

We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,900

Open access books available

186,000

International authors and editors

200M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com



Real-Time Cross-Layer Routing Protocol for Ad Hoc Wireless Sensor Networks

Khaled Daabaj¹ and Shubat Ahmeda²

¹*Murdoch University*

²*Alfateh University*

¹*Australia*

²*Libya*

1. Introduction

Reliable and energy efficient routing is a critical issue in Wireless Sensor Networks (WSNs) 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. Typically, WSNs provide the ability to gather information cheaply, accurately and reliably over both small and vast physical regions. Unlike other large data network forms, where the ultimate input/output interface is a human being, 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 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. The routing problem in ad hoc WSNs is nontrivial issue because of sensor node failures due to restricted resources. 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 the literature, such as in (Rahul et al., 2002), (Woo et al., 2003), (TinyOS, 2004), (Gnawali et al., 2009) and (Burri et al., 2007) 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 keep small message overhead and balanced energy dissipation. Recent studies on energy efficient routing in multihop WSNs have shown a great reliance on radio link quality in the path selection process. 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 as stated and resolved in and (Daabaj 2010). This chapter presents an empirical study on how to improve energy efficiency for reliable multihop communication by developing a real-time cross-layer lifetime-oriented routing protocol and integrating useful routing information from different layers to examine their

joint benefit on the lifetime of individual sensor nodes and the entire sensor network. The proposed approach aims to balance the workload and energy usage among relay nodes to achieve balanced energy dissipation, thereby maximizing the functional network lifetime. The obtained experimental results are presented from prototype real-network experiments based on Crossbow's sensor motes (Crossbow, 2010), i.e., Mica2 low-power wireless sensor platforms (Crossbow, 2010). The distributed real-time routing protocol which is proposed in this chapter aims to face the dynamics of the real world sensor networks and also to discover multiple paths between the base station and source sensor nodes. The proposed routing protocol is compared experimentally with a reliability-oriented collection-tree protocol, i.e., the TinyOS MintRoute protocol (Woo et al., 2003). The experimental results show that our proposed protocol has a higher node energy efficiency, lower control overhead, and fair average delay.

2. Motivations

While the majority of WSN-related research activities have used open-source network simulators such as ns-2 (ISI, 2010) and OMNeT++ (Omnetpp, 2010), others have used well-controlled indoor remote access testbeds such as Motelab (Werner-Allen, 2005) 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 asymmetrical 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. In low power WSNs, the unreliability of the links and the limitations of all resources bring considerable complications to the routing scheme. Even in the presence of static topology of fixed 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. Furthermore, sensor nodes are typically battery-powered, 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.

In this chapter, the above points will be addressed by describing the proposed protocol, and presenting a detailed analysis of the protocol's performance using a physical WSN platform. In the proposed protocol, the decision where to forward data packets depends on many potential factors from different layers. The implementation section will include a detailed discussion of the performance of the proposed protocol which is benchmarked against the baselines in an indoor using wireless platform. Standalone evaluation of routing efficiency is impracticable, as temporal dynamics prevent knowing what the optimal route would be for data dissemination. Therefore, routing efficiency is evaluated as a comparative measure. The proposed protocol is benchmarked with the updated version of TinyOS MintRoute (Woo et al., 2003) implementation using Crossbow's Mica2 sensor motes. Since MintRoute protocol (Woo et al., 2003) has been used in recent WSNs deployments, it is considered a reasonable evaluation. The testbed environment is conducted indoor using Crossbow's

Mica2 868/916MHz (Crossbow, 2010). Currently, Mica2 motes represent the lowest cost wireless sensor platform based on commercial off-the-shelf hardware components for low power sensor networks. Mica2 platform is running with the TinyOS (TinyOS, 2004) development environment.

3. Related work

In the literature, many reliability-based routing protocols have been proposed. However, the main disadvantages of the existing TinyOS collection routing protocols based on link quality are that they are unaware of the energy status of nodes and do not explicitly pursue balanced energy usage in their routing schemes; thereby diverting load to sensor nodes with low energy capacity. As a result, this chapter focuses on balanced energy dissipation scheme for lifetime maximisation by taking the advantage from reliability-oriented routing schemes, i.e., MintRoute (Woo et al., 2003), MultihopLQI (TinyOS, 2004) and CTP (Gnawali et al., 2009) collection protocols, and traditional energy-aware routing schemes, i.e., Energy-Aware Routing (EAR) protocol (Rahul et al., 2002). Although the main objective of load balancing routing is the efficient utilization of network resources, none of the studies reviewed above takes jointly link reliability and energy-wise metrics into account with load balancing. There is no doubt that a better distribution of load leads to the more efficient use of bandwidth, which means that there is less contention and consequently less energy is consumed, but it is not self-contained for achieving complete energy efficiency. WSNs are not necessarily energy-homogeneous, and there is thus insufficient information about the sensor nodes' load tasks to enable the energy-wise selection of the paths. The current load of a given sensor node can be used to estimate the future dissipation of energy but it does not contain a record of past activities and the residual energy level of the sensor node remains hidden. The proposed routing algorithm allows a child sensor node dynamically searches for a new reliable parent node with more residual energy and takes in account the tradeoffs between latency and energy. This dynamic adaptation strategy can alleviate the energy hole problem. The chapter aims to improve the performance evaluation of the proposed routing scheme by extending the experiments to indoor, outdoor, and simulations on larger networks.

4. Description of the real-time cross-layer routing protocol

4.1 Overview

Since the communications overheads are the major energy consumer during a sensor node's operation, the proposed routing protocol, a simple but reliable routing algorithm, aims to cause minimal communication overheads for network configuration and multi-hop data dissemination. As shown in figure 1, the proposed protocol uses Channel State Information (CSI) and residual energy capacity with other overheard parameters, e.g., aggregation load, sensor node-*id*, and tree-level, to form a cost function for selecting the most reliable and energy-efficient route towards the base station. In low power sensor networks, the unreliability of the links and the limitations of all resources bring considerable complications to routing. Even though most deployed sensor networks use stationary nodes or have low mobility, the channel conditions vary because of various effects such as asymmetrical radio performance, or multipath fading effects which modify the patterns of radio wave reflections. Since sensor nodes are typically battery-powered, and ongoing maintenance may be impracticable, the progressive reduction of the available residual

power needs to be considered jointly with other factors as a crucial factor in the parent selection process to control nodes' energy drain for the extension of the lifetime of the individual nodes and for the achievement of load balancing and consistent energy usage within the entire network. The proposed protocol is a hybrid, reactive and proactive, routing algorithm 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 run-out of energy. In the proposed protocol, 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 Channel State Information (CSI) can be measured directly from the radio hardware circuitry of the wireless platform in form of signal quality or computed by software at the receiver based on the successfully received packets; 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. These parameters with other overheard local information are used to form a cost function for the selection of the most efficient route. Moreover, the presence of a time constraint requires the network to favour routes over a short path with minimum 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 proposed protocol is a tree-based routing protocol where a child sensor node forms a routing tree to its 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 nodes by choosing a next hop based on the selection parameters.

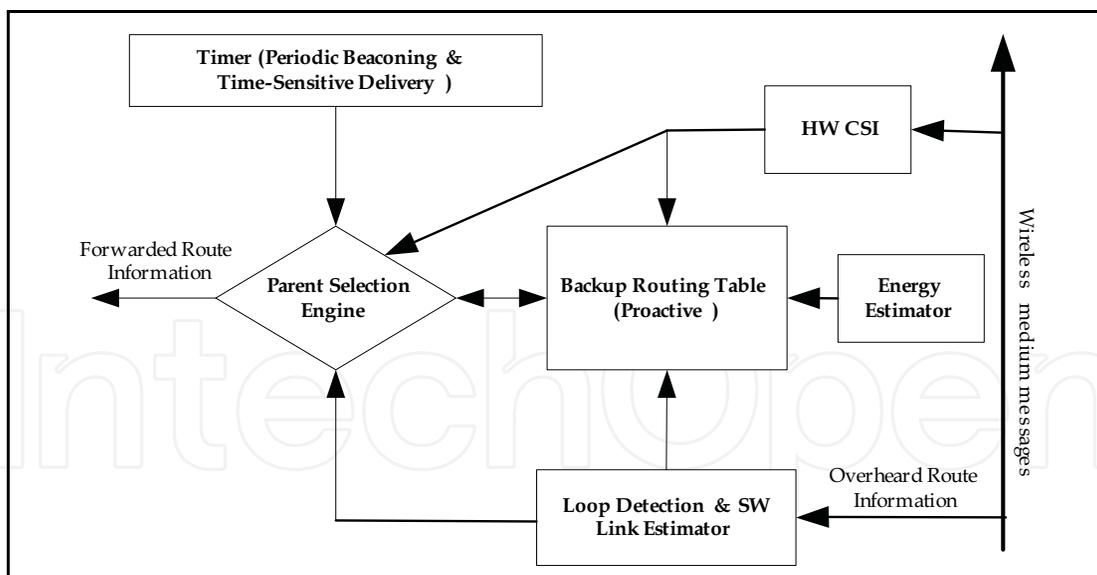


Fig. 1. Routing protocol framework

The proposed protocol mutually employs hardware-based Channel State Information (CSI) such as the Received Signal Strength Indicator (RSSI) and the Link Quality Indicator (LQI), to evaluate the signal quality of the wireless link and software-based link quality estimates of set of adjacent neighbours such as the Packet Reception Ratio (PRR), 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 the

