Empirical Study and Enhancements of Industrial Wireless Sensor–Actuator Network Protocols

Mo Sha, *Member, IEEE*, Dolvara Gunatilaka, Chengjie Wu, and Chenyang Lu, *Fellow, IEEE*

***Abstract*—Wireless sensor–actuator networks (WSANs) offer an appealing communication technology for process automation applications to incorporate the Internet of Things (IoT). In contrast to other IoT applications, process automation poses unique challenges for industrial WSAN due to its critical demands on reliable and real-time communication. While industrial WSANs have received increasing attention in the research community recently, most published results to date have focused on the theoretical aspects and were evaluated based on simulations. There is a critical need for experimental research on this important class of WSANs. We developed an experimental testbed by implementing several key network protocols of WirelessHART, an open standard for WSANs that has been widely adopted in the process industries based on the HART. We then performed a series of empirical studies showing that graph routing leads to significant improvement over source routing in terms of worst-case reliability, but at the cost of longer latency and higher energy consumption. It is therefore important to employ graph routing algorithms specifically designed to optimize latency and energy efficiency. Our studies also suggest that channel hopping can mitigate the burstiness of transmission failures; a larger channel distance can reduce consecutive transmission failures over links sharing a common receiver. Based on these insights, we developed a novel channel hopping algorithm that utilizes far away channels for transmissions. Furthermore, it prevents links sharing the same destination from using channels with strong correlations. Our experimental results demonstrate that our algorithm can significantly improve network reliability and energy efficiency.**

***Index Terms*—Channel hopping, Internet of Things (IoT), wireless sensor–actuator networks (WSANs), WirelessHART.**

I. Introduction

PROCESS automation is crucial for process industries such as oil refineries, chemical plants, and factories. Today's industry mainly relies on wired networks to monitor and control their production processes. Cables are used for connecting sensors and forwarding sensor readings to a control room where a controller sends commands to actuators. However, these wired systems have significant drawbacks. It is very costly to deploy and maintain such wired systems, since numerous cables have to be installed and maintained, which often requires laying cables underground in harsh environments. This severely complicates efforts to reconfigure systems to accommodate new production process requirements.

Wireless sensor–actuator network (WSAN) technology is appealing to process automation applications because it does not require any wired infrastructure. WSANs can be used to easily and inexpensively retrofit existing industrial facilities without the need to run dedicated cabling for communication and power. IEEE 802.15.4-based WSANs are designed to operate at a low data rate and low power, making them a good fit for industrial automation applications where battery life is often important. In the dawning of the industrial Internet [10] and industry 4.0 [13], significant effort is being made to integrate industrial WSANs to the Internet [1]. In contrast to other Internet of Things (IoT) applications, process automation poses unique challenges to industrial WSAN due to its critical demands on reliable and real-time communication. Violation of WSAN's reliability and real-time requirements may result in plant shutdowns, safety hazards, or economic/environmental impacts.

To meet the stringent requirements on reliability and predictable real-time performance, industrial WSAN standards such as WirelessHART [36] made a set of unique network design choices.

1) The network should support both source routing and reliable graph routing: source routing provides a single route for each data flow, whereas graph routing provides multiple redundant routes based on a routing graph.
2) The network should also adopt a multichannel time division multiple access (TDMA), employing both dedicated and shared time slots, at the medium access control (MAC) layer on top of the IEEE 802.15.4 physical layer. Only one transmission is scheduled in a dedicated slot, whereas multiple transmissions can share the same shared slot. The packet transmission occurs immediately in a dedicated slot, while a carrier sense multiple access with collision avoidance (CSMA/CA) scheme is used for transmissions in a shared slot.

Recently, there has been increasing interest in developing new network algorithms and analysis to support industrial applications. However, there remains a critical need

Manuscript received July 1, 2016; revised November 4, 2016 and December 8, 2016; accepted December 30, 2016. Date of publication January 16, 2017; date of current version June 15, 2017. This work was supported by the National Science Foundation (NSF) under Grant 1320921 (NeTS), Grant 1144552 (NeTS), Grant 1035773 (CPS), Grant 1646579 (CPS), and Grant 1657275 (CRII). *(Mo Sha and Dolvara Gunatilaka contributed equally to this work.)*

M. Sha is with the Department of Computer Science, State University of New York at Binghamton, Binghamton, NY 13902 USA (e-mail: msha@binghamton.edu).

D. Gunatilaka, C. Wu, and C. Lu are with the Department of Computer Science and Engineering, Washington University in St. Louis, St. Louis, MO 63130 USA (e-mail: dgunatilaka@wustl.edu; wu@cse.wustl.edu; lu@cse.wustl.edu).

Digital Object Identifier 10.1109/JIOT.2017.2653362

for experimental testbeds to validate and evaluate network research on industrial WSANs. Without sufficient experimental evaluation, industry has shown a marked reluctance to embrace new solutions.

To meet the need for experimental research on WSANs, we have built an experimental testbed for studying and evaluating WSAN protocols. Our testbed supports a suite of key network protocols specific to the WirelessHART standard and a set of tools for managing wireless experiments. We then present a comparative study of the two routing approaches adopted by WirelessHART, namely source routing and graph routing,[1] and an empirical study on the impact of channel hopping on the burstiness of transmission failures. Our studies have led to two major insights on the development of resilient industrial WSANs.

1) Graph routing leads to significant improvement over source routing in term of worst-case reliability, at the cost of longer latency and higher energy consumption. It is therefore important to employ graph routing algorithms specifically designed to optimize latency and energy efficiency.
2) Channel hopping can mitigate the burstiness of transmission failures; a larger channel distance can reduce consecutive transmission failures over links sharing a common receiver.

Based on these insights, we developed a novel channel hopping algorithm for graph routing that causes senders to utilize far-away channels between consecutive transmissions over the same link. It further prevents links sharing the same destination from using channels with strong correlations. Our experimental results demonstrate that our algorithm can significantly improve network reliability and energy efficiency.

The rest of this paper is organized as follows. Section II introduces the features of the WirelessHART networks and Section III describes our implementation of WirelessHART protocols. Section IV presents our empirical studies. Section V evaluates our channel hopping algorithm. Section VI reviews related work and Section VII concludes this paper.

