and local load balancing routing mechanisms in wireless sensor networks," in *Proc. of the Sixth Advanced International Conference on Telecommunications*, 2010.

122. J. Luo, P. Eugster and J. Hubaux, "route driven gossip: Probabilistic reliable multicast in ad hoc networks," in *Proc. of INFCOM'03*, 2003.

123. R. Jurdak, P. Baldi and C. Lopes, "Energy-aware adaptive low power listening for sensor networks," in *Proc. of the 2nd International Workshop on Networked Sensing Systems*, San Diego, CA, June, 2005.

124. U. Malesci and S. Madden, "A measurement-based analysis of the interaction between layers in TinyOS," in *Proc. of the 3rd European conference on Wireless Sensor Networks*, 2006.

125. C. Barrett, S. Eidenbenz, L. Kroc, M. Marathe, and J. Smith, "Parametric probabilistic sensor network routing," in *Proc. of The Second ACM International Workshop on Wireless Sensor Networks and Applications* (*WSNA'03*), San Diego, CA, Sept. 2003.

126. C. Barrett, S. Eidenbenz, L. Kroc, M. Marathe, and J. Smith, "Probabilistic multi-path vs. deterministic single-path protocols for dynamic ad-hoc network scenarios," in *Proc.*

of the 2005 ACM Symposium on Applied Computing (*SAC'05*), Santa Fe, NM, March 2005.

127. K. Langendoen, A. Baggio, and O. Visser, "Murphy loves potatoes: Experiences from a pilot sensor network deployment in precision agriculture," in *Proc. of WPDRTS*, 2006.

128. M. Nassr, J. Jun, S. Eidenbenz, A. Hansson and A. Mielke, "Scalable and reliable sensor network routing: Performance study from field deployment," in *Proc. of INFOCOM'07*, 2007, pp. 670–678.

129. M. Marina, and S. Das, "On-demand multipath distance vector routing in ad hoc networks," in *Proc. of the 9th IEEE International Conference Network Protocols* (*ICNP'01*), 2001, pp. 14-23.

130. A. Shuaib, A. Aghvami, "A routing scheme for the IEEE-802.15.4-enabled wireless sensor networks", in *Proc. of the IEEE Transations on Vehicular Technology*, Vol. 58, No. 9, November 2009.

131. C. Perkins, E. Belding-Royer, and S. Das, "Ad hoc On-demand Distance Vector (AODV) Routing," *IETF*, *RFC 3561*, July 2003.

132. V. Rai and R. N. Mahapatra, "Lifetime modeling of a sensor network," in *Proc. of the Conference on Design, Automation and Test in Europe* (*DATE'05*), Washington, DC, USA: IEEE Computer Society, 2005, pp. 202–203.

133. Yunxia Chen and Qing Zhao, "On the Lifetime of Wireless Sensor Networks", *IEEE COMMUNICATIONS LETTERS*, Vol. 9, NO. 11, November 2005.

134. H. Zhang and J. Hou, "On deriving the upper bound of lifetime for large sensor networks," in *Proc. of the 5th ACM international symposium on Mobile ad hoc networking and computing* (*MobiHoc'04*), ACM Press, New York, NY, USA, 2004, pp. 121–132.

135. E. J. Duarte-Melo and M. Liu, "Analysis of energy consumption and lifetime of heterogeneous wireless sensor networks," in *Proc. of the Global*

Telecommunications Conference (*GLOBECOM'02*), IEEE, vol. 1, Nov. 2002, pp. 21–25.

136. W. B. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan, "An application-specific protocol architecture for wireless microsensor networks," *IEEE Transactions on Wireless Communications*, vol. 1, no. 4, October 2002, pp. 660–670.

137. R. Madan and S. Lall, "Distributed algorithms for maximum lifetime routing in wireless sensor networks," *IEEE Transactions on Wireless Communications*, vol. 5, no. 8, Aug. 2006, pp. 2185–2193.

