

Research Article

REALFLOW: Reliable Real-Time Flooding-Based Routing Protocol for Industrial Wireless Sensor Networks

Kan Yu,¹ Zhibo Pang,² Mikael Gidlund,² Johan Åkerberg,² and Mats Björkman¹

¹ School of Innovation, Design and Engineering, Mälardalen University, P.O. Box 883, 72123 Västerås, Sweden

² ABB AB, Corporate Research, Tegnér, 72178 Västerås, Sweden

Correspondence should be addressed to Kan Yu; kan.yu@mdh.se

Received 12 December 2013; Accepted 23 May 2014; Published 15 July 2014

Academic Editor: Shancang Li

Copyright © 2014 Kan Yu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Wireless technologies have been increasingly applied in industrial automation systems due to flexible installation, mobility, and cost reduction. Unlike traditional wireless sensor networks (WSNs), industrial wireless sensor networks (IWSNs), when expanding from wireless monitoring to wireless control, have more stringent requirements on reliability, real-time performance, and robustness in a number of industrial applications. Successive transmission failures or deadline misses in these applications may severely degrade the control quality and result in serious economic losses and safety problems. Therefore, when deploying IWSNs in harsh industrial environments, to achieve reliable and deterministic end-to-end transmissions is critically important. In this paper, we explain the primary challenges of designing appropriate routing protocols and present a reliable real-time flooding-based routing protocol for IWSNs (REALFLOW). Instead of traditional routing tables, related node lists are generated in a simple distributed manner, serving for packet forwarding. A controlled flooding mechanism is applied to improve both reliability and real-time performance. A seamless transition in the event of topology change can be achieved by REALFLOW. Performance evaluations via simulations verify that significant improvements of reliability, real-time performance, and network recovery time can be achieved by REALFLOW, compared with traditional routing protocols.

1. Introduction

Nowadays, wireless sensor networks (WSNs) have been exhibiting their attractive advantages over traditional wired counterpart for industrial automation systems, such as the avoidance of cabling and flexible installation. Industrial wireless sensor networks (IWSNs) can serve a number of purposes, such as monitoring and control. Currently, wireless monitoring via IWSNs for industrial applications has been widely applied [1]. Nevertheless, wireless control, as an essential part of automation, is still lacking support in IWSNs [2]. Compared with wireless monitoring, data transmissions for wireless control in many industrial applications should be sufficiently reliable [3] and deterministic with the latency of the order of seconds, or even milliseconds [2]. Transmission failures or deadline misses may result in disturbances to the process, degradation of the overall control performance, and even more serious economic losses or human safety problems. Once both wireless monitoring and control are

fully supported, the booming development of IWSN applications can be expected. Besides high reliability and real-time performance, low energy consumption is also an important issue [4–6]. However, compared with the former two requirements, reducing energy consumption should not be prioritized as high as reliable and deterministic transmission for different industrial applications.

Compared with environments for traditional WSNs, industrial environments for IWSNs are harsher and more dynamic due to a great number of metallic surfaces, extreme temperature, high vibrations, and mobility of nodes and other objects [7]. Measurements from [8, 9] have shown that deep fading and shadowing in industrial environments may result in extremely low *received signal strength indicator* (RSSI) values with high variances or even packet losses. Moreover, authors in [7, 10] pointed out coexisting communication systems in the same frequency band as another major source of disturbances to wireless industrial applications. Hence, the

major challenge is to achieve reliable and deterministic end-to-end transmissions in industrial environments by using IWSNs. Currently, several standards have been published for process measurements and control applications. Among them, Zigbee [11] is shown to be unsuitable for industrial automation because there are no frequency diversity, no path diversity, and the MAC unreliability problem [12, 13]. WirelessHART [14] and ISA 100.11a [15] are another two standards designed for process automation (PA). Although some successful deployment instances exhibit the confidence on these standards, there is still a long way to go to fulfill the reliability and real-time performance requirements for most of industrial applications [2]. The improvement can be achieved on different layers. On existing devices, limited efforts can be taken on the physical layer due to restrictions imposed by hardware and/or software limitations. On the MAC layer, a number of techniques, such as channel hopping and black listing, are used to improve reliability, but in harsh industrial environments, transmission failures still occur quite often. Although automatic repeat-request (ARQ) is applied in most of the standards, and it is a straightforward method to increase reliability, the real-time performance is degraded due to retransmissions. The previous empirical results [16] exhibit serious consequences such as network congestions, caused by excessive unexpected retransmissions. On the network layer, the routing protocol plays an extremely important role in achieving high reliability and low latency [1], but the current standards fail to provide sufficient guidance on how to guarantee reliable and real-time transmission by using appropriate routing protocols. Although a number of research efforts have been taken to design a reliable real-time routing protocol for IWSNs [17–22], there is still no sufficient evidence to show that the problem has been fully solved.

There are several challenges to design an appropriate routing protocol for IWSNs. Firstly, if high reliability cannot be achieved by the lower layers, multipath diversity can be applied by routing protocols on the network layer to increase reliability. Retransmission is an effective method to increase reliability, but the transmission delay is also prolonged. For many industrial applications where hard deadline is strictly required, outdated packets are of limited use for their destinations. Therefore, the second challenge is that real-time performance should be provided by a routing protocol. Thirdly, routing protocols should be tolerant of sudden topology changes, such as link failures and node halting. Once the topology changes, the routing protocol should provide an alternative path as soon as possible; otherwise data transmissions may be terminated during the path recalculation period. The last challenge is that the workload of nodes should be also considered. Different from traditional WSNs, the network structure of an IWSN is typically centralized, since operators in the central control room must have the knowledge of the status of the whole network. Thus, the gateway or network manager, as the central device, is involved in most of network activities. The gateway or network manager should not spend too long time on calculating routing information, since partial communication in the network may be halted during the calculation period due to the outdated routing information.

Therefore, when designing a routing protocol for IWSNs, these challenges should be addressed.

In this paper, we propose a reliable real-time flooding-based routing protocol for IWSNs (REALFLOW) to address all challenges mentioned above. REALFLOW consists of a routing establishing and maintenance part and packet forwarding part. The gateway periodically broadcasts discovering messages, named as list-updated messages, to discover the current network topology. After list-updated messages are received, all nodes should send corresponding responses, named as list-updated messages, back to the gateway. The list-response message transmission stage has two purposes: one is that it helps the gateway obtain the current network topology information; the other is that related node lists are generated in all nodes based on receiving list-response messages from other nodes. For both uplink and downlink transmissions, packets are forwarded according to the related node lists in all intermediate nodes. In order to improve reliability, REALFLOW is based on the controlled flooding mechanism to provide multipath diversity. The real-time performance is also considered, so outdated packets are automatically discarded in intermediate nodes. Due to redundant paths and flooding mechanism, REALFLOW can be tolerant of parts of network topology changes. Since related node lists are distributively generated in all nodes, the workloads of the gateway are greatly reduced.

Due to flooding mechanism, more resources may be required than traditional routing protocols in WSNs. However, as we emphasized previously, in many industrial applications, especially those from industrial automation systems, reliability and real-time performance should be prioritized for IWSNs, rather than energy consumption. Moreover, it is proven that energy consumed by local processing, such as sensing, dominates the overall energy consumption [23]. Although additional communication is introduced in the network due to flooding, the increment of energy consumption caused by additional communication is not obvious. Therefore, our work bridges the gap of current research works to provide reliable and real-time end-to-end transmissions in IWSNs.

The rest of this paper is organized as follows: Section 2 presents the previous work on routing protocols in WSNs. In Section 3, we briefly describe the current IWSN architecture as background knowledge. In Section 4, we outline the proposed routing protocol REALFLOW. The simulation settings and scenarios are described in Section 5, followed by the simulation result analysis in Section 6. Finally, we conclude the paper in Section 7.

2. Related Work

Designing reliable routing protocols in WSNs, as one of the most challenging research topics, has attracted a great number of research interests. Traditional routing protocols, such as dynamic source routing protocol (DSR) [24] and ad hoc on-demand distance vector protocol (AODV) [25], have been successfully applied in traditional WSNs, such as

Zigbee networks. For this set of routing protocols, route discovery messages are broadcast to discover all available paths. In order to maintain the updated routing information, all nodes have to frequently exchange information for being aware of channel status. If any link is broken or node halts, all corresponding nodes have to recalculate the routing information and try to establish another new path. During this period, parts of the network may not work properly, which makes the network suffer from higher latency and lower reliability. Several extensions of traditional routing protocols were also proposed, for instance, the ad hoc on-demand multipath distance vector protocol (AOMDV) [26], the ad hoc on-demand distance vector multipath routing protocol (AODVM) [27], and split multipath routing (SMR) [28]. These extensions intend to improve reliability by identifying multiple nodes or edge-disjoint paths. However, primary drawbacks, such as excessive control messages and unpredictable network recovery time, still exist. Besides these extensions, multipath routing protocols have been extensively studied in other research works aiming for high reliability [17–20]. Authors in [19] split each packet into several subpackets by using erasure codes and send them instead of the whole packet. This scheme might be effective in traditional WSNs, but in IWSNs there exist very short packets with the strict deadline; this scheme may be hardly used in IWSNs, since the real-time performance is not fully considered. A reliable information forwarding using multiple paths (ReInForM) protocol proposed in [20] can achieve controlled reliability by sending redundant copies of a packet along multiple paths. However, this protocol can only be used under a number of assumptions. For instance, every node has knowledge of the local channel error and is able to compute the information content and importance of sensed events, which can hardly be obtained in many practical scenarios.

Unlike traditional routing protocols and all their extensions, flooding is an effective approach to increase reliability by multipath diversity. However, previous flooding-based routing protocols fail to satisfy industrial requirements. Authors in [29] try to avoid the drawbacks of flooding by randomizing the selection of retransmitters. Their approach may hardly be used for a centralized TDMA-based IWSN, since it may bring difficulties for scheduling. A random routing strategy based on flooding proposed in [30] aims for low energy consumption. However, according to the evaluation results, even higher end-to-end transmission latency may be introduced, although lower energy consumption is provided by their proposed routing algorithm.