II. Features of WirelessHART Networks

To meet the stringent requirements on reliability and predictable real-time performance, industrial WSAN standards such as WirelessHART [36] made a set of unique network design choices that distinguish industrial WSANs from traditional wireless sensor networks (WSNs) designed for best effort services. In particular, we focus on several key network mechanisms supported by WirelessHART, a major industrial wireless standard widely used in process industries today.

A WirelessHART network consists of a gateway, multiple access points, and a set of field devices (sensors and actuators). The access points and field devices are equipped with half-duplex omnidirectional radio transceivers (compatible with the IEEE 802.15.4 physical layer) [9] and form a wireless mesh network. The access points are connected with the gateway through wired links and serve as bridges between the gateway and wireless field devices.

[1]This paper focuses on investigating the graph routing and source routing because they are the two routing approaches adopted by WirelessHART standard.

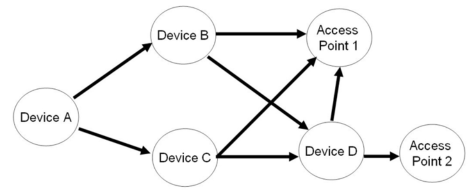

Fig. 1. Example of graph routing.

WirelessHART networks adopt a centralized network management architecture that enhances the predictability and visibility of network operations at the cost of scalability. The network manager, a software module running on the gateway, is responsible for managing the wireless network. The network manager collects the network topology information from the devices, determines the routes between itself and all devices and the transmission schedule of the network. It then disseminates the routes and the schedule to all devices.

WirelessHART supports both source routing and graph routing. Source routing provides a single route for each data flow, whereas graph routing first generates a reliable graph in which each device should have at least two neighbors to which they may send packets and then provides multiple redundant routes based on the graph. Fig. 1 shows an example. To send a packet to access points, device A may transmit the packet to device B and device C. From those devices, the packet may take several alternate routes to reach the access points. Compared to source routing, graph routing is designed to enhance network reliability through diversity and redundancy.

WirelessHART adopts a multichannel TDMA at the MAC layer. Compared to CSMA/CA, TDMA can provide predictable packet latency, which makes it attractive for real-time communication. All devices' clocks are synchronized, and time is divided into 10 ms slots that are classified into dedicated and shared slots. In a dedicated slot, only one sender is allowed to transmit. In a shared slot, multiple sensors can attempt to transmit, and these senders contend for the channel using CSMA/CA.

To enhance network capacity and to combat interference, WirelessHART networks can use up to 16 channels operating in 2.4 GHz ISM band, which are specified in IEEE 802.15.4 standard, and each device switches its channel in every slot. Specifically, after transmitting a packet on channel x in time slot k, a device can hop to the channel corresponding to logical channel $(x + 1) \bmod m$, where m is the number of available channels, for the next transmission in time slot $k + 1$. The logical channel is then mapped to a physical channel. Channel blacklisting is an optional feature that allows the network operator to restrict the channel hopping of field devices network-wide to selected channels in the wireless band. In each dedicated time slot, the total number of concurrent transmissions cannot exceed the number of available channels.

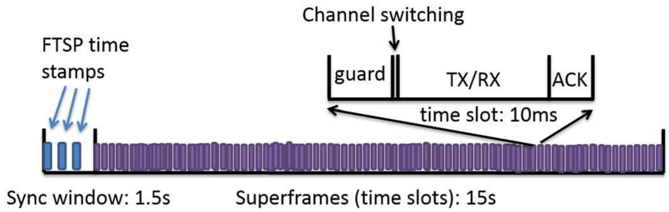

Fig. 2. Time frame format of RT-MAC.

III. Implementation of WirelessHART Protocols

We have implemented a WSAN system comprising a network manager running on a server and a protocol stack running on TinyOS 2.1.2 [35] and TelosB motes [34]. Our network manager implements a route generator and a schedule generator. The route generator is responsible for generating source routes or graph routes based on the collected network topology. We use Dijkstra's shortest-path algorithm[2] to generate routes for source routing and follow the algorithm proposed in [7] to generate reliable graphs. The schedule generator uses the rate monotonic scheduling algorithm [15] to generate transmission schedules.

Our protocol stack adopts the CC2420x radio driver [5] as the radio core, which provides an open-source implementation of IEEE 802.15.4 physical layer in TinyOS [35] operating over TI CC2420 radios. The CC2420x radio stack takes care of the low-level details of transmitting and receiving packets through the radio hardware. On top of the radio core, we have developed a multichannel TDMA MAC protocol, RT-MAC, which implements the key features of WirelessHART's MAC protocol. As shown in Fig. 2, RT-MAC divides the time into 10 ms slots following the WirelessHART standards and reserves a Sync window (1.5 s) in every 1650 slots.

Flooding time synchronization protocol (FTSP) [17] is executed during the Sync window to synchronize the clocks of all wireless devices over the entire network. Our microbenchmark experiment shows that an FTSP's time stamp packet can finish traversing of entire 55-node testbed within 500 ms. Therefore, RT-MAC configures the FTSP to flood three time stamps with 500 ms intervals over the network in each Sync window to adjust the local clocks of all devices to a global time source, which is the local time of the mote attached to the network manager. The time window following the Sync window consists of recurring superframes (a series of time slots) and idle intervals. We reserve 2 ms of guard time in the beginning of each slot to accommodate the clock synchronization error and channel switching delay, since our micro-benchmark experiments show that more than 95% of field devices over the entire network can be synchronized with errors less than 2 ms, and channel switching takes only a few microseconds to write to the registers. The rest of the field devices may disconnect from the network due to larger clock synchronization errors, but they will be reconnected in the next Sync window after they catch the new time stamps generated by FTSP. RT-MAC supports both dedicated and shared slots. In a dedicated slot, only one sender is allowed to transmit, and the packet transmission occurs immediately after the guard time. In a shared slot, more than one sender can attempt to transmit, and these senders contend for the channel using CSMA/CA.

[2] An alternative is to use expected transmission count (ETX) as the routing metric. In practice, a shortest path is usually close to a minimum-ETX path in a WirelessHART network because of link blacklisting using a high threshold (e.g., 80%).

