Implementation and Experimentation of Industrial Wireless Sensor-Actuator Network Protocols

Mo Sha[1,2,*], Dolvara Gunatilaka[1,*], Chengjie Wu[1], and Chenyang Lu[1]

[1] Cyber-Physical Systems Laboratory, Washington University in St. Louis,
St. Louis, MO, USA
[2] National Renewable Energy Laboratory, Golden, CO, USA

Abstract. Wireless sensor-actuator networks (WSANs) offer an appealing communication technology for process automation applications. However, such networks pose unique challenges due to their critical demands on reliability and real-time performance. While industrial WSANs have received attention in the research community, most published results to date 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 widely adopted in the process industries, including multi-channel TDMA with shared slots at the MAC layer and reliable graph routing supporting path redundancy. We then performed a comparative study of the two alternative routing approaches adopted by WirelessHART, namely source routing and graph routing. Our study shows that 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.

1 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 effort to reconfigure systems to accommodate new production process requirements.

---

[*] Mo Sha started this research project while at Washington University in St. Louis and currently works at National Renewable Energy Laboratory. The first two authors contributed equally to this work.

T. Abdelzaher et al. (Eds.): EWSN 2015, LNCS 8965, pp. 234–241, 2015.

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.

Industrial WSANs pose unique challenges due to their critical demands on reliable and real-time communication. Violation of their reliability and real-time requirements may result in plant shutdowns, safety hazard, or economic/environmental impacts.

To meet the stringent requirements on reliability and predictable real-time performance, industrial WSAN standards such as WirelessHART [15] made a set of unique network design choices.

- The network should support both source routing and reliable graph routing: the source routing provides a single route for each data flow, whereas the graph routing provides multiple redundant routes based on a routing graph.
- The network should also adopt a multi-channel Time Division Multiple Access (TDMA), employing both dedicated and shared time slots, at the 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 a same shared slot. The packet transmission occurs immediately in a dedicated slot, while a 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 is a critical need for experimental testbeds for validating and evaluating network research on industrial WSANs. Without sufficient experimental evaluation, industry consequently has shown a marked reluctance to embrace new solutions.

To meet the need for experimental research on WSANs, we have developed 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 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. This study leads to our insight on the development of resilient industrial WSANs that 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.

The rest of the paper is organized as follows. Section 2 describes our implementation of WirelessHART protocols and Section 3 presents our experimentation of source and graph routing. Section 4 reviews related work and Section 5 concludes the paper.

2 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 and TelosB motes. 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[1] to generate routes for source routing and follow the algorithm proposed in [6] to generate the reliable graphs. The schedule generator uses rate monotonic scheduling algorithm [7] to generate transmission schedules.

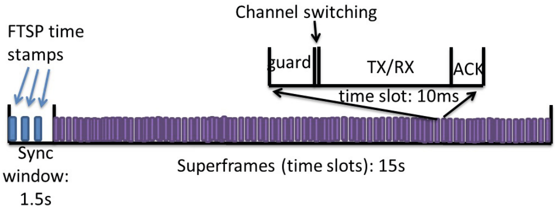

Fig. 1. Time frame format of RT-MAC

Our protocol stack adopts the CC2420x radio driver [2] as the radio core, which provides an open-source implementation of IEEE 802.15.4 physical layer in TinyOS [1] 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 multi-channel TDMA MAC protocol, *RT-MAC*, which implements the key features of WirelessHART's MAC protocol. As shown in Figure 1, 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) [8] is executed during the Sync window to synchronize the clocks of all wireless devices over the entire network. Our micro-benchmark experiment shows that a FTSP's time stamp packet can finish the traversing of the 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

---

[1] 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%).

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 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.

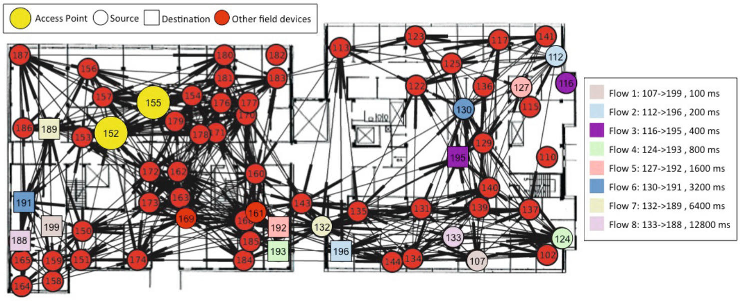

Fig. 2. Locations of access points and field devices. The bigger yellow circles denote the access points which communicate with the network manager running on the server through the wired backbone network. The other circles and squares denote the field devices. The source and destination of a flow are 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 $2^{0\sim 7}$ seconds, which falls within the common range of periods used in process industries.

