A Study on Polling-based Communication Scheme for Industrial IoT

Submitted to
Graduate School of Information Science and Technology
Osaka University

April 2019

Yuichi IGARASHI

List of Publications

Journal Papers

1. Yuichi Igarashi, Yoshiki Matsuura, Minoru Koizumi, Naoki Wakamiya, “Priority-Based Dynamic Multichannel Transmission Scheme for Industrial Wireless Networks," *Wireless Communication Mobile Compututing*, 2017, doi:10.1155/2017/9124858.

2. Yuichi Igarashi, Ryo Nakano, Naoki Wakamiya, “A Polling-Based Transmission Scheme Using a Network Traffic Uniformity Metric for Industrial IoT Applications," *Sensors*, 2019, 19(1), 187, doi:10.3390/s19010187.

Oral Presentation

1. Yuichi Igarashi, Ryo Nakano, Naoki Wakamiya, “Environment-aware data collection mechanism for industrial sensor networks with dynamic requirements,” *Vietnam-Japan Joint Workshop on Ambient Intelligence and Sensor Networks*, Nov. 2018.

Preface

Internet of Things (IoT) applications, such as environmental monitoring and big data analysis based on sensing data, have been becoming popular since the term IoT was first used in 1999. IoT makes it possible to collect heterogeneous data from sensors (end devices) to determine new and better actions after analyzing the data. When we look at IoT, it is clear that the only simple way how to connect a number of end devices to the Internet is using wireless technologies. There are a lot of different wireless network technologies for IoT, such as Low-power Wireless Personal Area Networks and Low Power Wide Area (LPWA) networks. Most network technologies for IoT are suitable for one-way communications from end devices to a central device.

Industrial systems and their applications also need wireless technologies to collect data and analyze that for the purpose of improving production efficiency, ensuring optimal resource consumption, and operating systems more efficiently and economically. Such Industrial IoT (IIoT) applications require higher reliable and more stable wireless networks than IoT applications. Industrial Wireless Sensor Networks (IWSNs) have been emerging as a new means of wireless communication for IIoT and several protocols for IWSNs, such as WirelessHART and ISA100.11a, have been developed and standardized. Fundamental IIoT applications, such as Supervisory Control And Data Acquisition (SCADA) periodically collect data from end devices as well as IoT applications. In addition, they generate unpredictable on-demand communications in order to collect additional data from end devices or control end devices remotely. However, the current standard IWSN technologies are not suitable for such IIoT applications, because they have difficulty in dealing with collecting data at high success ratio and guaranteeing latency for unpredictable on-demand communications at the same time.

In this thesis, we focus on polling-based communication schemes over IWSNs for Industrial IoT applications. We begin this thesis with clarifying requirements of our target industrial applications and stating issues. To solve them, we propose a new central control scheme first. Our proposed scheme generates uniform network traffic load for heterogeneous multi-cycle periodical data collection.

The uniform network traffic load distribution enables all devices to have fair opportunities to communicate with a central device leading to high success ratio of communication. Through simulation experiments, we confirmed the success ratio of collecting periodic data packets achieves higher than 90%. Moreover, the traffic distribution also leaves uniformly distributed time slots available to unpredictable on-demand communication. Once the need for a central device to obtain data from an end device arises, it can easily and promptly find an empty time slot and send a request without interruption of periodic data collection. Therefore, our proposal also realizes unpredictable on-demand communications between a central device and end devices within guaranteed deadlines.

Some industrial applications, such as building automation and factory automation, require shorter delay and more reliable communication than other fundamental IIoT applications whose main purpose is monitoring the IIoT systems. A feasible technique is to assign bandwidth to delay-sensitive communication to avoid interference and collision with other general periodic communication. However, the amount of required bandwidth cannot be predicted or estimated beforehand, because they are unpredictable and on-demand communication. In addition, control packets for network management, e.g. routing and topology maintenance, must have certain amount of bandwidth to make an IWSN reliable and stable. To improve performance of polling-based communication over IWSNs, we propose a central-autonomous decentralized hybrid scheme, which is incorporating central control by a central device and autonomous decentralized control by end devices in an IWSN. We confirmed that on-demand communications can be delivered within a guaranteed deadline and periodical data collection can receive the satisfactory quality of service at the same time, where available bandwidth of transmitting network control packets is at least 36%, which is larger than the bandwidth that standard IWSNs are originally assigned.

Through the discussions in this thesis, we conclude that polling-based communication schemes, which provide periodical data collection at high success ratio and unpredictable on-demand communication within guaranteed short delay, are fundamental technologies in IIoT era, because the stable data collection from end devices is a source value for all the IIoT applications.

Acknowledgments

This thesis could not have been accomplished without the assistance of many people, and I would like to acknowledge all of them.

Firstly, I would like to express my greatest appreciation to my supervisor, Professor Naoki Wakamiya, Graduate School of Information Science and Technology, Osaka University, for his insightful comments, comprehensive advice, and valuable discussions. He directed me to the appropriate perspective in this domain and inspired me to aim at higher goals.

I am deeply grateful to the members of my thesis committee, Professor Hideo Matsuda, Professor Hiroshi Shimizu, Professor Fumio Matsuda, and Professor Taro Maeda of Graduate School of Information Science and Technology, Osaka University, for their multilateral reviews, perceptive comments from various perspectives and encouragement.

Also, I would like to express my sincere appreciation for Dr. Michitaka Okuno, Department Manager of Hitachi Ltd., Center for Technology Innovation - Electronics, Connectivity Research Department, Dr Ryosuke Fujiwara, Unit Leader at Center for Technology Innovation - Electronics, Connectivity Research Department, Dr. Hiromitsu Kato, Senior Manager of Management Planning Office of Hitachi Ltd., Dr. Seishi Hanaoka, Senior Manager of Management Planning Office of Hitachi Ltd., for giving me the opportunity to study for a doctorate at Osaka University.

Furthermore, my appreciation also goes to my colleagues Dr. Minoru Koizumi, Mr. Ryo Nakano, Mr. Shinichi Tsunoo, Mr. Takuma Nishimura, Mr. Takahiro Yamada, Mr. Yoshimi Fujimata, Mr. Hiroshige Kashiwabara, Mr. Reiso Sasaki, Mr. Jumpei Honda, Mr. Nobuhiro Gotouda, Mr. Kazuya Monden, and Mr. Hiroki Satoh of Hitachi Ltd., and Mr. Yoshiki Matsuura, GROOVE X, Inc. for their valuable discussions, advice, support, and encouragement during this study.

Finally, I want to give my deepest gratitude of thanks to my wife, Rie, my son Motoi, my son Akira, my daughter Sumire, my father Ken, my mother Miyoko, my brother Koji, my parents in low, my uncle Shin, and all the member of my relatives, for their understanding and hearty support and encouragement in my daily life. This work would not have been achieved without them.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Background

1.1.1 Industrial Internet of Things

The term Internet of Things (IoT) was first used in 1999 to describe a system in which objects in the physical world could be connected to the Internet by devices such as sensors, machines, or actuators. This connectivity makes it possible to collect heterogeneous data from devices to determine new and better actions after analyzing the data. A lot of research papers and books about IoT have been published in order to understand what IoT is. Many definitions of IoT have been independently introduced by both individuals and companies [1, 2]. One of definitions for the IoT would be a group of infrastructures, interconnecting connected objects and allowing their management, data mining and the access to data they generate where connected objects are sensor(s) and/or actuator(s) carrying out a specific function that are able to communicate with other equipment [3]. The basic concept of IoT is to connect things together, thus enabling these things to communicate with each other and enabling people and machines to communicate with them. IoT has found applications in several areas, such as smart home, smart energy, smart agriculture, monitoring environments, health care, connected car, and smart grid [4].

Recently, industrial applications and social infrastructure applications use wireless network technologies, machine learning for Big Data analysis, and Cyber-Physical Systems (CPS) to improve production efficiency, ensure optimal resource consumption, and operate systems more efficiently and economically [5–7]. Typical industrial IoT applications are listed below.

- **Advanced metering infrastructure.** Advanced metering infrastructure (AMI) is an integrated

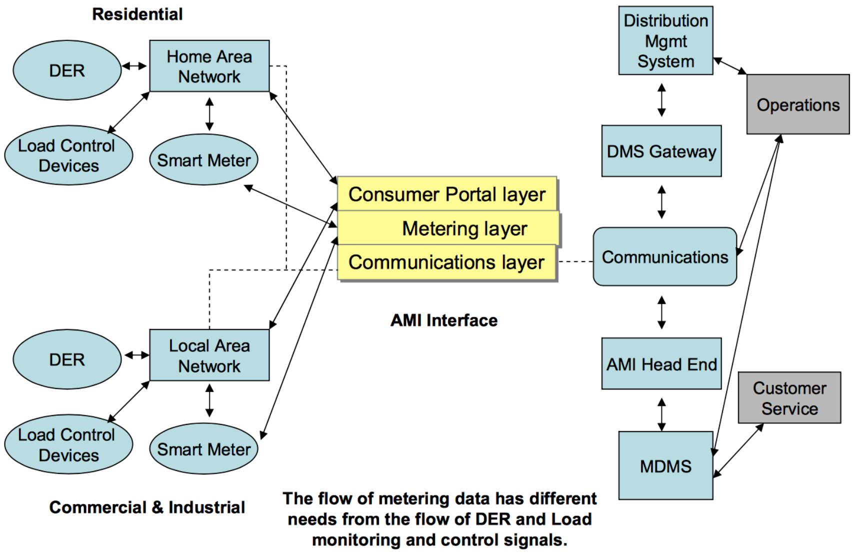

Figure 1.1: An illustration of AMI [8].

system of smart meters, communications networks, and data management systems that enables two-way communication between utilities and customers [8]. AMI applications gather energy usage information in near real-time. The purpose of AMI can be remote meter reading for error free data, network problem identification, load profiling, energy audit and partial load curtailment in place of load shedding. At the consumer level, smart meters communicate consumption data to both the user and the service provider.

- **Distribution automation.** Distribution automation (DA) optimizes utilitys operations and improves the reliability of its distribution power system. DA uses sensors and switches with advanced control and communication technologies to automate feeder switching; voltage and equipment health monitoring; outage, voltage, and reactive power management. Automation can improve the speed, cost, and accuracy of these key distribution functions to deliver reliability improvements and cost savings to customers [9, 10].

- **Process automation.** PA (Process automation) is a kind of optimized factory. The purposes of PA are continuous monitoring and controlling devices including the main hardware elements such as various field sensing devices and process controllers. According to the monitoring results, a central host handles equipments and instruments in a factory [11, 12].

- **Factory automation.** Factory automation is an industrial automation. Industrial automation

Figure 1.2: An illustration of a Distribution Automation System [10].

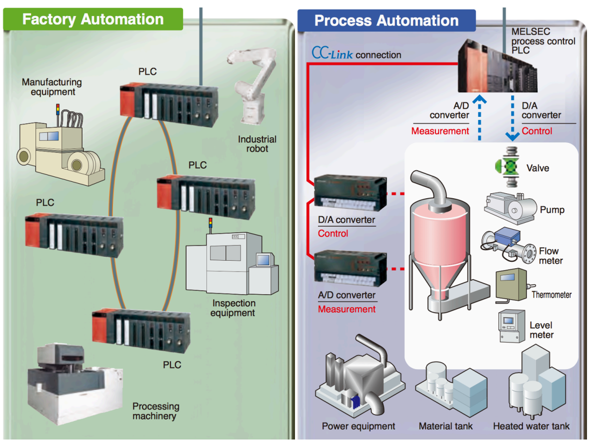

Figure 1.3: An illustration of FA and PA [11].

is the complete automation of the entire process done an industry where there is least human intervention in process, apart from that the only job a human does in monitoring and controlling the whole automation. As compared to PA, faster response time is required. [11]

- **Predictive Maintenance.** Predictive maintenance (PM) is a maintenance strategy driven by predictive analytics. PM focuses on predicting when device failure will occur and preventing that occurrence of failure with the help of maintenance monitoring so that maintenance can be planned before an issue manifests. PM can help in preventing expensive failures from occurring. When expensive failures are happen, the system loses production hours until parts and supplies are fixed. Predictive maintenance can minimize issues with reliability or quality [13].

In industrial fields, industrial control systems, such as SCADA (Supervisory Control And Data Acquisition), refer to the centralized systems that control and monitor the entire sites as shown in Figure 1.4. In DAs, a SCADA system performs operations like bus voltage control, bus load balancing, circulating current control, overload control, transformer fault protection, and bus fault protection [14]. All the control actions are automatically performed by an central device, such as a control server or programmable logic controllers (PLCs). As shown in Figure 1.5, first message is sent out to an end device (sensor) from a central device (PLC) and a central device waits for a reply. A response is sent back to the central device within a deadline specified in the request. Such a mechanism exchanging a request and a response is called "polling". In IIoT systems, polling is always initiated by a central device to obtain data and frequency, interval, or timing of polling is controlled by a central device. The process of polling can be periodically happen every couple of seconds, minutes, hours, days, months, or years. It is dependent on the system and applications.

In a typical AMI, DLMS/COSEM (Device Language Message Specification/Companion Specification for Energy Metering) [15] is used at the application layer. It is responsible for polling smart meters connected to the network and for sending the retrieved data to the management system from smart meters.