IV. Empirical Studies

Our empirical studies are conducted on our WSAN testbed, including a four-tier hardware architecture that consists of field devices, microservers, a server, and clients. The field devices in the testbed are 55 TelosB motes [34], a widely used wireless embedded platform integrating a TI MSP430 microcontroller and a TI CC2420 radio compatible with the IEEE 802.15.4 standard. A subset of the field devices can be designated as access points in an experiment. The field devices and access points form a multihop wireless mesh network running WSAN protocols. A key capability of our testbed is a wired backplane network that can be used for managing wireless experiments and measurements without interfering with wireless communication. The backplane network consists of USB cables and hubs connecting the field devices and microservers, which are in turn connected to a server through the Ethernet. The microservers are Linksys NSLU2 microservers running Linux, and they are responsible for forwarding network management traffic between the field devices and the server. The server runs network management processes, gathers statistics on network behavior, and provides information to system users. The server also serves as a gateway and runs the network manager of the WSAN. The clients are regular computers that users employ to manage their wireless experiments and collect data from the experiments through the server and the backbone network.

Following the practice of industrial deployment, the routing algorithms used in this paper consider only reliable links with pulse-repetition rate higher than 80%. We use eight data flows in our experiments. We run our experiments such that each flow can deliver at least 500 packets from its source to its destination. Fig. 3 shows the network topology along with a set of flows used in this paper. We also repeat our experiments with two other network configurations by varying the location of access points, sources, and destinations.

A. Experimentation of Source and Graph Routing

We conduct a comparative study of the two alternative routing approaches adopted by WirelessHART, namely source routing and graph routing. Specifically, we investigate the tradeoff among reliability, latency, and energy consumption under the different routing approaches. We run two sets of experiments, one with the source routing and one with the graph routing. We repeat the experiments under a clean environment, a noisy environment, and a stress testing environments.

1) *Clean:* We blacklist the four 802.15.4 channels overlapping with our campus Wi-Fi network and run the experiments on the remaining 802.15.4 channels.

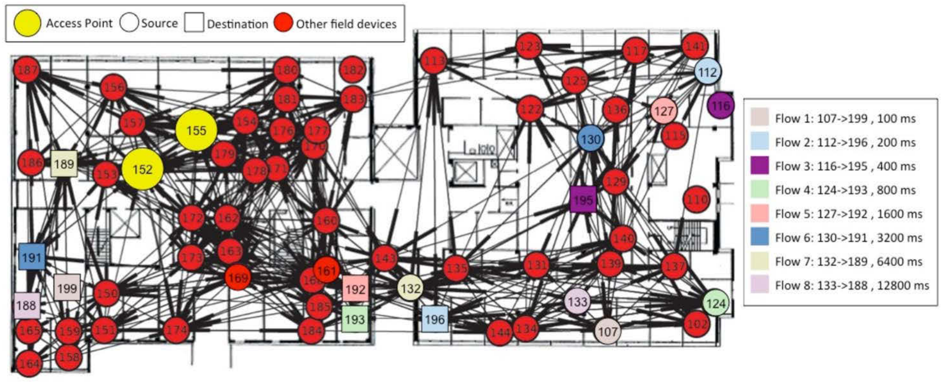

Fig. 3. Locations of access points and field devices. The bigger yellow circles denote the access points that communicate with the network manager running on the server through the wired backbone network. The other circles and squares denote the field devices. The source and destination of a flow are represented as a circle and a square, respectively. The pair of source and destination of a same flow uses the same color. The period of each flow is randomly selected from the range of $2^{0\sim7}$ s, which falls within the common range of periods used in process industries.

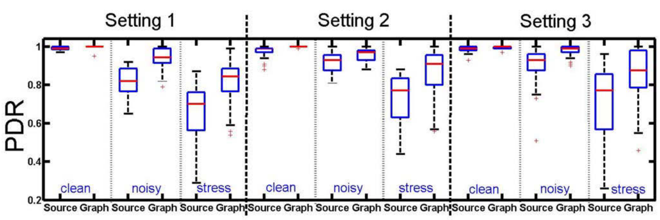

Fig. 4. Box plot of the PDR of source routing and graph routing in the clean, noisy, and stress testing environments. The central mark in the box indicates median; the bottom and top of the box represent the 25th percentile (q_1) and 75th percentile (q_2), respectively; crosses indicate outliers ($x > q_2 + 1.5 \cdot (q_2 - q_1)$ or $x < q_1 - 1.5 \cdot (q_2 - q_1)$); whiskers indicate range excluding outliers. Vertical lines delineate three different network configurations.

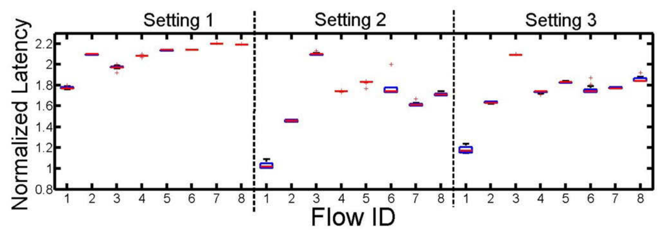

Fig. 5. Box plot of the normalized latency of source routing and graph routing of each flow under graph routing over that under source routing.

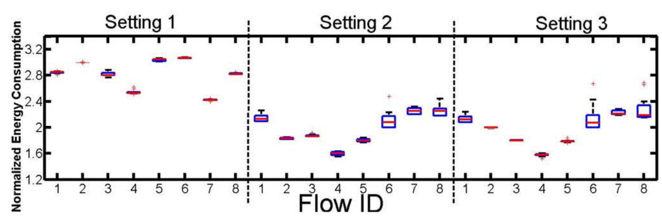

Fig. 6. Box plot of the normalized energy consumption of source routing and graph routing of each flow under graph routing over that under source routing.

2) *Noisy:* We run the experiments by configuring the network to use channels 16 to 19, which overlap with our campus Wi-Fi network.[3]
3) *Stress Testing:* We run the experiments with channels 16 to 19 under controlled interference, in the form of a laptop and an access point generating 1 Mb/s UDP traffic over Wi-Fi channel 6, which overlaps with 802.15.4 channels 16 to 19.