proposed protocol adaptive to unforeseen traffic changes. The proposed protocol does also exploit the benefit from in-network processing mechanisms in term 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 time-sensitive amount of aggregation load. The proposed protocol 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, the proposed protocol is designed to perform the dynamic adaptation at the network or routing layer.

Although the main objective of load balancing routing is the efficient utilization of WSN resources, the load balancing is advantageous technique for evening out the distribution of loads in terms of efficient energy consumption. As a result, maximising lifetime of each sensor node can be achieved with fair battery usage. The cost function takes into account not only the current energy capacity of the sensor nodes and the channel state but also considering other factors like deployment pattern, event patterns. In other words, the proposed protocol considers the overall distribution of the delay-sensitive aggregation load along the routes by means of load balancing benefits for ensuring the even distribution of traffic, which translate into more efficient energy utilization reliable packet delivery.

Sensor nodes with the best link quality average values are considered first in the initial stages of parent selection process, while 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 traffic load, e.g., aggregation load, increases, the remaining battery capacity of each sensor node is also accounted as the second prime metric in the parent/route selection process to choose the routes along which all sensor nodes have the actual available battery capacity levels exceeding a given threshold. The cost function 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 by utilizing the principle of max-min cost function as explained in Conditional Max-Min Battery Capacity Routing. To ensure a longer network lifetime, the strategy of minimising the variance in energy levels is employed to dissipate up all batteries powers uniformly to avoid some nodes suddenly 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.

From energy cost point of view, the residual energy capacity defines the refusal or readiness of intermediate sensor nodes 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.

4.2 Routing tree formation

The routing tree is a directed non-cyclical graph which relays packets towards the base station over multiple paths. 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 2. The valid parent is elected by the routing metrics used in the routing cost function, i.e., link quality, residual energy, hop-count, aggregation load or latency. Obviously, any path from source sensor nodes to the base station is the most efficient path in the resulting routing tree. The routing tree starts

with the easily-constructed shortest path tree (SPT), and then allows each sensor node to pick a new parent node if it appears to provide better routing cost with a higher link quality. Using the broadcast nature of the contention-based wireless medium, a sensor node can easily observe its neighbourhood by receiving and overhearing periodic beacon packets which initially originate by the base station.

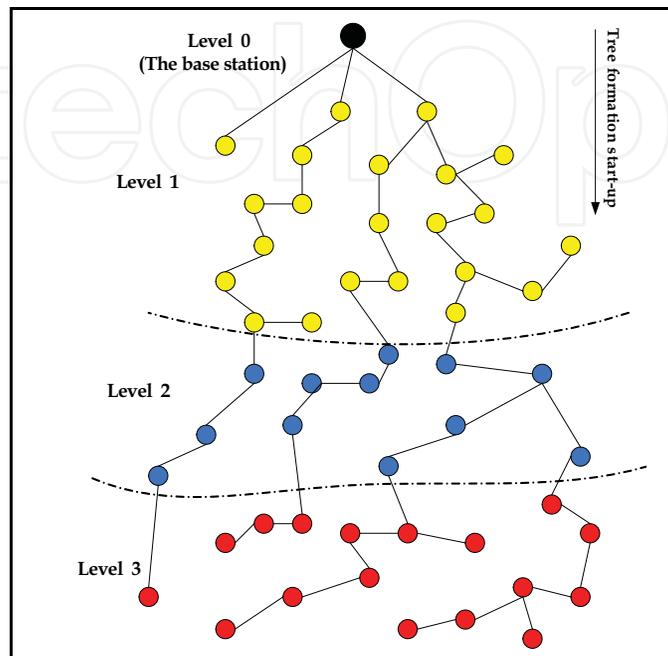


Fig. 2. Routing tree formation

4.3 Routing tree construction phases

The construction of the routing tree is performed in three overlapped phases: *Tree setup*, *Data transmission*, and *Route maintenance*.

In the *tree setup phase*, 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 the periodic beaconing during the *reactive* route maintenance phase in order to adapt with link dynamics. Therefore, the receiving sensor nodes determine all routes with their updated cost parameters to be used in 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 setup message. The intermediate sensor nodes at level one, for example, 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, for example, two-hops from the base station. Sensor nodes that have a higher cost compared with other peer sensor nodes, for example, 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 nodes and all nodes know their depth and the tree is fully defined.

In the *data transmission phase*, 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. Consecutively, 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. More challenging is the case when the time it takes a sensor node to deliver its local measurements or its own aggregates to its parent in the tree and there is also other costs involved in the waiting time decision according to topology changes and time-sensitive application. *Data aggregation load* is considered in this phase in order to maintain delay-sensitive data delivery. 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 the start-up time, an aggregating parent sensor node starts aggregating data from its own, if any, and from its children that have participated in aggregation. Later this aggregator node will forward the so far aggregated data to its parent. The amount of aggregated data is a function increasing in participating sensor nodes and decreasing in the waiting time. Moreover, sensor node within its communication range can exploit unavoidable overhearing or eavesdropping on neighbouring nodes' traffic to improve the selection of parent nodes and data aggregation. This feature is kept optional and application-specific in the proposed routing scheme as it can be enabled or disabled based on the application. Since this distributed parent selection process is performed dynamically whenever there is a packet to send, this approach can adaptively change the topology of aggregation according to different situations based on the aggregation load.

Route maintenance is the most important phase, which is performed using periodic beacons to handle link dynamics and disconnection failures and all valid routes are *reactively* kept on-demand available before any data packet transmissions. Hence, the routing tree is sustained and the neighbour routing tables are also kept updated to avoid relays with lower energy and unreliable links. To achieve reliable data packets 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 sensor nodes, their residual energy, depth or node *id*, and other helpful routing information. The rationale behind maintaining neighbour table is to proactively keep track of possible efficient routes to the base station and be able to order them on the basis of a joint metric favouring high-quality links, relays with good energy resources above predetermined threshold, and low number of hops. 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.

4.4 Avoidance of routing loops

During routing tree formation, specifically, in tree setup phase, a sensor node can only pick its parent in the same level or lower according to its communication range and routing metrics. Routing loops are prevented at the same level using a tiebreaker i.e., sensor node *id*. 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 loops in the whole routing tree, the parent selection of a sensor node is restricted to neighbours which are not farther away than its level. For instance, if a source sensor node and its parent candidate are in the same level, sensor node's *id* is used as a *tiebreaker* to prevent loop at this level. 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. In figure 2 shown earlier, sensor nodes can select parents from sensor nodes in the same level and one level downwards and no upward selection is allowed. As sensor node *id* is used as a *tiebreaker*, sensor nodes in the same level have an *ascending* ordering in the priority of being selected as parents, i.e., sensor node with larger *id* is selected as a parent for sensor nodes

with smaller *id*. Therefore, no loop can be formed within the same level. As a result, this prevents the routing scheme from creating loops within the entire network.