Both AMI and DA use polling schemes to exchange messages between a central device and end devices. In addition, a real-time system such as DA or PA reacts or responds to the applications within a fixed amount of time in order to avoid system failures. Both processing and reacting should be done within a pre-determined deadline that includes communication latency. Traditional PA systems use data collection protocols like Modbus that were originally designed for use with low-bandwidth wired communications such as serial communications and tolerant of long communication latency [16]. Traditional dedicated wired communication protocols for industrial applications have been well

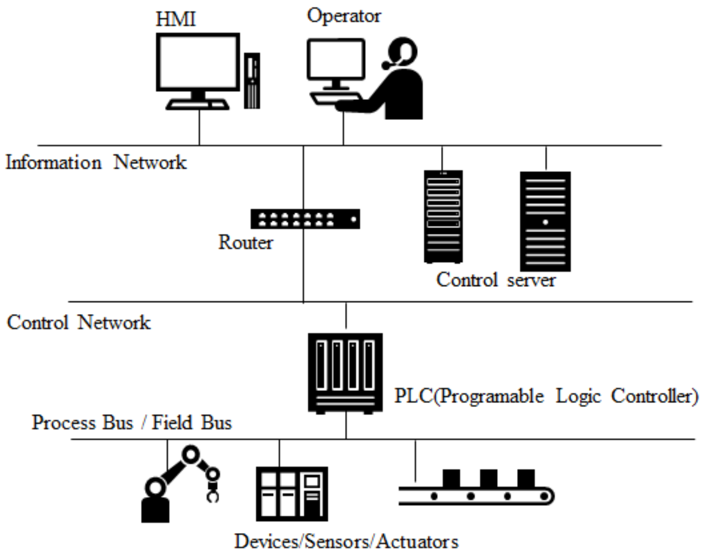

Figure 1.4: An overview of control system architecture

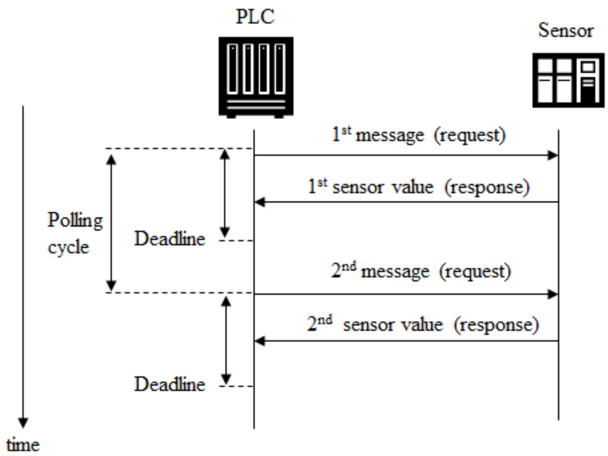

Figure 1.5: An example of data collection sequence by a polling scheme

designed to meet requirements regarding the deadline. Despite the high reliability of such wired communication protocols have been proven over many years, wired communication are expensive, time consuming, and difficult to install. Thus, wireless communication is fundamentally required for IIoT.

1.1.2 Wireless network technologies for IoT and IIoT

When we look at the IoT applications, it is clear that the only simple way how to connect a number of devices to the Internet is using wireless technologies. Consequently, one of the main IoT technologies is how to build IoT networks. There are a lot of different wireless network protocols for IoT like ZigBee [17], Z-wave [18], Bluetooth Low Energy [19], LoRa [20], Sigfox [21], etc. A brief overview of these related IoT wireless network protocols are presented as follows;

- **ZigBee.** ZigBee is an open, global standard for low-power Wireless Personal Area Networks. ZigBee uses the IEEE 802.15.4 standard for its PHY and MAC(CSMA/CA), and ZigBee Network, Application Support, Application Framework, ZigBee Device Object, and Security Service Provider are defined over the specification. ZigBee operates in the Industrial, Scientific and Medical radio bands and the exact frequency will depend where devices are in the world. It can use the 868 MHz band in much of Europe, 915 MHz in the USA and 2.4 GHz in many other locations. Within a ZigBee network architecture, there are 3 device types as shown in Figure 1.6: The ZigBee Coordinator (Coordinator), ZigBee Router (Rourter), and ZigBee End Device (End Device). Star, tree, and mesh topologies are supported. The mesh networking is to reduce the need of infrastructure and to increase coverage of the network.

- **Z-Wave.** Z-Wave is a low-power, IoT wireless technology, primarily designed for home automation. It is originally developed by Zensys and Z-Waves PHY and MAC(CSMA/CA) have been ratified by the ITU-T G.9959 standard. The Z-Wave Alliance and hundreds of international companies that use Z-Wave technology in their products. Z-Wave operates on a single channel in the <1GHz band (868MHz band for Europe and 915MHz band for North America and Australia) and offers reliable and low-latency communication with data rates up to 100kbit/s. Z-wave also supported mesh network topologies as well as ZigBee.

- **Bluetooth Low Energy.** Bluetooth Low Energy (BLE) is an wireless technology developed by the Bluetooth Special Interest Group for short range communication. BLE operates in the 2.4 GHz ISM band and defines 40 Radio Frequency channels with 2 MHz channel spacing.

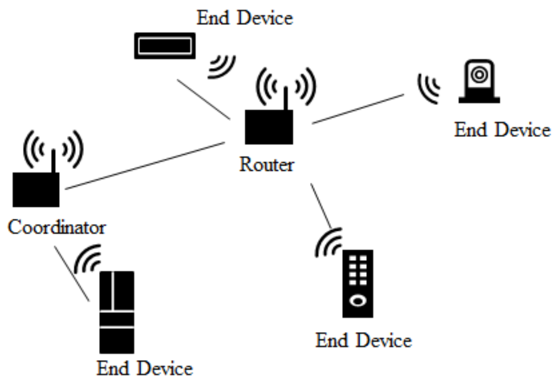

Figure 1.6: ZigBee Network Architecture

BLE has been designed as a low energy consumption for control and monitoring applications such as healthcare, smart home, smart energy, and security. As compared with Z-wave and ZigBee, BLE supports a single hop solution and provides rapid simple paring functions.Two device types, such as the master and the slave, are defined. In order to save energy, slaves are in sleep mode by default and wake up periodically to listen for receiving packets from the master. The master determines and coordinates communication timing by using a Time Division Multiple Access (TDMA) scheme. In addition, in order to avoid interference and wireless propagation issues, such as multi-path and fading, BLE uses an adaptive frequency hopping mechanism for data transmission.

- **LoRa.** LoRa, which stands for Long Range, is a long-range wireless communications system, promoted by the LoRa Alliance. It is designed to optimize LPWA networks for battery lifetime, capacity, range, and cost. A typical LoRa network is a star-of-stars topology, which includes three different types of devices, as shown in Figure 1.7.The LoRa physical layer operates in the unlicensed <1GHz frequency ISM band and uses spread spectrum technology so that transmitters are less likely to interfere with each other. Spread spectrum technology also provides a coding gain over narrow band communications. This results in strong and long communication.

- **Sigfox.** Sigfox is a one of LPWA networks, developed and delivered by the company Sigfox. It operates on the <1GHz frequency band, and uses low data rate transmission and sophisticated signal processing to avoid interference. Sigfox supports bidirectional communication. An end device transmits a message to base station(s) (uplink communication) and then the end

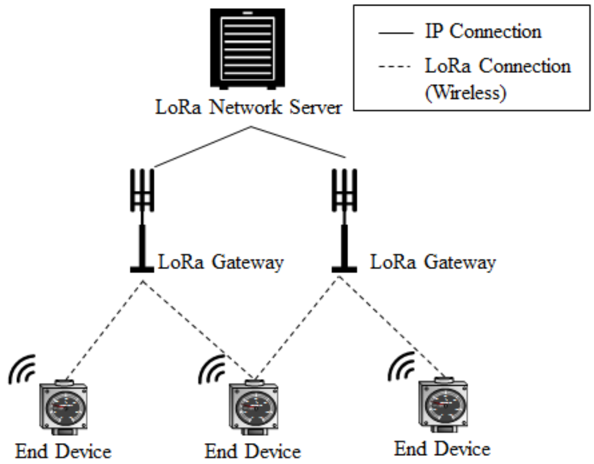

Figure 1.7: LoRa Network Architecture

device listens for a short duration in case there are data which a network server sent to the end device(downlink communication). The amount of bidirectional communication a day is limited: Each end device can transmit 140 message with a payload size of 12 octets to a network server and a network server can transmits 4 downlink message per device a day. In addition, both uplink and downlink communication are always initiated by a device. This results in low power and long range communication. However, it is not effective for communications from a network server to end devices. Sigfox would be ideal for one way monitoring applications with sensors.

From the coverage area perspective, we can roughly categorize the wireless networks into two groups. First one is shot range such as ZigBee, Z-wave, and BLE. Another is long range such as LoRa and Sigfox. Although ZigBee and Z-wave has multi hop routing functions to expand coverage, when IoT devices are located in wide/large area, long range groups should be better than short range ones.

For such long range connectivity, cellular technologies like 2G, 3G and 4G has been widely used in Machine-to-Machine systems. However, cellular networks are originally designed for voice and low-latency communications, which are not typical requirements of most IoT applications. From communication range with very low power operation perspective, LPWA networks are ideal for the IoT applications. LPWA networks technologies use very low data rates, e.g., $100 - 200[bps]$ to achieve the long range connectivity. The lower the data rate is, the longer it takes to transmit a packet. Of course, there are trade offs. It is not ideal for IoT applications that require guaranteed quality of service, or require low latency. Moreover, as we mentioned above, LPWA is not suitable for IoT

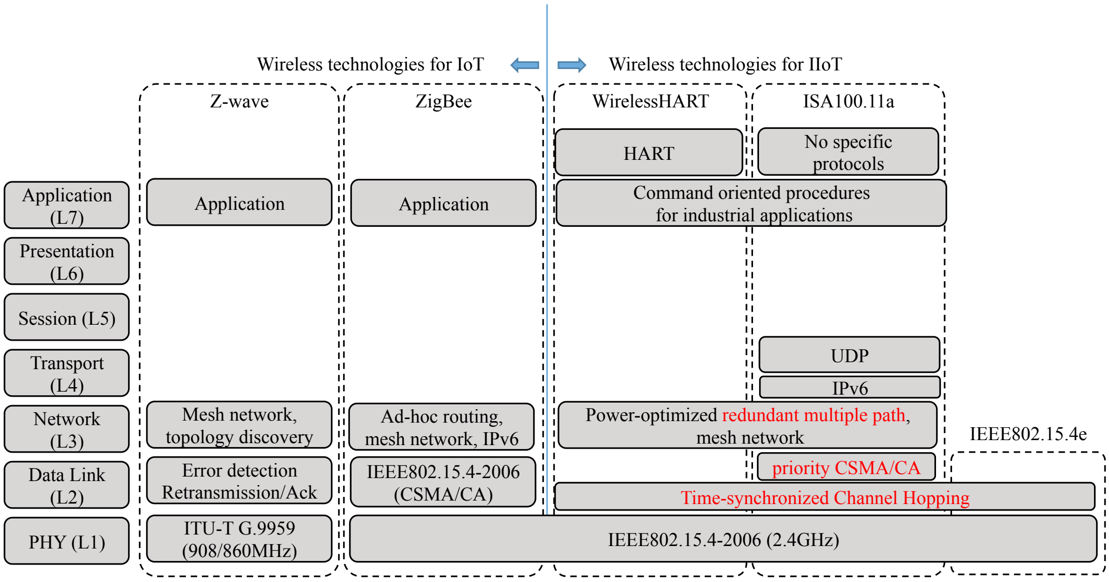

Figure 1.8: An overview of protocol stack of IoT and IIoT network technologies

applications that require bidirectional communications.

Short range groups are useful to applications, which monitor environments and control devices, e.g., smart home, smart city, smart health care. The applications require low power short/middle range networks and bidirectional and decentralized communications between devices. In addition, it is also important to guarantee interoperability among IoT devices which are provided by different companies. Then, many solutions have been proposed using short range open standard networks with multi-hop routing. IoT network technologies should be selected by characteristics of the protocols to meet requirements of IoT applications.

IIoT applications basically gather information from remote devices i.e. sensors, check device status or circumstance, and control devices, i.e. actuators, based on the gathered information. For remote monitoring devices, the main purpose is periodic collection of device status or sensor data. At the same time, industrial applications also require on-demand communication for data collection and operation of devices by a remote control server, within specific end-to-end deadlines. For instance, an AMI system requires a deadline of 20 – 60 seconds when a remote control server requests on-demand meter reading. Although such request/reply type of communication is unpredictable, IIoT network technologies must guarantee a maximum communication delay for both of periodic and unpredictable packets. Although IoT and IIoT have many technologies in common, including

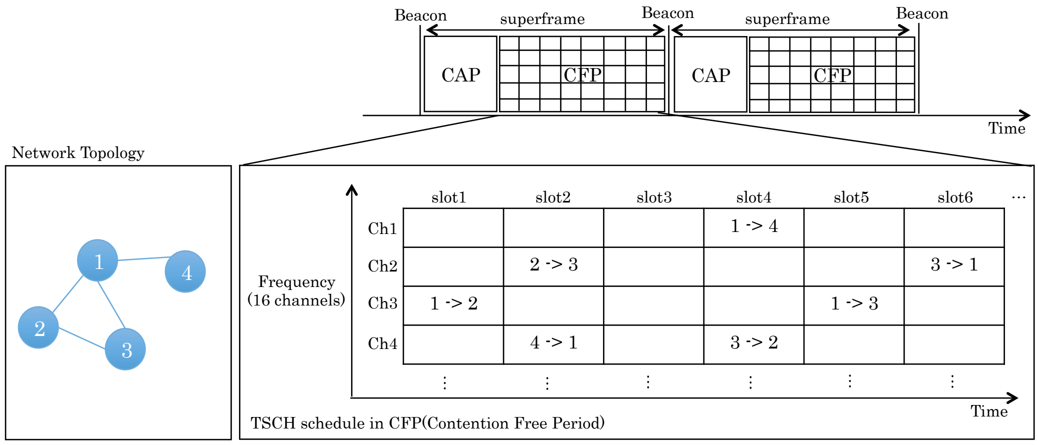

Figure 1.9: An example of TSCH schedule

cloud platform and wireless communications, IIoT, which is a subset of the large IoT, focuses on the specialized requirements of industrial applications.

The Industrial applications need connectivity for devices and Industrial Wireless Sensor Networks (IWSNs) [22,23] have been emerging as a new means of wireless communications for the IIoT. Recently, several protocols for IWSNs, such as WirelessHART [24], ISA100.11a [25], and IEEE802.15.4e [26], have been developed and standardized in order to increase the communication reliability in a wireless network with frequent packet loss and big latency under the heavy network utilization as shown in Figure 1.8. Wireless technologies for IoT, such as Z-wave and ZigBee often operate on a single channel and CSMA/CA to resolve packet collisions on the channel. On the other hand, in the standards for IIoT, a central network coordinator defines a cycle of repeated superframes that have a Contention Access Period (CAP) and a Contention Free Period (CFP). In a CFP, these standards use Time-Synchronized Channel Hopping (TSCH) mechanism that enables reliable and low power wireless networks [27]. TSCH is different in that a communication schedule orchestrates all communication in the wireless network. The schedule indicates when each wireless device transmits, listens, or sleeps and which channel each node uses for communication among neighbors. Both application data packets and network control packets in a wireless network are transmitted according to the assignment, as shown in Figure 1.9. This mechanism allows the standards to decrease packet loss due to collisions in a large scale network. WirelessHART, ISA100.11a and IEEE802.15.4e use the TSCH mechanism as an access control method.