We use the packet delivery rate (PDR) as the metric for network reliability. The PDR of a flow is defined as the percentage of packets that are successfully delivered to their destination. Fig. 4 compares the network reliability under source routing and graph routing in the three environments. As shown in Fig. 4, under the first network configuration, compared to source routing, graph routing improves the median PDR by a margin of 1.0% (from 0.99 to 1.0), 15.9% (from 0.82 to 0.95), and 21.4% (from 0.70 to 0.85) in the clean, noisy, and stress testing environments, respectively. Graph routing shows similar improvement over source routing under the other two network configurations. More importantly, graph routing delivers a significant improvement in min PDR and achieves a smaller variation of PDR than source routing, which represents a significant advantage in industrial applications that demand predictable performance. The improvements in min PDR are 35.5% and 63.5% in noisy and stress testing, respectively. This result shows that graph routing is indeed more resilient to interference due to route diversity. However, as shown in Fig. 5, route diversity incurs a cost in term of latency, with graph routing suffering an average of 80% increase in end-to-end latency. We also estimate the energy consumption based on timestamps of radio activities and the radio's power consumption in each state. As Fig. 6 shows, graph routing consumes an average of 130% more energy than source routing.

Observation 1: Graph routing leads to significant improvement over source routing in term of worst-case reliability, at the cost of longer latency and higher energy consumption. It is therefore important to employ graph routing algorithms specifically designed to optimize latency and energy efficiency.

[3]Co-existence of WirelessHART devices and WiFi is common in industrial deployments since WiFi is often used as backhauls to connect multiple WSANs.

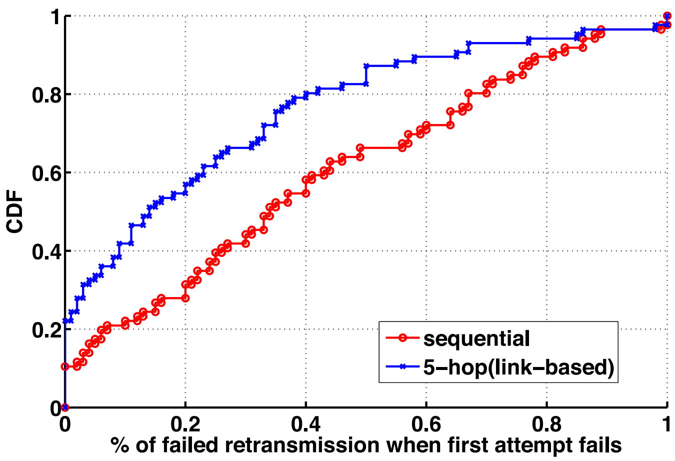

Fig. 7. CDF of percentage of a failed retransmission when first transmission attempt fails.

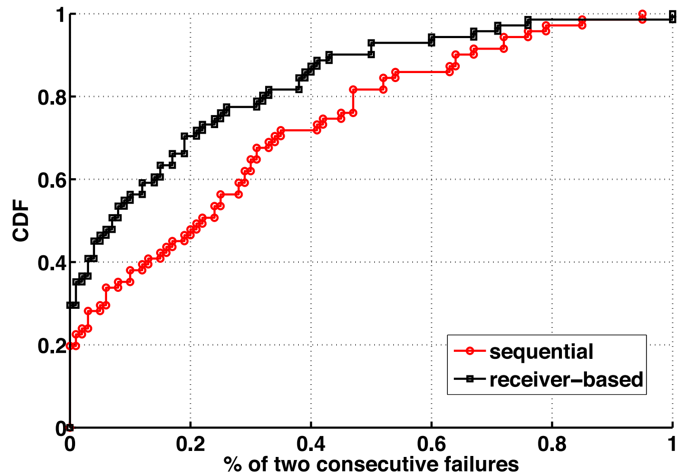

Fig. 8. CDF of percentage of two consecutive transmission failures over links that share a common receiver.

B. Impact of Channel Hopping on Burstiness of Transmission Failures

As shown in the previous studies [8], [18], [22], [29], [31]–[33] (and confirmed on our testbed), the burstiness of transmission failures significantly compromises the reliability and energy efficiency of WSANs. WirelessHART mitigates this issue by adopting spectrum diversity through sequential channel hopping in each time slot. Notably, while burstiness of transmissions in a same channel have been studied extensively, there have been few empirical studies of burstiness under channel hopping. To explore the impact of channel hopping, we run experiments on multiple links using 16 IEEE 802.15.4 channels under controlled interference generated by a Wi-Fi access point and a laptop. We run the experiments with twenty different pairs of senders and receivers. In each run, each sender transmits 500 packets. If a transmission fails, a sender hops to the next channel and retransmits a packet. Our previous study [29] showed that adjacent channels may suffer from bursty transmission failures due to significant correlation between adjacent channels. We hypothesize that increasing hopping distance can also improve the reliability of WSANs. Therefore, we conduct another study by increasing to 5 the distance between the channels used for a transmission and its retransmission.

Fig. 7 shows the cumulative distribution function (CDF) of the failure ratio of retransmissions following failed transmissions. Under sequential channel hopping, 33.5% of links have a 50% retransmission failure, while under a channel hopping distance of 5, only 13.5% of links have a 50% retransmission failure. To recover from a failed transmission, hopping over a large channel distance is more effective than sequential channel hopping. This may be due to the fact that interference often span multiple adjacent channels. For example, WiFi signals usually overlap with four channels of IEEE 802.15.4.aa In addition to burstiness-of-transmission failures over a link, we also observe that links sharing the same receiver can also suffer from strong correlations of transmission failures. We apply channel hopping to links with a common receiver. For each setup, we pick three links that involve the same receiver. Each sender takes turns sending out a packet and switch to a channel that is one channel (sequential) or five channels away from the channel used by the previous sender. Each experiment lasts 500 rounds.

Fig. 8 presents the CDF of the percentage of two consecutive transmission failures over links that share a common receiver. Again, increasing the channel distance used for channel hopping effectively reduces consecutive failures over links to a same receiver, and therefore mitigates the correlations of links to the same receiver.

Observation 2: A larger channel distance can reduce consecutive transmission failures over links sharing a common receiver.

We have shown the limitation of sequential channel hopping over either the same link or links sharing a receiver. In practice, the effect of channel distance may vary in different wireless environments depending on the source of interference and on wireless conditions. The channel distance therefore should be treated as a tunable parameter that needs to be selected based on field testing and knowledge about existing interference and wireless environments.