Except for reliable routing protocols in traditional WSNs, there also exist a number of newly proposed routing protocols to address the challenges from IWSNs. MERLIN protocol proposed in [21] utilizes multicast for both uplink and downlink transmissions to improve low latency and energy consumption, but high reliability is not the main scope of this protocol. Authors in [31] proposed EARQ, an energy aware routing protocol also for reliable and real-time communications in IWSNs. Next hop selections are based on the estimations of energy consumption, reliability, and deadlines. However, according to their evaluation settings, this protocol may be more suitable for WLAN or Zigbee networks, not

centralized TDMA IWSNs. Both authors in [32, 33] proposed a two-hop information-based routing protocol, aiming for enhancing real-time performance with energy efficiency. The routing decision in [32] is based on the two-hop velocity integrated with energy balancing mechanism, whereas the routing decision in [33] is based on the number of hops from source to gateway and two-hop information. However, reliability is not fully considered in these two protocols. Authors from [22] also proposed an entire reliable graph routing scheme for broadcast, uplink, and downlink transmissions in IWSNs with promising evaluation results. One drawback of their routing protocol is the too high workload of the gateway or the network manager to calculate all routing graphs [34].

It is notable that energy efficiency and low power consumption have been stressed by a number of previous research works. However, for many time-critical industrial applications, the high reliability and real-time performance are prior to all other requirements. If there are many packet losses or packets cannot arrive at their destinations before deadlines, optimizing energy efficiency and reducing power consumption become meaningless to the whole system. Reliability, availability, and usability of IWSNs should take top priority for those industrial applications. Therefore, different from previous research works, we mainly focus on improving reliability and real-time performance for IWSNs.

3. Industrial Wireless Sensor Network Architecture

Unlike traditional WSNs, the network structure of an IWSN is generally centralized. Operators in the central control room should have the knowledge of the status of the whole network. A comprehensive IWSN usually contains a number of components, such as network manager, access point, security manager, and field devices. However, the scope of this work is to design routing protocol for reliable and real-time transmissions in IWSNs, so simplified components are considered in this work. Figure 1 shows a typical topology of an IWSN considered in this work. Three basic device types are involved in the formation of an IWSN.

- (i) Gateway: it is responsible for managing the whole network, including identifying routing, distributing resources, and scheduling decision making. It also connects the control system to field devices.
- (ii) Sensor nodes: as one type of field devices, the responsibility of a sensor node is to collect all kinds of measurement data and upload to the gateway.
- (iii) Actuator nodes: as another type of field devices, an actuator node performs basic functions of actuating after receiving data or commands from the gateway.

As shown in Figure 1, in an industrial process control system, a programmable logic controller (PLC) connected to a gateway periodically acquires measurement data from sensor nodes at a certain refresh rate. After executing the control applications, the PLC periodically sends the output values to actuator nodes for actuating. Usually sensor and actuator nodes are deployed in a scattered manner in a

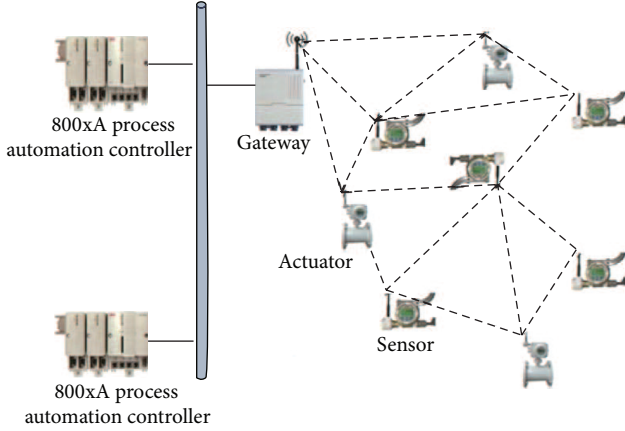


FIGURE 1: A typical topology of an IWSN.

large area. According to different wireless channel conditions, uplink and downlink communication may involve one hop or multihop packet transmissions. In order to provide reliable and real-time communication for industrial applications, an appropriate routing protocol serving packet forwarding plays an extremely important role in finding available transmission paths. We should also notice that in many industrial applications the data transmission period is extremely short, for example, 250 ms or 500 ms [2]. Due to the schedulability of transmission timing slots, the network size and node hop numbers are severely limited. Therefore, in many industrial applications an IWSN containing hundreds of nodes and a node with many hops away from the gateway can hardly be seen in those industrial applications.

4. Proposed Reliable Real-Time Flooding-Based Routing Protocol

To achieve higher reliability, flooding method is used in our proposed routing protocol to introduce multipath diversity. However, the inherent drawback of uncontrolled flooding is that the network resources will be rapidly exhausted. Thus, our proposed routing protocol is based on a controlled flooding mechanism in which flooding behaviors are severely restricted within a certain range, so the usage of network resources can be much more efficient. To assist packets in arriving at their destinations, different routing protocols apply different mechanisms. For instance, routing tables are commonly used in a great number of traditional routing protocols, whereas a graph ID embedded in a message is used by graph routing protocols for forwarding messages in the standard WirelessHART. As we mentioned previously, to generate all routing or graph information at the gateway or network manager may require extremely long computation time and lead to overwhelming workload of these devices [34]. Additionally, once the network topology changes, routing tables and graphs need to be recalculated again. During the calculation and recalculation period, partial communication in the network may be terminated due to the outdated routing

information. To avoid this problem, we apply a different mechanism and introduce new routing parameters.

In this section, we present the details of REALFLOW. We first introduce the definitions and notations for our proposed protocol. Then we describe the route establishing and maintenance method and packet forwarding method in Section 4.2 and Section 4.3, respectively, followed by an example of our proposed scheme for comprehensive understanding. Finally, a theoretical analysis is given in Section 4.5.

4.1. Definitions and Notations. Provided an IWSN consists of m nodes (sensors and actuators), the node set \mathbb{N} is written as $\mathbb{N} = \{N_1, N_2, \dots, N_m\}$. In order to control the flooding behavior efficiently, we introduce a new concept, called *relevant*, in our proposed routing protocol. This concept is defined as below.

Definition 1. If the node N_i ($1 \leq i \leq m$) in the network is involved in forwarding packets transmitted between the node N_k ($1 \leq k \leq m, k \neq i$) and the gateway G , the node N_i is a relevant node to the node N_k .

In our proposed routing protocol, a node is only able to forward packets sent from or to its relevant node; otherwise packets from or to irrelevant nodes will be dropped by this node. In order to distinguish relevant nodes from irrelevant nodes, each node in the network should maintain a list, named the related node list \mathbb{L} . The *related node* list of the node N_k in the network is defined as \mathbb{L}_k . The list \mathbb{L}_k consists of a set of nodes with their source addresses. Provided the length of the list \mathbb{L}_k is p , $\mathbb{L}_k = \{A_1, A_2, \dots, A_p\}$.

Definition 2. The source address node N_i is A_i , and the node N_k has the list \mathbb{L}_k . If $A_i \in \mathbb{L}_k$, N_i is a relevant node of the node N_k .

Sensor and actuator nodes are usually deployed dispersedly in a large area. Due to the communication distances, interferences, obstacles, and other aspects, packets may require more than one transmission hop to arrive at the destination. To describe the network topology, we define parent, child, and sibling nodes as below.

Definition 3. If N_i is the relevant node of N_k , and N_i has smaller hop number away from the gateway than N_k , N_i is a parent node of N_k and N_k is a child node of N_i .

Definition 4. If N_i is the relevant node of N_k , and N_i has same hop number away from the gateway than N_k , N_i is a sibling node of N_k .

The key point of REALFLOW is to obtain and maintain the related node list \mathbb{L} in each node. Several steps are required before regular data transmissions. The establishment and maintenance of routing information and related node lists are described in detail in the following subsections.

4.2. Establishing and Maintaining Routes. In order to obtain the current network topology for the gateway and generate

related node lists \mathbb{L} in all nodes in the network, the gateway periodically broadcasts discovering messages. The discovering messages are named as *list-updated messages*. Each list-updated message contains three important parameters.

- (1) r_{pkt} (the absolute accumulated RSSI): it is initialized as zero. When a node receives a message, the RSSI value obtained from the radio chip register will be accumulatively added to the previous value. Thus, this parameter is updated at every hop.
- (2) h_{pkt} (the packet hop): it is initialized as zero. Every time when a message goes through an intermediate node, this parameter is increased by one.
- (3) A_{prev} (the previous node address): every time when a message is forwarded, the address of this forwarding node will be saved in this parameter.

When a node receives a list-updated message, it will rebroadcast the message as needed. Before the message is forwarded, all those three parameters will be updated. The node also needs to record the information obtained from the list-updated message. In order to establish the related node list, each node needs to maintain three important parameters:

- (1) h_{node} (the node hop number): this parameter represents the hop number between the current node and gateway,
- (2) C (the number of received list-updated messages): once the node receives a new list-updated message, this parameter should be increased by one,
- (3) \mathbb{V} (the parent or sibling node record data set): once a node receives a list-updated message from its parent or sibling node, the address of this node, as well as the r_{pkt} value, will be added to this data set. This record history is extremely important for the next related node list generation stage.

Once the gateway broadcasts a list-updated message, this message should be rebroadcast by all other nodes until it propagates in the entire network. The rebroadcasting procedure is summarized in Algorithm 1. The notations mentioned in the algorithm are summarized as below

- (1) T_{link} : a predefined link threshold to filter out packets with weak signal strength,
- (2) r_{link} : the absolute value of measured RSSI of the current received message,
- (3) A_{current} : the address of the current node,
- (4) K_{max} : a predefined maximum allowed parent node number.