In addition, WirelessHART and ISA100.11a have reliable multi hop routing protocols that provide multiple paths between devices. In Figure 1.9, if communication from device 3 to device 1 fails on

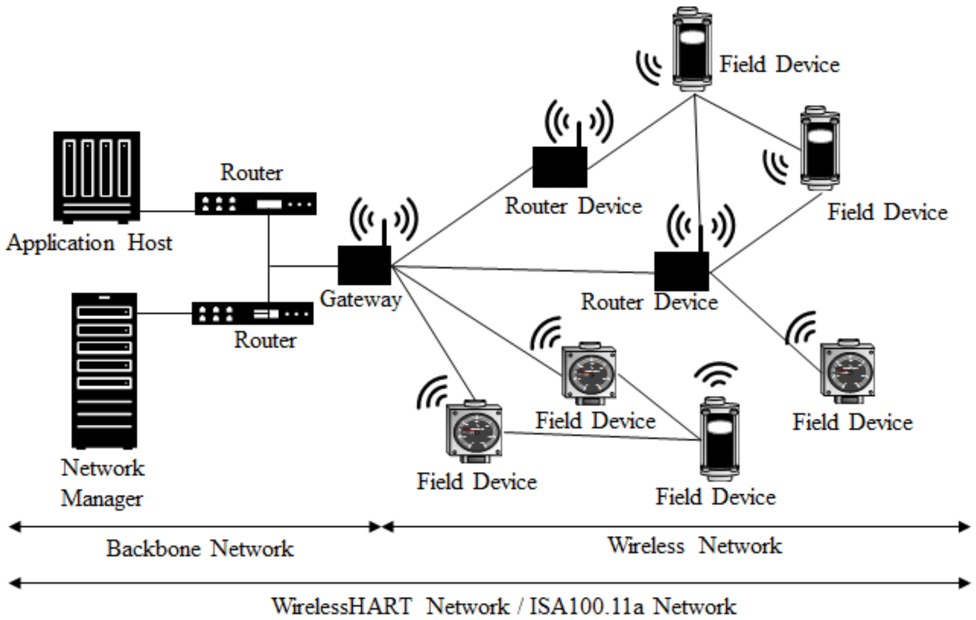

Figure 1.10: An overview of WirelessHART/ISA100.11a Network

path 3-1, the device will retry on a different path, for example, 3-2-1. Providing redundant multiple paths allows messages to be be routed around physical obstacles, broken links and interference. These redundancy mechanism, such as multiple radio channels (frequency diversity), multiple timing possibilities (time diversity), and multiple paths among devices (spatial diversity), increase reliability.

WirelessHART and ISA100.11a support a centralized network management architecture as shown in Figure 1.10. The network manager is the centralized "brain" of a network. Its responsibility is to manage everything related to a wireless network, e.g., scheduling, network path configuration, etc. Below elements of the standards are briefly explained:

- **Application Host.** This is the machine that application programs, such as monitoring or controlling devices, are running on

- **Network Manager.** A network manager is a centralized entity responsible for configuring and scheduling the wireless network. It accepts joining requests to the wireless network from field devices via a gateway device. A network manager that is connected to the gateway is responsible for creating schedule and routes for the entire wireless network. The schedule and routes are centrally determined to meet requirements of reliability, bandwidth, delay etc, by the network manager. Then, the network manager distributes the schedules among the devices by providing only the slots to each individual device for which the device has to transmit or receive.

- **Security Manager.** A security manager is responsible for creating and storing keys of authentication and cryptography used in the wireless network. The network manager uses the security manager for key management.
- **Gateway Device** A gateway device connects the wireless network such as WirelessHART network or ISA100.11a network to an IIoT application system. It provides the application host with connectivity to the field devices.
- **Field Devices.** Field devices are sensors or actuators. All field devices are time-synchronized and assigned communication timings by a central network manager. It decreases both interference within the wireless network and interference from other wireless networks using the same channels, thus increasing the rate of successful communication.

A brief summary of WirelessHART and ISA100.11a are presented as follows [28];

- **WirelessHART.** WirelessHART is the first open wireless standard protocol for the process control. The protocol supports operation in the 2.4GHz ISM band using IEEE802.15.4 standard radio. In WirelessHART, communication are precisely scheduled based on TSCH. Scheduling is performed by a centralized network manager that uses overall network routing information. In addition, a network manager provides multi-hop multiple paths between all devices and the central node. The application layer is HART that is global standard protocol for sending and receiving digital data across analog wires between field devices and control/monitoring system.
- **ISA100.11a.** ISA100.11a was developed through the International Society of Automation. The network architecture of the standard is similar to wirelessHART, but the network and transport layers are based on IPv6 and UDP standards, while wirelessHART supports proprietary network and transport layers. Regarding application layer, ISA100.11a does not define a process automation protocol. It only specifies tools for constructing an interface for industrial applications. ISA100.11a is more flexible than wirelessHART.

From operating IIoT systems perspective, these centralized wireless network standards have higher affinities than wireless network technologies for IoT. The network manager can be integrated into the gateway, host application or a controller in industrial control systems. Moreover, the standards for IIoT overcome the problem of packet collisions in the network. In fact, the data collection ratio for WirelessHART reaches more than 99%, because SmartMesh WirelessHART devices basically perform retransmission twice at most [29]. This performance seems sufficiently high for remote monitoring purposes. The data collection ratio is normally one of the most important parameters for evaluating reliability of a wireless network.

On the other hand, when an unpredictable on-demand packet has to be sent to or from an end device, the packet either consumes assigned bandwidth or waits several seconds until the next assigned bandwidth comes available. This can cause random latency, which further depends on TDMA scheduling, retransmission timing, and wireless radio conditions. it is difficult for even TDMA-based MAC protocols to support real time communication for large scale networks like AMI [30]. Guaranteed deadlines should be considered for IIoT networks, because industrial applications require real time processing.

1.2 Challenges

As described in Section 1.1, most traditional industrial applications use polling schemes to exchange messages between a central host application/device and end devices. Moreover, industrial applications and social infrastructure applications have an intense need for standards to guarantee interoperability between different vendors [31, 32]. Then, designing a new polling-based communication scheme which is stable for IIoT applications and compatible with standard IWSNs an important research challenge.

In this thesis, we address two problems specific to the communication schemes: There are two types of IIoT application data traffic which an application host generates. First one is scheduled periodic data collection from end devices. Industrial applications basically gather heterogeneous periodic information from end devices in the IIoT system. Then, polling requests, which are generated at different cycles, are transmitted in an IWSN. Another is unpredictable on-demand communication to get additional sensor data from end devices or control devices remotely from a application host.

The first issue is how to manage transmission timing of polling requests for all the application data traffic. When polling-based protocols for industrial applications are used over IWSNs, the number of retransmissions of polling queries should be also considered in order to maintain a high reliability of industrial systems. At the same time, their timing of the retransmission should be carefully considered in order to keep the deadline. Moreover, when the unpredictable on-demand data communication occurs, the unpredictable traffic may be given high priority to over periodical scheduled communication and a central device transmits a polling packet for the on-demand request.

Consequently, a periodical polling request from a central node to a node is dropped by the unpredictable on-demand data communication. In polling-based communication over current standard IWSNs, such as ISA100.11a and WirelessHART have difficulty in realizing both periodic data collection at high success ratio and unpredictable on-demand communication with short latency.

Regarding the first issue, to achieve both heterogeneous periodic data collection with high success ratio and unpredictable on-demand communication within a deadline over IWSNs, in Chapter 2, we propose a data traffic control scheme for polling-based communication in IWSNs and a scheduler which uniformly distributes network load over slots. Our proposed scheme incorporates multiple heterogeneous periodic data collection schedules in a single schedule. By uniformly distributing network traffic including periodic polling requests and their retries, all devices can have fair opportunities to communicate with a central device leading to high success ratio of communication. Furthermore, such uniform load distribution leave uniformly distributed time slots available to unpredictable on-demand communication. Once the need for a central device to obtain data from an end device arises, it can easily and promptly find an empty time slot and send a request without interruption of periodic data collection.

In the proposed scheme, a central device controls the transmission timing of all polling-based communication in accordance with a schedule that is determined by a Genetic Algorithm. Our proposed scheme incorporates multiple heterogeneous periodic data collection schedules in a single schedule. In the single schedule, communication of both periodic and unpredictable on-demand data collection are uniformly assigned. Simulation results show that network traffic is generated uniformly and a center node can collect periodic data from nodes at high success ratio. The average success probability of periodical data collection is 97.4% and the lowest probability is 95.2%.

The second issue is how to control several kinds of packets, such as, on-demand request/response packets, data collection request/response packets, and network control packets. Although the standards assign communication timing of both application and network control, it is difficult to guarantee the latency of on-demand and multi-hop communication at any time. Normally, on-demand requests/replies packets are defined as higher priority packets than regular periodical data collection packets. Because ISA100.11a is more flexible to improve network performance than WirelssHART and ISA 100.11a can transfer a higher-priority packet by applying the priority CSMA/CA scheme among single-hop neighbors, in this thesis we focus on ISA100.11a as a standard IWSN. A priority CSMA/CA scheme enables a priority control within single-hop communication and decreases the probability of collision among transmission of packets of different priority [33]. When the higher priority packet is forwarded on the multi-hop route, the forwarding node may transmit that to a neighbor after waiting several seconds until the next assigned bandwidth comes available. This causes random latency. Moreover, for network configuration perspective, it is also important to exchange network control information among devices in a wireless network.

To solve the second issue, we propose a priority-based dynamic multichannel transmission scheme for IIoT networks in Chapter 3. The scheme prioritizes packets in accordance with application requirements. Packet transmission is scheduled in a slotted manner, but detailed slot allocation as in usual TDMA-based protocols is not performed. More specifically, a central host only determines when it transmits on-demand request packets for remote control and when devices transmit data packets for periodic monitoring. On the other hand, packet forwarding is not scheduled at all. In a slot, which we call SlotFrame, priority CSMA/CA-like packet transmission is performed at each device. The scheme operates over a MAC layer and does not rely on any specific MAC protocol. We also discuss compatibility with ISA100.11a in this thesis.

We consider one type of packets for periodic communication and three types of packets for unpredictable communication. We first define three priorities of packets depending on their type and then assign one dedicated channel to each priority. The highest priority is given to unpredictable and on-demand packets that a central host sends to a device for control or information retrieval, that is, downward packets. Upward reply packets are also given the highest priority, because request/response communication for device control requires a very short end-to-end delay. The second priority is given to periodic packets used for regular data collection. Network control packets are then set to the lowest priority. In the proposal, a central host can thus transmit the highest-priority packets at any time but still control the transmission timing through centralized administrative control, as in ISA100.11a. More specifically, a device replies to a request packet at the time specified by a central host.

On the other hand, each devices decides the time of forwarding a packet by autonomous decentralized radio channel control. A device scans three communication channels in descending order of priority and dynamically decides which channel to use. For example, when a device having a reply packet finds that a request packet is to be sent by a neighbor, it defers transmission of the packet for a certain duration of time to avoid collision among downward and upward packets over the high priority channel. If there is no transmission of high-priority packets in the vicinity, a device can transmit a periodic data packet using another channel for the middle priority. Only when there is no packet transmission on either of high and middle priority channel, a node can transmit a network control packet. In this thesis, simulation evaluation shows lower priority packets belonging to periodic data gathering and control can receive the satisfactory quality of service, where the collection ratio of periodic data packets is higher than 45% and the lower bound of bandwidth available to control packets is larger than 36% at the worst case scenario.

In this thesis, we point out issues that standard IWSNs are not suitable for Industrial applications, which use polling-based communication schemes, such as SCADA, AMI, etc. To solve the issues, we propose poling-based communication schemes for IIoT applications.

1.3 Outline of this Thesis

In this thesis, we propose the new poling-based communication schemes over standard IWSNs for IIoT applications, which require both periodic data collection with high success ratio and unpredictable on-demand communication within short guaranteed delay as shown in Figure 1.11.

The first proposal is a simple central control schemes and it does not depend on any specific IWSN protocols at all. This schemes lies between applications and an interface of IWSNs. Some Industrial applications require similar requirements for IWSNs, but applications in different domains may use different IWSN protocols, which hinders interoperability of various IIoT systems. For example, AMI often uses IEEE802.15.4e and SCADA uses WirelessHART or ISA100.11a. When a company provides several different IIoT systems, such independent schemes are desirable. In addition, when an IIoT system uses sensor devices, which are provided by multiple vendors, the IIoT system should use a standard IWSN as it is. Our first proposal contributes to these situations.

Some industrial applications, such as building automation and factory automation, require shorter delay and more reliable communication than other IIoT applications whose main purpose is monitoring the IIoT systems. To improve performance of polling-based communication over IWSNs, our second proposal is to be introduced in an IWSN mechanism. Thus, the goal of the second proposal is accomplished by incorporating central control by a central device and autonomous decentralized control by end devices in an IWSN. Although we propose additional functions which enable a standard IWSN (ISA100.11a) to use for IIoT applications, we show how to adapt the proposal to ISA100.11a. in this thesis.

The rest of the thesis is organized as follows. Chapter 2 explains a polling-based transmission scheme using a network traffic uniformity metric. Chapter 3 describes a priority-based dynamic multichannel transmission scheme. Finally, Chapter 4 concludes the thesis.