V. ENHANCED CHANNEL HOPPING FOR WIRELESSHART

In this section, we present and evaluate configurable channel stride (CCS), a novel channel hopping algorithm designed to improve the reliability and energy efficiency of WirelessHART networks. CCS has several salient features that distinguish itself from existing channel hopping approaches in WirelessHART and other networks. First, CCS enforces a specified channel distance between transmissions, thereby avoiding adjacent channels with strong correlations. Second, CCS combines link-based and receiver-based channel hopping, further enhancing its effectiveness in reducing bursty transmission failures. Finally, CCS is specifically tailored for WirelessHART protocols such as graph routing and per-slot channel hopping.

A. Configurable Channel Stride Algorithm

In our empirical study in Section IV-B, we observed that consecutive retransmissions over a same link on adjacent channels cannot effectively eliminate transmission failures on primary routing paths due to strong channel correlation. We also observed a large number of consecutive failures when multiple senders transmit packets back to back using adjacent channels to a shared destination. To mitigates both per-link and per-receiver burstiness of failures, our algorithm combines two channel hopping approaches: 1) link-based channel hopping for links located on primary routing paths and 2) receiver-based channel hopping for links sharing the same destination.

Algorithm 1 shows the pseudo-code of our CCS algorithm. The input of our algorithm is both a desired channel hopping distance and a transmission schedule for the superframe, which is generated by the routing and scheduling algorithm, and which also specifies a set of transmissions scheduled for each time slot. Within a slot, transmissions are ordered according to their flow priority. For instance, tx_{ij} denotes the transmission j scheduled in slot i. $dest_{ij}$ denotes the receiver of tx_{ij}. $flow_{ij}$ denotes the flow tx_{ij} belongs to. $ChannelPool_i$ denotes a channel pool including all current available channels that can be scheduled in slot i. The output of the algorithm is the channel assigned to each tx_{ij}, denoted by $channel_{ij}$.

Our channel-hopping algorithm separates the channel used for a packet transmission from the one used for its retransmission such that the distance between them is at least h hop away. For instance, if the transmission tx_{ij} is the second transmission attempt (retransmission) over a link located on a primary path, a channel is chosen from the $ChannelPool_i$ such that it is at least h hop away from $channel_{1st\_tx}$, where $channel_{1st\_tx}$ is a channel assigned to the first transmission attempt over this link. Our receiver-based channel hopping requires that when a transmission tx_{ij} is on a backup path and there exists a prior transmission of $flow_{ij}$ to a receiving node $dest_{ij}$, then a selected channel must be at least h hop away from the channel used by $flow_{ij}$'s last transmission to $dest_{ij}$. In both cases, if such a channel does not exist, we assign tx_{ij} to use a channel in a $ChannelPool_i$ with a maximum spectral distance instead.

Algorithm 1: Configurable Channel Stride Algorithm

```
h: target channel distance per channel hop;
tx_ij: the j^th transmission assigned to time slot i;
channel_ij: the channel assigned to tx_ij;
flow_ij: the flow that tx_ij belongs to;
dest_ij: the receiver node of tx_ij;
channel_1st_tx: the channel assigned to the first
transmission attempt;
ChannelPool_i: the set of available channels for time slot i;
for each time slot i within a superframe S do
    for each transmission tx_ij scheduled in the time slot i
    do
        if tx_ij is on a primary path then
            if tx_ij is the first attempt for a transmission
            then
                channel_ij ← first channel in
                ChannelPool_i;
                channel_1st_tx ← channel_ij;
            else /* tx_ij is a retransmission */
                if there exists a channel c in
                ChannelPool_i that is at least h hop away
                from channel_1st_tx then
                    channel_ij ← c;
                else
                    channel_ij ← channel in ChannelPool_i
                    with a maximum channel distance
                    from channel_1st_tx;
        else    /* tx_ij is on a backup path */
            if there is no prior transmission to dest_ij from
            flow_ij then
                channel_ij ← first channel in
                ChannelPool_i;
            else
                if there exists a channel c in ChannelPool_i
                that is at least h hop away from flow_ij's
                last transmission to dest_ij then
                    channel_ij ← c;
                else
                    channel_ij ← channel in ChannelPool_i
                    with a maximum channel distance
                    from flow_ij's last transmission to
                    dest_ij;
        Remove channel_ij from ChannelPool_i;
```

B. Evaluation

We ran our experiments under three different network configurations by varying the location of access points, sources, and destinations. Under each configuration, we performed four experimental runs: a first run (sequential) using the sequential channel hopping approach suggested by the WirelessHART standard [36], a second run (link-based) using our algorithm with only link-based channel hopping enabled, a third run (receiver-based) enabling only our receiver-based channel hopping approach, and a fourth run (link+receiver-based) enabling both our channel hopping approaches. We performed the experiments on our testbed under controlled interference (see stress testing setup in Section III-B), use all 16 channels in 2.4 GHz, and set channel hopping distance h to 5.[4]

[4]Our empirical study shows that the probability of simultaneous channel failures drops off as channel distance increases to more than 3.

Fig. 9 shows the CDF of the PDR for different channel assignment approaches. Each data point represents a percentage of flows with 100 generated packets that have a PDR less than or equal to x. Under sequential channel hopping, only 55.2% of flows achieve a PDR larger than 90%. However, under receiver-based, link-based, and integration (link+receiver-based) policy, 69.5%, 81.9%, and 85.1% of flows, respectively, attain a PDR larger than 90%. These results demonstrate the effectiveness of a larger channel hopping distance and the complementary benefits of link-based and

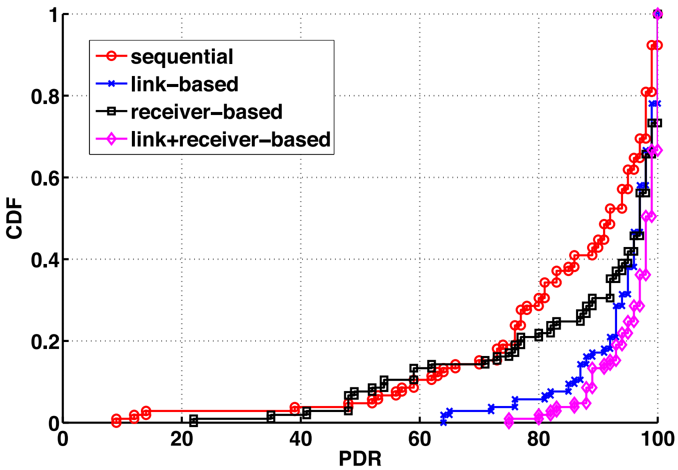

Fig. 9. CDF of PDR under four channel assignment approaches.