4.5 Neighbourhood participation policy

The high-level algorithm shown below describes how a sensor node selects its valid parent. To perform the algorithm, routing information can be easily acquired through periodic beaconing or packet overhearing to be maintained in the routing tables. While the information maintained in the routing tables is used for the *proactive* quick rerouting, the periodic broadcasting packets are also necessary for updating routing tables and the *reactive* routing to be used for route dynamics. The routing information required for the routing tree of the proposed algorithm is added into the original beacon packets' headers, so that sensor nodes can have the necessary neighbour information to modify the routing path up on request. In network start-up, the network is initially considered as fully identified and the values of route metrics are initially obtained in the routing table and ready for route maintenance.

High-Level Algorithm:

Initialization: network start-up

For Each Node

 If (ParentLossTime < WaitingTime) then

 For Beacon_recieved

 Update Route_infomation

 Send in next beacon

 End loop

 Else

 If (linkQuality & EnergyCapacity > Threshold) then

 If (ParentLevel <= NodeLevel) & (Parent_id > Node_id) then

 Set Parent

 End if

 End if

 End if

End loop

5. Experimental methodology

5.1 Overview

This section describes in details the indoor experimental testbed platform, and performance parameters used to evaluate the operation of the sensor network by means of the proposed routing protocol. 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 forwarding sensor nodes.

The wireless sensor testbed comprises a wireless platform of Mica2 a link layer of B-MAC (Polastre, 2004) in indoor channels. The proposed protocol is compared with the official TinyOS implementation of MintRoute collection protocol on Mica2 motes. In the conducted indoor experiments, all sensor nodes are homogeneous with fixed low transmission powers in each run, and commence with the same residual power capacity. Mica2 sensor nodes

(Crossbow, 2010) are equipped with Omni-directional whip antennas. On Mica2 sensor nodes, the standard TinyOS B-MAC MAC layer (Polastre, 2004) is used for CC1000 radio. B-MAC is a contention-based MAC protocols. Since the TinyOS-1.x version has several differences from its newer version TinyOS-2.x, the TinyOS-2.x version is not fully backward compatible with version TinyOS-1.x. Hence, the official stable release TinyOS-2.0.2 that supports different platforms is used for all indoor and outdoor experiments.

5.2 Implementation platform

To develop an understanding of sensor nodes' indoor communications performance, this section investigates the implementation challenges in the tiny wireless sensors by means of the proposed routing scheme. The implementation was done indoor using the low-power Mica2 (MPR400CB) wireless sensor network platform (Crossbow, 2010) with the component-based operating system TinyOS (TinyOS, 2004) which is written in an event-driven language called network embedded systems C-like language (nesC). The implementation is based on a real world testbed of wireless sensor nodes, specifically, the Berkeley's Mica2 motes which are popular due to their simple architecture, open source development and commercially available from Crossbow® Technology. UC Berkeley Mica2 Motes utilise a powerful Atmel® ATmega128L microcontroller and a frequency tuneable radio with extendable range. The UC Berkeley Mica2 Mote Module is a third generation tiny, wireless platform for smart sensors designed specifically for deeply embedded low power sensor networks. Table 1 reveals the specifications of a typical radio/processor platform Mica2 (MPR400CB) (Crossbow, 2010) which is powered by AA batteries. Mica2 is built with an 8-bit, 7.3828MHz Atmel® ATmega 128L processor, 128Kbytes of in-system program memory, 4Kbytes of in-system data memory, and 512Kbytes of external flash (serial) memory for measurements storage. Figure 3 shows the overall block diagram of Mica2 mote (Crossbow, 2010). A sensor node can be configured as a base station to route over standard serial port interface by attaching the interface board MIB520. The base station serves as the traffic sink.

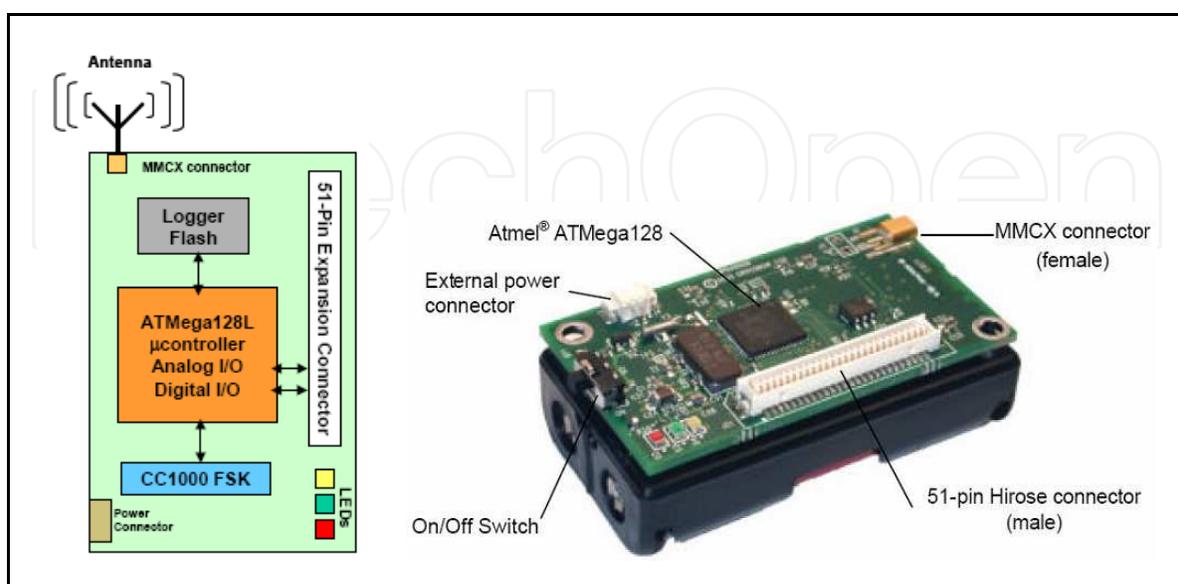


Fig. 3. Crossbow Mica2 868/916MHz Mote (MPR400CB) (Crossbow, 2010)

These resources seem unfit for computationally expensive or power-intensive operations. Explicit energy saving techniques are necessary to extend battery lifetime as much as possible. Communication is much more expensive than computation on wireless sensor devices. For instance, the Mica2 radio component when transmitting draws 30% more current than the CPU when it is active. 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.

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 with receive sensitivity of -98dBm
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

Table 1. Crossbow Mica2 mote (MPR400CB) specifications (Crossbow, 2010)

5.3 Experimental features of the deployed wireless platform

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 for WSN applications. MoteWorks™ (Crossbow, 2010) enables the development of custom sensor applications and is specifically optimised for low-power and battery-operated networks. MoteWorks™ is based on the open-source TinyOS operating system and provides reliable, ad-hoc mesh networking, over-the-air-programming capabilities, cross development tools, server middleware for enterprise network integration and client user interface for analysis and configuration. 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 (Crossbow, 2010).

Mica2 contains a processor and radio Platform (MPR400CB) which is based on the Atmel ATmega128L. The ATmega128L is a low-power microcontroller which runs MoteWorks™ 2.0 platform from its internal flash memory. A single processor board (MPR400CB) can be configured to run sensor application/processing and the network/radio communications stack 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 (Crossbow, 2010). Any Mica2 Mote can function as a base station when it is connected to a standard PC interface or gateway board. 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. MIB520 supports USB for the Mica2 Motes for both communication and in-system programming. Finally, Mica2 Motes can be integrated with sensing board or data acquisition board that supports a variety of sensor modalities that support environmental monitoring (e.g., Ambient light, humidity, temperature, 2-axis accelerometer, and barometric pressure) for Mica2 with built-in sensors and an optional GPS (Crossbow, 2010).

5.4 Programming environment (TinyOS)

The firmware of sensor nodes and the base station is based on TinyOS (TinyOS, 2004) which is the de-facto operating system and programming environments for sensor motes. The experimental implementations use various API and libraries provided by TinyOS-2.0.2. TinyOS-2.0.2 is implemented using the nesC-1.2.8 (networked embedded systems-C) event-programming language. Typically, TinyOS is an open source component-based operating system specifically designed for embedded WSNs, which was initially released in 2000 under Berkeley Software Distribution (BSD) licenses. It is supported by nesC's component-based programming model. TinyOS applications are a collection of components, where each component has three computational abstractions: *commands*, *events* and *tasks*. TinyOS deals with limited resources of severe energy constraints, very small and efficient code in memory storage of kilobytes, and CPU speed of less 10 MHz. The nesC is a static programming language where applications are made by writing and assembling components which reduces the used memory footprint. The nesC is an extension of C language, a new event-driven programming environment developed for networked embedded systems such as sensor networks. The nesC supports a programming model that integrates reactivity to environment, concurrency and communication. TinyOS defines a number of concepts that are expressed in nesC. First, nesC applications are built out of components with well defined bidirectional interfaces. Second, nesC defines a concurrency model, based on tasks and hardware event handlers and detects data races at compile time. *The nesC application* consists of one or more components linked together to form an executable. *Components* are the basic building blocks for nesC applications and classified as provides and uses interfaces components. A provide interface is a set of methods calls the upper layers while uses interface is a set of methods calls the lower layer components. An interface declares a set of functions called *commands* that the interface provider must implement, and another set of functions called *events* that the interface user must implement. There are two types of components in nesC *modules and configurations*: A *Module* is a nesC component consisting of application code written in a C-like syntax. A *Configuration* is a component that wires 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.

