IM2PR: Interference-Minimized MultiPath Routing Protocol for Wireless Sensor Networks

Marjan Radi[1,2,*], **Behnam Dezfouli**[1,2], **K. A. Bakar**[1], **S. A. Razak**[1], **Tan Hwee-Pink**[2]

[1] *Department of Computer Science, Faculty of Computing, Universiti Teknologi Malaysia (UTM), Malaysia*
[2] *Networking Protocols Department, Institute for Infocomm Research (I²R), A*STAR, Singapore*

Abstract

With respect to the inherent advantages of multipath routing, nowadays multipath routing is known as an efficient mechanism to provide even network resource utilization and efficient data transmission in different networks. In this context, several multipath routing protocols have been developed over the past years. However, due to the time-varying characteristics of low-power wireless communications and broadcast nature of radio channel, performance benefits of traffic distribution over multiple paths in wireless sensor networks are less obvious. Motivated by the drawbacks of the existing multipath routing protocols, this paper presents an Interference-Minimized MultiPath Routing protocol (IM2PR) which aims to discover a sufficient number of minimum interfering paths with high data transmission quality between each event area and sink node in order to provide efficient event data packet forwarding in event-driven wireless sensor networks. Extensive performance evaluations show that IM2PR presents improvements over the Micro Sensor Multipath Routing Protocol (MSMRP) and Energy-Efficient data Routing Protocol (EERP) as follows: 50% and 70% in term of packet reception ratio at the sink, 44% and 80% in term of goodput, 33% and 40% in term of packet delivery latency, 40% and 57% in term of energy consumption, 50% and 60% in term of packet delivery overhead.

Keywords: Wireless Sensor Networks, Multipath Routing, Interference, Event-Driven

1. Introduction

According to the event-driven nature of wireless sensor networks, the main responsibility of sensor nodes in different applications is to forward the sensed data from a target area towards the sink node. In this regard, designing efficient routing protocols to establish high-quality multi-hop paths from each event area towards the sink is one of the most important issues in developing wireless sensor networks [14, 17, 29, 37]. Over the past decade, numerous multipath routing protocols have been developed for wireless sensor and ad hoc networks [24, 31]. Due to the broadcast nature of shared wireless channel and unreliability of low-power radio communications, the main purpose of developing most of the existing multipath routing protocols was to support fault tolerance and reliable data delivery [1, 24, 31]. While, concurrent multipath routing for even network resource utilization has received less attention from the research community. The reason is that the wireless interfere which causes by simultaneous utilization of multiple nearby paths, impedes the performance benefits that can be achieved through distributing network traffic over several paths. In fact, in wireless networks lower number of non-interference paths demonstrate better performance compared to the more number of interfering paths [12, 28]. However, most of the proposed multipath routing protocols for wireless network have not considered this issue in their route construction mechanism or constructing several paths with low interference imposes a high overhead to the network [24, 31].

The inter-path interference problem which is the result of concurrent data transmission over adjacent paths is known as the *route coupling* effect. There are a few works on measuring the wireless interference level between different paths for path selection in multipath routing protocols. *Correlation factor* is one of the proposed metrics for this purpose [36]. The correlation factor of every path pair is defined as the number of links which are connecting the paths to each other. Authors in [20] introduce another metric, which is named *route coupling factor* to measure the amount of interference among multiple paths. The route coupling factor of a set of paths is defined as the average number of nodes along the paths which cannot send or receive any packet during data transmission over a link in one of the paths. Since both of these factors require the general network connectivity graph for identifying a set of minimum interfering paths between every node pair, calculation of these metrics in large-scale wireless sensor networks causes a high computational overhead [24, 31]. Alternatively, other techniques such as location-aware routing [34, 40], directed antenna [27], multi-channel data transmission [5, 30, 38], and specific MAC layer protocols [8, 33] have been utilized to avoid from route coupling problem. However, as these approaches require special hardware equipments and particular mechanisms at different layers of the network protocol stack, they are unsuitable for resource constrained wireless sensor networks. For

*Corresponding author:
Email address: radi@ieee.org (**Marjan Radi**)

Wireless Netw (2014) 20:1807–1823
DOI: 10.1007/s11276-014-0710-5

instance supporting multi-channel communications requires a specific MAC layer mechanism that supports channel switching.

In addition to the wireless interference, time-varying properties of low-power wireless communications impose more challenges in supporting efficient packet transmission in wireless sensor networks [15, 16, 25]. In fact, route selection without considering the data transmission quality of wireless links may result in the construction of the paths with low-quality links which in turn increases the number of packet transmissions required for successful packet delivery to the sink [2]. Moreover, in the situation that the link layer provides a limited number of transmission attempts for packet delivery over individual links, order of the links along the paths highly influences their respective data transmission cost. The reason is that packet drops due to the limited number of transmissions at the link layer on the links near the destination are very costly, as the packets have traversed several links before they are dropped [13].

With respect to the aforementioned issues the contributions of this paper are as follows:

Firstly, we propose an Interference-Minimized MultiPath Routing protocol (IM2PR) which aims to discover a sufficient number of minimum interfering paths with high data transmission quality between each event area and sink node in order to provide efficient event data packet forwarding in event-driven wireless sensor networks. IM2PR takes into account the maximum number of offered link layer transmission attempts at individual links through considering the relative position of the links along the paths and their respective packet delivery probability in the presence of active interfering links.

Secondly, we propose a load distribution algorithm in conjunction with the proposed multipath routing protocol in order to regulate the traffic rate of individual paths. The aim of this load distribution algorithm is to improve resource utilization and mitigate the negative effects of network congestion during the event reporting period. Furthermore, the developed load balancing algorithm enables every source node to keep track of the path quality changes during data transmission over various paths and redistribute the network traffic over available paths based on the latest data transmission quality of each path.

Thirdly, we implement the proposed multipath routing protocol and evaluate its performance compared to the recently proposed multipath routing protocols in wireless sensor networks. Extensive performance evaluations show that IM2PR presents improvements over the Micro Sensor Multipath Routing Protocol (MSMRP) and Energy-Efficient data Routing Protocol (EERP) as follows: 50% and 70% in term of packet reception ratio (PRR) at the sink, 44% and 80% in term of goodput, 33% and 40% in term of packet delivery latency, 40% and 57% in term of energy consumption, 50% and 60% in term of packet delivery overhead.

The rest of this paper is organized as follows: Section 2 provides an overview on the existing multipath routing protocols in wireless sensor networks. We introduce IM2PR in Section 3. Performance evaluation and comparison of IM2PR against MSMRP and EERP is performed in Section 4. We conclude in Section 5.

2. Related Work

Nowadays multipath routing approach is introduced as an effective technique for improving sensor and ad hoc networks performance in terms of energy consumption, fault tolerance, reliability and throughput. Multipath routing is a class of routing protocols which enables every source node to discover several paths towards the destination. Discovered paths can be utilized concurrently to distribute network traffic over several paths or the source node can use only one path for data transmission and then switches to an alternative path when a node or link failure occurs along the active path for fault tolerance purpose [24]. In the following we present some of the recently proposed multipath routing protocols in wireless sensor networks.

N-to-1 Multipath Routing Protocol [18] is proposed according to the convergecast traffic pattern of wireless sensor networks to improve data transmission reliability. In this protocol, each node identifies multiple paths towards the sink node through constructing a spanning tree rooted at the sink node. Through N-to-1 Multipath Routing Protocol, all the nodes utilize single-path forwarding strategy for transmitting every data segment, while they also use an adaptive per-hop packet salvaging technique to provide fast data recovery from node or link failures. Since all the paths identified in the tree routing topology are located physically proximal to each other, concurrent data transmission over these paths causes high inter-path interference which in turn degrades the network performance.

Multi-Constrained QoS Multipath Routing Protocol (MCMP) [10] is mainly designed to provide soft-QoS guarantee in terms of reliability and latency. Through this protocol each source node establishes multiple partially disjoint paths which can provide latency and reliability demands of the intended application. Therefore, to achieve the reliability demand of individual applications, every node should forward multiple copies of each packet over different paths. However, this data redundancy is in contrast with the resource constraints of sensor networks. Additionally, since partially disjoint paths are usually located near each other, concurrent data transmission over these paths causes a high inter-path interference which results in a high packet loss ratio.

Interference-Minimized Multipath Routing Protocol (I2MR) [32] aims to improve network throughput through transmitting every source node's traffic over zone disjoint paths which are constructed using location information of nodes and employing special hardware equipments. I2MR assumes there are several gateway nodes in the network which are serving as final destinations and they are connected directly to the command center using non-interfering links with high-capacity. With this assumption, a given source node should discover three zone-disjoint paths towards three of them. Since this protocol is designed based on a specific network structure with particular components, it can not be easily employed in many applications.

AOMDV-Inspired Multipath Routing Protocol [11] is designed based on the AOMDV [19] to provide low-latency and energy-efficient data delivery in wireless sensor networks through exploiting some information from the MAC layer. In fact, the aim of this protocol is to enable every node to select one of its next-hop neighbors towards the destination that wakes up earlier than others. Since this protocol is designed based on AOMDV, same as the ad hoc-based routing protocols the whole path information should be propagated throughout the network during the path establishment process. However, due to the resource constraints of low-cost sensor nodes, propagating the whole path information in the network through Route REQuest (RREQ) packets is not feasible.