In order to establish appropriate routing paths, the received signal strength of the list-updated message r_{link} should be larger than T_{link} ; otherwise this path is considered to be unstable and unreliable. T_{link} is predefined according to different wireless environments. In order to prevent list-updated messages from endless rebroadcasting, h_{pkt} will be checked at every hop. If h_{pkt} is larger than h_{node} , it means

```

(1) Extract  $r_{\text{pkt}}$ ,  $h_{\text{pkt}}$ , and  $A_{\text{prt}}$  from the list-updated message
(2) Obtain  $r_{\text{link}}$  from RF part
(3) if  $r_{\text{link}} < T_{\text{link}}$  then
(4)   if  $h_{\text{pkt}} \leq h_{\text{node}}$  then
(5)      $r_{\text{pkt}} = r_{\text{pkt}} + r_{\text{link}}$ 
(6)      $C = C + 1$ 
(7)     if  $C < K_{\text{max}}$  then
(8)       if  $h_{\text{pkt}} = h_{\text{node}}$  then
(9)          $\mathbb{V} = \mathbb{V} \cup \{A_{\text{prt}}, r_{\text{pkt}}, \text{sibling}\}$ 
(10)      else if  $h_{\text{pkt}} < h_{\text{node}}$  then
(11)         $\mathbb{V} = \mathbb{V} \cup \{A_{\text{prt}}, r_{\text{pkt}}, \text{parent}\}$ 
(12)         $h_{\text{node}} = h_{\text{pkt}} + 1$ 
(13)      end if
(14)    else if  $C \geq K_{\text{max}}$  then
(15)      if  $r_{\text{pkt}} < \max(\mathbb{V}, r_i)$  then
(16)         $\mathbb{V} = \mathbb{V} - \{r_i\}$ 
(17)        if  $h_{\text{pkt}} = h_{\text{node}}$  then
(18)           $\mathbb{V} = \mathbb{V} \cup \{A_{\text{prt}}, r_{\text{pkt}}, \text{sibling}\}$ 
(19)        else if  $h_{\text{pkt}} < h_{\text{node}}$  then
(20)           $\mathbb{V} = \mathbb{V} \cup \{A_{\text{prt}}, r_{\text{pkt}}, \text{parent}\}$ 
(21)           $h_{\text{node}} = h_{\text{pkt}} + 1$ 
(22)        end if
(23)      else
(24)        Drop
(25)      end if
(26)    end if
(27)     $h_{\text{pkt}} = h_{\text{pkt}} + 1$ 
(28)     $A_{\text{prt}} = A_{\text{current}}$ 
(29)    Forward
(30)  else if  $h_{\text{pkt}} > h_{\text{node}}$  then
(31)    Drop
(32)  end if
(33) else if  $r_{\text{link}} \geq T_{\text{link}}$  then
(34)   Drop
(35) end if

```

ALGORITHM 1: List-updated message rebroadcasting procedure.

that this message comes from a child node and should be discarded immediately. Different K_{max} value indicates different multipath transmission properties. If more parent and sibling nodes are selected, more paths will be involved in data transmissions. Thus, by varying K_{max} , the transmission reliability may also be changed. Although enlarging K_{max} will increase the reliability performance, more network resources are also needed due to incremental transmission paths. Therefore, there is a trade-off between the reliability and network resource consumption when determining the value K_{max} .

After receiving list-updated messages, each node shall broadcast a respond to the gateway, named *list-response messages*. A list-response message also includes three necessary parameters:

- (1) s_{pkt} (the packet sequence number): as a unique indicator to avoid packet duplications,

```

(1) Extract  $s_{\text{pkt}}, A_{\text{src}}$  and  $N_{\text{fwd}}$  from the list-response message
(2) if  $\{s_{\text{pkt}}, A_{\text{src}}\} \notin \mathbb{H}$  then
(3)    $\mathbb{H} = \mathbb{H} \cup \{s_{\text{pkt}}, A_{\text{src}}\}$ 
(4)   if  $A_{\text{current}} \in N_{\text{fwd}}$  then
(5)     if  $A_{\text{current}}$  is a parent node type then
(6)        $N_{\text{fwd}} = V_{\text{current}}$ 
(7)     else if  $A_{\text{current}}$  is a sibling node type then
(8)       Find  $\{A_i, r_i, \text{parent/sibling}\}$  where  $r_i$  is minimum in  $V_{\text{current}}$ 
(9)        $N_{\text{fwd}} = \{A_i, r_i, \text{parent/sibling}\}$ 
(10)    end if
(11)     $\mathbb{L} = \mathbb{L} \cup A_{\text{src}}$ 
(12)    Forward
(13)  else if  $A_{\text{current}} \notin N_{\text{fwd}}$  then
(14)    Drop
(15)  end if
(16) else if  $\{s_{\text{pkt}}, A_{\text{src}}\} \in \mathbb{H}$  then
(17)   Drop
(18) end if

```

ALGORITHM 2: List-response message forwarding and related node list generation procedure.

- (2) N_{src} (the accepted parent and sibling node set): containing the addresses of accepted parent and sibling node obtained from previous list-updated messages,
- (3) N_{fwd} (the next hop node set): containing the next hop node addresses, as well as node types.

In order to identify different list-response messages from a same node, the sequence number s_{pkt} is introduced and always increased by one after each response. N_{src} should be reported to the gateway to calculate the current network topology, and N_{fwd} is used to find out the next hop.

The list-updated message transmission has two purposes. Firstly, the gateway is able to obtain sufficient information to calculate the current network topology according to the content of these messages. Secondly, the related node list \mathbb{L} generation in each node is based on list-response messages. Both list-response messages forwarding procedure and related node list-generation procedure in a node are summarized in Algorithm 2. The notations that appeared in the algorithm are summarized as below:

- (1) A_{src} : the source address of the list-response message sent by the node N_{src} ,
- (2) V_{current} : the parent and sibling node set of the current node obtained from the previous list-updated message stage,
- (3) \mathbb{H} : a history table to record all seen messages in order to prevent duplicate forwarding.

Since the combination of s_{pkt} and the source address of the message can identify different list-updated messages, if these messages are not found in the current history table, they can be accepted for the next step. N_{fwd} contains the next hop node addresses. Thus, if the current node address A_{current} is seen in N_{fwd} , this message is allowed to be forwarded. The relationship between the current node and previous node

determines the next hop behavior. If the current node is a parent node of the previous node, N_{fwd} is simply replaced by V_{current} . If the current node is a sibling node of the previous node, the current node needs to find one of its parent or sibling nodes with the minimum accumulative RSSI r_i as the next hop node. The last step before forwarding is to add the source address of this list-updated message A_{src} to the local related node list \mathbb{L} . The arrival range of the list-updated message directly determines the flooding range of the node N_{src} . In order to limit the flooding range and avoid excessive packet flooding, the arrival range of the list-updated message should be severely restricted. If the list-updated message arrives at a sibling node, this message is still the same hops away from the gateway. Therefore, we only choose the node with the minimum accumulative RSSI to be the next hop to prevent enlarging the flooding range. After this stage, the gateway is able to get the latest network topology, and a related node table is, respectively, generated in each node.

According to the existing IWSN standards, IWSNs are typically centralized, and TDMA mechanism is applied on the MAC layer. Thus, a TDMA scheduling decision shall be made by the gateway, so each node in the network has the knowledge of its available timeslots for sending and receiving packets. Since designing a TDMA scheduling scheme is out of the scope of this paper, a simplified TDMA scheduling is utilized. Apparently, a parent node requires more timeslots than its child node, since the parent node is involved in forwarding packets for its child node. The number of timeslots required by a parent node depends on the number of its child nodes, the data refresh rate of its child nodes, and the data refresh rate of itself. Therefore, to make an appropriate scheduling decision, the network topology needs to be generated at the gateway first.

As we defined previously, an IWSN consists of m nodes, excluding the gateway. A_k is the address of the node N_k . N_k is the parent node set of the node N_k , obtained from its

```

(1)  $n = m$ 
(2)  $\mathbb{M} = \mathbb{M} \cup \{A_{gw}\}$ 
(3) while  $n \neq 0$  do
(4)   for all  $A_i$  such that  $A_i \in \mathbb{M}$  do
(5)     for all  $\mathbb{N}_k, k = 1, \dots, m$  do
(6)       if  $A_i \in \mathbb{N}_k$  then
(7)         Set  $N_k$  as a child or sibling of  $N_i$ .
(8)          $\mathbb{N}_k = \mathbb{N}_k - \{A_i\}$ 
(9)         if  $\mathbb{N}_k = \emptyset$  then
(10)           $n = n - 1$ 
(11)        end if
(12)        if  $A_k \notin \mathbb{M}$  then
(13)           $\mathbb{M} = \mathbb{M} \cup \{A_k\}$ 
(14)        end if
(15)      end if
(16)    end for
(17)  end for
(18) end while

```

ALGORITHM 3: Network topology tree generation.

list-response message. \mathbb{M} is the node set already processed by the topology generation algorithm. The generation of the network topology tree is summarized in Algorithm 3. The generation procedure is quite straightforward, since sufficient information can be extracted from list-response messages to describe the relationship of two nodes in the network. Afterwards the simplified scheduling scheme can also be calculated based on the latest network topology. It is notable that exploring the optimal scheduling scheme of the routing protocol to maximize the resource efficiency is out of the scope of this paper.

Once the scheduling decision is made, the gateway sends a *list-confirmed message* to each node as a new routing confirmation. Moreover, the list-confirmed message also contains the latest scheduling decision. The forwarding scheme of list-confirmed messages is identical to the packet forwarding scheme, which is described in detail in the next subsection. Once a node receives its list-confirmed message, both new related node list and scheduling decision will be applied. As a centralized network, before sending out *list-confirmed messages*, the gateway is able to change the network topology. If the gateway intends to change the topology, it needs change the calculated topology information locally. Then the gateway encloses a command in the list-confirmed message to inform the corresponding node to delete a certain node in its related node list \mathbb{L} .

4.3. Packet Forwarding Method. After all nodes successfully generate and apply their latest related node list and scheduling decisions, they are able to forward packets. As we described before, the inherent drawback of flooding transmission is excessive packet forwarding. Thus, related node lists are used in our proposed routing protocol to restrict the flooding range. Now we can formulate packet forwarding

criteria of REALFLOW in detail. The first forwarding criterion is written as

$$A \in \mathbb{L}. \quad (1)$$

This criterion indicates that if a node receives a packet with the source address or destination address A , which can be found in its local related node list \mathbb{L} , this packet can be considered to be forwarded.