3 Experimentation of Source and Graph Routing

Our WSAN testbed includes 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 [4], 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 Ethernet. The microservers are Linksys NSLU2 microservers running Linux. Microservers 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 used by users 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 our study only consider reliable links with PRR higher than 80%. We use 8 data flows in our experiments. We run our experiments long enough such that each flow can deliver at least 500 packets from its source to its destination. Figure 2 shows the network topology along with a set of flows used in our study. We also repeat our experiments with two other network configurations by varying the location of access points, sources, and destinations.

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, once with the source routing and once 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.
2. Noisy: we run the experiments by configuring the network to use channels 16 to 19, which overlap with our campus Wi-Fi network[2].
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 Mbps 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. Figure 3 compares the network reliability under source routing and graph routing in the three environments. As shown in Figure 3, 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 Figure 4, route diversity incurs a cost in term of latency, with graph routing suffers an average

---

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

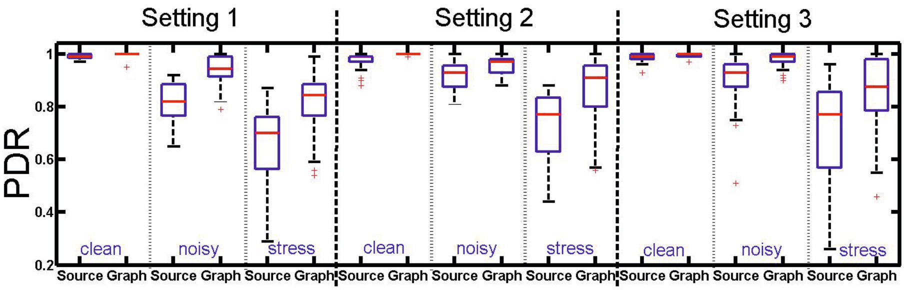

Fig. 3. Box plot of the PDR of source routing and graph routing in the clean, noisy, and stress testing environments. Central mark in box indicates median; bottom and top of box represent the 25th percentile (q_1) and 75th percentile (q_2); crosses indicate outliers ($x > q_2 + 1.5 \cdot (q_2 - q_1)$ or $x < q_1 - 1.5 \cdot (q_2 - q_1)$); whiskers indicate range excluding outliers. Vertical lines delineate three different network configurations.

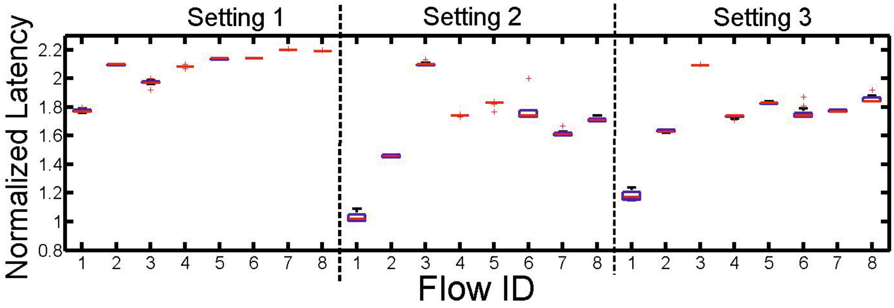

Fig. 4. Box plot of the normalized latency of source routing and graph routing of each flow under graph routing over that under source routing

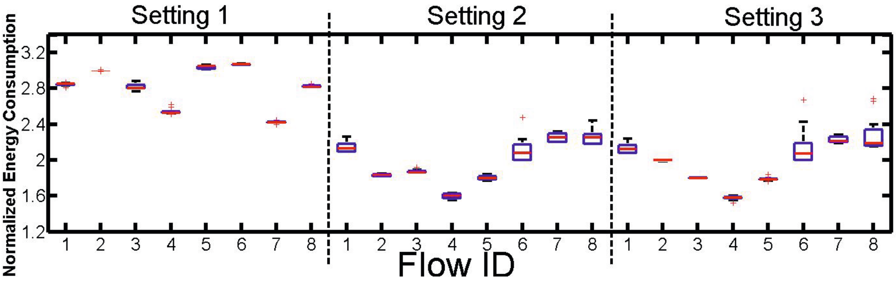

Fig. 5. Box plot of the normalized energy consumption of source routing and graph routing of each flow under graph routing over that under source routing

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 Figure 5 shows, graph routing consumes an average of 130% more energy over source routing.