Energy-Efficient and Collision-Aware Multipath Routing Protocol (EECA) [34] is proposed to construct two collision free paths in both sides of the straight line between every source-sink pair using location information of network nodes. With respect to the main operation of this protocol, all the nodes should be equipped with GPS. Furthermore, every node should be aware about the exact location of their neighbors for making routing decisions. However, these requirements increase the network deployment cost and intensify the communication overhead, specifically in large and dense wireless sensor networks.

Energy-Efficient and QoS-based Multipath Routing Protocol (EQSR) [3] aims to satisfy the latency and reliability requirements of real-time applications. In order to fulfill the latency requirements of various applications, this protocol utilizes a service differentiation technique through a queuing model to manage real-time and non-real-time traffic. Furthermore, EQSR improves data transmission reliability through using a lightweight XOR-based Forward Error Correction (FEC) mechanism. However, the utilized FEC mechanism imposes a significant computational overhead to the resource constrained sensor networks for computing the error correction codes and retrieving the original messages.

Low-Interference Energy-Efficient Multipath ROuting Protocol (LIEMRO) aims to construct minimum interfering paths from every event area towards the sink [22, 23]. Since overhearing of the RREP packets enable nodes to update their interference levels, this protocol does not allow the source node to establish minimum interfering paths concurrently. Furthermore, the path cost function of LIEMRO estimates data transmission cost of a given path through summation of the links' ETX values along that particular path [6]. However, as the ETX metric assumes the link layer provides an infinite number of transmission attempts over individual links, this protocol may not be able to identify efficient paths in the cases where the link layer offers a limited number of transmissions per packet delivery.

Micro Sensor Multipath Routing Protocol (MSMRP) [9] is developed to extend network lifetime through distributing the traffic generated by a given source node over two node-disjoint paths. In MSMRP, every source with event data packets to transmit which has not identified any path towards the sink, initiates the route discovery process by broadcasting RREQ packets. Each node that receives a RREQ packet, updates the hop-count and path quality indicator fields of the received packet and rebroadcasts the updated packet. Upon reception of the RREQ packets by sink node, it selects two of the best discovered paths based on the hop-count and path quality indicator fields of the received RREQ packets. According to the operation of this protocol, it does not consider the effects of inter-path interference on the packet delivery performance of individual paths.

Energy-Efficient Data routing Protocol (EERP) [4] aims to prevent network from being disjointed through multipath routing. In EERP, sink node initializes the route construction precess by flooding a *"Signalization"* packet. Whenever a node receives a *"Signalization"* packet, it calculates the energy cost of data transmission towards the sink through the sender and rebroadcasts this packet with the updated energy cost. Upon occurrence of an event in the sensor field, the selected source node transmits the event data towards the sink through the lowest cost path. Moreover, EERP tries to maintain the network connectivity for a maximum possible period by enabling nodes to switch to an alternative path whenever they realize the remaining battery level of their next-hop nodes are below a certain threshold. Although this protocol aims to improve the network lifetime through enabling nodes to switch to different paths during the data transmission process, it suffers from restricted capacity of a single path.

3. Interference-Minimized Multipath Routing Protocol

The proposed IM2PR consists of three phases: (1) Initialization phase, (2) Path establishment phase, and (3) Data transmission and path maintenance phase. At the initialization phase every node estimates data transmission quality of its links to its neighbors. Furthermore, all the nodes also calculate the data transmission cost towards the sink through their neighboring nodes. Every node uses these information for selecting its best next-hop node towards the sink during the path establishment process. Detection of an event in the sensor field, triggers the path establishment phase in order to construct an adequate number of paths towards the sink. In this phase, the selected source node in the event area starts to construct the first two paths concurrently, and then additional paths will be constructed if their concurrent utilization improves the Data Reception Rate (DRR) of the sink node. During the path construction process, every node uses a path cost function which considers probability of successful packet transmission over every link in the presence of other interfering links, and residual battery level of the sensor nodes to select the best next-hop node toward the sink. Finally, the data transmission and path maintenance phase takes care of data transmissions from event area towards the sink and handles path failures. The following sections describe theses phases in detail. All the notations used in this paper are presented in Table 1.

3.1. Initialization Phase

The initialization phase aims to enable every node to estimate data transmission quality of its connections to its neighbors and construct a minimum cost data gathering tree rooted at the sink node. At the beginning of the initialization phase individual

Table 1: Notations and their descriptions.

symbol	Description
$l_{i,j}$	Outgoing link from node n_i to node n_j
r	Maximum number of offered link layer transmission attempts at each link
ι	Data packet size in bit
R	Radio bit rate
$p_{i,j}$	Packet delivery probability over link $l_{i,j}$ with a single transmission effort
$p^{in}_{i,j}$	Probability of successful packet transmission from node n_i to node n_j in the presence of active interfering neighbors
γ_j	Set of active interfering neighbors of node n_j
$E[l_{i,j}]$	Expected number of packet transmissions required for successful packet delivery over link $l_{i,j}$
ω	Minimum path cost value towards the sink
$\Psi(l)$	Weighting function which reflects the influence of link positions on the data transmission cost of a path
$RP_{j,i} \to \omega$	Minimum path cost value towards the sink included in the routing packet received at node n_i from node n_j
$RP_{j,i} \to \Psi(l)$	The $\Psi(l)$ value included in the routing packet received at node n_i from node n_j
$SFTC^j_{i,sink}$	Successful or Failed packet Transmission Cost (SFTC) from node n_i towards sink through node n_j
$PathCost^j_{i,sink}$	Data transmission cost from node n_i towards the sink through node n_j
$IACost_j$	Real interference-aware data transmission cost from node n_j towards the sink
IBL_i	Initial battery level of node n_i
RBL_i	Remaining battery level of node n_i
$Rate_k$	Data transmission rate of the *kth* path
ϖ_k	Data transmission cost of the *kth* path
τ	Current state of the IM2PR protocol
τ_{ne}	No-event state
τ_{fp}	First path construction state
τ_{sp}	Second path construction state
τ_{ap}	Additional path construction state
τ_{dt}	Data transmission state
τ_{ic}	Experienced interference level calculation state
τ_{wic}	Waite for collecting the nodes experienced interference levels state

nodes broadcast a fixed number of beacon messages and record the number of received beacon messages from their neighboring nodes to estimate data transmission quality of their links based on their respective PRR [7, 26].

The second step of this phase aims to enable every node to calculate data transmission cost towards the sink node through its individual neighboring nodes in term of the required number of transmissions for every successful packet delivery. Since packet drops due to the bounded number of link layer transmissions on the links near the destination is very costly, order of the links along the paths from source nodes towards the sink plays an important role to reduce the number of packet transmissions for every single packet delivery. In this regard, this phase employs a data gathering tree construction process to enable every node to find out the packet transmission cost of the available paths towards the sink based on the location of the links which may cause packet drops due to the limited number of link layer transmissions. Therefore, at the second step of this phase sink node initiates the data gathering tree construction process through broadcasting a routing packet to the network. During this process, whenever an intermediate node n_i receives a routing packet, it calculates its data transmission cost towards the sink through the sender node (e.g., n_j) as:

$$SFTC^j_{i,sink} = (RP_{j,i} \to \omega) + (E[l_{i,j}] \times (RP_{j,i} \to \Psi(l)))$$
where
$$E[l_{i,j}] = (\sum_{k=1}^{r} k(1-p_{i,j})^{k-1} p_{i,j}) + r(1-p_{i,j})^r \tag{1}$$

where $SFTC^j_{i,sink}$ is the Successful or Failed packet Transmission Cost (SFTC) from node n_i towards sink node through node n_j, $E[l_{i,j}]$ is the expected number of transmissions needed for successful packet transmission over link $l_{i,j}$ with assuming the link layer provides at most r transmission attempts per packet at each link, $p_{i,j}$ is the probability of successful packet transmission from node n_i to node n_j with a single transmission effort, $RP_{j,i} \to \omega$ is the minimum path cost value towards the sink included in the received routing packet, and $RP_{j,i} \to \Psi(l)$ is the $\Psi(l)$ value included in the received routing packet. $\Psi(l)$ is a weighting function that scales the required number of transmission attempts at each link in order to reflect the influence of link positions and their respective data transmission quality on the data transmission cost of a path. The insight behind using this weighting function is that, it scales the expected number of link layer transmissions on a given link $l_{i,j}$ based on the ratio of the required number of packet transmissions for successful data delivery over every link traversed from sink node till node n_i to the maximum number of offered link layer transmission attempts. Note that sink node initializes the ω and $\Psi(l)$ fields of the routing packet to 0 and 1 respectively.