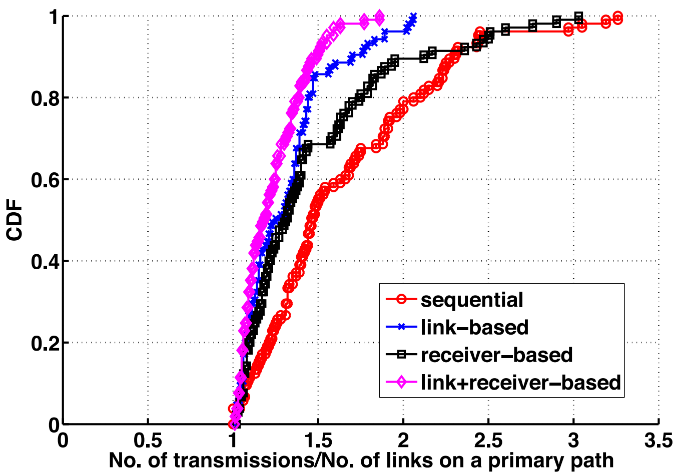

Fig. 10. CDF of the number of transmission attempts per the number of links on a primary path under four channel assignment approaches.

receiver-based channel hopping. Overall the full CCS algorithm increased the number of reliable flows (with a PDR above 90% PDR) by 54% compared to the sequential channel hopping approach. Furthermore, CCS drastically improved the reliability of the flows, seeing the worst PDR among all the flows. Under sequential channel hopping, the least reliable flow experienced a PDR of only 9%. In contrast, under the full CCS algorithm, the least reliable flow still achieved a PDR of 75%. This shows that our CCS policy benefits industrial applications that demand a high degree of reliability and predictability.

Fig. 10 presents a CDF of the number of transmission attempts per number of links on a primary path. With the standard channel assignment, 44.8% of flows requires more than 1.5*X* transmission attempts to achieve a desired PDR. Our receiver-based, link-based, and integration approaches are proved to be more efficient, with 31.0%, 14.0%, and 8.6% of flows requiring 1.5*X* transmission attempts, respectively. In a worst-case scenario, sequential, receiver-based, link-based, and integration policies yield at most 3.3*X*, 3.0*X*, 2.1*X*, and 1.9*X* transmission attempts. Hence, our channel hopping policy provides a notable reduction in the number of transmission attempts of each flow to achieve a desired PDR, which indicates better link quality and can result in lower energy consumption.

VI. Related Works

In recent years, there has been increasing interest in studying industrial WSANs. Previous research mostly focused on network algorithms and theoretical analysis. Zhang *et al.* [38] designed a link scheduling and channel assignment algorithm for a simplified linear network model, while Soldati *et al.* [30] studied the same problem for tree network models. Rao *et al.* [24] studied the tradeoff between energy consumption and network performance. Franchino and Buttazzo [6] proposed a real-time energy-aware MAC layer protocol. Han *et al.* [7] presented a graph routing algorithm. Saifullah *et al.* [25], [26] presented a series of theoretical results on real-time transmission scheduling, rate selection for wireless control, and delay analysis [27], [37]. Readers are referred to a recent review article for comprehensive survey on these works [16]. Real-time transmission scheduling algorithms have also been studied in the context of WSNs [3], [12], [21]. All these works are based on theoretical analysis and simulation studies. In contrast to the existing research focused on theoretical aspects of industrial WSANs, this paper presents an experimental study of WSAN protocols on a physical testbed that implements a set of network mechanisms of the WirelessHART standard. This paper is therefore complementary to previous work in this area.

There has been recent work that implemented and evaluated real-time WSN protocols experimentally. Recently, O'donovan *et al.* [20] developed the GINSENG system, which uses WSN to support mission-critical applications in industrial environments and shared their valuable experience during real-world deployments. Munir *et al.* [18] designed a scheduling algorithm that produces latency bounds of the real-time periodic streams and accounts for both link bursts and interference. Pöttner *et al.* [23] designed a scheduling algorithm to meet application requirements in terms of data delivery latency, reliability, and transmission power. While valuable insights can be drawn from the aforementioned efforts, the novelty of this paper lies in its focus on key aspects of the WirelessHART standard, such as graph routing, that were not studied in earlier works. Our results are therefore complementary to earlier findings on other aspects of real-time WSANs.

There have been recent empirical studies that investigated the burstiness of transmission failures and 802.15.4 channel performance in various wireless environments and network settings. Srinivasan *el al.* [31]–[33] performed a series of link studies to quantify the burstiness of intralink and interlink performance on their office testbed. Sha *et al.* [29] performed a spectrum study in the 2.4 GHz band as well as a link study of IEEE 802.15.4 channels in residential environments. Hauer *et al.* [8] conducted a multichannel measurement of body area networks. Ortiz and Culler [22] evaluated the multichannel behavior of 802.15.4 networks in a machine room, a computer room, and an office testbed and found path diversity to be an effective strategy to ensure reliability. In contrast to these studies, our own study is specific to WirelessHART's key mechanisms such as graph routing and sequential slot-based channel hopping. Therefore, our results are complementary to these earlier findings.

In recent years, there has been increasing interest in using channel hopping to enhance network reliability. Navda *et al.* [19] proposed a rapid channel hopping scheme to protect a network from jamming attacks. Le *et al.* [14] designed a control theory approach to dynamically allocate channels in a distributed manner. Sha *et al.* [28] designed an opportunistic channel hopping algorithm to avoid jammed channels. Industry standards such as Bluetooth's AFH [4] leveraged constant hopping in a pseudorandom fashion across channels to avoid persistent interference. In contrast to these works, our channel hopping algorithm CCS is designed to improve the reliability and energy efficiency of WirelessHART networks with several features that distinguish itself from existing channel hopping approaches in WirelessHART and other networks. First, CCS enforces a specified channel distance between transmissions, thereby avoiding adjacent channels with strong correlations. Second, CCS combines link-based and receiver-based channel hopping, further enhancing its effectiveness in reducing bursty transmission failures. Finally, CCS is specifically tailored for WirelessHART protocols such as graph routing and per-slot channel hopping.