6. Underlying layers

6.1 Physical layer

At the Physical Layer, Mica2 mote uses a low powered radio “Chipcon CC1000 RF transceiver” 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 (Crossbow, 2010). Mica2 (MPR400CB) has a digitally programmable/tuneable output radio power levels ranges 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. CC1000 radio uses Frequency Shift Keying (FSK) modulation with an effective data rate or throughput of 38.4Kbps. 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 (Crossbow, 2010).

6.2 Mac layer

TinyOS operating system 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, e.g., B-MAC with low-power listening (LPL) scheme, the default TinyOS MAC protocol developed at the UC Berkeley (Polastre, 2004). TinyOS CC1000 has 128bytes maximum MAC frame size and employs Frequency Shift Keying (FSK) modulation Scheme. Due to the highly dynamic and untethered nature of WSNs, the inherent advantages of contention-based protocols, i.e., B-MAC (Polastre, 2004), 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 totally 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 wakeup during the preamble transmission time. It also employs an adaptive preamble sampling scheme to reduce 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. Therefore, in addition to the receiver, all the other neighbours of the sender will receive the packet, even the packet is not addressed to them, e.g., 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 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. Finally, B-MAC is aware to the protocols that run above it and offers control to the protocols that sit on top of it, allowing to 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 sensor node to overhear packets transmitted by its neighbours; this allows high layer network

protocols, i.e., routing protocols, to employ snooping for the sake of link quality estimation, and in-network processing and 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 very important MAC feature for time-sensitive data aggregation provided by the proposed routing scheme. The method of adaptation is an application-dependent. B-MAC does not perform link-level retransmission or hidden terminal 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 (Polastre, 2004).

Moreover, B-MAC uses the energy detect indicator as a carrier sense mechanism 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 radio, which provides low-level access to the individual bits received by the radio. Hence, B-MAC can generate long preambles with CC1000 radio but the recommended preamble length in B-MAC is 100ms, which is used in the deployed WSN experiment. Even though the official version of B-MAC suffers from the inevitable overhearing, and the long preamble dominates the energy usage, the modified version of B-MAC, provided by TinyOS, 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 (consistently delivers 98.5% of packets), latency, and for most cases energy consumption (Polastre, 2004).

8. Experimental performance evaluation

8.1 Performance metrics and observed entities

The real WSN is evaluated considering different performance metrics that are observed by the base station, relayed to the attached laptop, and saved in log files for later analysis using Matlab scripts. Particularly, the results show how the *link quality measurements* in the considered scenarios and the network behaviour was characterized in terms of: *packet delivery performance* to assess the significance of wireless link reliability on packet loss probability; *average dissipated energy* to figure out how sensor nodes deplete their energy to achieve multihop communication; and *average end-to-end delay* to evaluate the multihop data aggregation and hop count effect on data delivery time.

Received Signal Strength Indicator (RSSI): RSSI represents the amount of signal energy received by the sensor node. It can be measured by either Chipcon radio chips, CC100 on Mica2. RSSI readings 1000 have a range from -100 dBm to 0 dBm and the maximum error (accuracy) is 6 dBm. It is calculated over 8 symbol periods.

Packet Delivery Performance: One of the basic metrics used for evaluating packet delivery performance and to measure link quality is *Packet Reception ratio (PRR)* (also known as *packet delivery fraction*) which is the percentage of successfully received packets to packets transmitted. In other words, the PRR is the ratio of the total number of packets received by

the base station that successfully passes the Cyclic Redundancy Check (CRC) to the total number of packets originally sent (considered) by the source sensor nodes as expressed in equation 1.

$$PRR = \frac{\text{Successfully received packets}}{\text{Sent packets}} \times 100 \quad (1)$$

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. During sensor node's operation, the estimation of average dissipated energy is computed per sensor node using equation 2 and used in the cost function in favour of the most efficient route. Where V_{batt} is the battery voltage of the sensor mote, and I_{drawn} is the current consumed by the mote system. The time spent per *CPU cycle* depends on the type of the mote system, for example, (1/7.3828) μ s for Mica2. The number of *CPU cycles* spent during mote's tasks is counted based on the *average dissipated energy profile* of mote system.

$$\text{Energy} = V_{batt} \times I_{Drawn} \times \text{CycleTime} \times \text{Cycles Count} \quad (2)$$

Average End-to-End Delay: 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.

Cyclic Redundancy Check (CRC) field: CRC indicates whether the packet received pass the CRC checking as TinyOS has a CRC field in its radio packet. Chipcon radio chip (CC1000 or CC2420) has an automatic CRC checking capability and the CRC scheme used in is CRC-16.

8.2 Experimental testbed

In order to evaluate the suitability of the proposed routing scheme for indoor WSN, a set of indoor experiments are run on the testbed network for a particular topology. This small indoor testbed consists of 20 Mica2 motes deployed on paved floor inside roofed showground-like building as shown in figure 4. The surrounding conditions and Mica2's antenna orientation have a significant impact on radio performance. To minimize this effect, for a given topology, testing scenarios were performed many times and the average of these runs was recorded. Data packets were set with fixed size to maintain the same transmission and receiving time for each data packet. The motes are organised and the radio power is configured such that the maximum network diameter is three to five hops. While the operating radio frequency is digitally programmable, external hardware attached to the CC1000 is set to operate in one frequency band. That is, a board set to operate in 868/916MHz bands will not operate at the 433MHz band. The operating frequency range of a Mica2 mote is labelled on the mote. Mica2 (MPR400CB) motes are built to operate in the 868/916MHz bands, i.e., 868-870MHz (up to 4 channels) and 902-928MHz (up to 54 channels). Thus, these Mica2 motes are unlikely to create interfere particularly with 802.11 devices that operate in 2.4GHz ISM band. The actual number of possible channels is higher for all the Mica2 motes (Crossbow, 2010). However, the adjacent channel spacing is to be at least 0.5MHz to avoid adjacent channel interference thereby reducing the number of available channels. The sensor node that acts as the base station is connected to MIB520

programmer, and a RS-232 to USB converter is used to connect the laptop and MIB520 to collect messages sent within the network. Sensor nodes are placed indoor in the way they can only communicate with adjacent neighbours with low transmission powers; however, there is still a probability of opportunistic connections for longer distances (Crossbow, 2010).

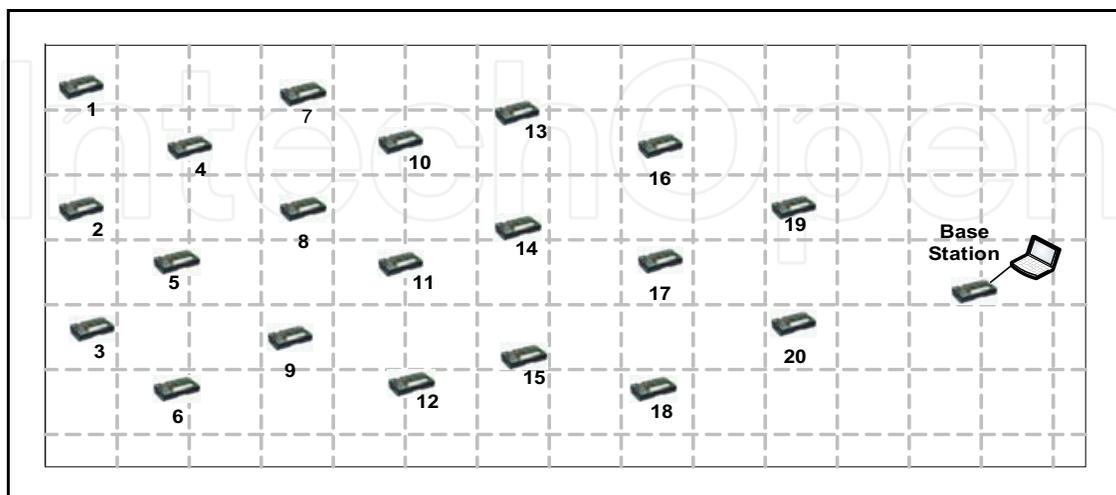


Fig. 4. Indoor testbed topology with perimeter base station

The source sensor nodes broadcast generated data packets towards the base station. The base station acts as a bridging device between sensor nodes 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 various metrics and measurements such as RSSI, CRC, time-stamp, sequence number, and appending them to each received packet. Then, the packet logger/parser program in the laptop processes these received packets, and save them to a log file for thorough analysis using Matlab scripts. TinyOS-2.0.2 is used as the Mica2 CC1000 radio library for earlier TinyOS-1.x releases doesn't support the time-stamping interface. If the local clocks on sensor nodes had the exact frequency 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 $40\mu\text{s}$ per second. 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.