When node n_i calculates the data transmission cost towards the sink through node n_j, it preserves the routing information through this node in its routing table. Furthermore, if the newly calculated SFTC value by node n_i is lower than the minimum SFTC value which has been calculated by this node so far, it should rebroadcast the received routing packet with updated ω and $\Psi(l)$ values. In this regard node n_i updates the $\Psi(l)$ value as follows:

$$\Psi(l) = \begin{cases} RP_{j,i} \to \Psi(l) \times 1 & \text{if } \frac{(1/p_{i,j})}{r} \leq 1 \\ RP_{j,i} \to \Psi(l) \times (\frac{1/p_{i,j}}{r}) & \text{if } \frac{(1/p_{i,j})}{r} > 1 \end{cases} \tag{2}$$

where r is the maximum number of offered link layer transmission attempts at individual links, and $p_{i,j}$ is the probability of successful packet transmission from node i to node j. The reason for selecting the above options to update the $\Psi(l)$ value is that, in order to have a successful packet delivery over a given link through performing at most r transmission attempts, the ratio of the required number of packet transmissions for successful packet delivery over that link to the r value should be less or equal to 1. If this ratio is higher than 1, there is a probability of packet drop after performing r transmission attempts.

This phase ends whenever the entire nodes identify the data transmission cost towards the sink through their immediate neighboring nodes.

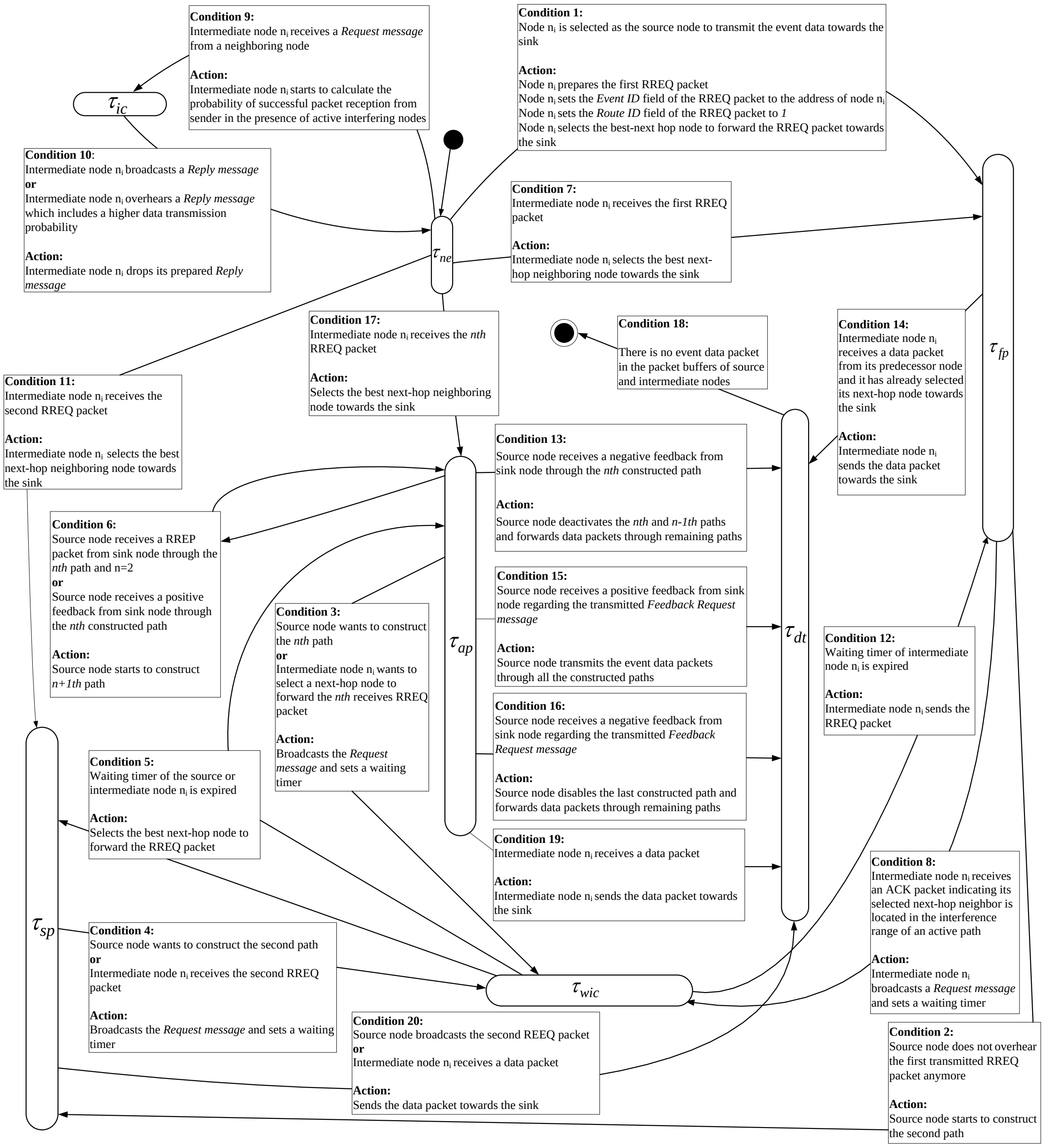

Figure 1: The state diagram of the IM2PR protocol.

3.2. Path Establishment Phase

As demonstrated in Figure 1, IM2PR protocol performs the path discovery process through different states. During normal network operation, all the nodes are in the τ_{ne} state. Upon occurrence of an event in the sensor field the selected source node initiates the route discovery process by moving to the τ_{fp} state. The selected source node which is in the τ_{fp} state begins to discover sufficient number of minimum interfering paths by forwarding the first RREQ packet towards the sink node. In this regard, the selected source node calculates its data transmission cost towards the sink through its neighboring nodes to find a best-next hop node which will cause minimum data delivery cost during the data transmission process. In IM2PR, source and every intermediate node n_i calculate the data transmission cost towards the sink through neighbor n_j as:

$$PathCost^{j}_{i,sink} = (SFTC^{j}_{i,sink}) \times (\frac{IBL_j}{RBL_j}) \qquad (3)$$

where $PathCost^{j}_{i,sink}$ is the data transmission cost from node n_i (i.e., the node wants to find a minimum cost neighbor) towards the sink through neighbor n_j, $SFTC^{j}_{i,sink}$ is the SFTC value from node n_i towards sink node through neighbor n_j, IBL_j is the initial battery level of neighbor n_j, and RBL_j is the remaining battery level of neighbor n_j. When the source node finds its minimum cost neighbor through Equation 3, it creates the first RREQ packet and sends this packet to the selected neighbor towards the sink (lines 1-10 of Algorithm 1). Figure 2 shows the

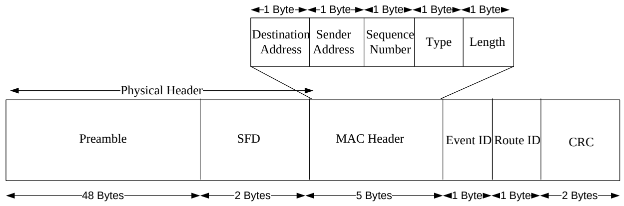

Figure 2: RREQ packet format

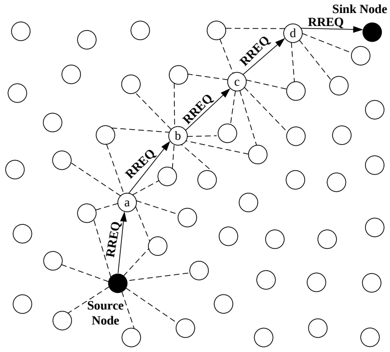

Figure 3: Transmission of the RREQ packet from source node towards the sink and overhearing of this packet by different nodes. Dashed lines show the packet overhearing between node pairs.

format of the RREQ packet. The *Event ID* field of the RREQ packet indicates the ID of the occurred event which has triggered the route establishment process. Moreover, to distinguish between paths constructed by a same source node, the *Route ID* field of each RREQ packet indicates the ID of the path that is being constructed through that RREQ packet. In IM2PR, source node sets the *Route ID* field of the first created RREQ packet to *1* in order to indicate this packet is transmitted for establishing the first path and it also assigns its ID to the *Event ID* field.

Every intermediate node n_i which receives the first RREQ packet changes its state to the τ_{fp} in order to continue the path establishment process. After that, node n_i forwards the received RREQ to its minimum cost neighbor towards the sink which is selected through Equation 3. Furthermore, every node which forwards the RREQ, preserves the ID of its selected next-hop node towards the sink and ID of the node from which it has received the RREQ to establish the reverse path towards the source node, as well as, the included *Event ID* to indicate it belongs to a route from a given event area (lines 1-7 of Algorithm 2).

Moreover, as IM2PR utilizes the broadcast nature of wireless channel to construct minimum interfering paths, every node that overhears a RREQ packet should update the preserved neighborhood information regarding the sender of the overheard packet. As shown in Figure 3, each node that overhears a RREQ packet marks the sender of the overheard packet as an active interfering neighbor in its neighborhood table (lines 22-24 of Algorithm 2). Furthermore, it assigns the included *Event ID* in the overheard packet to the path membership variable of this neighbor to indicate that neighbor belongs to a path from a specific event area. The path membership variable allows the nodes to establish node-disjoint paths from a given event area towards the sink.