However, since flooding is applied rather than unicasting, duplicated packets may appear. In order to filter out duplicated packets and avoid unnecessary packet forwarding, each outgoing data packet must contain a unique identifier. A unique pair P is defined as the identifier. P consists of the sequence number s_{pkt} and the source address A_{src} of the current node, $P = (s_{pkt}, A_{src})$. After each data transmission, the sequence number s_{pkt} is increased by one. Each node should also maintain a history table \mathbb{H} . Once a new data packet is received, its P is recorded in \mathbb{H} . If P appears in \mathbb{H} , it indicates that a duplicate packet is received. Then, the second forwarding criterion can be expressed as

$$P \notin \mathbb{H}. \quad (2)$$

As mentioned previously, real-time performance is one of the stringent requirements of IWSNs. Thus, data packet should arrive at its destination before the deadline. In many industrial applications, outdated packets are of limited use for industrial systems. It is more reasonable to discard outdated packets in the intermediate node to save the network resources for other transmissions. The final forwarding criterion of our scheme is

$$T_{age} < T_{refresh}, \quad (3)$$

where T_{age} is the packet age and $T_{refresh}$ is the refresh interval of its originating node.

Theoretically the refresh interval $T_{refresh}$ equals the packet deadline, so a packet should arrive at the destination within $T_{refresh}$. The packet age should be checked before being sent out, since packet buffering in the intermediate node may introduce additional delay.

Finally, the entire packet forwarding scheme for both uplink and downlink is summarized in Algorithm 4.

So far the details of our proposed routing protocol REALFLOW have been presented. In summary, four proceeding steps and four message types are involved in REALFLOW. First, the gateway broadcasts list-update messages to discover the latest network topology. Then each node replies with a list-response message to report their parent nodes and add itself into the related node lists in all intermediate nodes. After receiving all responses from all nodes, the gateway is able to obtain the latest network topology and calculate the latest TDMA scheduling decisions and send them back to all nodes by list-confirm messages. Finally, all nodes are able to communicate with the gateway based on controlled flooding according to our forwarding rules for both uplink and downlink. A summary of all steps and corresponding messages is shown in Figure 2.

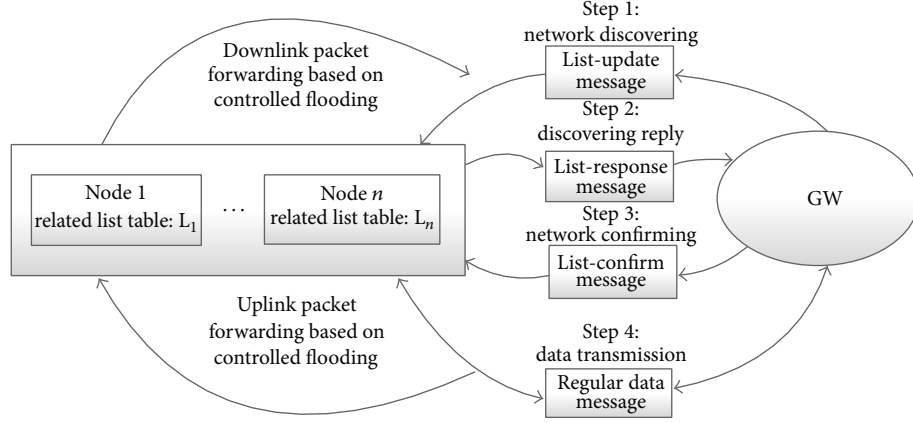


FIGURE 2: REALFLOW proceeding summary.

```

(1) Extract  $A_{src}$ ,  $A_{dst}$  and  $s_{pkt}$  from the received packet
(2) if  $A_{src} = A_{gw}$  then
(3)   if  $A_{dst} \in \mathbb{L}$  then
(4)     Construct the unique pair  $P = (s_{pkt}, A_{dst})$ 
(5)     if  $P \notin \mathbb{H}$  then
(6)       Update  $T_{age}$ 
(7)       if  $T_{age} < T_{refresh}$  then
(8)          $\mathbb{H} = \mathbb{H} \cup \{P\}$ 
(9)       Forward
(10)    end if
(11)   end if
(12)  end if
(13) else if  $A_{dst} = A_{gw}$  then
(14)   if  $A_{src} \in \mathbb{L}$  then
(15)     Construct the unique pair  $P = (s_{pkt}, A_{src})$ 
(16)     if  $P \notin \mathbb{H}$  then
(17)       Update  $T_{age}$ 
(18)       if  $T_{age} < T_{refresh}$  then
(19)          $\mathbb{H} = \mathbb{H} \cup \{P\}$ 
(20)       Forward
(21)     end if
(22)   end if
(23) end if
(24) end if
(25) Drop

```

ALGORITHM 4: Packet forwarding procedure.

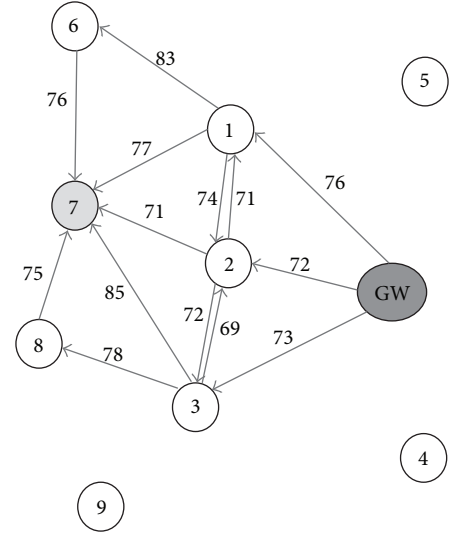


FIGURE 3: An example of REALFLOW: route establishment and packet forwarding between the gateway and Node 7.

4.4. Example: Reliable Real-Time Flooding-Based Routing Protocol. Figure 3 illustrates an example of REALFLOW. Transmissions between Node 7 and the gateway are described in detail. In the beginning, the gateway broadcasts a list-updated message. Propagation routes of the list-updated message to Node 7 are shown as solid arrows in the figure. Numbers on the arrows are the absolute values of RSSI from one node to another. We define K_{max} as 2, since multipath diversity is achieved and not too much network resource is required. Node 7 gets five copies of list-updated messages from Nodes 1, 2, 3, 6, and 8. Compared with other nodes, r_{pkt} from Nodes 1 and 2 are minimum. Thus, Node 7 chooses

Nodes 1 and 2 to be its parent nodes. From the figure, Node 1 considers the gateway and Node 2 as its parent nodes, and Node 2 selects the gateway and Node 3 to be its parent nodes. After receiving the list-updated message, Node 7 shall reply a list-response message to the gateway. When Nodes 1 and 2 receive the list-response message, since its address is included in \mathbb{N}_{fwd} , so they add the address of Node 7 to their related node lists and forward it to their parent nodes. After all list-response messages are sent to the gateway, related node lists are generated in all nodes, and the gateway is able to calculate the latest network topology. Then the gateway needs to send list-confirmed messages to all nodes with the confirmation of related node lists and scheduling decisions.

After the related node lists are confirmed by the gateway, when Node 7 sends a packet to the gateway by flooding, only Nodes 1 and 2 are able to forward the packet, and other nodes will drop the packet automatically. By this means, packets are transmitted from Node 7 to the gateway via two paths,

so the multipath diversity is achieved. Moreover, flooding transmission is severely controlled and restricted within a small range to avoid excessive packet forwarding.

Usually, many research works reach a consensus that flooding method is much less efficient at packet transmission than unicasting, since much more network resources may be required for flooding. However, it is not always the truth. In this example, as TDMA is used on the MAC layer, for uplink, only 3 timeslots are required to achieve two-path diversity. For traditional unicast routing protocol, four timeslots are needed to achieve the same diversity. A similar situation happens in the reversed direction for the downlink. More analyses of required timeslots, as well as complexity and reliability, are discussed in the following subsection.

4.5. Theoretical Analysis of REALFLOW. In this subsection, the complexity of the algorithm, the network resource efficiency, and reliability performance are analyzed by theory.

4.5.1. Algorithm Complexity. REALFLOW consists of four algorithms as shown above. Algorithms 1, 2, and 4 run distributively in each node in the network without requiring the gateway to participate, including related node list generation. The complexity of these three algorithms is $T(n) = O(1)$, where n is the number of nodes in the network excluding the gateway. In a centralized IWSN, the gateway is a bottleneck for the whole network. Thus, these algorithms release the gateway from high workload. Even if the network size increases significantly, the workload from these three algorithms for each node including the gateway remains the same.

Only Algorithm 3 should run at the gateway to generate the network topology. The parameter K_{\max} , which is the maximum allowed parent node number, can affect the complexity of the algorithm from Lines 4 and 5. It is a predefined value and can be considered as a constant. In the worst case, there is only one node directly connected to the gateway, and all other nodes are able to find K_{\max} parent nodes. Therefore, the worst-case complexity of Algorithm 3 is:

$$T(n) = 5K_{\max}n^2 + 16K_{\max}n + 2n + 8 - K_{\max} = O(n^2). \quad (4)$$

Compared with other routing protocols for IWSNs, such as [34], where the complexity is $O(n^3)$, REALFLOW hugely decreases the time complexity. In centralized IWSNs, the latest routing and scheduling decisions are all made by the gateway or network manager. Thus, low complexity is severely important for centralized IWSNs, since it may take the gateway or network manager excessively long time for the calculations if the routing algorithm is overcomplicated. During the processing period, the whole network may behave abnormally, since all nodes in the network do not have the latest routing information. Although the optimal scheduling algorithm is not specified in this paper, the complexity of the simplified scheduling scheme is still no more than that of the topology generation algorithm. Therefore, the overall complexity remains at the same level.

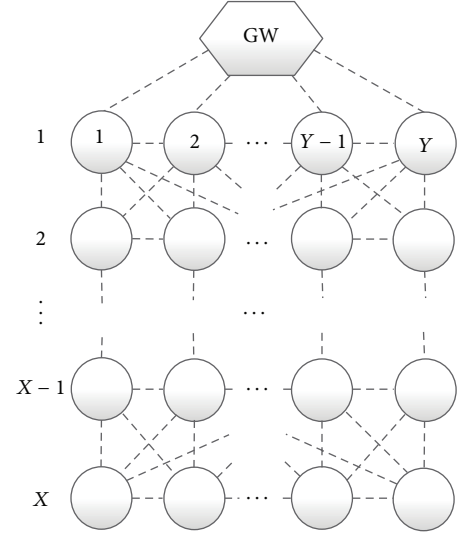


FIGURE 4: An example of a multihop IWSN for analysis.