To limit the radios transmission range, the motes were placed directly on the floor and to determine the distance which provides a reliable delivery performance but minimises the possibility of a mote transmitting further than to adjacent motes; motes were placed at varied distance and the delivery rate recorded. Then, the distance that provided the most reliable packet delivery performance, e.g., PRR is greater than %90, is used. In indoor environment, where space is more limited, the transmitting power of the sensor nodes is initially set to be at the lowest output power level of -20dBm ($10\mu\text{W}$) and then increased to -15 , -10 , -5 and 0 dBm and variable in-between spaces are been allowed to provide a reliable delivery performance within 1, 2, or 3 hops and to minimise opportunistic reception. However, it is still likely that some reliable long distance links will form. The Chipcon CC1000 can select a minimum output power level using a variable power radio such that

messages are transmitted successfully to their destination, possibly using less power than the default setting. With variable separating spaces between adjacent nodes, adjacent nodes are within the transmission range of each other to allow multi-hop communications. As transmission distance has to be exceeded to make multi-hop more energy efficient than direct transmission. While the network is operating, the source nodes are transmitting packets periodically; the number of packets received by the base station is recorded for each run and the average of these runs is taken. The proposed routing scheme sets up a spanning tree towards the base station and is configured to operate with packet sending rate of one packet apiece 100 ms per source sensor node.

Due to the jitter in the testbed network, transmission start times vary with a mean of few milliseconds. Further, obtaining reliable signal strength measurements for link state indicator can take up to 7ms as this is not a controllable parameter in the CC1000 radios. Therefore, the times at which the signal strength is measured need to be carefully chosen at the receiver to ensure any intended collision. Mica2 motes use CSMA-based MAC protocol, i.e., TinyOS B-MAC that perform 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. This is to avoid MAC layer overhead and to focus on the routing layer performance. Signal strength measurements are taken in the middle of long packet transmission periods so substantial jitter can be tolerated. In Mica2 CC1000 radio implementation; the data path does not implement software Manchester encoding but it is provided by the CC1000 hardware. The data path interfaces to the radio at the byte level. The CC1000 hardware takes care of bit-synchronization and clocking. The bytes coming off the radio, however, are not necessarily aligned with the data packet. The data path determines and executes the necessary bit shifts on the incoming stream. The CRC computations are running on the received data packet at the base station.

Finally, Mica2 motes are labelled with numbers and placed in predetermined locations on the ground. The base station mote is placed on the MIB520 Mote Interface Board which powered by an AC power supply and attached to a laptop to collect the data of interest. The residual battery capacity is measured and calculated instantaneously and fed into the software in order to be used in the routing cost function in favour of the most energy efficient route. For the initial set of the experiments, all sensor nodes begin with equal battery power levels, roughly 3Volts. The rates at which the data packets are transferred tracked, and the amount of energy required to get the data packets to the base station is monitored.

9. Results and empirical observations

The results obtained experimentally in this section have been worked out based on a real sensor network field which is more important and effective than pure simulation-based approach. Also performance analysis of scenarios in areal sensor field is valuable, satisfactory and possesses academic and practical values on WSN field. Observations and results obtained from the experimental testing are presented and thoroughly analysed using Matlab scripts. This empirical research in the context of WSNs has given a good understanding of the complex irregular behaviour of low-power wireless links in WSNs.

Although the WSN is positioned in indoor environment with very limited ambient noise, multihop WSN has several challenges which represent in: the wireless link limits the

number of data packets that can be in flight concurrently from source to destination due to unreliable wireless transmission at each hop and MAC protocol contention problems from hidden nodes and/or exposed nodes; the physical-layer properties that may constrain the throughput achievable over a multihop route; end-to-end reliable delivery of data requires each packet to traverse one or more intermediate hops from the source sensor node towards the base station.

9.1 Link reliability

The RSSI readings are measured at the receiver sensor node based on forward channel. The RSSI is measured indoor within different distances and mote's antenna orientations, then the averaged results are recorded. Figure 5 (a) shows the overall tendency of RSSI measurements as a function of transmission distance and mote/antenna orientation at the highest transmission power. As an overall, it has been observed that in the indoor environment the wireless link reliability estimations based on RSSI doesn't vary significantly with sensor node placement or density within the same space as the hardware-based RSSI provided by CC1000 radio may be inadequate for predicting the link reliability and connectivity. However, different deployment topologies and node density have an observable effect on the overall link reliability of the sensor nodes.

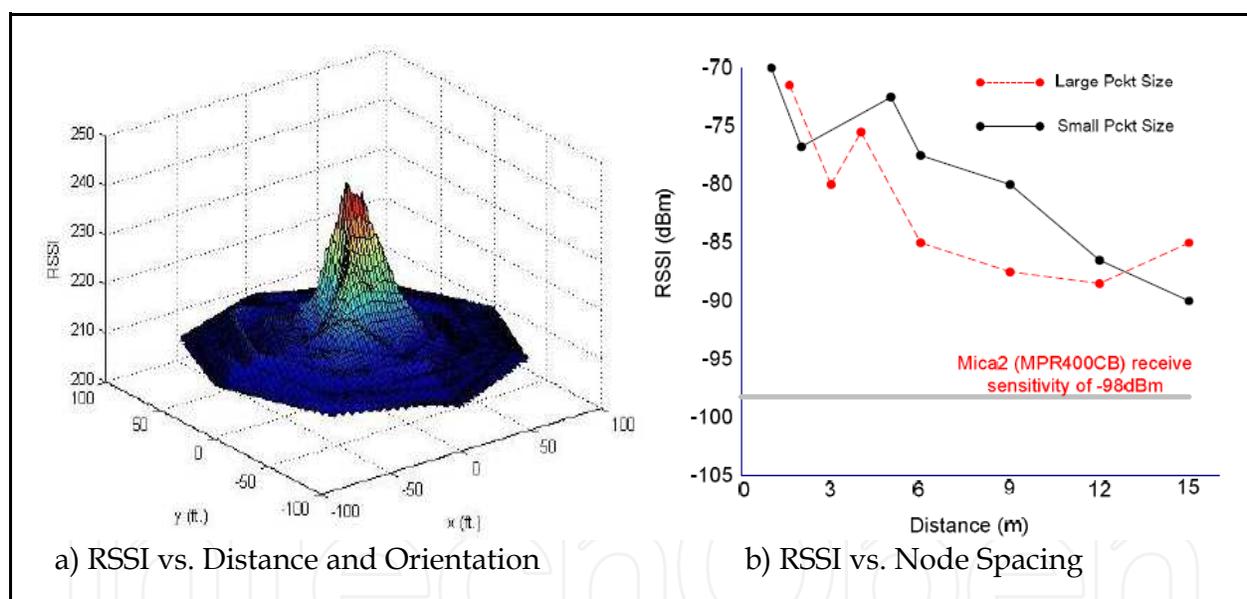


Fig. 5. RSSI readings measured indoor

The RSSI values decrease as the distance between sensor nodes increase with various packet sizes. Although the indoor experiment is performed with stationary sensor nodes, the RSSI values have a tendency to fluctuate as shown in Figure 5 (b) where the values presented are average values from the packets that are received and do not imply a steady link with various packet sizes. It was observed that within short distances of few meters, the RSSI of small size packets were generally stronger than with the larger size packets with a small packet loss. For longer distances, longer than 13 meters, the larger size packets tend to have stronger RSSI readings. However, the RSSI readings follow an exponential diminishing while the successful packet reception ratio is high; after approximately 20 meters, the signal is noisier and its strength deteriorates to the minimum sensitivity of the CC100 transceiver.