IM2PR also considers the case where multiple events coexist in the sensor field. In this regard, if during construction of the first path from a given event area intermediate nodes realize there exist another active event in the sensor field, they try to construct minimum interfering paths through the nodes which have experienced lower interference level from the existing active paths. In this context, intermediate nodes which are pursuing the construction process of the first path for transmitting the data packets of the just occurred event towards the sink should be aware about the data transmission activity of their neighbors which are engaged with transmitting the data packets generated by another event. To this aim, every intermediate node n_j which changes its state to the τ_{fp} because of receiving the first RREQ packet, first searches its neighborhood table to see whether it overheard any RREQ packet related to another active event area. If node n_j finds out it has not overheard any packet, it selects a minimum cost neighbor as its next-hop node towards the sink through Equation 3 and forwards the received RREQ packet to the selected next-hop node. While, if node n_j realizes it has experienced interference from an active path which is transmitting the event data of another source node, it notifies the sender of the RREQ packet by including an additional field in its ACK packet. Upon reception of such ACK packet by an intermediate node (e.g., node n_i), it perceives its selected next-hop node is located in the interference range of an active path which is transmitting the data packets generated by another event. In this situation, intermediate node n_i moves to the state τ_{wic} to collection information about the data delivery performance of its neighbors in the presence of active interfering nodes for selecting the best next-hop node towards the sink (lines 38-43 of Algorithm 2). The operational details of the state τ_{wic} will be given later (during describing the construction process of the second path). The transmission of the first RREQ packet by intermediate nodes will be continued until the sink node receives this packet.

As mentioned earlier, in IM2PR every source node constructs the first two paths concurrently. However, in order to establish minimum interfering paths, every node that wants to select a next-hop node towards the sink during construction of the second path should be aware about the interference level experienced by its neighboring nodes. In this context, source node should postpone the construction process of the second path to allow its neighbors to update their experienced interference level from the nodes along the first path. As demonstrated in Figure 4, source node waits before transmitting the second RREQ packet, as long as it does not overhear the first RREQ packet anymore. After that, source node starts to construct the second path by moving to the τ_{sp} state (lines 11-13 of Algorithm 1). When the source node moves to the τ_{sp} state, it first broadcasts a *Request message* in order to be aware about the probability of successful data transmission to its next-hop neighbors towards the sink in the presence of active interfering nodes. Then it sets a waiting timer based on the multiplication of the number of its neighboring nodes and two way message

Algorithm 1 IM2PR algorithm at a source node.

1: **if** (node n_i is selected as the source to transmit the event data towards the sink) **then** ▷ Condition 1
2: $\tau = \tau_{fp}$
3: **for** (every neighbor n_j of the source node) **do**
4: calculate $PathCost^{j}_{source,sink}$
5: **end for**
6: create a RREQ
7: add ID of the source to the RREQ as the *Event ID*
8: set the *Route ID* field of the RREQ to 1
9: send the RREQ to the minimum cost neighbor
10: **end if**
11: **if** (the first REEQ is not overheard any more) **then** ▷ Condition 2
12: $\tau = \tau_{sp}$
13: **end if**
14: **if** $((\tau == \tau_{sp} || \tau == \tau_{ap})$ **then** ▷ Condition 3 or 4
15: broadcast a *Request message*
16: $\tau = \tau_{wic}$
17: set a waiting timer
18: **end if**
19: **if** (waiting timer is expired & there exists n path(s)) **then** ▷ Condition 5
20: source node creates a RREQ
21: add ID of the source to the RREQ as the *Event ID*
22: set the *Route ID* field of the RREQ to *n+1*
23: transmit the RREQ to the neighbor which is not a member of any path from the same event area and has the highest $IACost_j$ value
24: **if** (construction of the secound path is not complited yet) **then** ▷ Condition 20
25: $\tau = \tau_{dt}$
26: start to transmit the event data packets towards the sink through the selected neighbor over the first path
27: **end if**
28: **end if**
29: **if** (a RREP is received from the second path) **then** ▷ Condition 6
30: transmit the event data packets through first and second paths
31: $\tau = \tau_{ap}$
32: **end if**
33: **if** (a positive feedback is received from nth path & $n > 2$) **then** ▷ Condition 6
34: transmit the event data packets through n constructed paths
35: $\tau = \tau_{ap}$
36: **end if**
37: **if** (a negative feedback is received from nth path & $n > 2$) **then** ▷ Condition 13
38: disable n and *n-1*th paths
39: $\tau = \tau_{dt}$
40: transmit the event data packets through remaining paths
41: **end if**
42: **if** (a positive feedback regarding the transmitted *Feedback Request message* is received) **then** ▷ Condition 15
43: $\tau = \tau_{dt}$
44: transmit the event data packets through all the constructed paths
45: **end if**
46: **if** (a neagtive feedback regarding the transmitted *Feedback Request message* is received) **then** ▷ Condition 16
47: disable the last constructed path
48: $\tau = \tau_{dt}$
49: transmit the event data packets through remaining paths
50: **end if**
51: **if** (the nth transmitted RREQ is returned) **then**
52: transmit a *Feedback message* through *n-1*th path
53: **end if**

Algorithm 2 IM2PR algorithm at an intermediate node.

1: **if** (a RREQ with *Route ID*==1 is received by node n_i) **then** ▷ Condition 7
2: **for** (every neighbor n_j of node n_i) **do**
3: calculate $PathCost^{j}_{i,sink}$
4: **end for**
5: $\tau = \tau_{fp}$
6: transmit the RREQ to the minimum cost neighbor
7: **end if**
8: **if** (a RREQ with *Route ID* ≠ 1 is received) **then**
9: **if** (*Route ID*==2) **then** ▷ Condition 11
10: $\tau = \tau_{sp}$
11: **else** ▷ Condition 17
12: $\tau = \tau_{ap}$
13: **end if**
14: **if** (there exist next-hop nodes which are not member of any path from the same source node) **then**
15: broadcast a *Request message* ▷ Condition 3
16: $\tau = \tau_{wic}$
17: set a waiting timer
18: **else**
19: return the RREQ to its predecessor node along the reverse path
20: **end if**
21: **end if**
22: **if** (a RREQ is overheard) **then**
23: mark the sender of the overheard RREQ as an active interfering node
24: **end if**
25: **if** (a *Request message* is received from neighbor n_j & this node is not a member of any path from a same source node) **then** ▷ Condition 9
26: $\tau = \tau_{ic}$
27: calculate $p^{in}_{j,i}$ and $IACost_i$
28: broadcast a *Reply message*
29: $\tau = \tau_{ne}$
30: **end if**
31: **if** (a *Reply message* is overheard & its $IACost_x \geq IACost_i$) **then** ▷ Condition 10
32: Drop the *Request message*
33: $\tau = \tau_{ne}$
34: **end if**
35: **if** (a RREP packet is received) **then**
36: forward the RREP packet to the source node through the reverse path
37: **end if**
38: **if** (an ACK packet with *Active Interfering flag*==true is received) **then** ▷ Condition 8
39: **if** (there exist next-hop nodes which are not member of any path from a same source node) **then**
40: $\tau = \tau_{wic}$
41: broadcast a *Request message*
42: set a waiting timer
43: **else**
44: return the RREQ to its predecessor node along the reverse path
45: **end if**
46: **end if**
47: **if** (the waiting timer is expired) **then** ▷ Condition 5 or 12
48: transmit the RREQ to the neighbor which has the highest $IACost_j$
49: **end if**
50: **if** (the nth transmitted RREQ is returned from node n_k) **then**
51: **if** (there exist other next-hop nodes which are not member of any path from a same source node) **then**
52: try to forward the RREQ towards the sink through another qualified neighbor
53: **else**
54: return the RREQ to its predecessor node along the reverse path
55: **end if**
56: **end if**

transmission latency. After that, source node changes its state to the τ_{wic} in order to wait for receiving *Reply messages* from its neighbors (lines 14-18 of Algorithm 1). In order to avoid nodes being shared between various paths from a same event area, every node which receives this *Request message*, first checks its path membership status to see whether it is a member of any path belong to the announced *Event ID* in the received *Request message*. The nodes which are not member of any path constructed from the same event area change their state to the τ_{ic} to calculate the probability of successful packet reception from the sender of the *Request message* based on their packet reception probability from their active interfering neighbors. Every node n_j which has received a *Request message* from node n_i and it is in the τ_{ic} state, first calculates the probability of successful packet reception from the sender of that message (i.e., n_i) in the presence of active interfering neighbors as:

$$p^{in}_{i,j} = p_{i,j} \times \zeta$$
$$where\ \zeta = \begin{cases} \prod_{k \in \gamma_j} (1 - p_{k,j}) & if\ \gamma_j \neq \emptyset \\ 1 & if\ \gamma_j = \emptyset \end{cases} \quad (4)$$

where $p^{in}_{i,j}$ is the probability of successful packet transmission from node n_i (i.e., sender of the *Request message*) to node n_j in the presence of interfering neighbors which are dealing with packet transmission, $p_{i,j}$ is the packet delivery probability from

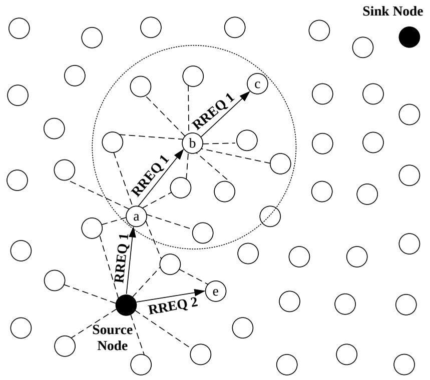

Figure 4: Transmission of the second RREQ packet concurrent with the construction of the first path. Dashed lines show the packet overhearing between node pairs, and the dotted line circle shows the transmission range of node b.

node n_i to node n_j without considering active interfering nodes (which is calculated during the initialization phase), and γ_j is the set of active interfering neighbors that node n_j overheard a packet from them. When node n_j calculates the packet reception probability of its incoming link from sender of the *Request message* (i.e., $p^{in}_{i,j}$), then it should calculate its real interference-aware data transmission cost in order to reply to the sender of the *request message* as:

$$IACost_j = \frac{SFTC_i - SFTC_j}{SFTC_i} \times (1 + p^{in}_{i,j}) \qquad (5)$$

where $SFTC_i$ is the data transmission cost of the sender of the *request message* towards the sink.