4.5.2. Network Resource Efficiency. Regarding a general case, a TDMA IWSN consisting of $X \cdot Y$ nodes is constructed, as shown in Figure 4. On each hop, there exist Y nodes, and the furthest nodes are X hops away from the gateway. For simplicity, we assume that a node is able to communicate with any parent node, child node, or sibling node, and each node periodically communicates with the gateway with the same refresh interval. Here, network resource efficiency refers to the usage of time slots. Flooding-based transmission is deemed to be inefficient due to excessive transmissions. In order to explore the network resource efficiency of REALFLOW, we intend to calculate the upper and lower bound of the total required timeslots of the entire network. To obtain the upper bound, we assume that any node is able to find K_{\max} parent nodes or sibling nodes, and we also assume that all nodes directly connected to the gateway are always involved in forwarding packets from nodes more than one hop away. Then the maximum required timeslots R_{up} for the whole network is calculated as

$$R_{\text{up}} = \begin{cases} Y(K_{\max} + (1 + Y) + (1 + Y + K_{\max}) + \dots + (1 + Y + K_{\max} + K_{\max}^2 + \dots + K_{\max}^{X-2})), & Y \geq K_{\max}^{X-2}, X > 2 \\ Y(K_{\max} + (1 + Y) + (1 + Y + K_{\max}) + \dots + (1 + 2Y + K_{\max} + K_{\max}^2 + \dots + K_{\max}^{X-3})), & K_{\max}^{X-3} \leq Y < K_{\max}^{X-2}, X > 3 \\ Y(K_{\max} + (1 + Y) + (1 + Y + K_{\max}) + \dots + (1 + 3Y + K_{\max} + K_{\max}^2 + \dots + K_{\max}^{X-4})), & K_{\max}^{X-4} \leq Y < K_{\max}^{X-3}, X > 4 \\ \vdots \\ Y(K_{\max} + (1 + Y) + \dots + (1 + (X-1)Y)), & 1 \leq Y < K_{\max} \end{cases} \quad (5)$$

and (5) can be further simplified as

$$R_{up} = \begin{cases} Y \left[K_{max} + Y(X-1) + \frac{K_{max}(K_{max}^{X-1} - 1)}{(K_{max} - 1)^2} + \frac{1 - X}{K_{max} - 1} \right], & Y \geq K_{max}^{X-2}, X > 2 \\ Y \left[K_{max} + Y(X-1) + \frac{(i-1)iY}{2} + \frac{K_{max}(K_{max}^{X-i} - 1)}{(K_{max} - 1)^2} + \frac{(i-1)K_{max}^{X-i+1} - X + 1}{K_{max} - 1} \right], & K_{max}^{X-i} \leq Y < K_{max}^{X-i+1}, \\ & X > i, i = 3, \dots, X. \end{cases} \quad (6)$$

Different from upper bound calculation, there are two theoretical lower bounds for REALFLOW. To calculate the first lower bound, we still assume that all nodes can find K_{max} parent nodes or sibling nodes, but we also assume that redundant paths overlap with each other at maximum extent, which means that nodes with the same hops away from the gateway share the same parent nodes. Thus, the first lower bound of required timeslots R_{low1} is calculated as

$$R_{low1} = \begin{cases} Y(1 + (1 + K_{max}) + (1 + 2K_{max}) + \dots + (1 + (X-1)K_{max})) & Y \geq K_{max} \\ Y(1 + (1 + Y) + (1 + 2Y) + \dots + (1 + (X-1)Y)), & Y < K_{max} \end{cases} \quad (7)$$

and (7) can be further simplified as

$$R_{low1} = \begin{cases} XY + \frac{(X-1)XYK_{max}}{2}, & Y \geq K_{max} \\ XY + \frac{(X-1)XY^2}{2}, & Y < K_{max} \end{cases} \quad (8)$$

In reality, a node cannot always find K_{max} parent or sibling node due to interferences or network topology. The second theoretical lower bound can be obtained if each node can only find one available path. Thus, we assume that each node is only able to find one parent or sibling node. The required time slots for the whole network R_{low2} is

$$R_{low2} = Y + 2Y + \dots + X \cdot Y = \frac{(1+X)XY}{2}. \quad (9)$$

From (6), since K_{max}^{X-1} , K_{max}^{X-i} and K_{max}^{X-i+1} determine the increment of R_{up} , when the hop number of the node X increases, the maximum required timeslots also grow rapidly. It indicates that REALFLOW may not be very resource efficient in an extremely large size of a network, where nodes

are many hops away from the gateway. However, REALFLOW can be applied in IWSNs for two reasons. The first reason is that in a great number of industrial applications the refresh rate is extremely fast, at the order of seconds, or even milliseconds [2]. Thus, the network size is seriously constrained and cannot be too large. Secondly, since the gateway is allowed to send commands to shrink the related node list in each node. The flooding range can be further restricted. Even if the network size is large with hundreds of nodes, for industrial applications with the slow refresh rate, REALFLOW can still be resource efficient. Moreover, (6) is calculated based on the worst case in which every node has different parent nodes. However, in reality, many nodes may share the same parent nodes, so the worst case happens very rarely.

When nodes with the same hops away from the gateway share the same parent nodes, (8) is obtained. According to (8), even if the hop number X increases, the total required time slots will not grow extraordinarily. Equation (9) is an extreme case in which all nodes can only find one path to the gateway. The required timeslot number is the same as unicast transmission. Thus, it is unnecessary to discuss.

In reality, the required timeslot number of the whole network is most probably between (6) and (8). Therefore, the network resource consumption when applying REALFLOW is definitely not overwhelming to the network, although the control-flooding mechanism is applied.

4.5.3. Reliability Performance. As we emphasized previously, high reliability is one of the most important requirements of IWSNs. REALFLOW is initially designed to achieve high reliability by high multipath diversity.

We use the same network topology (Figure 4) for the reliability analysis. The packet delivery ratio (PDR) of a node X hops away from the gateway is calculated as follows. We assume that there are Z_i nodes, which are i hops away from the gateway and are intermediate nodes to the destination node. Also for simplicity, we assume that any node out of these Z_i nodes is able to directly find K_{max} parent nodes, if this node is more than one hop away from the gateway. Furthermore, we assume that all channels are symmetric. This assumption is valid if the bandwidth, transmission power, and hardware for both uplink and downlink are the same. $P_d^{(k)}(j)$ is the PDR of the j th node to its k th parent node if this node is more than one hop away from the gateway; otherwise, $P_{ds}^{(k)}(j)$ is the PDR of the k th redundant path from the j th node to the gateway. $P_d^{prt}(i, j)$ is the overall PDR of the j th node from i th hop to all its parent nodes at $(i-1)$ th hop. $P_d^{hop}(i)$ is the overall PDR of nodes from i th hop to $(i-1)$ th hop. Then we have

$$P_d^{prt}(i, j) = \begin{cases} P_d^{(1)}(j) + (1 - P_d^{(1)}(j))P_d^{(2)}(j) + \dots + \left(\prod_{m=1}^{K_{max}-1} (1 - P_d^{(m)}(j)) \right) P_d^{(K_{max})}(j), & i \neq 1 \\ P_{ds}^{(1)}(j) + (1 - P_{ds}^{(1)}(j))P_{ds}^{(2)}(j) + \dots + \left(\prod_{m=1}^{K_{max}-1} (1 - P_{ds}^{(m)}(j)) \right) P_{ds}^{(K_{max})}(j), & i = 1, \end{cases} \quad (10)$$

$$P_d^{\text{hop}}(i) = P_d^{\text{prt}}(i, 1) + (1 - P_d^{\text{prt}}(i, 1))P_d^{\text{prt}}(i, 2) + \dots + \left(\prod_{m=1}^{Z_i-1} (1 - P_d^{\text{prt}}(i, m)) \right) P_d^{\text{prt}}(i, Z_i). \quad (11)$$

Finally the overall PDR between the gateway and the j th node with l hops away from the gateway $P_d^{\text{overall}}(l)$, $l > 1$, can be calculated as

$$P_d^{\text{overall}}(l) = P_d^{\text{prt}}(l, j) \prod_{m=1}^{l-1} P_d^{\text{hop}}(m). \quad (12)$$

It is apparent that from (10) $P_d^{\text{prt}}(i, j)$ will increase if K_{max} increases. In order to investigate if $P_d^{\text{hop}}(i)$ monotonically increases with $P_d^{\text{prt}}(i, j)$, the partial derivative can be calculated as

$$\frac{\partial P_d^{\text{hop}}(i)}{\partial P_d^{\text{prt}}(i, j)} = \prod_{m=1, m \neq j}^{K_{\text{max}}} (1 - P_d^{\text{prt}}(i, m)), \quad \forall j \in [1, K_{\text{max}}]. \quad (13)$$

Since $0 \leq P_d^{\text{prt}}(i, m) < 1$, it follows that $\partial P_d^{\text{hop}}(i) / \partial P_d^{\text{prt}}(i, j) > 0$. From (12), it is also apparent that $P_d^{\text{overall}}(l)$ will increase if $P_d^{\text{hop}}(m)$ increases. Thus, we can conclude that $P_d^{\text{overall}}(l)$ will definitely increase if K_{max} increases. Finally, we prove that, by increasing K_{max} , we can definitely improve the overall reliability performance of REALFLOW, given that there are more available paths to use.

However, according to (6) and (8), the required number of time slots will also increase at the same time if we increase K_{max} . Therefore, there is a trade-off between the reliability performance and the network resource efficiency.

5. Experimental Setup

The reliability and real-time performance of REALFLOW are evaluated and compared with those of three traditional routing protocols via simulations in the discrete event simulator QualNet. In this section, the experimental settings and simulation scenarios are described as follows.

5.1. Simulation Settings. We construct a centralized mesh IWSN placed within a 100×100 m area. Two types of devices are involved in our simulations, namely, a gateway and a number of nodes, with the complete communication protocol stack. On the network layer, we implemented REALFLOW in QualNet and utilize three other existing transitional routing protocols from QualNet. Besides the network layer, the configurations from other layers are summarized in Table 1.

The configuration of the physical layer is defined according to the standard IEEE 802.15.4 [35], where the communication operates at 2.4 GHz with the bit rate 250 kbps. In order to provide deterministic communication, TDMA mechanism is applied rather than traditional CSMA/CA mechanism. The timeslot duration is determined according to the standard WirelessHART. The reasons for turning off

TABLE 1: The protocol parameters.