138. J. Chang and L. Tassiulas, "Maximum lifetime routing in wireless sensor networks", *IEEE/ACM Transactions on Networking (TON)*, 12:609-619, Vol. 12 Issue 4, IEEE Press Piscataway, NJ, USA, August 2004.

139. Y. Xu, J. Heidemann, and D. Estrin, "Geography-informed energy conservation for ad hoc routing," in *Proc. of the 7th annual international conference on Mobile computing and networking* (*MobiCom'01*), ACM Press, NY, USA, 2001, pp. 70–84.

140. E. I. Oyman and C. Ersoy, "Multiple sink network design problem in large scale wireless sensor networks," *in the Proc. of IEEE International Conference on Communications* (*ICC'04*), vol. 6, June 2004, pp. 3663–3667.

141. M. Bhardwaj, T. Garnett, and A. P. Chandrakasan, "Upper bounds on the lifetime of sensor networks," in *Proc. of the IEEE International Conference on Communications, Communications* (*ICC'01*), vol. 3, Helsinki, Finland, 2001, pp. 785–790.

142. M. Bhardwaj and A. P. Chandrakasan, "Bounding the lifetime of sensor networks via optimal role assignments," in *Proc. of the 21st Annual Joint Conference of the IEEE Computer and Communications Societies* (*INFOCOM'02*), vol. 3, 2002, pp. 1587–1596.

143. T. Rapport, *Wireless Communications: Principles and Practice*, Prentice Hall, 1996. ISBN 0-13-375536-3.

144. F. Othman, N. Bouabdallah, and R. Boutaba, "Load-balanced routing scheme for energy-efficient wireless sensor networks," in *Proc. of IEEE Globecom'08*, New Orleans, LA USA, 2008.

145. S. Toumpis and S. Gitzenis, "Load balancing in wireless sensor networks using Kirchhoff's voltage law," in *Proc. of IEEE INFOCOM'09*, Rio de Janeiro, Brazil, 2009.

146. J. Gao and L. Zhang, "Load balanced short path routing in wireless networks," in *Proc. of IEEE INFOCOM'04*, Hong Kong, China, 2004.

147. Z. Wang, E. Bulut, and B. Szymanski, "Energy efficient collision aware multipath routing for wireless sensor networks", in *Proc. of IEEE ICC'09*, Dresden, Germany, 2009.

148. L. Popa, C. Raiciu, I. Stoica, and D. Rosenblum, "Reducing congestion effects by multipath routing in wireless networks", in *Proc. of the 14th IEEE International Conference on Network Protocols* (*ICNP'06*), Santa Barbara, USA, 2006, pp. 96-105.

149. C. Wu, R. Yuan, and H. Zhou, "A novel load balanced and lifetime maximization routing protocol in wireless sensor networks", in *Proc. of IEEE Vehicular Technology Conference* (*VTC'08*) *Spring*, Singapore, 2008, pp. 113-117.

150. V. Gungor, C. Sastry, Z. Song, and R. Integlia, "Resource-aware and link quality based routing metric for wireless sensor and actor networks", in *Proc. of IEEE International Conference on Communications* (*ICC'07*), Glasgow, Scotland, 2007.

151. M. Rashid, M. Alam, M. Razzaque, and C. Hong, "Reliable event detection and congestion avoidance in wireless sensor networks", in *Proc. of High Performance*

Computing Conference (*HPCC'07*), *LNCS 4782*, Houston, Texas, USA, 2007, pp. 521-532.

152. R. Min, M. Bhardwaj, S.-H. Cho, N. Ickes, E. Shih, A. Sinha, A. Wang, and A. Chandrakasan, "Energy-Centric Enabling Technologies for Wireless Sensor Networks," *IEEE Wireless Communications*, vol. 9, Aug. 2002, pp. 28–39.

153. K. Daabaj, "Energy-Aware Reliability-Oriented Scheme to Deliver Time-Sensitive Data in Sensor Networks," in *Proc. of the 2010 IEEE International Conference on Service-Oriented Computing and Applications* (*SOCA 2010*), Perth, WA, Australia, December 13-15, 2010, pp. 1-8.