Since every *Reply message* includes the corresponding $IACost_j$ value, every node that overhears a *Reply message* including a higher $IACost_j$ compare to its calculated value refuses to broadcast its prepared *Reply message* (lines 31-34 of Algorithm 2). Whenever, the waiting timer of the sender of the *Request message* expires, it changes its state to the τ_{sp} and transmits the second RREQ packet to a neighboring node which is not a member of any path belong to the same event and has the highest $IACost_j$ value (lines 19-23 of Algorithm 1 and lines 47-49 of Algorithm 2). This procedure will be repeated between all the intermediate nodes which receive the second RREQ packet until reception of this packet by the sink node. If during the path construction process, an intermediate node which has received a RREQ packet, finds out it cannot establish a node disjoint path, it should inform its predecessor node about the failure in forwarding the RREQ packet (lines 18-20 and 43-45 of Algorithm 2). Afterward, the sender of the RREQ packet tries to forward this packet to another qualified next-hop neighboring node, otherwise it should forward this packet to its predecessor node in the reverse direction towards the source node (lines 50-56 of Algorithm 2). This back-pressure mechanism continues until an intermediate node finds another qualified next-hop neighbor towards the sink or the RREQ packet reaches the source node. Notice that, receiving a RREQ packet by the source node indicates that it cannot establish another node-disjoint path.

In order to reduce the latency of event reporting, when the

Algorithm 3 IM2PR algorithm at the sink node.

1: **if** (a RREQ packet with *Route ID*==2 is received) **then**
2: create a RREP packet
3: transmit the RREP towards the source through the second path
4: **end if**
5: **if** (*n*th RREQ packet with *Route ID*> 2 is received) **then**
6: **if** (DRR through *n-1* paths> DRR through *n-2* paths) **then**
7: create a RREP packet which indicates a positive feedback and transmit towards the source node through the *n*th path
8: **else**
9: create a RREP packet which indicates a negative feedback and transmit towards the source node through the *n*th path
10: **end if**
11: **end if**
12: **if** (a *Feedback message* is received through *n*th path) **then**
13: **if** (DRR through *n* paths ≥ DRR through *n-1* paths) **then**
14: transmit a positive feedback towards the source through the *n*th path
15: **else**
16: transmit a negative feedback towards the source through the *n*th path
17: **end if**
18: **end if**

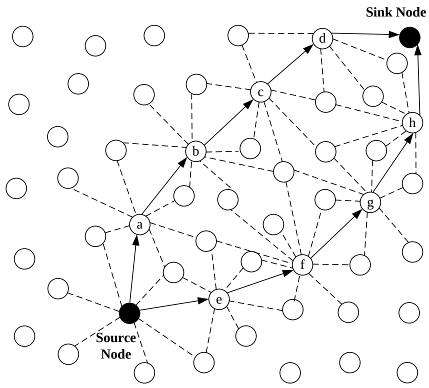

Figure 5: The first two paths constructed by IM2PR.

source node sends the second RREQ packet, it changes its state to the τ_{dt} and starts to transmit the event data packets towards the sink through its selected next-hop node along the first path (lines 24-28 of Algorithm 1). Upon reception of the second RREQ packet by the sink node, it forwards a RREP packet towards the source over the second constructed path (lines 1-4 of Algorithm 3). When the source node receives a RREP packet from the second path, it transmits its event data packets through both of the constructed paths, while it also starts to construct the third path by moving to the τ_{ap} state (lines 29-32 of Algorithm 1). Every intermediate node along the second path which receives a data packet for the first time should change its state to the τ_{dt}. The construction process of the third path follows the same procedure described for establishing the second path. The only difference is that, the sink node allows the source to use the newly discovered path (i.e., third path) for data transmission, if it realizes concurrent data transmission over two paths results in a higher DRR compared to using only one path. Therefore, when sink node receives the third RREQ packet, it first compares its achieved DRR through receiving data packets from two paths with that achieved through receiving data packets from one path. If concurrent utilization of the first two paths has increased the DRR of the sink node, it forwards a RREP packet over the third path which indicates its positive feedback for using the last discovered path concurrent with other paths (lines

5-7 of Algorithm 3). However, if simultaneous usage of the first two established paths has reduced the DRR of the sink node, it notifies the source node to disable the second path and transmit the event data through the first path (lines 8-10 of Algorithm 3). Whenever source node receives a positive feedback from the sink for its nth constructed path, it distributes its traffic over the n established paths and starts to construct the n+*1*th path (lines 33-36 of Algorithm 1). While, upon reception of a negative feedback at the source node regarding the nth path (i.e., third and subsequent paths), it finalizes the path construction process by disabling the last two constructed paths (i.e., nth and n-*1*th paths) and transmitting its event data packets though remaining paths (lines 37-41 of Algorithm 1). Notice that, in the case that source node fails to construct a new node-disjoint path and receives its nth transmitted RREQ packet, it sends a *Feedback Request message* towards the sink node through the last established path (i.e., n-*1*th path) (lines 51-53 of Algorithm 1). The aim of this *Feedback Request message* is to inform source node about the decision of the sink regarding the utilization of the last constructed path. Whenever, the sink node receives a *Feedback Request message* from nth path, it compares its achieved DRR through receiving data packets from n paths with that achieved through receiving data packets from n-*1* paths, and sends its feedback towards the source node (lines 12-18 of Algorithm 3). Figure 5 shows the first two paths constructed by the proposed IM2PR protocol.

3.3. Data Transmission and Path Maintenance Phase

In order to reduce the latency of event reporting, IM2PR allows the source node to transmit its collected event data towards the sink concurrent with the path construction process. As mentioned in Section 3.2, source node starts to transmit its data packets towards the sink through its selected next-hop node along the first path concurrent with the construction of the second path. During construction of further paths, whenever source node receives a RREP packet which indicates the positive feedback of the sink to utilize the last discovered path, it redistributes its traffic over the established paths towards the sink. Since paths discovered by the multipath routing protocol contain nodes with various packet delivery probability and remaining battery level, individual paths offer a distinct data transmission capacity. Therefore, to provide efficient data delivery (in terms of PRR, latency, energy consumption and goodput), source node should adjust the traffic rate of individual paths based on their data transmission performance. To this aim, all the RREP and ACK packets transmitted by the nodes in IM2PR, include some information about probability of successful packet transmission, and remaining battery level of the nodes along the traversed paths. During transmission of every RREP packet along a reverse path towards the source node these values are being updated. Since the initial route information collected by the source node will be changed during data transmission, ACK packets are utilized to provide source node with updated route information regarding the paths that are engaged with the data transmission process. Therefore, whenever source node receives a RREP packet from the sink node regarding the kth constructed path, it first calculates data transmission cost of the established path as:

$$\varpi_k = \frac{1}{1+\sum_{i=1}^{n-1} p_{i,i+1}^{in}} \times \frac{1}{1+\sum_{i=1}^{n-1} RBL_i} \tag{6}$$

where, $\sum_{i=1}^{n-1} p_{i,i+1}^{in}$ and $\sum_{i=1}^{n-1} RBL_i$ are the accumulated packet delivery probability and accumulated residual battery level along kth path with n nodes. In addition to the calculation of the ϖ value for the last established path (i.e., kth path), it also recalculates the ϖ of other available paths based on the achieved updates. Finally, source node updates the optimal traffic rate of the available paths as follows:

$$\begin{aligned} &\min(Rate_1 \times \varpi_1 = Rate_2 \times \varpi_2 = \ldots = Rate_n \times \varpi_n) \\ &\text{Subject to } \sum_{i=1}^{n} Rate_i = 1 \\ &Rate_k = \frac{1}{\varpi_k \sum_{j=1}^{n} \frac{1}{\varpi_j}} \end{aligned} \tag{7}$$

where, n is the number of available paths, $Rate_k$ and ϖ_k are data transmission rate and data transmission cost of the kth path respectively. The optimal data rate of different paths can be determined whenever the source node finalizes the path construction process and only conducts the data transmission process.