Parameter	Description
Application layer	Constant bit rate application
Network layer	REALFLOW/AODV/DSR/DYMO
MAC layer	TDMA
Timeslot duration	10 ms
Retransmission	Off
Physical layer	O-QPSK
Bit rate	250 kbps
Channel frequency	2.4 GHz
Fading model	Rayleigh fading
Shadowing model	Log-normal
Transmit power	10 dBm
Receiver sensitivity	-85 dBm

retransmissions are due to the stringent real-time performance requirement and fast refresh rates. Retransmissions bring significantly higher delivery latency, so packets may be already outdated when arriving at their destinations if the deadlines are very short. For many industrial applications, outdated packets are of limited use for destinations. In the standard WirelessHART and ISA 100a, channel hopping is used on the MAC layer as well. For simplicity, we also disable channel hopping in our simulation. On the network layer, besides REALFLOW, we choose three traditional routing protocols, AODV, DSR, and DYMO provided by QualNet. These traditional routing protocols have been already applied in traditional WSNs. Constant bit rate application is selected on the application layer to provide continuous data transmissions.

Besides the protocol stack, the channel parameters and node settings are also included in Table 1. As mentioned before, due to a great number of obstacles, NLOS communication dominates in many industrial environments [8, 9]. Therefore, we choose the Rayleigh fading model. We assume that industrial environments are relatively static, so log-normal model is chosen as the shadowing model. According to the standard WirelessHART, the maximum allowed radio output power is 10 dBm. To achieve the longest communication distance, we set the transmission power as 10 dBm. This radio output power has been already supported by existing devices, such as CC2591 [36]. We choose the receiver sensitivity as -85 dBm to filter out too weak signals. A great number of existing devices are able to support this sensitivity such as CC2591 and CC2531 [37]. In order to prove our simulation setting suitable, we place two nodes in QualNet with around 40 meters away and measure RSSI value at the receiver side. According to the measurement result, although the measured RSSI values from our simulation are around 10 dB less than those from the real environment, the variance of the signal strength follows the same trend. RSSI values may differ from different platforms, so we can consider the channel status to be relatively similar to that in a real industrial environment.

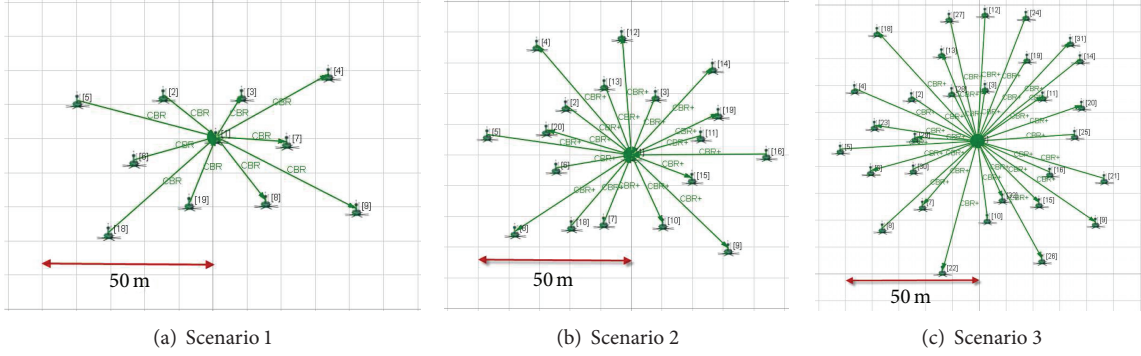


FIGURE 5: Three simulation scenarios. A mesh network is created in each scenario. The arrows between nodes and the gateway do not mean the direct links but are the QualNet configurations on the application layer for periodic data transmissions.

5.2. Simulation Scenarios. We set up three different scenarios to show the results with respect to reliability and real-time performance from REALFLOW. In each scenario, we consider both uplink and downlink transmissions. Thus, each node periodically sends packets to the gateway, and the gateway also periodically sends packets to all nodes. The refresh intervals from three scenarios are 250 ms, 500 ms, and 1 s, respectively. These refresh rates are commonly seen in existing industrial applications [2].

Due to the fast refresh rate, the size of the network is strictly constrained. The faster the refresh rate is, the fewer the timing slots and bandwidth resources are available for each node. If too many nodes exist in the network, the network resources may become unschedulable. It is also important that due to the fast application refresh rate and stringent reliability and timing requirements, the hop number of each node in the network cannot be too large. Not only does larger hop number require more network resources and timing for packet forwarding, but also the end-to-end transmissions are at the risk of higher failure rate. This is one of the major differences between time-critical IWSNs and traditional WSNs. Usually the data refresh rate of traditional WSNs is much lower. Moreover, traditional WSNs have a much higher tolerance to transmission failures. Therefore, the sizes of the network from three scenarios are randomly selected as 10, 18, and 30, respectively, excluding the network manager.

In each scenario, 50% are sensors and 50% actuators. Three scenarios are shown in Figure 5. It is very important to know that arrows in Figure 5 do not mean that all nodes are able to communicate with the gateway directly. These arrows are the configurations from QualNet to show that all nodes are configured to communicate with the gateway periodically. If nodes are farther away than one hop distance to the gateway, intermediate nodes will be involved automatically.

After mesh networks are established, all nodes periodically send packet to the gateway at a certain refresh rate, and the gateway also periodically sends packet to all nodes. Then we measure packet delivery ratio (PDR), end-to-end transmission latency, and network recovery time. PDR measurements indicate the reliability of different routing protocols. End-to-end latency is calculated from the time

when a packet is generated to the time when it arrives at the final destination, which reveals the real-time performance. The network recovery time means the time period spent by a routing protocol on recalculating a new path. It shows the robustness of routing protocols to be tolerant of network topology changes. These are the most important criteria to validate a well-qualified routing protocol for IWSNs.

6. Evaluation Results and Analysis

This section describes the simulation results from all three scenarios, covering the comparison between REALFLOW and three traditional routing protocols. Results of reliability, real-time performance, and network recovery time are analyzed separately in three subsections.

6.1. Packet Delivery Ratio. All PDRs from three scenarios are measured in two ways, namely, overall PDR and deadline PDR. Overall PDRs are measured based on all received packets at the destinations without considering application deadlines, whereas outdated packets are discarded when measuring deadline PDRs. Both PDRs from uplink and downlink are measured separately. Simulation results from three scenarios are shown in Figures 6, 7, and 8, respectively. From the results, we observe that PDRs measured from all scenarios for both uplink and downlink are much higher than the other three traditional routing protocols. Even in Scenario 3 where nodes are placed further away from each other, REALFLOW is still able to achieve more than 70% average PDRs for both uplink and downlink. This observation is mainly because packets are always sending via several paths from the sources to the destinations by our routing protocol, so the probability of packets arriving at the destinations is much higher. Therefore, even if the channel is in a bad status where broken links occur frequently, packets may be still able to find one path to the destination. It is also notable that overall PDRs and deadline PDRs from REALFLOW are the same, since all outdated packets have been discarded at every intermediate node at their respective deadlines.

Compared with REALFLOW, traditional routing protocols perform much worse in the Scenario 3. Not only are all deadline PDRs of traditional routing protocols for both

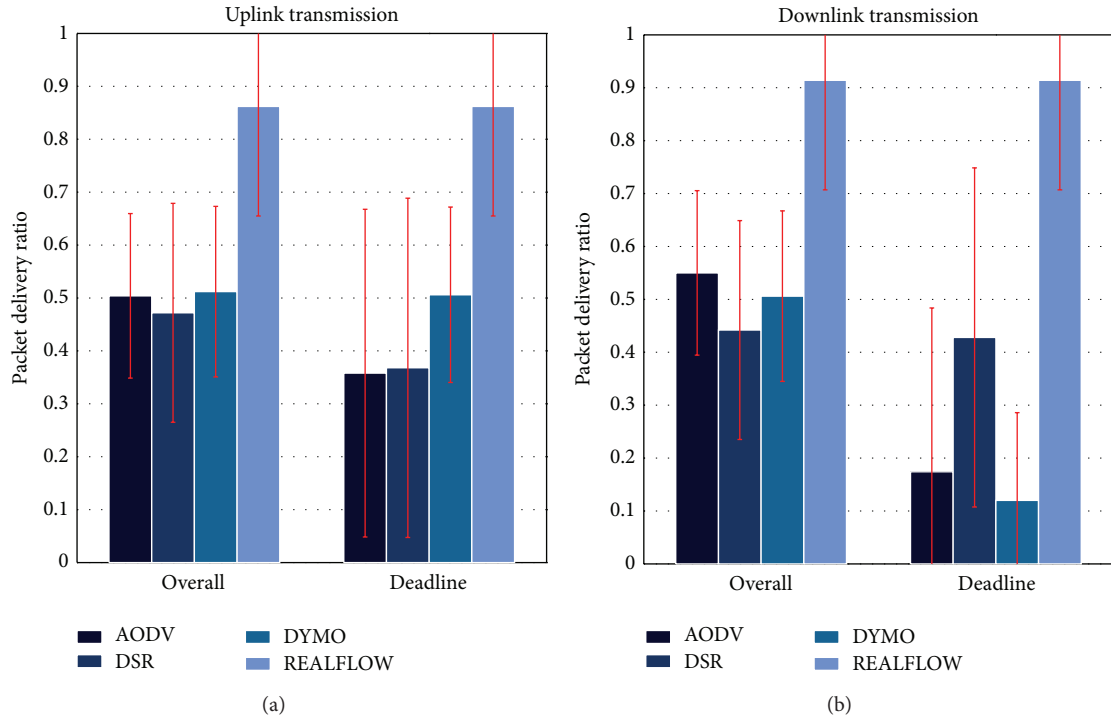


FIGURE 6: PDRs of different routing protocols in Scenario 1.