154. K. Daabaj, M. Dixon and T. Koziniec and K. Lee, "Reliable Routing for Low-Power Smart Space Communications," *the journal of IET Communications*, 2011.

155. I. Khemapech, A. Miller and I. Duncan, "A Survey of Transmission Power Control in Wireless Sensor Networks," in *Proc. of IKPGNET 2007*.

156. J. Heidemann and W. Ye, "Energy Conservation in Sensor Networks at the Link and Network Layers", USC/ISI Technical Report ISI-TR-599, November 2004.

157. J. Jeong, D. Culler and J-H Oh, "Empirical Analysis of Transmission Power Control Algorithms for Wireless Sensor Networks," in *Proc. of the Fourth International Conference on Networked Sensing Systems* (*INSS'07*), Braunschweig, June 6-8, 2007, pp. 27-34.

158. S. Panichpapiboon, G. Ferrari and O. Tonguz, "Optimal Transmit Power in Wireless Sensor Networks," IEEE Transactions on Mobile Computing, Vol. 5, No. 10, October 2006, pp. 1432-1447.

159. M. Kubisch, H. Karl, A. Wolisz, L. C. Zhong, and J. Rabaey, "Distributed algorithms for transmission power control in wireless sensor networks," IEEE WCNS, 2003.

160. D. Son, B. Krishnamachari, and J. Heidemann, “Experimental study of the effects of transmission power control and blacklisting in wireless sensor networks,” in *Proc. of IEEE SECON’04*, October 2004.

161. T. A. ElBatt, S. V. Krishnamurthy, D. Connors, and S. Dao, “Power management for throughput enhancement in wireless ad-hoc networks,” in *Proc. of IEEE ICC’00*, 2000.

162. S. Lin, J. Zhang, G. Zhou, L. Gu, J. A. Stankovic, and T. He, “ATPC: adaptive transmission power control for wireless sensor networks,” in *Proc. of ACM Sensys’06*, Colorado, November 2006.

163. D. Hwang, E. Kwon and J. Lim, “EASR: an energy aware source routing with disjoint multipath selection for energy-efficient multihop wireless ad hoc networks,” in *Proc. of Conference Networking 2006*, LNCS 3976, 2006, pp. 41-50.

164. F. Ovalle-Martínez, I. Stojmenovi´c, F. García-Nocetti and J. Solano-González, “Finding minimum transmission radii for preserving connectivity and constructing minimal spanning trees in ad hoc and sensor networks,” *Journal of Parallel and Distributed Computing*, 65(2):132–141, 2005.

165. S. Narayanaswamy, V. Kawadia, R. Sreenivas and P. Kumar, “Power control in ad-hoc networks: Theory, architecture, algorithm and implementation of the COMPOW protocol,” in *Proc. of European Wireless Conference*, February 2002, pp. 156–162.

166. F. Ingelrest, D. Simplot-Ryl and I. Stojmenovic, “Optimal Transmission Radius for Energy Efficient Broadcasting Protocols in Ad Hoc Networks,” *IEEE Transactions on Parallel and Distributed Systems*, June 2006.

167. L. Correia, D. Macedo, A. dos Santos d, A. Loureiro, J. Nogueira, “Transmission power control techniques for wireless sensor networks,” *Elsevier Computer Networks*, 51 (2007), pp. 4765–4779.

168. R. Wattenhofer, L. Li, P. Bahl, and Y. Wang, “Distributed topology control for power efficient operation in multihop wireless ad hoc networks”, in *Proc. of IEEE INFOCOM*, April 2001, pp. 1388–1397.

169. D. Blough, et al, “The k-neigh protocol for symmetric topology control in ad hoc networks”, in *Proc. of ACM MobiHoc*, June 2003, pp. 141–152.

170. P. McCullagh and J. Nelder, “Generalized Linear Models,” Second Edition. *Boca Raton*: *Chapman and Hall/CRC*, 1989, ISBN 0-412-31760-5.