Since in wireless sensor networks active paths may fail due to the link dynamics, energy depletion of nodes or physical damages, IM2PR provides a path maintenance mechanism during the data transmission process. In this regard, if an intermediate node n_i finds out there is no communication with its next-hop node for interval T during the data transmission period, it will notify the source node about the identified path failure. Since every node (e.g., n_i) knows the packet delivery probability of the outgoing link towards its next-hop node (e.g., n_j), therefore with assuming there is two transmission attempt over a link with $p_{i,j} = 1$, interval T can be calculated through the geometric distribution as:

$$T = (\frac{1}{p_{i,j}} + 1) \times (\frac{\iota}{R}) \tag{8}$$

where ι is the data packet size in bit and R is the radio bit rate. When an intermediate node detects a path failure during the data transmission process, it will forward an error message towards the source node. Upon reception of an error message by the source node, it disables the failed path and redistributes the traffic over the remaining paths. Furthermore, it initiates the route discovery process if there exists less than two active paths.

4. Performance Analysis

This section analyzes and compares performance of IM2PR against EERP [4] and MSMRP [9]. First we describe the simulation software employed for performance evaluations and its parameters. After that, we present the performance evaluation metrics. Finally, we study performance of the proposed IM2PR in comparison with the EERP and MSMRP protocols.

Table 2: Simulation parameters.

Radio	
Average noise power [dBm]	-106
Noise figure [dB]	13
Switch to TX/RX [μs]	250
Radio sampling [μs]	350
Evaluate radio sample [μs]	100
Noise bandwidth [Hz]	30000
Modulation	NC-FSK
Encoding	Manchester
Transmission power [dBm]	0
Standard deviation of transmission power heterogeneity	1.2
Standard deviation of noise floor heterogeneity	0.9
Radio speed after encoding [bits per second]	19200
Reference distance [m]	1
PL (d_0) [dB]	55
Environment	
Ambient temperature [$C°$]	27
Path loss exponent (outdoor)	4.7
Multipath channel variations (outdoor)	3.2
B-MAC	
Initial backoff [slots]	32
Congestion backoff [slots]	16
Sampling interval [ms]	20
Other parameters	
Network topology	Random
Number of nodes	100,400
Number of source nodes	2
Area size (network with 100 nodes in outdoor)	30m×30m
Area size (network with 400 nodes in outdoor)	65m×65m
Number of generated packets by each source	100

4.1. Simulation Setup

We have performed our performance evaluations using the OMNeT ++ simulation framework. In order to precisely simulate the characteristics of low-power wireless communications and improve the accuracy of the simulation results, we have developed a particularly accurate wireless channel model and a physical layer model that consider path loss, multipath effect, transmission power variations, noise floor variations and the capture effect based on the models presented in [35, 39, 41]. The radio parameters are chosen based on the Mica2 mote specifications. Furthermore, we have implemented B-MAC [21] as the underlying MAC protocol in our simulation software. Table 2 presents the default simulation parameters of this paper in detail.

4.2. Performance Parameters

We have evaluated and compared the performance of the IM2PR, MSMRP, and EERP protocols using following parameters:

i. **Packet reception ratio**: This metric reveals the ability of different protocols to enhance reliability of event reporting through measuring the ratio of the number of event packets received by the sink to the total number of event packets transmitted by source nodes.

ii. **Goodput**: This metric is defined as the ratio of the total number of data bits received by the sink node to the event reporting duration. Since each path has a limited capacity, this metric presents the significance of different protocols to improve network performance under various network traffic conditions.

iii. **Packet delivery latency**: This metric is measured as the average elapsed time for sending event data packets from source nodes to the sink. This metric demonstrates the efficiency of different protocols to reduce latency of event reporting.

iv. **Energy consumption for packet transmission**: This metric indicates the average energy consumed by individual nodes to transmit data packets to the sink node which is presented as the percentage of total battery capacity of a sensor node. Therefore, this metric compares the energy efficiency of different routing protocols.

v. **Packet delivery overhead**: This metric reveals the overhead cost of using different routing protocols in event-driven applications of wireless sensor networks by measuring the ratio of the number of data and control packets transmitted during the path establishment and data transmission processes to the number of data packets received by the sink node.

4.3. Performance Evaluation

This section analyzes and compares the performance of IM2PR, MSMRP, and EERP protocols in terms of the metrics presented in Section 4.2. In all the figures, each result point shows the median of 20 simulation runs, while the error bars present the upper and lower quartiles.

4.3.1. Packet Reception Ratio

The PRR achieved at the sink node through IM2PR, MSMRP, and EERP protocols in the networks with 100 and 400 nodes are presented in Figure 6. As can be seen from this figure, IM2PR improves PRR at the sink node about 50% and 70% compared to the MSMRP and EERP protocols in a network with 100 nodes. This observation reveals the effectiveness of the inter-path interference level measurement mechanism employed by the IM2PR protocol to evaluate the amount of interference level that every node may experience during the data transmission process. Based on Figure 6, by increasing the packet generation rate at the source nodes, reliability of event reporting through all of the protocols is reduced significantly. This performance degradation can be explained as follows: Firstly, it is obvious that increasing the number of generated event packets per second intensifies the chance of network congestion due to the packet buffer overflow. Secondly, raising the event packet generation rate of the source nodes increases the channel contention degree among the active nodes which in turn elevates packet loss ratio and network congestion degree during the data transmission process.

Figure 6 also shows that increasing the number of nodes from

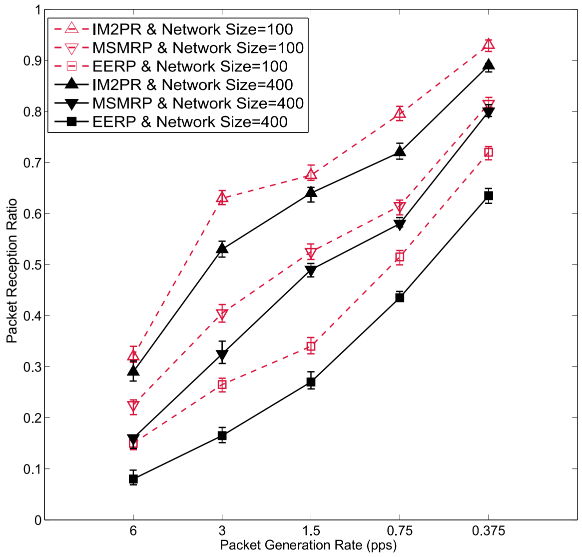

Figure 6: Packet reception ratios achieved by IM2PR, MSMRP, and EERP protocols versus packet generation rate.

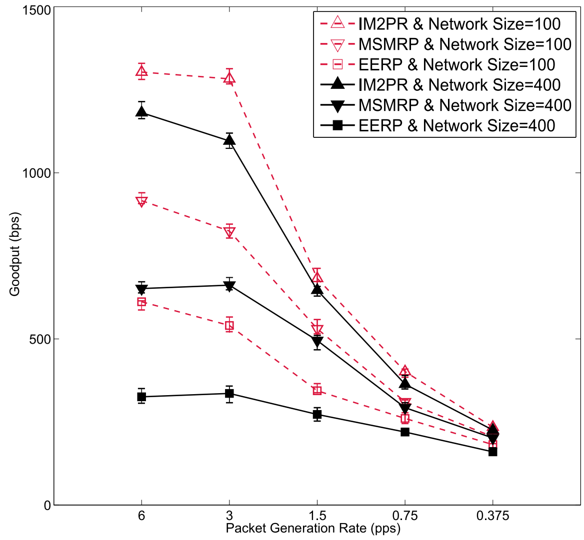

Figure 7: The goodput achieved by IM2PR, MSMRP, and EERP protocols versus packet generation rate.

100 to 400, reduces the PRR provided by all the protocols. The reason is that increasing the network size results in the construction of the paths with more number of hops which in turn elevates the channel contention level among the nodes, interference and packet loss ratio during the data transmission process. However, the proposed IM2PR protocol provides a higher PRR at the sink node compared to the MSMRP, and EERP protocols.

4.3.2. *Goodput*

Figure 7 shows the goodput achieved at the sink node through IM2PR, MSMRP, and EERP protocols against packet generation rate of the source nodes in two different networks with 100 and 400 nodes. The general observation from this figure is that increasing the packet generation rate elevates the goodput achieved through all the considered protocols. As can be seen from this figure, IM2PR improves goodput about 44% and 80% compared to the MSMRP and EERP protocols in a network with 100 nodes. These performance improvements are due to two reasons: Firstly, IM2PR constructs higher capacity paths compared to the other protocols through constructing paths that their concurrent utilization causes minimum inter-path interference level. Secondly, identifying the sufficient number of active paths based on the DRR of the sink node enables the IM2PR to improve goodput.

Furthermore, Figure 7 depicts the effects of network size on the achievable goodput at the sink through IM2PR, MSMRP, and EERP protocols. Generally, increasing the number of nodes from 100 to 400 reduces the goodput achieved through all the considered multipath routing protocols. This is due to the fact that increasing the network size elevates the amount of channel contention among nodes which in turn reduces the capacity of individual paths. However, the proposed IM2PR provides higher goodput compared to the MSMRP, and EERP protocols in the network with 400 nodes.