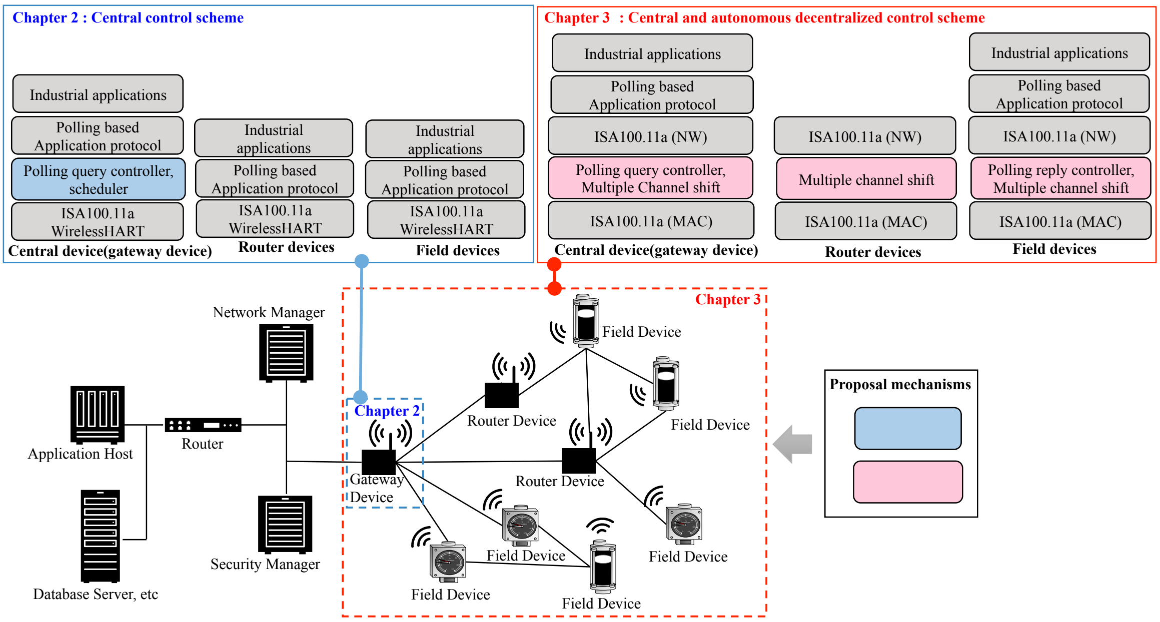

Figure 1.11: Configuration of this thesis. First, in Chapter 2, in order to realize both heterogeneous periodic data collection at high success ratio and unpredictable on-demand communication with short latency, we propose a central control scheme which can generate polling queries uniformly in a wireless network. To improve performance of polling-based communication over an IWSN, then in Chapter 3, we propose communication with dynamic multichannel transmission scheme which is accomplished by incorporating central control by a central device and autonomous decentralized control by end devices in an IWSN.

Chapter 2

A Polling-Based Transmission Scheme Using a Network Traffic Uniformity Metric

2.1 Introduction

IIoT applications, such as AMI, DA, PA, FA, and PM, gather sensor data from end devices in IIoT systems for increasing optimization, efficiency or productivity of the IIoT systems, as we described in Chapter 1. Some IIoT applications, such as DA and PA, have already continuously monitored whole systems and controlled devices via SCADA. In SCADA, the measurements are made by end devices and the information is transferred to a central device such as, an application host or a central server.

The IIoT systems adopt the centralized system and use a polling scheme for exchanging massages between a central device and a end device over traditional dedicated wired communications. Industrial applications basically gather heterogeneous periodic information from field devices via a polling communication. The transmission frequency of periodic information collection is normally static. At the same time, they require unpredictable communication for gathering on-demand data or operating devices by a center device within specific end-to-end deadlines [34–36]. In general, the industrial communication protocols over wired communications take into consideration maintaining the deadline even if the unpredictable communication occurs in the network. For frequent data collection in DA, the wired network load can be as much as ten times less than upper bound [37].

The IIoT applications over wireless networks, such as IWSNs, also adopt polling schemes. For

Table 2.1: IIoT application and system requirements in Chapter 2.

	920-MHz Band	2.4-GHz Band
Applications	- AMI and DA	- PA and PM
Application processing	- Remote Monitoring - Remote Operation	- Remote Monitoring
Communication type	- Publish/Subscribe - Request/Response	- Publish/Subscribe - Request/Response
Remote monitoring cycle	30 min (periodic data)	20 min@100 nodes (periodic data)
Maximum delay for remote operation	20 – 60 sec (unpredictable data)	20 – 60 sec (unpredictable data)
Number of nodes in wireless network	1 – 500 nodes	1 – 100 nodes
Packet length	500 – 600 B	90 B
Communication speed	50-100 kbps	250 kbps
Protocols	- IEEE802.15.4e	- ISA100.11a - WirelessHART

example, AMI also uses DLMS/COSEM as its application protocol, which is responsible for polling smart meters to collect data from a central server. Most standard IWSN protocols are suitable for gathering sensor data system, which end devices autonomously push sensing data to a central server, because bandwidth of standard IWSNs for upstream is assigned much more than for downstream in order to increase data collection at high success ratio. Therefore, when polling-based protocols for industrial applications are used over IWSNs, the number of retransmissions of polling queries over IWSN layer should be considered in order to maintain a high reliability of industrial systems. At the same time, their timing of the retransmission of the polling query should be carefully considered in order to keep the deadline. Table 2.1 summarizes IIoT applications and system requirements of this chapter [9, 12, 15, 32, 38].

Furthermore, as we discussed in Chapter 1, IIoT systems are often build by a number of devices, which are provided by multiple vendors. In this case, the IIoT system should use a standard IWSN as it is, because we have no room to modify the standard IWSN or replace with a new IWSN. Therefore, a scheme to satisfy the above-mentioned requirements must conform to the standard.

In this chapter, to achieve both heterogeneous periodic data collection with high success ratio and unpredictable on-demand communication within a deadline over standard IWSNs, we propose a data traffic control scheme for polling-based communication in IWSNs and a scheduler which uniformly distributes network load over slots .We adopt a centralized control system and the data traffic control scheme lies over IWSNs. The proposed scheme gives a central device opportunities of polling request transmission that are scheduled in a slotted manner. Transmission timing for all of the packet is in accordance with application requirements of data collection intervals and network qualities.

We consider three kinds of slot types to decide a schedule. The first slot type is transmitting a polling query which is generated by a central node to get application data from an end device. The second one is transmitting its retry packet in case of wireless communication failure or an unpredictable on-demand request interruption. The final one is letting some slots open for unpredictable on-demand request. Ideally, for a polling based communication scheme, these three slot types should be equally assigned in a schedule. It will be complicated to find an optimal schedule to meet the requirement when many end devices belong to an IWSN and a central device transmits polling requests to them at various cycles as described in Section 2.4.

In addition, we adopt a GA (genetic algorithm) as a heuristic to derive an optimal schedule [39]. GA is one of probabilistic search algorithms and optimization techniques based on the mechanisms of natural selection and evolution. In this thesis, we consider a GA-based algorithm, which could achieve a reasonable and feasible schedule with practically short computation time on an off-the-shelf computer.

The contribution is to propose a data traffic control scheme for polling-based communications and verify its performance from viewpoints of end-to-end communication success probability and balanced slot utilization. None of the conventional technologies for IWSNs focus on the problem that an unbalanced network bandwidth between uplink and downlink causes unexpected big latency of unpredictable on-demand communication or decrease of success probability for periodic data collection. Standard IWSN protocols like ISA100.11a typically have a scheduler for allocating network resources such as timeslots to all nodes. Since a network manager of IWSN protocols gathers information about network condition from all nodes, IWSN protocols normally require a large available bandwidth for upward traffic. In addition, the scheduler has to deliver the information to all nodes whenever a new node joins the network or a network topology changes. Therefore, it is difficult to support a polling-based unpredictable on-demand communication within deadline. In contrast, since our scheme only determines when to generate and transmit a packet at a root node for periodic data collection and unpredictable on-demand data collection. It does not need to adjust the schedule because the number of nodes does not change.

The rest of the chapter is structured as follows. We first describe assumptions regarding our proposal and challenges in Section 2.2, and Section 2.3 presents an overview of related work. In

Section 2.4 we propose the data traffic control scheme. Then, we evaluate uniformity and success ratio of collecting periodic data in Section 2.7. In Section 2.8, we discuss possibility of providing schedules when an application data traffic is high. Finally, we summarize this chapter and list the future work in Section 2.9.

2.2 Assumptions of Our Target System and Challenge

In this section, we provide assumptions of our proposed scheme first, and describe challenges of this thesis.

2.2.1 Assumptions

- **IIoT Application features.** As noted above, our target IoT applications are AMI, DA, PA, and so on. These typical industrial applications normally collect field data from end devices periodically within pre-determined deadlines that include communication latency and internal processing time on both a central node and an end device. In this thesis, we assume that the end devices periodically generate data in intervals, and a deadline that the central node should get data from an end device is equal to the next time that an end device generates data within the interval.

- **Polling-based communication.** Our target IoT applications normally use polling-based communication to keep a sequence of processes or simplify network management. In general, there are two different polling schedulings, called monocycle and multicycle polling scheduling.

- **Monocycle polling scheduling.** Monocycle polling scheduling is the most common and the simplest. In monocycle polling scheduling, a central node uses a single cycle for the polling required for all devices belong to a single group. Figure 2.1 shows an example of a monocycle polling scheduling. In this case, the deadline (cycle) is T_1 and three nodes ($n_{1,1}, n_{1,2}, n_{1,3}$) belong to the group.

- **Slot assignment for reliable polling communication.** A central node transmits the first queries at t_1, t_2, t_3 to the nodes respectively. Retransmission timings are also allocated in case of communication failure. For example, t_4 and t_7 are timing for retries of the first query to $n_{1,1}$. The number of the retries R_j for node j depends on communication success rate, for example

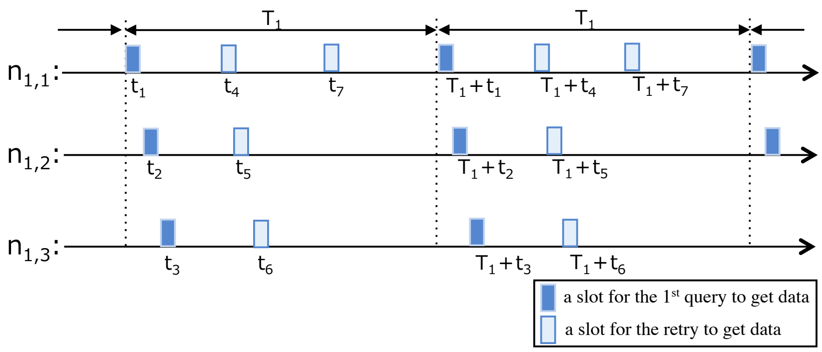

Figure 2.1: An example of polling schedule in a monocycle polling scheduling.

PER (Packet Error Rate) as shown in Equation (2.1);

$$1 - P_{PER_j}^{R_j} \geq Th_{success}, \tag{2.1}$$

where $P_{PER\,j}$ is an end-to-end PER between a central node and node j and $Th_{success}$ as one of system requirements is probability of collecting data from any nodes in a system.

- **Multicycle polling scheduling.** Multicycle polling scheduling is a set of heterogeneous monocycle polling schedulings. Figure 2.2 shows an example of multicycle polling patterns. In this example, three applications in a system have different cycles that are denoted by T_1, T_2, and T_3, respectively. The central node transmits the first queries and their retires in each cycle as well as a monocycle polling scheduling. In Figure 2.2, lower brightness areas represent some consecutive slots of first queries and higher ones do some consecutive slots of their retires. Besides, white areas stand for open slots for unpredictable on-demand request. The width of the areas depends on the schedules. For example, the first lower brightness area of G_1 includes 3 slots and the second one includes 5 slots.

- **Wireless communication for IWSNs.** To increase the reliability of wireless networks for industrial applications, Time Slotted Channel Hopping (TSCH) based IWSN protocols such as WirelessHART, ISA100.11a, and IEEE802.15.4e have been developed. They are time-synchronized and assigned communication timings for transmitting packets by a central network manager in order to decrease both interference within the wireless network and interference from other wireless networks using the same radio channels. The communication timing for both downlink traffic from a central node to end devices and uplink traffic from end devices to

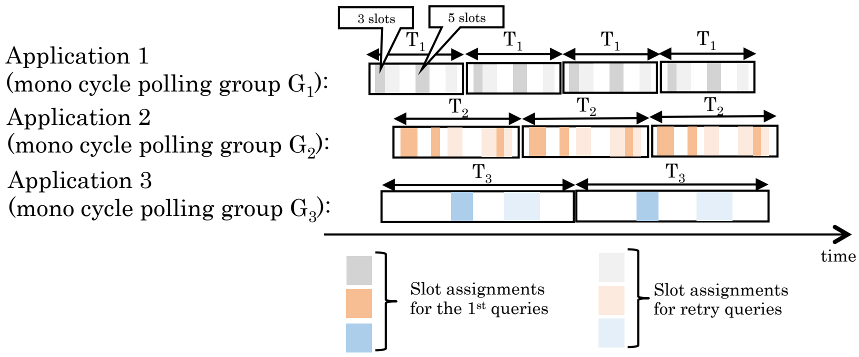

Figure 2.2: An example of multicycle polling patterns.

a central node are assigned. These bandwidths for downward and upward are normally fixed. In addition, the available bandwidth for upward traffic is bigger than that for downward.

- **Imbalanced bandwidth between upstream and downstream of IWSNs** Generally speaking, bandwidth of standard IWSNs for the downstream from a center node to end devices is quite less than that for upstream because main traffic for IWSNs is autonomously transmitting sensor data from devices. For example, SmartMesh IP [40] that is based on the 6LoWPAN and IEEE802.15.4e standards provides only one timeslot for downstream traffic from a center node to a device node in every 2 seconds by default settings, contrary to several timeslots for upstream from the nodes to a center node.

- **Traffic patterns.** Industrial applications basically gather heterogeneous periodic information from field devices. At the same time, they also require on-demand data communication for additional data collection and operation of devices by a remote server within specific end-to-end deadlines [34]. When the unpredictable on-demand data communication occurs, the unpredictable traffic may be given high priority to over periodical scheduled communication and a central node transmits a polling packet for the on-demand request. Consequently, a periodical polling request from a central node to a node is dropped by the unpredictable on-demand data communication. If an open slot is fortunately assigned when the on-demand request occurs, a central node can get data from a node by an on-demand request without any interruptions of periodic data collection. Therefore, it is better for IWSNs to generate network traffic load for periodical data collection uniformly and also keep available bandwidth for unpredictable traffic at any time.