Observation: *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.*

4 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. [17] developed a latency-optimal link scheduling policy and established performance bounds for convergecast in WirelessHART networks. Chraim et al. [3] studied the decentralized sequential hypothesis testing problems for the WirelessHART networks at the theoretical level. Franchino et al. [5] proposed a real-time energy-aware MAC layer protocol. Han et al. [6] presented a graph routing algorithm. Saifullah et al. presented a series of theoretical results on real-time transmission scheduling [13], rate selection for wireless control [12], and delay analysis [14,16]. All these works are based on theoretical analysis and simulation studies. In this paper, we present an experimental study of WSAN protocols on a physical testbed that implements a set of network mechanisms of the WirelessHART standard.

There has been recent work that implemented and evaluated real-time WSN protocols experimentally. Recently, O'donovan et al. [10] developed the GINSENG system which uses a WSN to support mission-critical applications in industrial environments. Munir et al. [9] designed a scheduling algorithm that produces latency bounds of the real-time periodic streams and accounts for both link bursts and interference. Pottner et al. [11] 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 those efforts, the novelty of our work lies in its focus on key aspects of the WirelessHART standard such as graph routing that was not studied in earlier work.

5 Conclusion

Industrial WSANs offer a promising technology for process automation while posing unique challenges due to their critical demands on reliable and real-time communication. Complementary to recent research on theoretical aspects of WSAN design, we have implemented a suite of network protocols of the WirelessHART standard in TinyOS and TelosB motes and then empirically studied the tradeoff among reliability, latency, and energy consumption under source and graph routing in a 55-node testbed. Our experimental results show that 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.

Acknowledgments. This work is supported, in part, by NSF through grants 1320921 (NeTS), 1144552 (NeTS) and 1035773 (CPS).

References

1. `http://www.tinyos.net/`
2. CC2420 radio stack, `http://www.tinyos.net/tinyos-2.x/doc/html/tep126.html`
3. Chraim, F., Pister, K.: Smart fence: Decentralized sequential hypothesis testing for perimeter security. In: ICST (2013)
4. Crossbow Technology. TelosB mote platform, `http://www.xbow.com/Products/Product_pdf_files/Wireless_pdf/TelosB_Datasheet.pdf`
5. Franchino, G., Buttazzo, G.: WBuST: A real-time energy-aware MAC layer protocol for wireless embedded systems. In: ETFA (2012)
6. Han, S., Zhu, X., Mok, A., Chen, D., Nixon, M.: Reliable and real-time communication in industrial wireless mesh networks. In: RTAS (2011)
7. Liu, J.W.S.: Real-time systems (2000)
8. Maróti, M., Kusy, B., Simon, G., Lédeczi, Á.: The flooding time synchronization protocol. In: SenSys (2004)
9. Munir, S., Lin, S., Hoque, E., Nirjon, S., Stankovic, J., Whitehouse, K.: Addressing burstiness for reliable communication and latency bound generation in wireless sensor networks. In: IPSN (2010)
10. O'donovan, T., Brown, J., Busching, F., Cardoso, A., Cecilio, J., Do, J., Furtado, P., Gil, P., Jugel, A., Pottner, W.-B., Roedig, U., Silva, J.S., Silva, R., Sreenan, C.J., Vassiliou, V., Wolf, T.V.L., Zinonos, Z.: The ginseng system for wireless monitoring and control: Design and deployment experiences. ACM Transactions on Sensor Networks 10(1) (November 2013)
11. Pottner, W.-B., Seidel, H., Brown, J., Roedig, U., Wolf, L.: Constructing schedules for time-critical data delivery in wireless sensor networks. ACM Transactions on Sensor Networks 10(3) (August 2014)
12. Saifullah, A., Wu, C., Tiwari, P., Xu, Y., Fu, Y., Lu, C., Chen, Y.: Near optimal rate selection for wireless control systems. In: RTAS (2012)
13. Saifullah, A., Xu, Y., Lu, C., Chen, Y.: Real-time scheduling for wirelesshart networks. In: RTSS (2010)
14. Saifullah, A., Xu, Y., Lu, C., Chen, Y.: End-to-end delay analysis for fixed priority scheduling in wirelesshart networks. In: RTAS (2011)
15. WirelessHART (2007), `http://www.hartcomm2.org`
16. Wu, C., Sha, M., Gunatilaka, D., Saifullah, A., Lu, C., Chen, Y.: Analysis of EDF Scheduling for Wireless Sensor-Actuator Networks. In: IWQoS (2014)
17. Zhang, H., Soldati, P., Johansson, M.: Performance bounds and latency-optimal schedules for convergecast in WirelessHART networks. IEEE Transactions on Wireless Communications 12(6) (June 2013)