While this paper focuses on WirelessHART, alternative industrial WSAN standards exist—such as ISA-100.11a [11], WIA-PA [39], and the recently approved IEEE 802.15.4e MAC enhancement standard [2]. These standards share many common approaches and mechanisms. For example, IEEE 802.15.4e specifies a time-slotted channel hopping mode which combines time slotted access, multichannel communication, and channel hopping to improve reliability and mitigate the effects of interference and multipath fading. Our insights and proposed algorithm therefore may be generalized to influence the implementation of WSANs based on these standards.

VII. Conclusion

Industrial WSANs offer an appealing communication technology for process automation applications to incorporate IoT while posing unique challenges due to their critical demands on reliable and real-time communication. Complementary to recent research on theoretical aspects of WSANs, we have implemented a suite of network protocols of the WirelessHART standard in TinyOS and TelosB motes and then performed a series of empirical studies on WSAN protocol designs. We further developed a novel channel hopping algorithm that prevents consecutive transmissions from using channels with strong correlations on a common link or to a common receiver. Experimental results demonstrate that our algorithm can significantly improve network reliability and energy efficiency.

References

[1] *6TiSCH: IPv6 Over the TSCH Mode of IEEE 802.15.4e*. Accessed on Sep. 25, 2014. [Online]. Available: https://datatracker.ietf.org/group/6tisch/charter/

[2] *IEEE 802.15.4e Enhancement Standard*. Accessed on Sep. 25, 2014. [Online]. Available: http://standards.ieee.org/getieee802/download/802.15.4e-2012.pdf

[3] T. F. Abdelzaher, S. Prabh, and R. Kiran, "On real-time capacity limits of multihop wireless sensor networks," in *Proc. RTSS*, Lisbon, Portugal, 2004, pp. 359–370.

[4] *Specification of the Bluetooth System, Version 4.0*. Accessed on Sep. 25, 2014. [Online]. Available: https://www.bluetooth.com/

[5] *CC2420 Radio Stack*. Accessed on Sep. 25, 2014. [Online]. Available: http://www.tinyos.net/ tinyos-2.x/doc/html/tep126.html

[6] G. Franchino and G. Buttazzo, "WBuST: A real-time energy-aware MAC layer protocol for wireless embedded systems," in *Proc. ETFA*, Kraków, Poland, 2012, pp. 1–8.

[7] S. Han, X. Zhu, A. K. Mok, D. Chen, and M. Nixon, "Reliable and real-time communication in industrial wireless mesh networks," in *Proc. RTAS*, Chicago, IL, USA, 2011, pp. 3–12.

[8] J.-H. Hauer, V. Handziski, and A. Wolisz, "Experimental study of the impact of WLAN interference on IEEE 802.15.4 body area networks," in *Proc. EWSN*, Cork, Ireland, 2009, pp. 17–32.

[9] *Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs)*, IEEE Standard 802.15.4a, 2006.

[10] P. C. Evans and M. Annunziata. (Nov. 26, 2012). *Industrial Internet: Pushing the Boundaries of Minds and Machines*. [Online]. Available: http://www.industrialinternetconsortium.org/white-papers.htm

[11] *ISA 100*. Accessed on Sep. 25, 2014. [Online]. Available: http://www.isa.org/isa100

[12] P. Jayachandran and M. Andrews, "Minimizing end-to-end delay in wireless networks using a coordinated EDF schedule," in *Proc. INFOCOM*, San Diego, CA, USA, 2010, pp. 1–9.

[13] H. Kagermann, W. Wahlster, and J. Helbig, "Recommendations for implementing the strategic initiative industrie 4.0," Dept. Commun. Promoters Group Ind.-Sci. Res. Alliance, acatech—German Acad. Sci. Eng., Munich, Germany, Tech. Rep., Apr. 2013. [Online]. Available: http://www.acatech.de/fileadmin/user_upload/Baumstruktur_nach_Website/Acatech/root/de/Material_fuer_Sonderseiten/Industrie_4.0/Final_report__Industrie_4.0_accessible.pdf

[14] H. K. Le, D. Henriksson, and T. Abdelzaher, "A practical multi-channel media access control protocol for wireless sensor networks," in *Proc. IPSN*, St. Louis, MO, USA, 2008, pp. 70–81.

[15] J. W. S. Liu, *Real-Time Systems*. Upper Saddle River, NJ, USA: Prentice-Hall, 2000.

[16] C. Lu *et al.*, "Real-time wireless sensor-actuator networks for industrial cyber-physical systems," *Proc. IEEE*, vol. 104, no. 5, pp. 1013–1024, May 2016.

[17] M. Maróti, B. Kusy, G. Simon, and Á. Lédeczi, "The flooding time synchronization protocol," in *Proc. ACM SenSys*, Baltimore, MD, USA, 2004, pp. 39–49.

[18] S. Munir *et al.*, "Addressing burstiness for reliable communication and latency bound generation in wireless sensor networks," in *Proc. IPSN*, Stockholm, Sweden, 2010, pp. 303–314.

[19] V. Navda, A. Bohra, S. Ganguly, and D. Rubenstein, "Using channel hopping to increase 802.11 resilience to jamming attacks," in *Proc. INFOCOM*, Anchorage, AK, USA, 2007, pp. 2526–2530.

[20] T. O'donovan *et al.*, "The GINSENG system for wireless monitoring and control: Design and deployment experiences," *ACM Trans. Sensor Netw.*, vol. 10, no. 1, Nov. 2013, Art. no. 4.

[21] R. S. Oliver and G. Fohler, "Probabilistic estimation of end-to-end path latency in wireless sensor networks," in *Proc. MASS*, 2009, pp. 423–431.

[22] J. Ortiz and D. Culler, "Multichannel reliability assessment in real world WSNs," in *Proc. IPSN*, Stockholm, Sweden, 2010, pp. 162–173.