Mica2 (MPR400CB) radio has a receive sensitivity of -98dBm. This extreme sensitivity can be interfered by another oscillator from an adjacent Mica2 node. A distance of at least 65cm should be maintained between adjacent mica2 nodes to avoid local oscillator interference. However, at low transmission power levels, the sensor nodes are still able to communicate with each other.

Using CC1000 RF chip's RSSI independently may not be adequate for predicting the link quality for reliable connectivity. Therefore, for better understanding of low-power wireless link reliability, a newer hardware-based link quality indicator (known as LQI) is used with RSSI for improved link quality estimations in the next experimental outdoor deployment using TelosB's Radio CC2420 that supports LQI measurements as LQI is not supported by Mica2's CC1000 radio.

The experience with the experimental work done has revealed several underlying issues that stem from the properties of the reliability-oriented cost-based routing layers, specifically, MintRoute combined with the resource constraints of the mote platform. Those issues include energy efficiency, long-term link estimations, count-to-infinity and routing loops. The proposed routing scheme considers the suitable countermeasures to address these issues. During the parent selection process, MintRoute uses the link quality estimations with the surrounding neighbours together with cumulated route quality estimation to the base station, and the hop count metric included in the route updates is completely ignored. 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 MT metric, which results suboptimal route that could be in some direction faraway 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 problem, e.g., routing hole. The natural occurrence of suboptimal routes is taken into account by the proposed scheme when performing parent selection by adopting, for instance, the least number of hops as a tier breaker; this advantage does not apply for MintRoute, also the proposed protocol is further enhanced in chapter four to avoid routing holes using large-scale simulations. In MintRoute, only next packets transmission may probably reduce the already perceived link quality, which makes the selective forwarder look less attractive. In other words, 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. To be selected, a sensor node must have both a good send and receive quality. To get a high send quality, the high value must be included in a route update sent by the relay sensor node that caused a routing hole. To get a high receive value, this relay sensor node will have to keep sending packets to prevent the decaying of the receive value by the sensor node. The number of packets that might be lost also lowers the receive quality. Figure 6 (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 suboptimal 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. In figure 6 (b), sensor nodes 11, 13 and 16 select node 14 as their parent with best ink quality using suboptimal routes that purely based on link quality estimations using MT metric. This leads MintRoute to cause a routing hole to the downstream nodes at node 14.

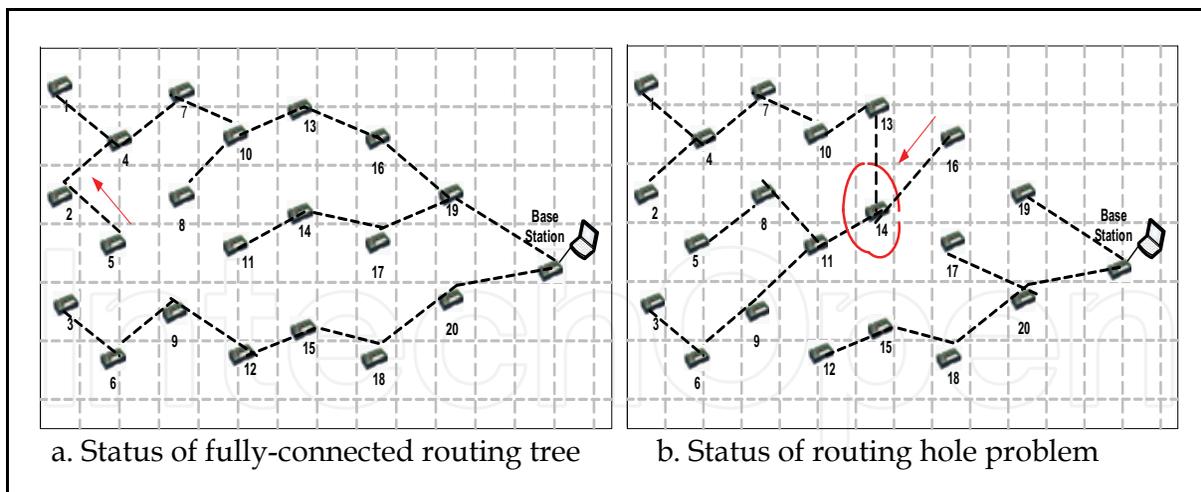


Fig. 6. Routing in minroute protocol

9.2 Average dissipated energy

In the indoor environment, the transmission power of sensor nodes is kept to the lower levels in order to keep the power consumption minimised as possible but the transmission power is increased gradually to maintain reliable multihop connectivity within a limited indoor space. It can be observed from Figure 7 that the average dissipated power by the sensor nodes for transmission and receiving during their operation instantaneously increases faster in MintRoute than in the proposed protocol as the in-between spacing between nodes increases. In terms of energy dissipation cost, since the rate of route message exchanges is low in MintRoute, its energy dissipation in can be minimised. However, MintRoute is more expensive than the proposed protocol at higher message exchange rates and spends a longer time to convey the topological changes to the entire network; during this time, most forwarded packets are routed through optimal paths based on link quality, this leads to additional energy consumption and thus offsets the benefit of energy balancing.

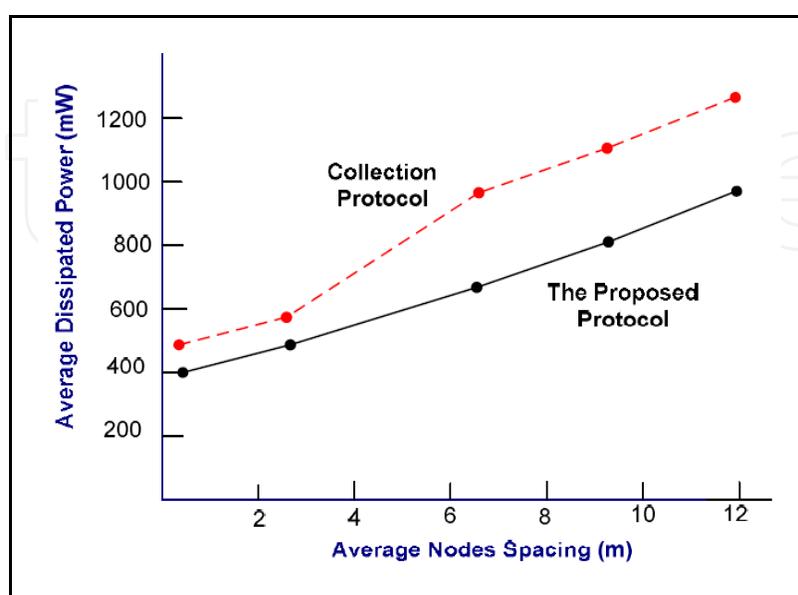


Fig. 7. Average dissipated power vs. inter-nodes spacing

Hence, the proposed routing scheme considers the acceleration of route message exchange rate for reactively adapting the topological changes. Although MintRoute protocol balances the traffic load with unintentional parent switching based on its default Minimum Transmissions (MT) metric, MintRoute protocol does not clearly apply a metric that considers workload balancing in its routing scheme.

The total network-wide energy expenditure is due to: the *parent selection process*; packet transmission; packet overhearing/receiving; failed packet reception, and updating routing table. In B-MAC, overhearing a packet consumes the same energy as receiving a packet because B-MAC requires sensor nodes to receive the whole packet before discarding failed ones. Failed packets reception that may result from packet collision or link failure requires packet retransmission to be successfully received at the destined recipient. Figure 8 shows the total dissipated energy consumption required for retransmissions due to packet loss or link failures. Since the proposed routing scheme has the feature of employing the implicit acknowledgements strategy for less communication overhead, packet transmission is less than that in MintRoute. The fewer packets sent results the 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 the proposed protocol than in MintRoute even though the proposed protocol requires only 0.48% of computation overhead for parent selection overhead. On average, the proposed protocol saves around 65% on energy consumption for communication less than MintRoute.