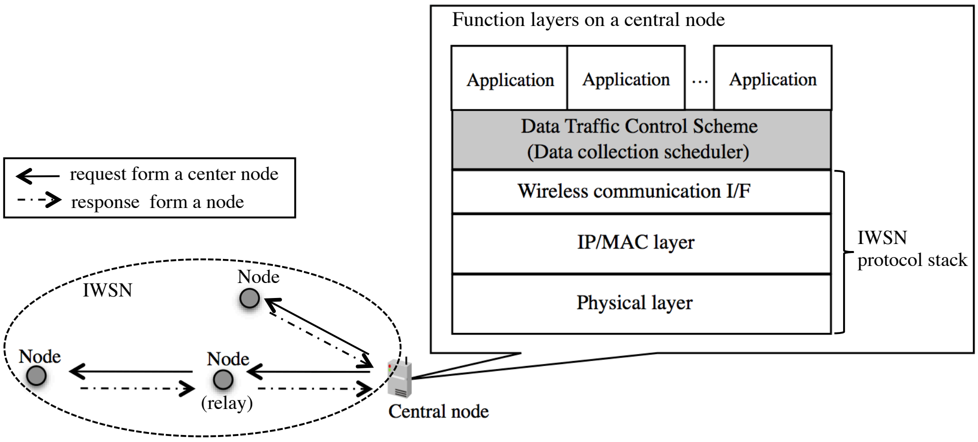

Figure 2.3: Position of our data traffic control scheme in function layers.

- **Traffic control scheme for multiple wireless communications.** Industrial applications require similar requirements for IWSNs, but applications in different domains may use different IWSN protocols. For example, AMI often uses IEEE802.15.4e and PA uses WirelessHART or ISA100.11a. We propose a data traffic control scheme for periodical data collection over IWSNs and our proposal does not completely depend on any specific protocols. Our proposed scheme is a technology that lies between applications and an interface of IWSNs, namely IP (network layer) or MAC (datalink layer) as shown in Figure 2.3. A center node has the data traffic control function in order to manage all polling traffic.

2.2.2 Challenge

As described above, it is important to realize both collecting multiple periodic data from sensors within deadlines and transmitting unpredictable on-demand packets. Transmitting opportunity for downstream of IWSNs such as ISA100.11a or WirelessHART, is normally lower than that for upstream, even though industrial applications use polling-based protocols that need equal opportunities for downward packets and upward packets. Therefore, one of our challenge is to provide a communication method that controls both downward and upward network load in IWSNs. The communication scheme should also ensure sufficient responsibility of unpredictable requests and stability of periodic data collection. The second challenge of this thesis is to provide a schedule which a central node normally generates uniform network load for periodic data collection by a polling-based communication protocol, in order to enable applications to transmit unpredictable packets effectively using the idle time of the IWSNs.

2.3 Related Work

In communication systems and wireless sensor network systems, many studies have been proposed to solve scheduling problems [41–46].

In a FieldBus environment, scheduling problems of dynamic or static collecting information flow are to be found in [41, 42]. Algorithms capable of solving this problem have to be able to calculate the transmission sequence for all groups. The solution is firstly to decide the primary cycle among multiple cycles. The primary cycle is the shortest period. Then, if there are two polling tasks of different groups at the same time, the group whose polling cycle is T_i always has higher priority than group whose polling cycle is T_j, where $T_i \le T_j$. In the papers, authors assume that deadline coincides with the polling cycle, and scheduling problem is to ensure that at least one transmission for each group will occur at least once in the cycle. They do not address features of wireless communication.

In [43], the authors propose TDMA link scheduling algorithms for the purpose of maximizing network throughput. In [44], an end-to-end real-time transmission scheduling over the wirelessHART networks is proposed. Both [43, 44] are focus on making conflict-free link level schedules. According to these schedules, end devices can autonomously transmit periodical sensor data to a center node within a deadline according to the schedule but this communication is not polling-based.

In [45], the authors show through analysis and experiments that conflict-free query scheduling has an inherent tradeoff between network throughput and latency. They propose real-time scheduling algorithms for prioritized conflict-free transmission scheduling in order to aim to balance network throughput and latency.

In [46], a new priority-based parallel schedule polling MAC protocol in WSNs combines polling orders with access policies to realize the priority-based scheme and reduce the overhead time through parallel schedule. Sensor nodes can be classified in different clusters and this MAC protocol coordinates all clusters and gives a chance to send data in order. To do this, the node to which a central node wants to transmit a priority packet is given the transmission chance first, and other nodes remain in a sleeping status to save energy.

As will be descried in Section 2.6, we adopt a GA (genetic algorithm) as a heuristic to derive an optimal schedule [39]. GA is one of probabilistic search algorithms and optimization techniques based on the mechanisms of natural selection and evolution. It has been applied to optimization of complex network management problems such as network load balancing management [47] and routing [48]. Although other heuristic algorithms such as simulated annealing, PSO (particle swarm optimization), or even machine learning can also be adopted, we consider a GA-based algorithm in

Table 2.2: Summary of related work in comparison with our work.

Purpose of Scheduling	**Work**	**Network Type**	**Communication Technique**	**Scheduling Type**	**Unbalanced Bandwidth**
Real-time data collection	[41]	Wired	Polling	Conflict-free	(Balanced Bandwidth)
Multicycle periodic data collection	[42]	Wired	Polling	Conflict-free	(Balanced Bandwidth)
Optimizing throughput	[43]	Wireless	Pushing	Conflict-free	No consideration
Minimizing latency	[44]	Wireless	Pushing	Conflict-free	No consideration
Minimizing latency and optimizing throughput	[45]	Wireless	Polling	Prioritized conflict-free	No consideration
Saving energy	[46]	Wireless	Polling	Prioritized	No consideration

this thesis, which could achieve a reasonable and feasible schedule with practically short computation time on an off-the-shelf computer.

Our work is different from related work in the following aspects. As shown in Table 2.2, other works focus on how to schedule transmitting queries without conflicts in order to minimize network latency, optimize network throughput, or save energy. However, none of them focuses on the problem that unbalanced network bandwidth between uplink and downlink causes unexpected big latency of unpredictable on-demand communication or decrease of success probability for periodic data collection. We firstly propose a polling-based data traffic control scheme which is available to any IWSN protocols that assign unequal bandwidth to upstream and downstream communication. In addition, we show a scheduling using a network traffic uniformity metric for IWSNs to realize both scheduled periodic data collection at high success ratio and unpredictable on-demand communications with short latency.

2.4 Data Traffic Control Scheme for Periodic Data Collection

In this section, we provide terminologies at first. Then, we present an outline of how our data traffic control scheme for periodic data collection works. We define two kinds of frames and Table 2.3 summarizes terminologies and notations.

- **QueueFrame** The first kind is the **QueueFrame (QF)**, which consists of N_{all} slots whose length is Δt_s as shown in Figure 2.4. Let N_i be the number of nodes which belong to group i and Let τ_g be the number of multicycle polling groups in an IWSN. Some nodes may belong

Table 2.3: Notation and description.

Notation	Description
N_i	N_i is the number of nodes in group i.
τ_g	τ_g is the number of multicycle polling groups in an IWSN.
N_{all}	N_{all} is the number of total nodes which belong to all groups.
QF	QueueFrame(QF) is the minimum frame which consists of N_{all} slots.
ΔT_{QF}	ΔT_{QF} is The length of a QF. A central node can transmit at most N_{all} polling requests in a QF.
Δt_s	Δt_s is the length of a slot. A central node can transmit a polling request in a slot.
T_i	T_i is a collection data interval of group i.
SF	ScheduleFrame is the minimum unit of multicycle polling schedule.
T_M	T_M is a multicycle polling interval.
N_{QF}	N_{QF} is the number of QFs in a SF.

to multiple different groups. Then N_{all} in an IWSN is as bellows;

$$N_{all} = \sum_{i=i}^{\tau_g} N_i. \tag{2.2}$$

We assume that a central node collects data from all nodes at multiple periodic cycle as shown in Figure 2.2. The length of a QF is $\Delta T_{QF} (= \Delta t_s \times N_{all})$. In our proposal, a central node generates at most N_{all} queries to all nodes in a QF, so that our data collection scheduler gives fair opportunities for all nodes even in a short time span (ΔT_{QF}). We consider the bandwidth of downward traffic to define polling cycle Δt_s. For example, a our preliminary experiment showed downward traffic for SmartMesh IP that is based on the 6LoWPAN and IEEE802.15.4e standards is assigned every about 2 seconds by the default setting. Δt_s should be longer than the frequency for pre-assigned downlink traffic of IWSN protocols. In this thesis, we set Δt_s to $3s$ for simulation setting. The order of queries to nodes are decided by a data collection scheduler. One of the simplest ways to do this is arranging nodes in ascending order of a unique id.

- **ScheduleFrame** The second kind is the **ScheduleFrame(SF)**, which consists of QFs, as shown in Figure 2.5. Let T_M be the least common multiple of collection cycle of all groups as shown in Equation (2.3). In the interval T_M, the SF consists of $N_{QF} (= T_M \div \Delta T_{QF})$ QFs. Therefore, a SF is the minimum unit of multicycle polling schedule of N_{all} nodes.

$$T_M = \text{lcm}(T_i), \text{for } i = 1, 2, ..., \tau_g. \tag{2.3}$$

Our data collection scheduler decides all polling orders in order to make network traffic of

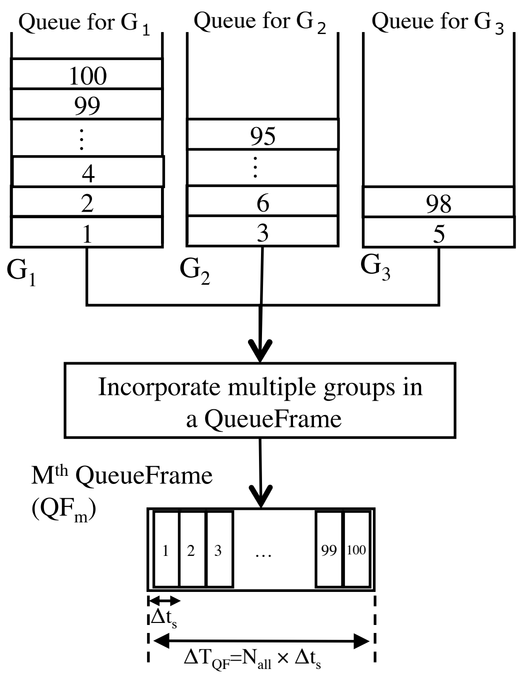

Figure 2.4: A composition of a QueueFrame (QF).

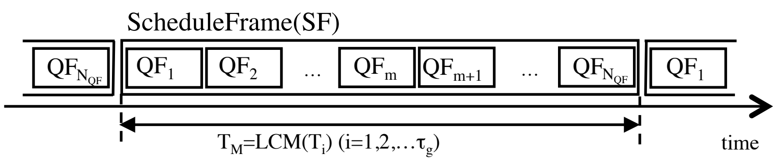

Figure 2.5: A composition of a ScheduleFrame (SF).

all applications uniform during the T_M and our data traffic control scheme generates polling queries every Δt_s according to the schedule.

Below, we give an outline of how our proposal works over any IWSN protocols that assign unequal bandwidth to upstream and downstream communication. In our scheme, we define two kinds of frames to decide all polling orders in both a short time span and a long time span.

Our scheme incorporates multiple groups in a single group as shown in Figure 2.4. A central node transmits at most N_{all} queries every Δt_s according to a sequence of nodes in a QF. This polling behavior is as same as a monocycle polling scheduling. We consider that a central node controls transmitting timing of a polling query so that down stream traffic from a central node is less than pre-assigned downlink traffic of an IWSN protocol.

Then, a scheduler provides a SF that is a multicycle polling schedule during the least common multiple of collection cycle for all groups as shown in Figure 2.5. In a SF, each QF may assign a different sequence of all nodes. For example, a sequence of all nodes in QF_1 may be assigned in ascending order and one in QF_2 may be assigned in descending order.

2.5 Problem Formulation of Data Collection Scheduler

A polling cycle of a group T_i consists of K_i QFs where

$$\begin{aligned} K_i &= T_i/\Delta T_{QF} && (2.4)\\ &= T_i/(\Delta t_s \times \sum_{k=1}^{\tau_g} N_k) && (2.5)\\ &= T_i/(\Delta t_s \times N_{all}). && (2.6) \end{aligned}$$

Let $C_{i,j}$ be a total number of combinations of selecting QFs which any node j in any group i uses for data transmit and its retries$R_{i,j}$ is

$$C_{i,j} = K_i \times \sum_{m=1}^{K_i} {}_{(K_i-m)}C_{min(K_i-m,R_{i,j})}, \tag{2.7}$$

where $min(K_i - m, R_{i,j})$ is the smallest number of the two arguments.

Total number of combinations of selecting QFs which all nodes in any group i use is

$$C_{i,all} = \prod_{j=1}^{N_i} C_{i,j}. \tag{2.8}$$

Then, total number of combinations of selecting QFs for all groups and all nodes $C_{all,all}$ is

$$C_{all,all} = \prod_{i=1}^{\tau_g} (C_{i,all})^{\frac{T_m}{T_i}}. \tag{2.9}$$

The number $C_{all,all}$ depends on the number of nodes N_{all} in an IWSN, the number of collection cycles τ_g, communication quality, i.e., the number of total communication retries $R_{i,j}$.