[23] W.-B. Pöttner, H. Seidel, J. Brown, U. Roedig, and L. Wolf, "Constructing schedules for time-critical data delivery in wireless sensor networks," *ACM Trans. Sensor Netw.*, vol. 10, no. 3, Apr. 2014, Art. no. 44.

[24] L. Rao, X. Liu, J.-J. Chen, and W. Liu, "Joint optimization of system lifetime and network performance for real-time wireless sensor networks," in *Proc. QSHINE*, Las Palmas, Spain, 2009, pp. 317–333.

[25] A. Saifullah *et al.*, "Near optimal rate selection for wireless control systems," in *Proc. RTAS*, Beijing, China, 2012, pp. 231–240.

[26] A. Saifullah, Y. Xu, C. Lu, and Y. Chen, "Real-time scheduling for WirelessHART networks," in *Proc. RTSS*, San Diego, CA, USA, 2010, pp. 150–159.

[27] A. Saifullah, Y. Xu, C. Lu, and Y. Chen, "End-to-end delay analysis for fixed priority scheduling in WirelessHART networks," in *Proc. RTAS*, Chicago, IL, USA, 2011, pp. 13–22.

[28] M. Sha, G. Hackmann, and C. Lu, "ARCH: Practical channel hopping for reliable home-area sensor networks," in *Proc. RTAS*, Chicago, IL, USA, 2011, pp. 305–315.

[29] M. Sha, G. Hackmann, and C. Lu, "Real-world empirical studies on multi-channel reliability and spectrum usage for home-area sensor networks," *IEEE Trans. Netw. Service Manag.*, vol. 10, no. 1, pp. 56–69, Mar. 2013.

[30] P. Soldati, H. Zhang, and M. Johansson, "Deadline-constrained transmission scheduling and data evacuation in WirelessHART networks," in *Proc. ECC*, Budapest, Hungary, 2009, pp. 4320–4325.

[31] K. Srinivasan, P. Dutta, A. Tavakoli, and P. Levis, "An empirical study of low power wireless," *ACM Trans. Sensor Netw.*, vol. 6, no. 2, 2010, Art. no. 16.

[32] K. Srinivasan *et al.*, "The k-factor: Inferring protocol performance using inter-link reception correlation," in *Proc. Mobicom*, Chicago, IL, USA, 2010, pp. 317–328.

[33] K. Srinivasan, M. A. Kazandjieva, S. Agarwal, and P. Levis, "The b-factor: Measuring wireless link burstiness," in *Proc. SenSys*, Raleigh, NC, USA, 2008, pp. 29–42.

[34] *TelosB: Telosb Mote Platform Datasheet Provided by MEMSIC Inc.* Accessed on Sep. 25, 2014. [Online]. Available: http://www.memsic.com/

[35] *TinyOS*. Accessed on Sep. 25, 2014. [Online]. Available: http://www.tinyos.net/

[36] *WirelessHART*. Accessed on Sep. 25, 2014. [Online]. Available: http://www.hartcomm2.org

[37] C. Wu *et al.*, "Analysis of EDF scheduling for wireless sensor-actuator networks," in *Proc. IWQoS*, Hong Kong, 2014, pp. 31–40.

[38] H. Zhang, P. Soldati, and M. Johansson, "Optimal link scheduling and channel assignment for convergecast in linear WirelessHART networks," in *Proc. WiOPT*, Seoul, South Korea, 2009, pp. 1–8.

[39] T. Zhong, C. Mengjin, Z. Peng, and W. Hong, "Real-time communication in WIA-PA industrial wireless networks," in *Proc. ICCSIT*, Chengdu, China, 2010, pp. 600–605.

Mo Sha (GS'09–M'14) received the B.Eng. degree from Beihang University, Beijing, China, in 2007, the M.Phil. degree from the City University of Hong Kong, Hong Kong, in 2009, and the Ph.D. degree in computer science from Washington University in St. Louis, St. Louis, MO, USA, in 2014.

He is an Assistant Professor with the Department of Computer Science, Binghamton University (the State University of New York at Binghamton) Binghamton, NY, USA. His current research interests include wireless networks, Internet of Things, embedded and real-time systems, and cyber-physical systems.

Dolvara Gunatilaka received the B.S. degree in information and communication technology from Mahidol University, Salaya, Thailand, in 2010, and the M.S. degree in computer science from Washington University in St. Louis, St. Louis, MO, USA, in 2013, where she is currently pursuing the Ph.D. degree in computer science.

Her current research interests include industrial wireless sensor-actuator networks.

Chengjie Wu received the B.S. and M.S. degrees from Tsinghua University, Beijing, China, in 2006 and 2008, respectively, and the Ph.D. degree in computer science from Washington University in St. Louis, St. Louis, MO, USA, in 2014.

He is a Research Scientist with Facebook, Menlo Park, CA, USA. His current research interests include real-time systems, cyber-physical systems, wireless sensor networks, and Internet of Things.

Dr. Wu was a recipient of the McDonnell International Scholarship, the Emerson Corporate Fellowship Award by Washington University in St. Louis and the Best Paper Nomination at the RTAS'12.

Chenyang Lu (M'02–SM'13–F'16) received the B.S. degree from the University of Science and Technology of China, Hefei, China, in 1995, the M.S. degree from the Chinese Academy of Sciences, Beijing, China, in 1997, and the Ph.D. degree from the University of Virginia, Charlottesville, VA, USA, in 2001.

He is currently a Fullgraf Professor with the Department of Computer Science and Engineering, Washington University in St. Louis, St. Louis, MO, USA. He has authored or co-authored over 150 research papers with over 14 000 citations and an H-index of 52. His current research interests include real-time systems, wireless sensor networks, cyber-physical systems, and Internet of Things.

Dr. Lu is the Editor-in-Chief of the *ACM Transactions on Sensor Networks*, an Area Editor of the IEEE INTERNET OF THINGS JOURNAL, and an Associate Editor of the *ACM Transactions on Cyber-Physical Systems and Real-Time Systems*. He also chaired premier conferences such as the IEEE Real-Time Systems Symposium, the ACM/IEEE International Conference on Cyber-Physical Systems, and the ACM Conference on Embedded Networked Sensor Systems.