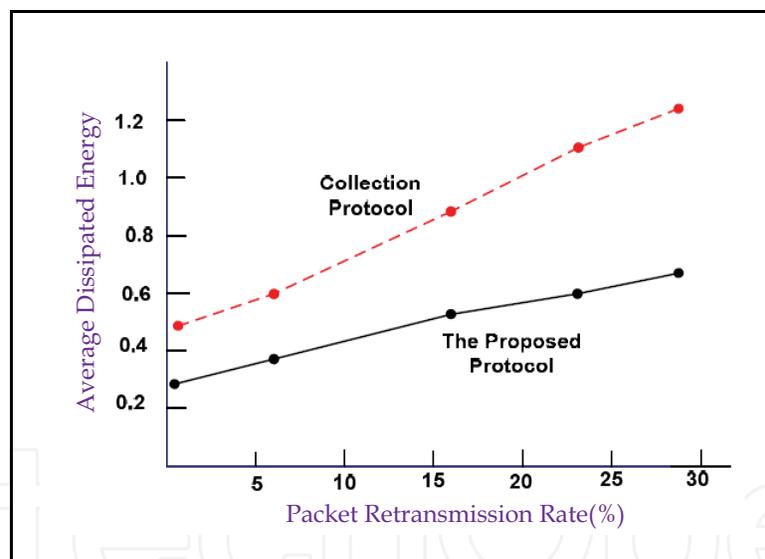


Fig. 8. Average dissipated energy vs. packet retransmissions rate

9.3 Packet delivery performance

In multihop WSN, the achieved throughput may be lower than the maximum achievable throughput for several reasons such as CSMA-based MAC protocol backoff waiting times at each wireless sensor node and packet retransmissions after detected collisions or packet loss. At the physical layer, indoor environment has unconstructive effect on packet delivery performance, especially when a higher transmission power is used, conceivably due to the effect of Multipath Rayleigh Fading Channel (MRFC). Besides that, Manchester coding has much more overhead and also has a negative effect on packet delivery performance in multihop settings, as shown per node transmission and reception overhead in Figure 9. In

addition, high signal strength is a necessary but not a sufficient condition for good packet reception ratio. Packet error cannot be distinguished if it was due to physical layer packet error or due to MAC layer collisions. At the MAC layer, about 50-75% of the energy is spent for repairing lost transmissions and the established links has over 50% link asymmetry problem in packet delivery ratio due to surrounding environmental conditions, and mote and antenna orientation.

Typically, there are many different ways for a packet to be corrupted in wireless communication and thereby packet is to be considered lost at the destined recipient. Firstly, a packet may be lost due to errors in the wireless transmission which results in an unsuccessful CRC or not received at all. The second possibility is that two sensor nodes send their packets at times so that the transmissions overlap in time which cases packet collision due to the hidden node problem; thereby resulting in two lost packets. Finally, a packet may be lost before it has been transmitted if a sensor node senses a channel as busy a maximum number of times. In this situation, the sensor node will simply discard the packet and move on to the next packet. As a result, predicting the source of the packet loss is complicated and unclear in terms of the hardware. In addition, previous experimental studies have indicated that radio connectivity is imperfect and non-uniform, even in ideal settings. Furthermore, a packet loss due to link failures is the most common in WSN channels. When data aggregation is enabled, a single link failure will result in an sub-trees of aggregated values being lost, The influence is significant If the failure is close to the base station.

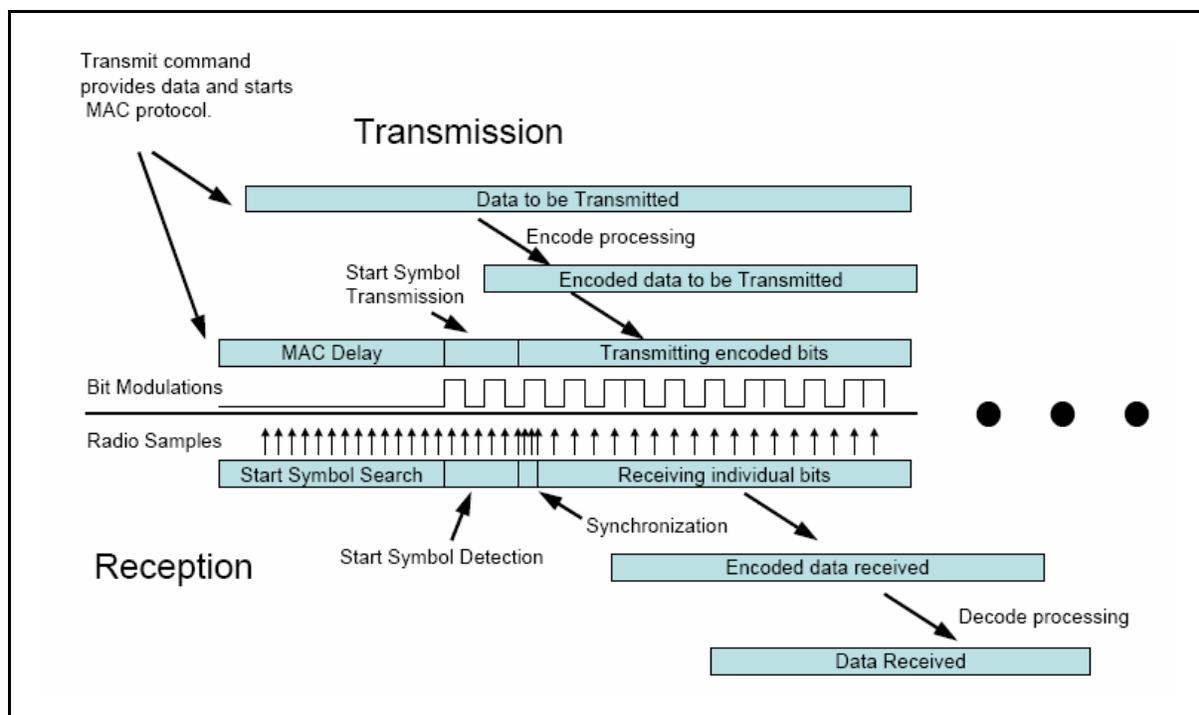


Fig. 9. Wireless communication phases per relay

9.4 Average end-to-end delay

Average end-to-end delivery delay is evaluated in terms of packet transfer rate between the transmitter and the receiver. The transmission rate at the source sensor node has been programmed prior to the experiment and the average of multiple runs with different

sending rates is considered. Figure 10 demonstrates how the proposed routing scheme outperforms MintRoute as the packets transfer rate changes through few hops from the source sensor node to the base station. In addition, figure 11 shows how that the packet reception rate for both protocols decreases as the number of hops increases by changing transmission range of the sensor nodes for a constant transmission rate of 7Kbitsps. MintRoute performs poorly in the deployed testbed topology due to the limitation of its route searching and maintenance phases compared to the proposed routing protocol.