On the other hand, maximum number of polling queries for periodic data collection of group i during a polling cycle T_i is$\sum_{j=1}^{N_i}(1 + R_{i,j})$. Let $Q_{i,x}$ be total polling queries of the xth QF of a group i during T_i. When the queries are uniformly generated during T_i, $Q_{i,x}$ is approximated by the

Equation (2.10).

$$Q_{i,x} \approx \frac{1}{K_i}\sum_{j=1}^{N_i}(1 + R_{i,j}) \tag{2.10}$$

The total polling queries of xth QF of all groups is

$$\begin{aligned} Q_{all,x} &= \sum_{i=1}^{\tau_g} Q_{i,x} & (2.11)\\ &\approx \sum_{i=1}^{\tau_g} \frac{1}{K_i}\sum_{j=1}^{N_i}(1 + R_{i,j}). & (2.12)\end{aligned}$$

Then, we find an optimal schedule from $C_{all,all}$ patterns of schedules in order to generate network traffic load uniformly for periodical data collection. The standard deviation of Equation (2.12) $\sigma(Q_{all,x})$ is a measure of how uniform polling queries are generated during T_M. Our objective function is

$$\alpha = \min\{\sigma(Q_{all,x})\}, where 1 \leq x \leq \frac{T_M}{\Delta t_s \times N_{all}}. \tag{2.13}$$

2.6 GA-Based Slot Assignment Algorithm

Our polling-based data traffic control scheme decides the order of all queries to all nodes according to a schedule that ensures sufficient responsibility of unpredictable requests and stability of periodic data collection. In other words, the slots of queries for periodical data collection and slots for unpredictable on-demand should be uniformly assigned. As we described in Section 2.3, Genetic Algorithm (GA) is classified one of heuristic functions which can find an optimum solution for above our problem, so that we propose a GA-based slot assignment algorithm.

The process of our GA-based slot assignment algorithm includes generating initial population, selection, mutation, crossover, repair, and evaluation as shown in Figure 2.6. We describe the specific process of the flow as follows.

2.6.1 Encoding

Our data collection schedule includes two problems for transmitting polling queries. One is how a central node generates polling queries for heterogeneous multiple periodic data collection with high success ratio. Another is how a central node can transmit polling queries for unpredictable on-demand communication within a deadline over IWSNs. We encode a sequence of nodes and

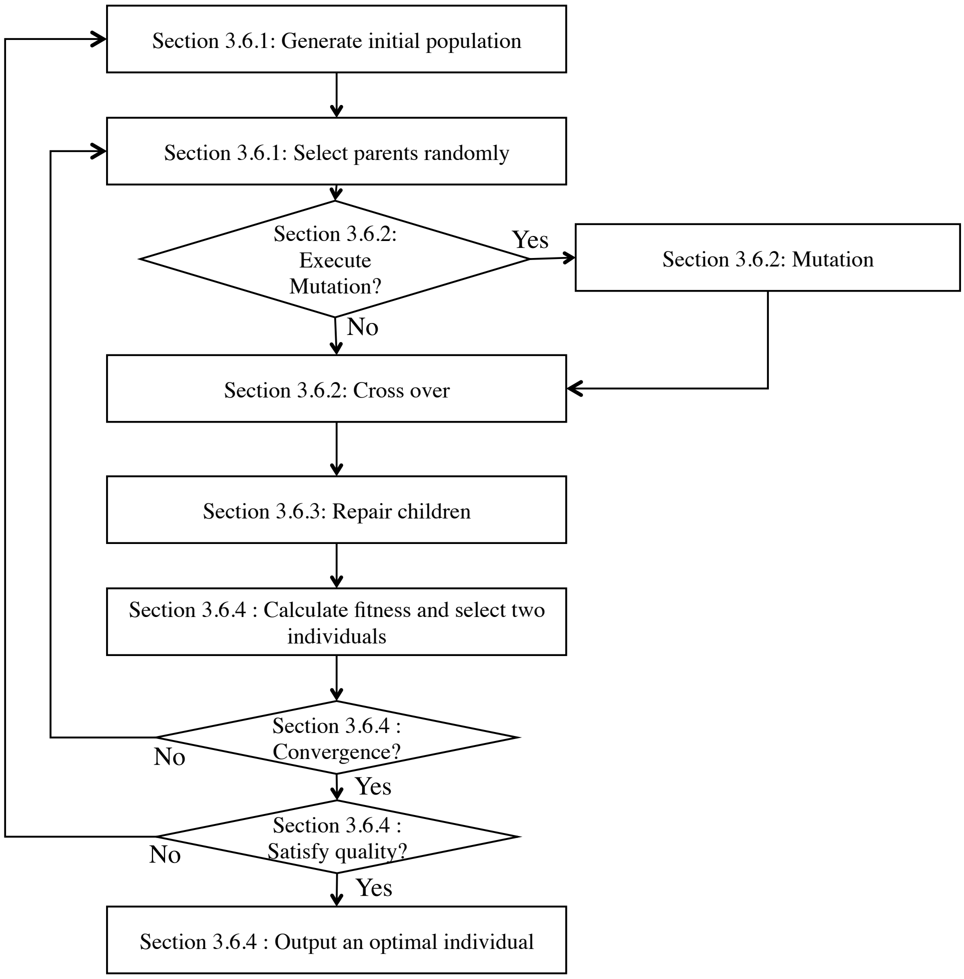

Figure 2.6: Flowchart of our Genetic Algorithm (GA)-based slot assignment algorithm.

Slot number	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	⋯		
Node number	1	2	3	4	5	6	2	1	3	5	6	4	3	1	4	6	⋯		
Slot type	1	1	1	3	3	1	2	2	3	1	3	1	2	2	3	1	⋯		

QF_1 QF_2 QF_3

Figure 2.7: An example of chromosome encoding.

kinds of packet (e.g., periodic data collection, retransmissions for periodic data collection, and unpredictable on-demand communication). In our proposal, each individual has two chromosomes. We describe coding chromosome structure at first and then, we present how to generate the initial population as shown in Figure 2.6.

Chromosome Structure

An example coding chromosome representing a schedule is shown in Figure 2.7. Each individual has two chromosomes corresponding to a sequence of node numbers and a sequence of slot types. Both chromosomes have N_{QF} QFs which consist of N_{all} slots as shown in Figure 2.4, respectively.

As the first chromosome, a sequence of node numbers represents transmission orders from a central node to nodes. In this thesis, we substitute a simple number like $1, 2, 3, \ldots$ as a node number for (Group number, node number) in order to simplify the notation.

As the second chromosome, a sequence of slot types represents kinds of packets that a central node sends to a node at the slot. We define three slot types in this thesis. First type “1” represents that a central node transmits an original polling request packet for periodic data collection. Second one “2” represents that a central node sends the retry packet of the original packet to an end node. Final one “3” means that a central node can transmit a query for an unpredictable on-demand collection to any node at the slot.

For example, Figure 2.7 shows that a central node transmits a request packet to collect periodical application data from node 1 at slot 1, and a retry request packet to node 1 at slot 8 if the central node did not receive the data corresponding to the previous request to node 1. If the central node received a reply from node 1 by slot 8, the central node does not transmit the retry packet to node 1 at slot 8. In addition, a central node may transmit a polling query to any node for an unpredictable on-demand collection at the slot 4 but a central node never transmits a request packet to node 4 for a periodic data collection at the slot.

Initialization

Each Individual of GA population has two chromosomes. At the beginning of the initialization, a data collection scheduler generates N_{QF} QFs such that all nodes in an IWSN are collated at random in order to encode a chromosome that represents an order of node IDs. We think that the random selection provides a search diversity. Our simulation experiment showed that it did not have a big impact on the performance. After that, a chromosome of slot type is assigned. For the slot type chromosome initialization, first of all, the first queries of all nodes to collect data while a periodic cycle are assigned. The assignment of slot type chromosome denotes 1 in the chromosome as shown in Figure 2.7. Then, the timing of transmitting queries for retries of the collection while a periodic cycle are assigned. The assignment denotes 2 in the slot type chromosome. The number of retries depends on a path quality (end-to-end PER) between a node and a central node. A central node gets the path quality from a network manager in an IWSN, because a network manager perceives network conditions. If a path quality between a node and a central node is 80%, a central node will require at least a retry. Finally, the slot type of other slots that are assigned to the node are 3. "3" means that unpredictable on-demand data traffic from central nodes can be generated at the slot.

2.6.2 Crossover and Mutation

Figure 2.8 shows an example of mutation, crossover and repair. At the crossover process, we select two individuals randomly as parents and pick a QF up from the node sequence chromosome of the parent, respectively. The QF of the first parent is exchanged with the QF of the second parent in order to generate two new children individuals. Let N_{indiv} be the number of individuals that are generated at the initial phase. Then we have at most $N_{indiv} + 2$ kinds of node sequence in T_M. In other words, there are at most $N_{indiv} + 2$ schedules at the moment.

The mutation operation also provides a search diversity by avoiding the local maximums. Mutation operation alters one individual. In our proposal, the alteration of one individual in each mutation operation is considered. A new individual is produced in the same way of the initialization phase. The new one is replaced to an individual of GA population. Mutation operation does not have to be executed before every crossover operation as shown in Figure 2.6.

2.6.3 Repair

Because each slot type chromosome corresponds to each node sequence chromosome, slot type chromosome of the first parent are also exchanged with one of the second one. However, the exchanges

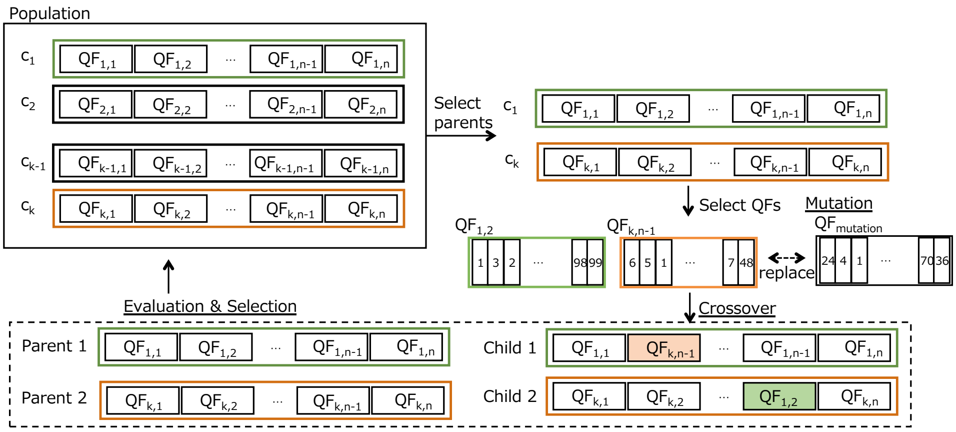

Figure 2.8: An example of our mutation and selection.

of slot type chromosome is more complicated than the one of node sequence chromosome because "1" (a query for collecting data) appears once during a data collection cycle T_i and "2" (a query for retries) should be assigned after the "1". As shown in Figure 2.9, slot number 1 and 2 of the slot type chromosome of a child 1 should be repaired after the exchange. In our proposal, the repair follows the 9 cases as shown in Figure 2.10;

- **Case 1-1**:

  The child chromosome does not need to be repaired at all.

- **Case 1-2**:

  After the crossover operation, a new child chromosomes has no "1" during the T_i. Then, an anterior "3" in the collection cycle should be changed to "1", and one of the "2" should be changed to "3" if there are more "2" than required. If there is no "3" before the exchanged slot number, the child chromosome does not need to be repaired at all.

- **Case 1-3**:

  After the crossover operation, a new child chromosome has no "1" during the T_i. Then, an anterior "3" in the collection cycle should be changed to "1". If there are no "3" before the exchanged slot number, the child chromosome does not need to be repaired at all.

- **Case 2-1**:

  After this crossover operation, a new child chromosome has additional "1" during the T_i. Then,

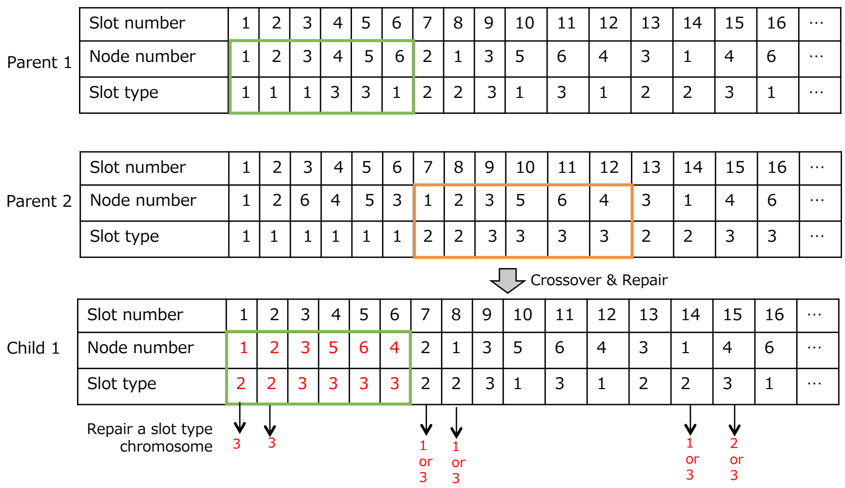

Figure 2.9: An example of the crossover process.

an anterior “1” and “2” should be changed to “3”. Also, one of “3” after the new “1” should be “2” in the case that there are fewer “2” than required.

- **Case 2-2**:

  The child chromosome does not need to be repaired at all.

- **Case 2-3**:

  One of “3” after “1” changes to “2”.

- **Case 3-1**:

  After the crossover operation, a new child chromosome has additional “1” during the T_i. Then, an anterior “1” and “2” should be changed to “3” or a rearward “1” is changed to “3”. Then, one of the “2” should be changed to “3” if there are more “2” than required.

- **Case 3-2**:

  one of “2” changes to “3” in case there are more “2” than required.

- **Case 3-3**:

  The child chromosome does not need to be repaired at all.

		Slot type after crossover operation		
		1	2	3
Slot type before crossover operation	1	Case 1-1	Case 1-2	Case 1-3
	2	Case 2-1	Case 2-2	Case 2-3
	3	Case 3-1	Case 3-2	Case 3-3

Figure 2.10: Cases of repairing genes after crossover slot type chromosomes.

2.6.4 Selection and Fitness Function

After the crossover, mutation, and repair operations, there are $N_{indiv} + 2$ individuals. The first N_{indiv} individuals, which have higher fitness values, are selected and transferred to the next generation. The fitness values of chromosomes are calculated from our fitness function (Equation (2.13)).

One of our objectives is to uniform the data traffic generation from a central node to nodes. In our proposal, each slot in a QF generates at most one query to collect data and the slot type chromosome represents how many queries are generated in a QF. At the same time, our scheduler tries to maximize communication success rate. When the fitness value is calculated, the communication success rate of all nodes are also calculated and checked. If the success rate of all nodes of a child chromosome does not satisfy our target probability, the individual is discarded.