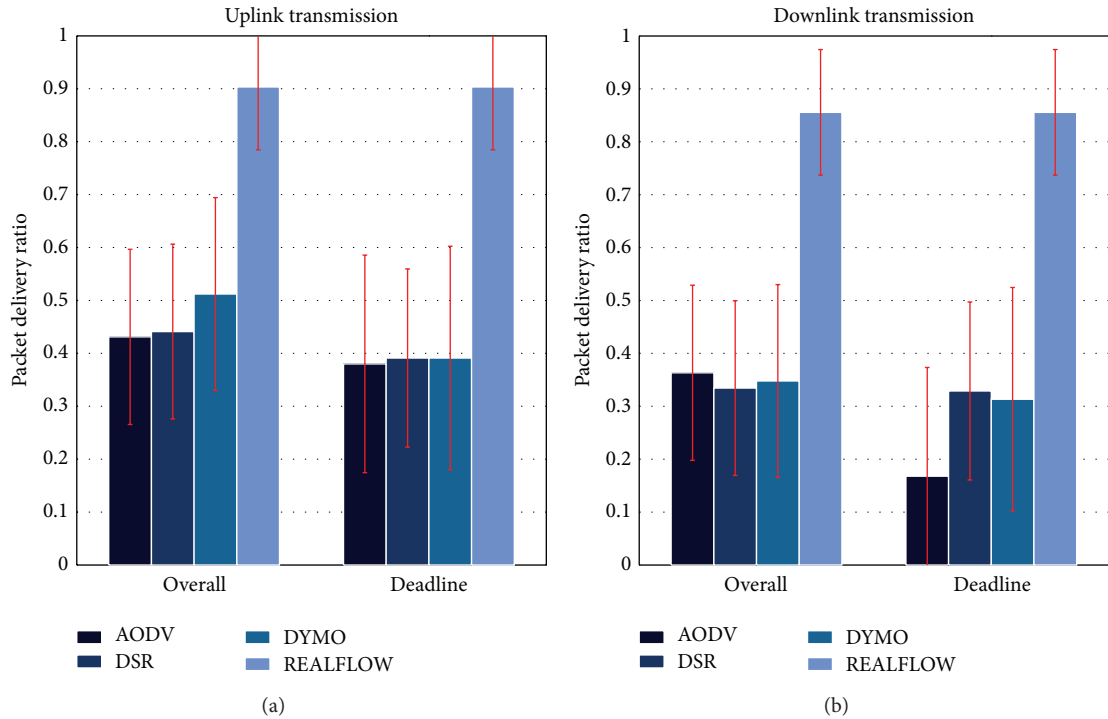


FIGURE 7: PDRs of different routing protocols in Scenario 2.

uplink and downlink much lower than those of REALFLOW, but also the overall PDRs of those routing protocols are also quite low. For instance, in Scenario 3, deadline PDRs from all three traditional routing protocols are around 10%, much lower than the PDRs from REALFLOW, 75%. The standard

deviations from all routing protocols are similar, which indicates the dynamic changes of the channel status. There are several reasons for the unsatisfying reliability performance of these three traditional routing protocols. The first reason is lack of redundant paths. These traditional routing protocols

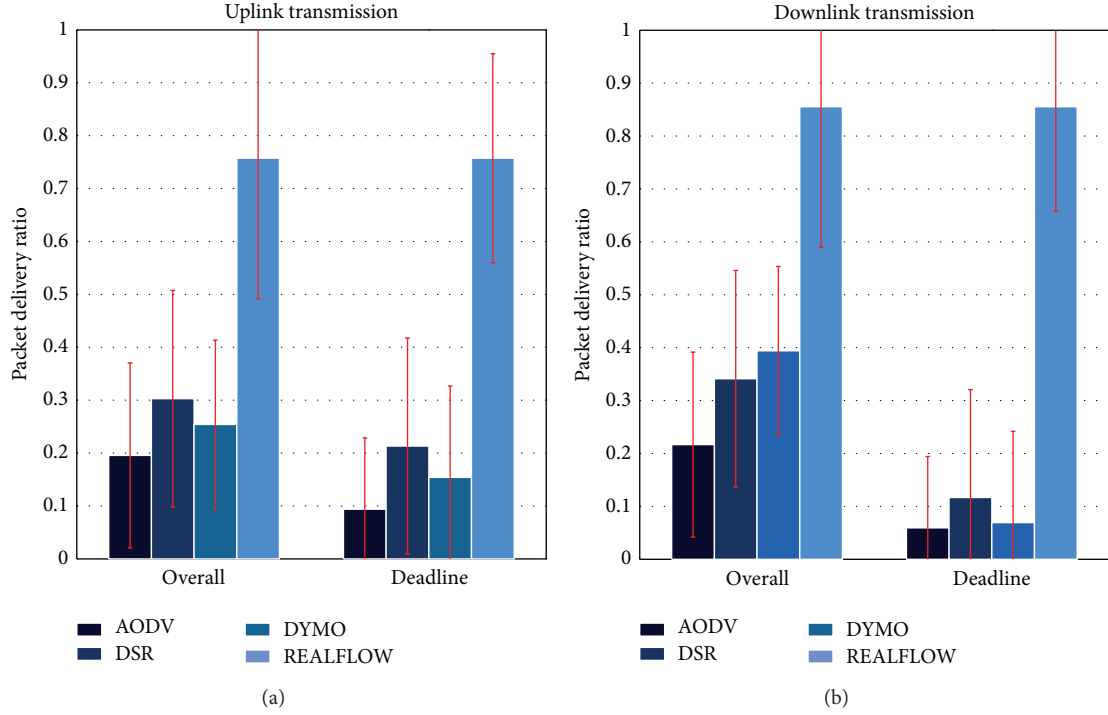


FIGURE 8: PDRs of different routing protocols in Scenario 3.

utilize only one route to forward all packets. Once any broken link occurs in this path, packets fail to be delivered to the destination. Secondly, traditional routing protocols are often used along with ARQ mechanism. Once transmission failure happens, the sender will retransmit the failed packets until they are successfully received or up to maximum retried times. As we described above, since retransmissions may introduce much longer delay, outdated packets are of limited use for destinations. Thus, ARQ is not used in our simulation. Without retransmissions, the traditional routing protocols can hardly achieve high reliability. The last reason is that these traditional routing protocols are initially designed for traditional ad hoc networks instead of industrial centralized networks. Therefore, under these IWSN settings, REALFLOW significantly outperforms traditional routing protocols in the reliability performance.

6.2. End-to-End Latency. Average uplink and downlink transmission latency from three scenarios for both uplink and downlink as well as their standard deviations are shown in Table 2. According to the simulation results, the average uplink and downlink delays of REALFLOW are much lower than the other three routing protocols. It is also notable that even if considering the standard deviations, the transmission latencies are still below the application deadlines. The decent real-time performance of REALFLOW can be explained as follows. Firstly, for REALFLOW, packets are only buffered at intermediate nodes until next available timeslot to transmit. Even if partial links of the network are broken, intermediate nodes will not spend any time on searching for the next hop, which greatly accelerates the packet forwarding procedures.

However, for traditional routing protocols, if the links are broken, it will take them extremely long time to find the next available hop. Finally, the packet age is calculated through the time when the packet is generated and checked by all intermediate nodes. Outdated packets will be discarded automatically. Therefore, more network resources can be used to forward valid packets, which also shorten the end-to-end transmission delay. Therefore, REALFLOW can achieve extremely low end-to-end transmission delay. However, these advantages are not held by three traditional routing protocols. Packet ages calculated by traditional routing protocols are based on the packet hop number instead of existing time. Moreover, according to traditional routing protocols, nodes frequently exchange information to update the latest route status. Thus, packets may be buffered at intermediate nodes for long time until the next hop is found. According to the comparison, REALFLOW is proven to be a well-qualified routing protocol for real-time transmissions in IWSNs.

6.3. Network Recovery Time. As we mentioned previously, a routing protocol should be tolerant of topology changes, such as intermediate nodes halting. Thus, in this experiment, we halt an intermediate node at a certain time during simulation. Then, the network recovery time is measured by observing the time when a new route is established. During the network recovery period, transmissions from certain nodes may be temporarily terminated, so a new path should be established as soon as possible. The network recovery time of different routing protocols from three scenarios is also shown in Table 2. Because REALFLOW always utilizes more than one path to forward packets, even if one intermediate node

TABLE 2: Simulation results of average latency and network recovery time for three scenarios.

Routing protocol	End-to-end delay				Network recovery time	
	Average uplink delay	Uplink standard deviation	Average downlink delay	Downlink standard deviation	Average recovery time	95% confidence interval
Scenario 1						
AODV	0.87 s	1.67 s	0.21 s	0.35 s	54.90 s	5.81 s
DSR	1.20 s	1.46 s	0.93 s	1.01 s	373.77 s	27.06 s
DYMO	1.51 s	2.71 s	0.90 s	0.71 s	63.74 s	4.48 s
REALFLOW	0.09 s	0.02 s	0.08 s	0.07 s	1 timeslot	0 s
Scenario 2						
AODV	12.00 s	26.77 s	36.15 s	72.94 s	49.45 s	10.10 s
DSR	14.79 s	41.76 s	0.20 s	0.28 s	219.91 s	10.79 s
DYMO	24.75 s	64.68 s	0.85 s	3.21 s	62.024 s	7.54 s
REALFLOW	0.30 s	0.09 s	0.11 s	0.10 s	1 timeslot	0 s
Scenario 3						
AODV	3.47 s	7.01 s	7.86 s	6.26 s	50.12 s	8.76 s
DSR	21.45 s	22.80 s	3.78 s	10.95 s	198.16 s	14.04 s
DYMO	18.86 s	37.67 s	1.17 s	1.69 s	66.76 s	17.10 s
REALFLOW	0.49 s	0.26 s	0.51 s	0.24 s	1 timeslot	0 s

halts or one path is blocked, the packets are still able to arrive at the destination via other paths without requiring to establish new paths. Therefore, a seamless transition of the topology change can be achieved by REALFLOW, shown in the results. However, for traditional routing protocols, routing information needs to be updated timely by all nodes. Once a previous path is blocked, traditional routing protocols need to recalculate the latest routing information and try to find a new route. Depending on the channel condition, the time consumption of finding a new path is unpredictable and not deterministic. According to the simulation results, we observe that it took extremely long time for traditional routing protocols to establish a new path. Since packet delivery is blocked, unexpected operations may be taken at the receiver side, which is unacceptable to a great number of industrial applications. Therefore, the result reveals that REALFLOW is more robust for industrial automation.