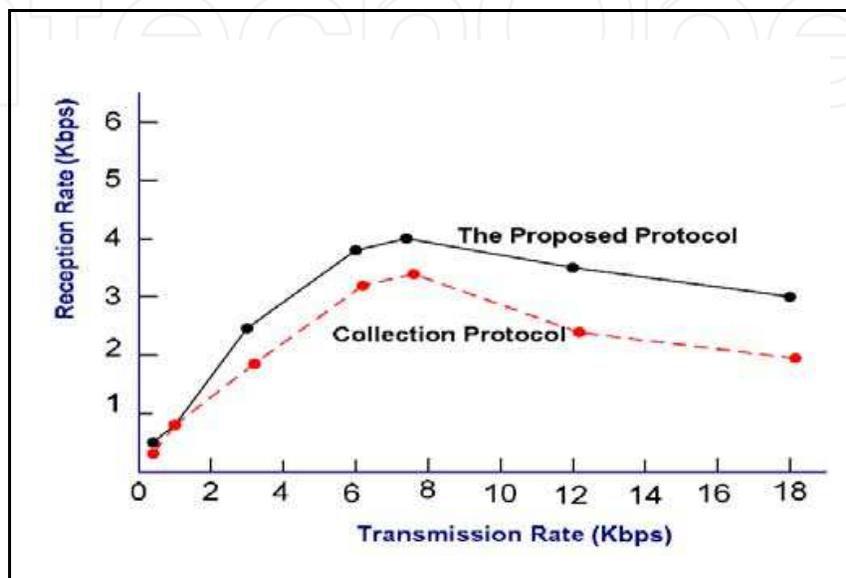


Fig. 10. Reception rate vs. transmission rate

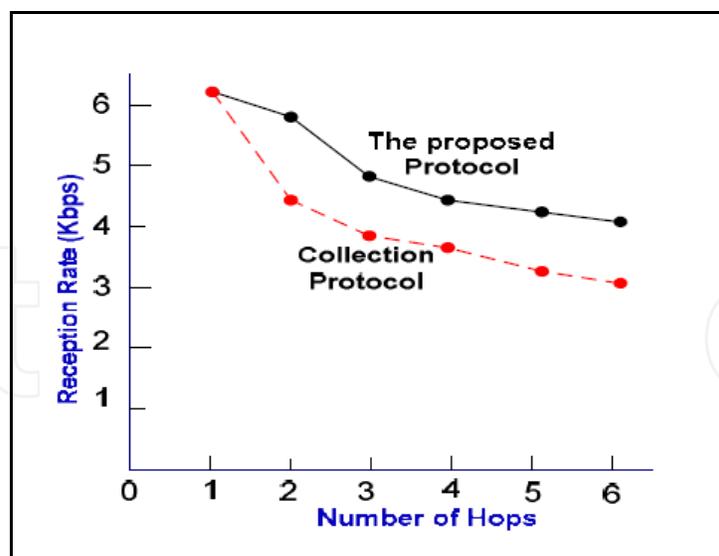


Fig. 11. Reception rate vs. number of traversed hops

10. Conclusion and future work

Some of these observations are well-known phenomena in low power wireless communications. These experiments allow to understanding the irregular behaviour of

wireless channel such as asymmetry. A series of experiments were carried with different node spacing. In link asymmetry, there is a noticeable variance in the corresponding packet delivery rate because of a fluctuation in the RSSI below the sensitivity threshold of the receiver due to interference, fading, or shadowing state or 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 rate 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.

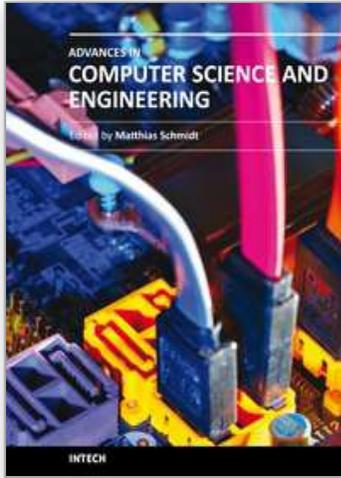
Although the indoor experiment was performed with stationary sensor nodes, the Received Signal Strength Indicator (RSSI) values have a tendency to fluctuate, and do not imply a steady link with various packet sizes. RSSI could yield a routing tree with additional number of hops and extra messages being sent and overheard at the same time as lower transmission power does not mean that the link quality is that such poor. As a result of an irregular low power radio neighborhood, the packet transfer rate is probabilistic and time-varying. Packet transfer rate also changes accordingly with number of hops passed toward the base station. In a multihop sensor network, if the number of hops increases the transfer or reception rate decreases for constant transmission rate due to packet process per relay (e.g., encoding and/or decoding) and wireless signal propagation delay. Moreover, since the radio communication cost is a function of the distance transmitted, it can be observed that the average power dissipated by the sensor nodes during their operation increases as the inter-nodes spacing increases. Since the nodes do not constantly communicate, it is optimal to reduce the time the radio spends in active mode and decreasing radio duty cycle is invaluable as an energy saving technique. However, the ability to use the sleep or idle modes depends on network and application behaviour and reducing the cost of each transmission by means of data aggregation is equally important to minimise the current used to power an active radio. Losing packets before reaching the base station not only wastes energy and network resources, but also degrades the quality of application. Another subtle issue is fairness where sensor nodes far-away from the base station are likely to have a lower end-to-end success rate than sensor nodes that are closer. The fall down of success rate by hops or distance verifies this behaviour. Finally, the performance achieved in the real environment is heavily affected by the number of hops that a packet needs to travel to reach the destination and directly affected by the surrounding environment. Finally, While the experiments conducted here have highlighted the substantial performance gains of the proposed scheme, more detailed experiments are needed under different topologies using the new generation of sensor nodes such as IRIS 2.4GHz, TelosB that use Chipcon CC2420 and are IEEE802.15.4 compliant. In order to confirm the experiments, analytical and simulation results are also derived. Comparisons using simulations will be addressed against existing state-of-the-art routing protocols for WSNs.

11. References

- Burri, N., Rickenbach, P. & Wattenhofer, R. (2007). Dozer: ultra-low power data gathering in sensor networks, *Proceedings of the 6th International Conference on Information Processing in Sensor Networks (IPSN'07)*, pp. 450–459, New York, NY, USA, 2007.
- Crossbow Technology, Inc. [Online]. Available: <http://www.xbow.com> (Accessed Sep. '10).
- Daabaj, K., (2010). Energy-Aware Reliability-Oriented Scheme to Deliver Time-Sensitive Data in Sensor Networks, *Proceedings of the IEEE International Conference on Service-*

- Oriented Computing and Applications (SOCA'10)*, December 13-15, 2010, Perth, WA, Australia.
- Gnawali, O., Fonseca, R., Jamieson, K., Moss, D. & Levis, P. Collection Tree Protocol. *Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems (SenSys'09)*, Berkeley, California, USA, 2009.
- ISI, the Network Simulator. ns-2. [Online]. Available: <http://www.isi.edu/nsnam/ns/index.html>
- Omnetpp, OMNeT++ Simulation Framework. [Online]. Available: <http://www.omnetpp.org>
- Polastre, J., Hill, J. & Culler, D. (2004). Versatile Low-Power Media Access for Wireless Sensor Networks, *Proceedings of the Second ACM Conference on Embedded Networked Sensor Systems (SenSys'04)*, pp. 95-107, November 2004, Baltimore, MD, USA.
- Rahul, S., Jan R. (2002). Energy Aware Routing for Low Energy Ad Hoc Sensor Networks, *Proceedings of IEEE Wireless Communications and Networking Conference (WCNC'02)*, pp. 350-355, Orlando, Florida, USA, March 2002.
- TinyOS. (2004). MultihopLQI collection protocol, [Online]. Available: <http://www.tinyos.net/tinyos-1.x/tos/lib/MultiHopLQI/>
- Werner-Allen, G., Swieskowski, P. & M. Welsh. (2005). Motelab: A wireless sensor network testbed, *Proceedings of the Fourth International Conference on Information Processing in Sensor Networks (IPSN'05)*, Special Track on Platform Tools and Design Methods for Network Embedded Sensors (SPOTS), 2005.
- Woo, A., Tong, T. & Culler, D., Taming the Underlying Challenges of Reliable Multihop Routing in Sensor Networks. *Proceedings of the 1st International Conference on Embedded Networked Sensor Systems (SenSys'03)*, Los Angeles, CA, USA, Nov. 2003.

IntechOpen



Advances in Computer Science and Engineering

Edited by Dr. Matthias Schmidt

ISBN 978-953-307-173-2

Hard cover, 462 pages

Publisher InTech

Published online 22, March, 2011

Published in print edition March, 2011

The book *Advances in Computer Science and Engineering* constitutes the revised selection of 23 chapters written by scientists and researchers from all over the world. The chapters cover topics in the scientific fields of Applied Computing Techniques, Innovations in Mechanical Engineering, Electrical Engineering and Applications and Advances in Applied Modeling.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Khaled Daabaj and Shubat Ahmeda (2011). Real-Time Cross-Layer Routing Protocol for Ad Hoc Wireless Sensor Networks, *Advances in Computer Science and Engineering*, Dr. Matthias Schmidt (Ed.), ISBN: 978-953-307-173-2, InTech, Available from: <http://www.intechopen.com/books/advances-in-computer-science-and-engineering/real-time-cross-layer-routing-protocol-for-ad-hoc-wireless-sensor-networks>

INTECH
open science | open minds

InTech Europe

University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China

Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

© 2011 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the [Creative Commons Attribution-NonCommercial-ShareAlike-3.0 License](#), which permits use, distribution and reproduction for non-commercial purposes, provided the original is properly cited and derivative works building on this content are distributed under the same license.

IntechOpen

IntechOpen