Finally, when the fitness values of the best 5 slot type chromosomes are not changed after crossover operation, and the best fitness value is less than a target threshold value, we consider the GA operation is convergent. After the maximum iteration, we also consider that the GA operation will not be converged. At that time, we return to the initialization procedure. Our experiments showed that after 100 experiments, the average execution time of slot assignment for 10 nodes was 1544 milliseconds. After the convergence, the best slot type chromosome that is ranked in the top 5 and that is expected the highest average communication success rate of all nodes, is selected as the best schedule.

2.7 Evaluation

2.7.1 Simulation Settings

We implemented our GA-based slot assignment algorithm in C using the gcc compiler and ran all experiments on Mac OS 10.12.6 with an 2.7 GHz Intel Core i5 CPU with 16GB RAM. To evaluate the performance impact of our data traffic control scheme, we performed a set of simulations with 10 end nodes placed statically and randomly in a square field. A central node was placed at the

Table 2.4: Simulation conditions.

Item	Notation	Value
The number of nodes in an IWSN	N_{all}	10 nodes
Data collection cycle 1	T_1	2 min
Data collection cycle 2	T_2	4 min
Data collection cycle 3	T_3	8 min
The least common multiple of collection cycles	T_M	8 min
The number of slots in a QF	S_{QF}	10
Slot length	Δt_s	3 s
QF length	ΔT_{QF}	30($S_{QF} \times \Delta t_s$) s
The number of QFs in a chromosomes	N_{QF}	16 ($T_M \div T_{QF}$)
The number of slots in a chromosomes	S_{gene}	160 ($N_{QF} \times S_{QF}$)
The number of individuals	N_{indiv}	20
The number of iteration of crossover operation	I_{max}	4000 times
Packet error rate	PER	0–20%
MAX retry count	R_{max}	2
Threshold of a path quality	Th_{path}	90%

lower left-hand corner of the field, and LOADng that is a routing protocol for low power and lossy networks [49] was applied to create multi-hop routes from all nodes to a central node with a shortest-path metric. A network topology was fixed during a simulation. We also assume that packet loss among neighbors is caused by several factors such as propagation models, signal processing technology, transmitting power, antenna characteristics, and reception sensitivity, but except signal interference from other nodes. We then determined PER of a path quality at random for every node. As shown in Table 2.4, we set the range of PER to 0–20% . Although the link PER dynamically changes in reality, we assume it is stable and constant in this thesis. Evaluation under dynamic environment is left as future work. After creating a network topology, our implementation uses the information and evaluates how a central node generates network traffic load for periodical data collection uniformly.

At first, we randomly divided 10 end nodes to three groups that have different data collection cycles $T_1 = 2$ min, $T_2 = 4$ min, $T_3 = 8$ min, respectively. According to our preliminary experiments, the average number of crossover operations for 100 network topologies was 1071 when our fitness values were converged. Then, we set the maximum number of crossover iteration to be 4000. Table 2.4 summarizes the details of the other parameter settings.

2.7.2 Simulation Results

As described in Sections 2.2 and 2.6.4, the standard deviation of Equation (2.12) is measure of how uniform polling queries are generated during T_M and Equation (2.13) is our fitness function. Therefore, we calculated fitness values by Equation (2.13) for evaluation.

First of all, we confirmed that random selection at the initialization step did not have a big impact on the performance. To evaluate this, we created a network topology and then we generated 200 kinds of schedules for a scenario as shown in Table 2.4. All the results were converged. Then we show one of the detail results as bellow. The fitness values of the top 20 individuals after the GA-based slot assignment are plotted in Figure 2.11. As shown in Figure 2.11, minimum standard deviation at the initial random selection where the number of GA operation is 0, was 0.186 and the final one was 0.049. In this case, our proposal generates a schedule for periodic multicycle data collection after 2000 times GA crossover operations. In contrast, the standard deviation of the conventional scheme [46] is 0.49. From uniforming application traffic of periodic data collection point of view, our proposal is superior to the conventional scheme.

The mean value of all slot type during T_M is calculated by the Equation (2.14). SI_{ideal} is the ideal value when the polling schedule includes all slot assignment of data transmission for all nodes and their retries. As shown in Figure 2.12, the best schedule in the 20 individuals assigns the ideal number of slots for data transmission and their retries.

$$SI_{ideal} = \frac{1}{N_{QF} \times N_{all}} \times \sum_{i=1}^{\tau_g} (\frac{T_M}{T_i} \times SI_i), \tag{2.14}$$

where the sum of slot type of all nodes which belong to group i is as follows;

$$SI_i = N_i + 2\sum_{j=1}^{N_i} R_{i,j} + 3(K_i - N_i - \sum_{j=1}^{N_i} R_{i,j}).$$

In addition, Figure 2.13 shows the initial E2E communication success provability of all nodes and Figure 2.14 shows E2E communication success provability of all nodes after 2,000 times crossover operations, and average one of all nodes in each individual. The E2E communication is calculated by the left-side hand of Equation (2.1). The abscissa of the graph represents the ID number of individuals, and the ordinate represents the E2E communication success provability. In this case, E2E communication success provability of all nodes does not satisfy our target probability (90%) as shown in Figure 2.14. Therefore, the individuals are discarded and we return to the initialization

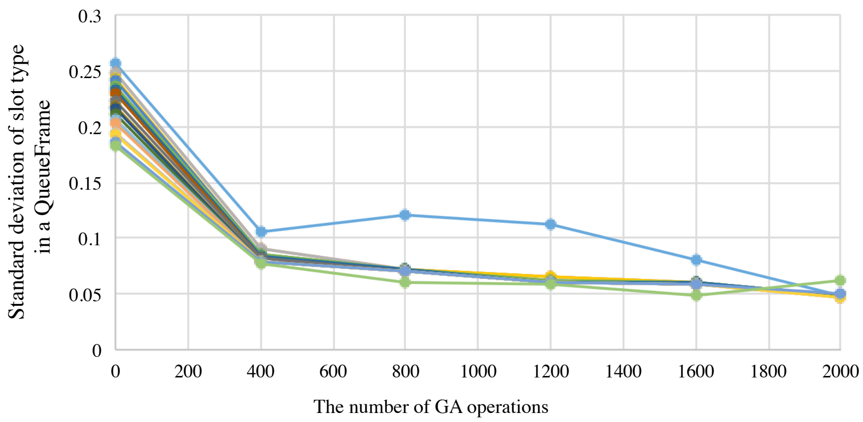

Figure 2.11: Transition of standard deviation slot type value of the top 20 individuals, where $N_{all} = 10$.

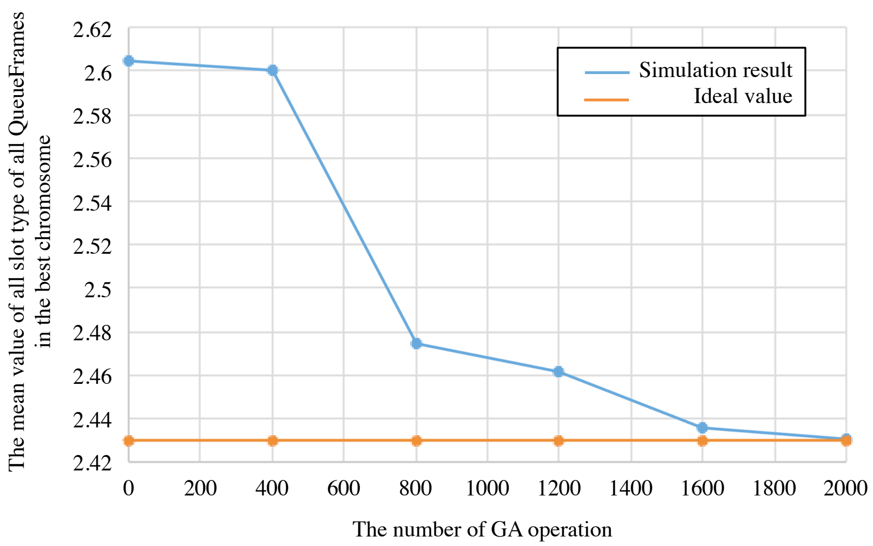

Figure 2.12: Comparison between the mean value of slot type of all QueueFrames between the best chromosome and ideal value.

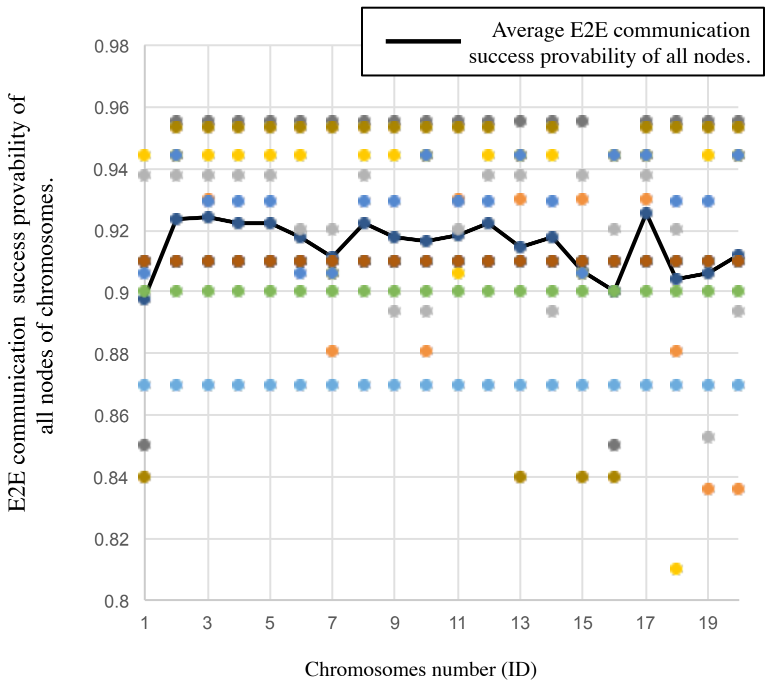

Figure 2.13: Initial E2E communication success provability of all nodes in each individuals and average one of all nodes in each individual.

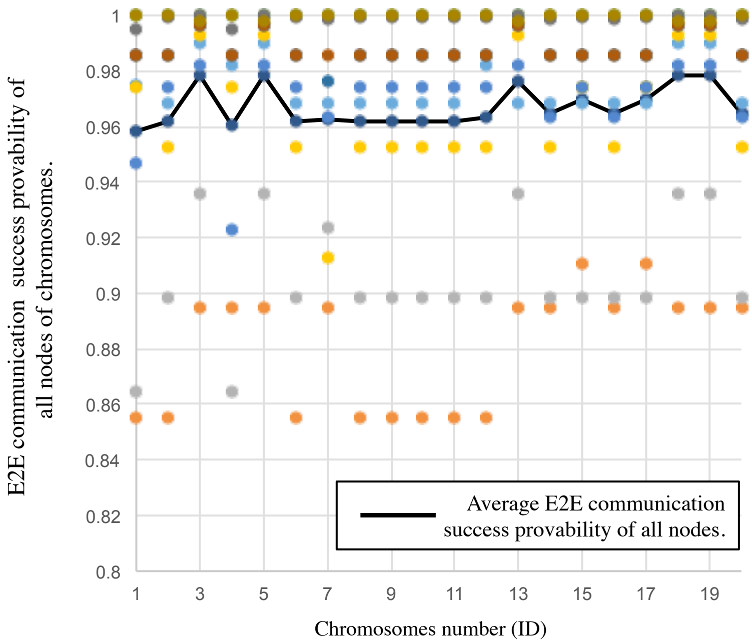

Figure 2.14: E2E communication success provability of all nodes in the top 20 individuals and average one of all nodes in each individual after 2,000 times crossover operations.

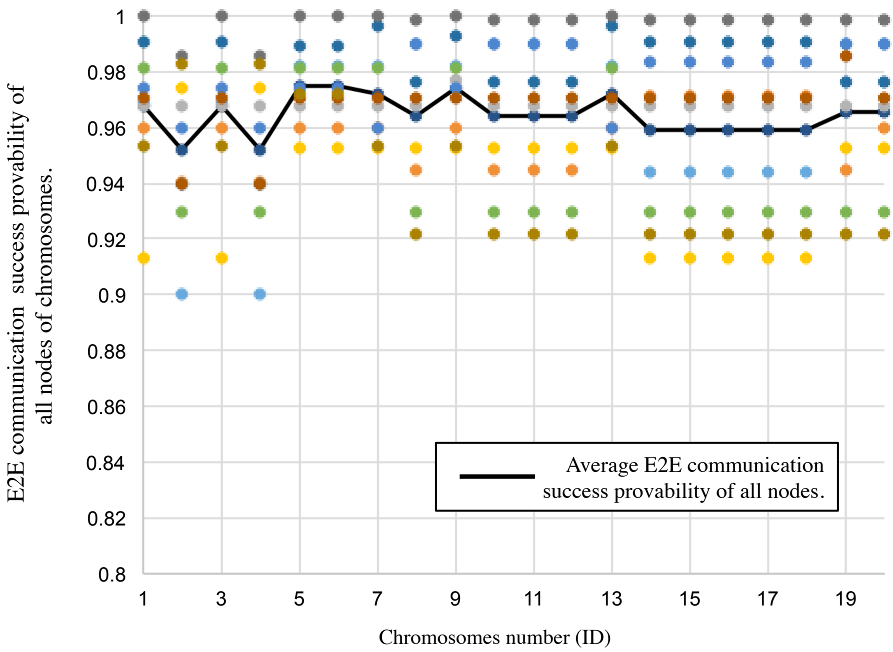

Figure 2.15: Final E2E communication success provability of all nodes in the top 20 individuals and average one of all nodes in each individual.