7. Conclusions and Future Work

In this paper, we abstract the challenges on how to design a usable and robust routing protocol in IWSNs for industrial automation. Then REALFLOW is proposed to address all these challenges of deploying IWSNs for industrial purpose. Reliability, real-time performance, robustness, and node workload are thoroughly considered as REALFLOW designed. Instead of traditional routing tables, a related node list is distributively generated in each node and utilized to assist with packet forwarding. Packet transmissions for uplink and downlink are based on the controlled flooding mechanism and several other forwarding criteria. The simulation shows the outstanding improvement in both reliability and real-time performance, compared with traditional routing protocols. Moreover, a seamless transition in the event

of topology change can be achieved by our routing protocol. Since our goal is to deploy IWSNs for industrial control systems, supporting hundreds of nodes in a large scale of networks is out of the scope of this paper.

In future work, REALFLOW can be further improved to support mobility of nodes, if industrial applications require. We shall also continue to look for an optimal scheduling scheme to enhance REALFLOW. Implementation of REALFLOW in real devices and conducting experiments in practical industrial environments is also of the most importance. Although low energy consumption is not the main issue of REALFLOW, to evaluate and analyze the overall energy consumption, compared to other routing protocols, are also parts of our future work.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

References

- [1] A. Willig, "Recent and emerging topics in wireless industrial communications: a selection," *IEEE Transactions on Industrial Informatics*, vol. 4, no. 2, pp. 102–122, 2008.
- [2] J. Akerberg, M. Gidlund, and M. Bjorkman, "Future research challenges in wireless sensor and actuator networks targeting industrial automation," in *Proceedings of the 9th IEEE International Conference on Industrial Informatics (INDIN '11)*, pp. 410–415, Caparica, Portugal, July 2011.
- [3] V. C. Gungor and G. P. Hancke, "Industrial wireless sensor networks: challenges, design principles, and technical approaches," *IEEE Transactions on Industrial Electronics*, vol. 56, no. 10, pp. 4258–4265, 2009.

- [4] C. Caione, D. Brunelli, and L. Benini, "Distributed compressive sampling for lifetime optimization in dense wireless sensor networks," *IEEE Transactions on Industrial Informatics*, vol. 8, no. 1, pp. 30–40, 2012.
- [5] J. Ploennigs, V. Vasyutynskyy, and K. Kabitzsch, "Comparative study of energy-efficient sampling approaches for wireless control networks," *IEEE Transactions on Industrial Informatics*, vol. 6, no. 3, pp. 416–424, 2010.
- [6] G. Anastasi, M. Conti, and M. di Francesco, "Extending the lifetime of wireless sensor networks through adaptive sleep," *IEEE Transactions on Industrial Informatics*, vol. 5, no. 3, pp. 351–365, 2009.
- [7] P. Ångskog, C. Karlsson, J. F. Coll, J. Chilo, and P. Stenumgaard, "Sources of disturbances on wireless communication in industrial and factory environments," in *Proceedings of the Asia-Pacific Symposium on Electromagnetic Compatibility (AP EMC '10)*, pp. 281–284, April 2010.
- [8] J. Åkerberg, M. Gidlund, F. Reichenbach, and M. Björkman, "Measurements on an industrial wireless HART network supporting PROFIsafe: a case study," in *Proceedings of the 16th IEEE Conference on Emerging Technologies & Factory Automation (ETFA '11)*, pp. 1–8, Toulouse, France, September 2011.
- [9] D. Sexton, M. Mahony, M. Lapinski, and J. Werb, "Radio channel quality in industrial wireless sensor networks," in *Proceedings of the Sensors for Industry Conference*, pp. 88–94, February 2005.
- [10] A. Sikora and V. F. Groza, "Coexistence of IEEE802.15.4 with other systems in the 2.4 GHz-ISM-band," in *Proceedings of the IEEE Instrumentation and Measurement Technology Conference (IMTC '05)*, vol. 3, pp. 1786–1791, Ottawa, Canada, May 2005.
- [11] Zigbee alliance, <http://www.zigbee.org/>.
- [12] T. Lennvall, S. Svensson, and F. Hekland, "A comparison of wirelessHART and ZigBee for industrial applications," in *Proceedings of the 7th IEEE International Workshop on Factory Communication Systems*, pp. 85–88, May 2008.
- [13] G. Anastasi, M. Conti, and M. Di Francesco, "A comprehensive analysis of the MAC unreliability problem in IEEE 802.15.4 wireless sensor networks," *IEEE Transactions on Industrial Informatics*, vol. 7, no. 1, pp. 52–65, 2011.
- [14] Hart 7 specification, 2010, <http://www.hartcomm.org/>.
- [15] Industrial society of automation, <http://www.isa.org/>.
- [16] J. Åkerberg, M. Gidlund, J. Neander, T. Lennvall, and M. Björkman, "Deterministic downlink transmission in WirelessHART networks enabling wireless control applications," in *Proceedings of the 36th Annual Conference of the IEEE Industrial Electronics Society (IECON '10)*, pp. 2120–2125, Glendale, Ariz, USA, November 2010.
- [17] S. Mueller, R. Tsang, and D. Ghosal, "Multipath routing in mobile ad hoc networks: issues and challenges," in *Performance Tools and Applications to Networked Systems*, vol. 2965 of *Lecture Notes in Computer Science*, pp. 209–234, Springer, 2004.
- [18] M. Tarique, K. E. Tepe, S. Adibi, and S. Erfani, "Survey of multipath routing protocols for mobile ad hoc networks," *Journal of Network and Computer Applications*, vol. 32, no. 6, pp. 1125–1143, 2009.
- [19] S. Dulman, T. Nieberg, J. Wu, and P. J. M. Havinga, "Trade-off between traffic overhead and reliability in multipath routing for wireless sensor networks," in *Proceedings of the IEEE Wireless Communications and Networking (WCNC '03)*, pp. 1918–1922, New Orleans, La, USA, 2003.
- [20] B. Deb, S. Bhatnagar, and B. Nath, "Reinform: reliable information forwarding using multiple paths in sensor networks," in *Proceedings of the 28th Annual IEEE International Conference on Local Computer Networks*, pp. 406–415, October 2003.
- [21] A. G. Ruzzelli, G. M. P. O'Hare, M. J. O'Grady, and R. Tynan, "MERLIN: a synergetic integration of MAC and routing protocol for distributed sensor networks," in *Proceedings of the 3rd Annual IEEE Communications Society on Sensor and Ad hoc Communications and Networks (SECON '06)*, vol. 1, pp. 266–275, Reston, Va, USA, September 2006.
- [22] S. Han, X. Zhu, A. K. Mok, D. Chen, and M. Nixon, "Reliable and real-time communication in industrial wireless mesh networks," in *Proceeding of the 17th IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS '11)*, pp. 3–12, Chicago, Ill, USA, April 2011.
- [23] G. J. Pottie and W. J. Kaiser, "Wireless integrated network sensors," *Communications of the ACM*, vol. 43, no. 5, pp. 51–58, 2000.
- [24] D. B. Johnson and D. A. Maltz, "Dynamic source routing in ad hoc wireless networks," in *Mobile Computing*, pp. 153–181, Kluwer Academic Publishers, New York, NY, USA, 1996.
- [25] C. E. Perkins and E. M. Royer, "Ad-hoc on-demand distance vector routing," in *Proceedings of the IEEE Workshop on Mobile Computing Systems and Applications*, 1997.
- [26] M. K. Marina and S. R. Das, "On-demand multipath distance vector routing in ad hoc networks," in *Proceedings of the 9th International Conference on Network Protocols (ICNP '01)*, pp. 14–23, November 2001.
- [27] Z. Ye, S. V. Krishnamurthy, and S. K. Tripathi, "A framework for reliable routing in mobile ad hoc networks," in *Proceedings of the 22nd Annual Joint Conference on the IEEE Computer and Communications Societies (INFOCOM '03)*, vol. 1, pp. 270–280, March-April 2003.
- [28] S.-J. Lee and M. Gerla, "Split multipath routing with maximally disjoint paths in Ad hoc networks," in *Proceedings of the International Conference on Communications (ICC '01)*, vol. 10, pp. 3201–3205, IEEE, June 2000.
- [29] K. Akkaya and M. Younis, "A survey on routing protocols for wireless sensor networks," *Ad Hoc Networks*, vol. 3, no. 3, pp. 325–349, 2005.
- [30] M. Hamdi, N. Essaddi, and N. Boudriga, "Energy-efficient routing in wireless sensor networks using probabilistic strategies," in *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC '08)*, pp. 2567–2572, IEEE, Las Vegas, Nev, USA, April 2008.
- [31] J. Heo, J. Hong, and Y. Cho, "EARQ: energy aware routing for real-time and reliable communication in wireless industrial sensor networks," *IEEE Transactions on Industrial Informatics*, vol. 5, no. 1, pp. 3–11, 2009.
- [32] Y. Li, C. S. Chen, Y. Song, Z. Wang, and Y. Sun, "Enhancing Real-time delivery in wireless sensor networks with two-hop information," *IEEE Transactions on Industrial Informatics*, vol. 5, no. 2, pp. 113–122, 2009.
- [33] P. T. A. Quang and D.-S. Kim, "Enhancing real-time delivery of gradient routing for industrial wireless sensor networks," *IEEE Transactions on Industrial Informatics*, vol. 8, no. 1, pp. 61–68, 2012.
- [34] S. Han, A. K. Mok, M. Nixon, D. Chen, L. Waugh, and F. Stotz, "Utilizing parallelization and embedded multicore architectures for scheduling large-scale wireless mesh networks," in *Proceedings of the 38th Annual Conference on IEEE Industrial Electronics Society (IECON '12)*, pp. 3244–3251, October 2012.

- [35] IEEE 802.15.4 Wpan Task Group, <http://www.ieee802.org/15/pub/TG4Expert.html>.
- [36] Cc2591 2.4 GHz RF front end (rev. a), <http://www.ti.com/lit/ds/symlink/cc2591.pdf>.
- [37] "A usb-enabled system-on-chip solution for 2.4-ghz ieee 802.15.4 and zigbee applications," <http://www.ti.com/lit/ds/symlink/cc2531.pdf>.