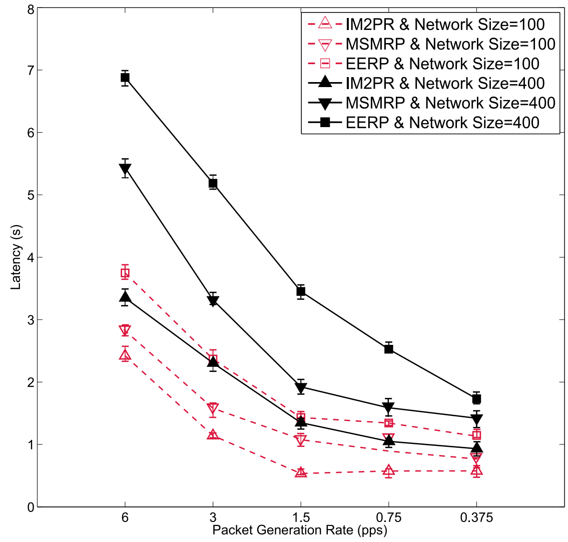

Figure 8: Packet delivery latency of IM2PR, MSMRP, and EERP protocols versus packet generation rate.

4.3.3. *Packet Delivery Latency*

The average packet delivery latency through IM2PR, MSMRP, and EERP protocols in networks with 100 and 400 nodes are depicted in Figure 8. The general observation that can be drawn from this figure is that the average packet delivery latency reduces as the packet generation rate decreases. This is due to the fact that, elevating the network traffic rate causes more packet collisions, wireless interferences, and channel access contentions which in turn intensify the packet delivery latency. Furthermore, as the network traffic load increases, data packets suffer from longer queuing latency at individual nodes. As expected, IM2PR reduces the packet delivery latency by 33% and 40% compared to the MSMRP, and EERP protocols in the network with 100 nodes. Furthermore, in the network with 400 nodes IM2PR provides about 54% and 63% lower

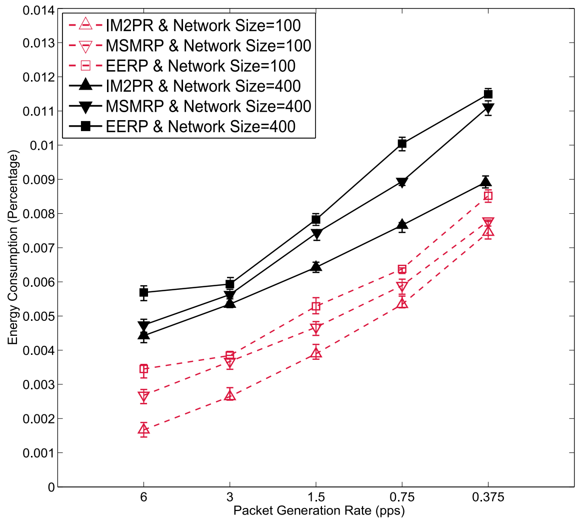

Figure 9: Percentage of the average energy consumed for packet transmission towards the sink node in IM2PR, MSMRP, and EERP protocols versus packet generation rate.

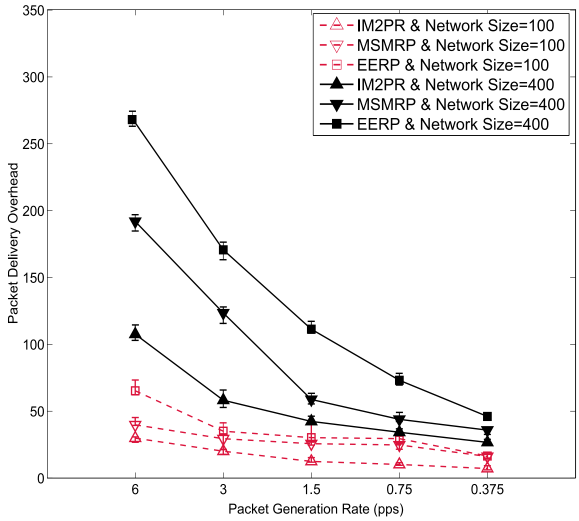

Figure 10: Packet delivery overhead of IM2PR, MSMRP, and EERP protocols versus packet generation rate.

packet delivery latency compared to the MSMRP, and EERP protocols respectively. These performance improvements are direct result of distributing network traffic load over minimum interfering paths. In fact, concurrent data transmission over minimum interfering paths established by IM2PR, highly reduces the channel access contention degree, packet corruption rate and queuing latency. Consequently, IM2PR causes a lower packet delivery latency compared to the other protocols.

4.3.4. *Energy Consumption for Packet Transmission*

Figure 9 presents the percentage of the average energy consumed by the network nodes to transmit the event packets to the sink node through IM2PR, MSMRP, and EERP protocols in the networks with 100 and 400 nodes. Since in all the experiments, source nodes generate identical number of packets with different packet generation rates, reducing the packet generation rate increases the packet generation duration. Therefore, as the packet generation rate reduces, the percentage of the average energy consumed by sensor nodes to transmit data event packets towards the sink node is elevated. The main reason is that reducing the packet generation rate elevates the data transmission duration which in turn makes the network nodes busy for a longer period. As can be seen from this figure, IM2PR reduces the average energy consumed by network nodes in a network with 100 nodes about 40% and 57% compared to the MSMRP, and EERP protocols respectively.

According to Figure 9, increasing the network size elevates the average energy consumption of nodes through all the considered protocols. This behavior is due to the fact that raising the network size causes more packet collisions, wireless interferences, and channel access contentions. Therefore, network nodes along the constructed paths should spend more time at send or receive states, which in turn increases the energy consumption at individual node for transmitting data packets towards the sink. However, in the network size with 400 nodes IM2PR causes lower energy consumption at individual nodes compared to the MSMRP, and EERP protocols. These performance improvements are the direct result of distributing data packets over minimum interfering paths.

4.3.5. *Packet Delivery Overhead*

Figure 10 demonstrates the overhead of running IM2PR, MSMRP and EERP protocols through measuring the ratio of the total number of control and data packets transmitted during the path establishment and data transmission phases to the number of data packets received by the sink node. Based on this figure, increasing the network size elevates the packet delivery overhead caused by all the considered protocols. These incremental trends are due to the fact that increasing the number of sensor nodes causes more number of packet transmissions for route discovery process. Furthermore elevating the network size, results in the construction of the paths with more number of hops which cause data transmission over long distance paths.

As demonstrated in Figure 10, IM2PR reduces the packet delivery overhead by 50% and 60% compared to the MSMRP, and EERP protocols in the network with 100 nodes. Moreover, IM2PR also decreases the packet delivery overhead in the network with 400 nodes about 40% and 57% compared to the MSMRP, and EERP protocols. This can be explained as follows: Firstly, both of the MSMRP and EERP protocols utilize a flooding mechanism to identify several paths from each event area towards the sink node. While, the proposed IM2PR tries to reduce the number of packet transmissions for path construction process through engaging a subset of network nodes in the path establishment process. Secondly, IM2PR transmits the event data packets over minimum interfering paths which causes a lower number of packet corruptions due to the channel access contentions, interference and network congestion. Thirdly, since IM2PR considers the relative position of the links along the paths with respect to their packet delivery probability, it constructs paths which incur lower number of transmissions per packet delivery compared to the other protocols.

5. Conclusion

This paper proposed a multipath routing protocol to provide efficient event packet forwarding in event-driven wireless sensor networks by using different mechanisms during the path construction and data transmission phases. First of all, IM2PR exploits the broadcast nature of wireless communications to construct minimum interfering paths from every event area towards the sink in a localized manner without requiring specific hardware equipments or particular assumptions. Secondly, it considers the limitation on the number of offered link layer transmission attempts at individual links and the relative position of the links along the paths in order to select the paths that incur a minimum number of packet transmissions for every single packet delivery. Thirdly, to achieve the maximum possible network performance, IM2PR determines the efficient number of paths that can be used simultaneously based to the DRR of the sink node. Finally, in IM2PR every source node adjusts the traffic rate of individual paths based on their data delivery probability in the presence of active interfering links and battery capacity.

Simulation comparison studies show the higher performance of the proposed protocol compared to the MSMRP, and EERP protocols in terms of PRR, goodput, latency, energy consumption and packet delivery overhead. The achieved results reveal that constructing minimum interfering paths with high packet transmission quality improves performance of packet delivery in event-driven wireless sensor networks.

As future work, we are intended to enhance the IM2PR protocol to provide fair event packet delivery from different event areas to the sink in the cases where mulitple events coexist in the sensor field. Moreover, according to the operation of the proposed IM2PR protocol, every node that wants to select a next-hop node towards the sink should be aware about the interference level experienced by its neighboring nodes in order to establish minimum interfering paths. In order to reduce the number of packet transmissions during the path construction process, we plan to design efficient mechanisms to adjust the transmission time of the update messages on the amount of wireless interference experienced by individual nodes.

References

[1] Alwan H, Agarwal A (2009) A Survey on Fault Tolerant Routing Techniques in Wireless Sensor Networks. In: Proceedings of the 3th International Conference on Sensor Technologies and Applications (Senosrcomm '09), Athens/Glyfada, Greece, pp 366–371

[2] Baccour N, Kouba A, Mottola L, Zuniga MA, Youssef H, Boano CA, Alves M (2012) Radio Link Quality Estimation in Wireless Sensor Networks : A Survey. ACM Transactions on Sensor Networks 8(4):183–217

[3] Ben-Othman J, Yahya B (2010) Energy Efficient and QoS based Routing Protocol for Wireless Sensor Networks. Journal of Parallel and Distributed Computing 70(8):849–857

[4] Boulfekhar S, Benmohammed M (2013) A Novel Energy Efficient and Lifetime Maximization Routing Protocol in Wireless Sensor Networks. Wireless Personal Communications 72(2):1333–1349

[5] Cheng H, Xiong N, Vasilakos AT, Yang LT, Chen G, Zhuang X (2012) Nodes Organization for Channel Assignment with Topology Preservation in Multi-Radio Wireless Mesh Networks. Ad Hoc Networks 10(5):760–773

[6] Couto D, Aguayo D, Bicket J, Morris R (2005) A High-Throughput Path Metric for Multi-Hop Wireless Routing. Wireless Networks 11(4):419–434

[7] Dezfouli B, Radi M, Razak SA, Whitehouse K, Bakar KA, Hwee-pink T (2014) Improving Broadcast Reliability for Neighbor Discovery, Link Estimation and Collection Tree Construction in Wireless Sensor Networks. Computer Networks 62:101–121

[8] ElBatt T, Andersen T (2006) Cross-Layer Interference-Aware Routing for Wireless Multi-Hop Networks. In: Proceedings of the nternational conference on Wireless communications and mobile computing (IWCMC '06), ACM, Vancouver, Canada, pp 153–158

[9] Gao D, Yang O, Zhang H, Chao HC (2011) Multi-Path Routing Protocol with Unavailable Areas Identification in Wireless Sensor Networks. Wireless Personal Communications 60(3):443–462

[10] Huang X, Fang Y (2007) Multiconstrained QoS Multipath Routing in Wireless Sensor Networks. Wireless Networks 14(4):465–478

[11] Hurni P, Braun T (2008) Energy-Efficient Multi-path Routing in Wireless Sensor Networks. In: Proceedings of the 7th Iternational Conference on Ad-hoc, Mobile and Wireless Networks (ADHOC-NOW '08), Sophia Antipolis, France, pp 72–85

[12] Jain K, Padhye J, Padmanabhan VN, Qiu L (2005) Impact of Interference on Multi-Hop Wireless Network Performance. Wireless Networks 11(4):471–487

[13] Jakllari G, Eidenbenz S (2012) Link Positions Matter : A Noncommutative Routing Metric for Wireless Mesh Networks. IEEE Transactions on Mobile Computing 11(1):61–72

[14] Li C, Zhang H, Hao B, Li J (2011) A survey on Routing Protocols for Large-Scale Wireless Sensor Networks. Sensors 11(4):3498–526

[15] Li M, Li Z, Vasilakos AT (2013) A Survey on Topology Control in Wireless Sensor Networks: Taxonomy, Comparative Study, and Open Issues. Proceedings of the IEEE 101(12):2538 – 2557

[16] Li P, Guo S, Yu S, Vasilakos AT (2012) CodePipe: An Opportunistic Feeding and Routing protocol for Reliable Multicast with Pipelined Network Coding. In: Proccedings of the 31st Annual IEEE International Conference on Computer Communications (INFOCOM '12), Orlando, FL,USA, pp 100–108

[17] Liu Y, Xiong N, Zhao Y, Vasilakos AT, Gao J, Jia Y (2010) Multi-Layer Clustering Routing Algorithm for Wireless Vehicular Sensor Networks. IET communications 4(7):810–816

[18] Lou W (2005) An Efficient N-to-1 Multipath Routing Protocol in Wireless Sensor Networks. In: Proceedings of IEEE International Conference on Mobile Adhoc and Sensor Systems, Washington DC, USA, pp 672–680

[19] Marina MK, Das SR (2001) On-Demand Multipath Distance Vector Routing in Ad Hoc Networks. In: Proceedings of the 9th International Conference on Network Protocols, Riverside, California, USA, pp 14–23

[20] Pearlman M, Haas Z, Sholander P, Tabrizi S (2000) On the Impact of Alternate Path Routing for Load Balancing in Mobile Ad Hoc Networks. In: Proceedings of the 1st Annual Workshop on Mobile and Ad Hoc Networking and Computing (MobiHOC'00), Boston, Massachusetts, pp 3–10

[21] Polastre J, Hill J, Culler D (2004) Versatile Low Power Media Access for Wireless Sensor Networks. In: Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems (SenSys '04), Maryland, USA, pp 95–107

[22] Radi M, Dezfouli B, Razak SA, Bakar KA (2010) LIEMRO: A Low-Interference Energy-Efficient Multipath Routing Protocol for Improving QoS in Event-Based Wireless Sensor Networks. In: Proceedings of the 4th International Conference on Sensor Technologies and Applications (SENSORCOMM '10), IEEE Computer Society, Venice, Italy, pp 551–557

[23] Radi M, Dezfouli B, Bakar KA, Razak SA, Nematbakhsh MA (2011) Interference-Aware Multipath Routing Protocol for QoS Improvement in Event-Driven Wireless Sensor Networks. Tsinghua Science & Technology 16(5):475–490

[24] Radi M, Dezfouli B, Bakar KA, Lee M (2012) Multipath Routing in Wireless Sensor Networks: Survey and Research Challenges. Sensors

12(1):650–685

[25] Radi M, Dezfouli B, Bakar KA, Razak SA, Lee M (2013) Network Initialization in Low-Power Wireless Networks: A Comprehensive Study. The Computer Journal In Press:1–24, DOI 10.1093/comjnl/bxt074

[26] Radi M, Dezfouli B, Bakar KA, Razak SA (2014) Integration and Analysis of Neighbor Discovery and Link Quality Estimation in Wireless Sensor Networks. The Scientific World Journal 2014:1–23

[27] Roy S, Bandyopadhyay S, Ueda T, Hasuike K (2002) Multipath Routing in Ad Hoc Wireless Networks with Omni Directional and Directional Antenna: A Comparative Study. In: Proceedings of the 4th International Workshop on Distributed Computing, Mobile and Wireless Computing (IWDC '02), London, UK, pp 184–191

[28] Son D, Krishnamachari B, Heidemann J (2006) Experimental Study of Concurrent Transmission in Wireless Sensor Networks. In: Proceedings of the 4th International Conference on Embedded Networked Sensor Systems (SenSys '06), Boulder, Colorado, USA, pp 237–250

[29] Spyropoulos T, Rais RNB, Turletti T, Obraczka K, Vasilakos AT (2010) Routing for Disruption Tolerant Networks: Taxonomy and Design. Wireless Networks 16(8):2349–2370

[30] Tam Wh, Tseng Yc (2007) Joint Multi-Channel Link Layer and Multi-Path Routing Design for Wireless Mesh Networks. In: Proceedings of the 26th IEEE International Conference on Computer Communications (INFOCOM '07), Anchorage, AK, pp 2081–2089

[31] Tarique M, Tepe KE, Adibi S, Erfani S (2009) Survey of Multipath Routing Protocols for Mobile Ad Hoc Networks. Journal of Network and Computer Applications 32(6):1125–1143

[32] Teo JY, Ha Y, Tham CK (2008) Interference-Minimized Multipath Routing with Congestion Control in Wireless Sensor Network for High-Rate Streaming. IEEE Transactions on Mobile Computing 7(9):1124–1137

[33] Wang X, Garcia-luna aceves JJ (2008) Embracing Interference in Ad Hoc Networks Using Joint Routing and Scheduling with Multiple Packet Reception. Ad Hoc Networks 7(2):460–471

[34] Wang Z, Bulut E, Szymanski BK (2009) Energy Efficient Collision Aware Multipath Routing for Wireless Sensor Networks. In: Proceedings of the 2009 IEEE International Conference on Communications (ICC'09), Dresden, Germany, pp 91–95

[35] Whitehouse K, Woo A, Jiang F, Polastre J, Culler D (2005) Exploiting the Capture Effect for Collision Detection and Recovery. In: Proceedings of The 2nd IEEE Workshop on Embedded Networked Sensors, Sydney, Australia, pp 45–52

[36] Wu K, Harms J (2001) On-Demand Multipath Routing for Mobile Ad Hoc Networks. In: Proceedings of the 4th European Personal Mobile Communications Conference (EPMCC'2001), Vienna, Austria, February, pp 14 – 23

[37] Xiang L, Luo J, Vasilakos AT (2011) Compressed Data Aggregation for Energy Efficient Wireless Sensor Networks. In: Proccedings of the 8th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON '11), Salt Lake City, UT, pp 46–54

[38] Yan B, Gharavi H (2006) Multi-Path Multi-Channel Routing Protocol. In: Proceedings of the 5th IEEE International Symposium on Network Computing and Applications (NCA '06), Cambridge, Massachusetts, pp 27–31

[39] Zamalloa MZn, Krishnamachari B (2007) An Analysis of Unreliability and Asymmetry in Low-Power Wireless Links. ACM Transactions on Sensor Networks 3(2):165–199

[40] Zeng Y, Xiang K, Li D, Vasilakos AT (2013) Directional Routing and Scheduling for Green Vehicular Delay Tolerant Networks. wireless Networks 19(2):161–173

[41] Zhou G, He T, Krishnamurthy S (2006) Models and Solutions for Radio Irregularity in Wireless Sensor Networks. ACM Transactions on Sensor Networks 2(2):221–262