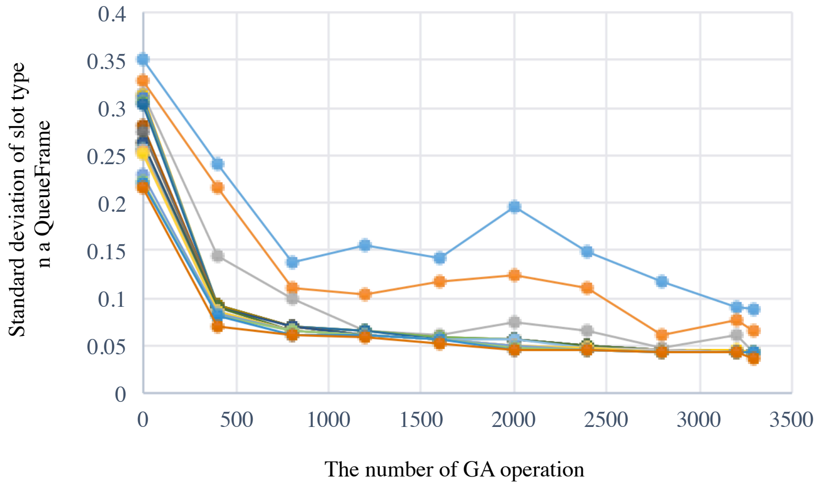

Figure 2.16: Transition of standard deviation slot type value of the top 20 individuals, where $N_1 = 8$, $N_2 = 1$, and $N_3 = 1$.

procedure as shown in Figure 2.6. Our experiments showed that after 100 experiments, the average iterations is 1.36. Then, the final E2E communication success provability of all nodes after the GA-based slot assignment, and average one of all nodes in each individual The E2E communication success provabilities are over 90% as shown in Figure 2.15. The average success probability is 97.4% and the lowest one is 95.2% where the 5th individual is selected. The results meet our target requirements, as listed in Table 2.4.

2.8 Discussion

2.8.1 Possibility of Finding Schedules When Application Data Traffic Is High

In general, the more nodes which belong to the shortest cycle group there are, the central device generates polling requests more frequently during T_M. In the case that most nodes belong to the shortest cycle group, a scheduler should determine a multicycle scheduling that generates polling transaction uniformly among less options than in the other cases. Then, we additionally evaluate our proposal in the worst case for 10 nodes that 8 nodes in T_1, 1 node in T_2 and 1 node in T_3 respectively. Other conditions listed in Table 2.4 are the same as the above evaluation. Our proposal scheduler found schedules that meet our target requirements as shown in Figures 2.16 and 2.17. The result of the E2E communication success provability depends not on the grouping ratio but on the network topology.

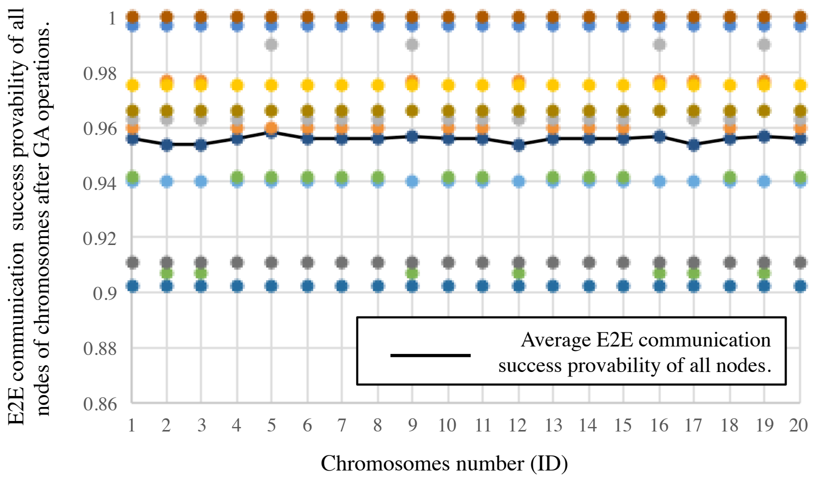

Figure 2.17: E2E communication of all nodes in the top 20 individuals and average one of all nodes in each individual, where $N_1 = 8$, $N_2 = 1$, and $N_3 = 1$.

2.8.2 Possibility for a Scalable IWSN

An IWSN may include over 100 end devices such as sensors or actuators [50], though we evaluated our proposal in Section 2.7 under a condition that an IWSN includes only 10 end nodes. To evaluate scalability of our proposal scheduler, we increased the number of nodes from 10 to 100. At the same time, we expand polling cycles because our proposal data traffic control scheme has a TDMA-like manner. To be more precise, we enlarged the length of a QF 10 times because the length of a QF is $\Delta T_{QF} (= \Delta t_s \times N_{all})$ as described in Section 2.4. Accordingly, we also enlarged the length of a SF 10 times. Simulation conditions are described in Table 2.5. Figure 2.18 shows that our proposal also provides a solution which has a scalability that is adaptive to a small scale network to a large scale one with the same uniformity. Furthermore, the average execution time of slot assignment for 100 nodes was 17.3 sec, which is roughly in proportion to the number of polling queries in a wireless network.

2.8.3 Power Consumption

In IIoT systems, many devices such as sensors are powered from batteries. From power consumption point of view, it is important to save energy while devices communicate with others. Due to wireless radio interferences from other systems, PER may increase in IWSNs. Though standard IWSN protocols such as ISA100.11a and WirelessHART increase the reliability of wireless networks by

Table 2.5: Simulation conditions.

Item	Notation	Value
The number of nodes	N_{all}	100 nodes
Data collection cycle 1	T_1	20 min
Data collection cycle 2	T_2	40 min
Data collection cycle 3	T_3	80 min
The least common multiple of collection cycles	T_M	80 min
The number of slots in a QF	S_{QF}	100
Slot length	Δt_s	3 s
QF length	ΔT_{QF}	$300(S_{QF} \times \Delta t_s)$ s
The number of QFs in a chromosomes	N_{QF}	160 $(T_M \div T_{QF})$
The number of slots in a chromosomes	S_{gene}	1600 $(N_{QF} \times S_{QF})$
The number of individuals	N_{indiv}	20
MAX iteration of crossover operation	I_{max}	$20,000$ times
Packet error rate	PER	0–20%
Max retry count	R_{max}	2
Threshold of a path quality	Th_{path}	90%

TSCH mechanism, our proposal scheme increases retry queries in harmful network conditions in order to collect data at high success ratio when PER increases. As we described in Section 2.7, we assume it is stable and constant in this thesis. Detailed evaluation is left as future work.

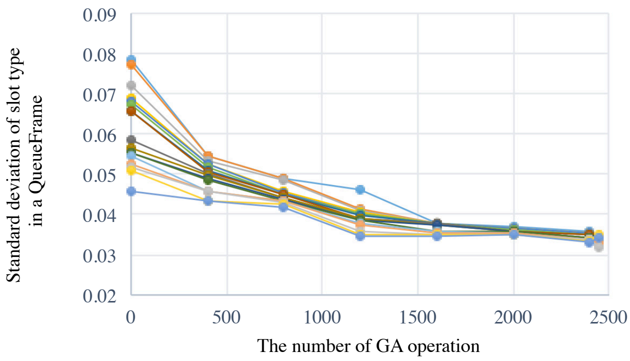

Figure 2.18: Transition of standard deviation slot type value of the top 20 individuals, where $N_{all} = 100$.

2.8.4 Strengths and weakness of our proposal

Our proposal is very simple way to generate the network traffic load uniformly over stable IWSNs. Our polling-based communication mechanism decides only transmission timing of all polling queries and does not need to reschedule even if a network topology changes. In addition, our proposal does not depend on any IWSN protocols, such as ISA100.11a, WirelessHART, and IEEE802.15.4e. Therefore, this is widely used for any IIoT applications. Above features are strong points of our proposal.

On the other hand, our data traffic control scheme generates polling queries every Δt_s according to the schedule. If IIoT applications require data more frequently than Δt_s, our proposal can not collect data from end devices, but IWSNs should be designed and selected to meet IIoT application requirements. Moreover, the minimum unit of multicycle polling schedule of all nodes is calculated by Equation (2.3). This value is the least common multiple of all polling cycles. Thus, if an IIoT system supports many kinds of polling cycles, the execution time of slot assignment would be long. In addition, as described in 2.8.2, our execution time is in proportion to the number of polling queries in wireless network. If the IWSNs are not stable and a number of retries of transmitting polling queries in order to collect data at high success ratio, the execution time of slot assignment would be also much longer than the execution time for stable IWSNs. In either case, we need a faster scheduler but our polling-based communication mechanism is still available.

2.9 Summary

This chapter introduce a data traffic control scheme over IWSNs for polling-based data collection from multiple IIoT applications and its data collection scheduling. Our algorithm enables the uniform generation of network traffic load for periodical data collection. Polling request packets from a central node to end nodes can be uniform even if the polling cycles are multiple. This is achieved through a data traffic control scheme that generates a polling query at a fixed interval according to a schedule which is decided to make the occurrence of transmitting a polling request for periodical data collection, its retry in case of wireless communication failure or an interruption of higher priority request, and non operation by a GA-based algorithm. At the same time, our scheduler can give maximum opportunities for retransmitting polling requests and data collection ratio of periodic data packets achieves higher than 90%. We adopted a GA based algorithm to create a schedule, but our data traffic control scheme, of course, allows an algorithm that is not GA-based to decide a schedule.

In this thesis, we did not address dynamic adaptation of our scheme to handle dynamic or unexpected changes in application and system requirements. For example, some applications will be installed after the deployment phase. In this case, a scheduler should modify a current schedule. Updating a whole schedule is one of the simplest solutions, but the applications should wait to execute this for at most T_M. Moreover, in our scenario, all nodes reply a response packet corresponding to a polling request, but an application such as error log monitoring on an end device may transmit more information (including system log data) than that which can be delivered in a reply packet. Therefore, in order to make the data traffic of all applications uniform, a scheduler should consider both uplink traffic and downlink traffic at the schedule creating phase. We plan to tackle these issues as future work.

Chapter 3

A Priority-based Dynamic Multichannel Transmission Scheme for Industrial Wireless Networks

3.1 Introduction

To guarantee a deadline for unpredictable on-demand request and periodical data collection at high success ratio in a IWSN over a polling-based communication, we proposed a centralized scheme in Chapter 2. In a polling-based scheme, all data exchanges must be made through a central device. However, a central device should wait at least a round trip time between a central device and an end device to receive a reply. Moreover, to operate IIoT systems safely, a central device may not transmit a new polling query until it receives a reply of the previous polling request, because some packets would be dropped in IWSNs due to signal interference, collision, etc. As such, reliable communication sacrifices promptness in conventional polling-based schemes.

Some industrial applications, such as building automation and factory automation, require shorter delay and more reliable communication for unpredictable on-demand polling than other IIoT applications whose main purpose is monitoring the IIoT systems. A feasible technique is to assign bandwidth to delay-sensitive communication to avoid interference and collision with other general periodic communication. However, the amount of required bandwidth cannot be predicted or estimated beforehand, because they are unpredictable and on-demand communication. In addition, control packets for network management, e.g. routing and topology maintenance, must have certain

Table 3.1: IIoT applications and system requirements in Chapter 3.

	920-MHz Band	2.4-GHz Band
Applications	- AMI - DA	- PA and FA - PM
Application processing	- Remote Monitoring - Remote Operation	- Remote Monitoring - Remote Operation
Communication type	- Publish/Subscribe - Request/Response	- Publish/Subscribe - Request/Response
Remote monitoring cycle	30 min (periodic data)	1 – 5 min (periodic data)
Maximum delay for remote operation	20 – 60 sec (unpredictable data)	5 – 20 sec (unpredictable data)
Max number of nodes in wireless network	500 nodes	500 nodes
Packet length	500 – 600 B	90 B
Communication speed	50-100 kbps	250 kbps
Protocols	IEEE802.15.4e	ISA100.11a

amount of bandwidth to make an IWSN reliable and stable. Table 3.1 summarizes IIoT applications and system requirements of this chapter [9, 12, 15, 32, 38].

In this chapter, we propose a priority-based dynamic multichannel transmission scheme for both a central device and end devices to improve reliability and performance of IWSNs and simultaneously satisfy different requirements on, guaranteed deadline for on-demand communication, low PER for periodic communication, and sufficient bandwidth for control. Our proposal has a compatibility of central control with autonomous decentralized control by end devices in an IWSN.

In our proposal, we define three priorities of packets depending on their type and then assign one dedicated channel to each priority. The highest priority is assigned for unpredictable and on-demand packets. A central host transmits polling requests for them, which are downward packets to end devices in this channel. The reply packets from end devices are also defined as the highest priority, because both the bi-directional packets, such as polling communication, should be transferred in an IWSN within a short delay. The second one is periodic packets from end devices, which regular packets, such as application data and health monitoring results regarding network conditions, are periodically transferred to a central device. The lowest priority is defined network control packets among neighbors, including routing packets, time synchronization packets, or beacon packets. A central device can transmit the highest-priority packets at any time but still control the transmission timing through centralized administrative control, as in ISA100.11a., because the bandwidth of downward packets in an IWSN is limited as we describe in Chapter 2. More specifically, a device replies to a request packet at the time specified by a central host. The highest priority is a central control scheme.

On the other hand, channel selection and decision of transmitting timing are autonomous decentralized control by end devices in an IWSN. More specifically, when each devices ftransmits periodical data or forward a packet according to a routing table, which is decided by a centralized administrative control manager, such as a network manager in ISA100.11a, the device firstly scans the highest priority channel by using CSMA/CA mechanism. When the channel is busy, a polling query or its reply packet is transmitting by a neighbor and the device wait to send the packet until the priority channel is free. If there is no transmission of high-priority packets in the vicinity, the device can transmit the packet to its neighbor. In addition, only when there is no packet transmission on either of high and middle priority channel, a node can transmit a network control packet. These mechanisms are autonomous decentralized one by devices in order to avoid collision packets over higher priority channels.

The contribution is to propose a priority-based multi-channel transmission scheme that determines when and what packets should be transmitted on which channel. Through simulation, we validate our proposal for two industrial applications: AMI and industrial process monitoring and control. We also theoretically estimate the lower bounds of available bandwidth for middle and low priority packet transmission. Moreover, TDMA-based protocols like ISA100.11a typically have a scheduler for allocating network resources such as time slots to all nodes. This scheduling process is often complicated and the scheduler has to deliver the information to all nodes whenever a new node joins the network or a network topology changes. In contrast, since our scheme only determines when to generate and transmit a packet at a root node for remote control and at nodes for periodic monitoring, it does not need to adjust a schedule as far as the maximum number of hops and the number of nodes do not change. We discuss this advantage in more details in 3.6.3.

The rest of the chapter is structured as follows. In this chapter, we explain common assumptions of our target IIoT systems first and Section 3.3 presents an overview of related work. In Section 3.4, we propose the priority-based transmission scheme. Then, we evaluate the communication delay for highest-priority packets and available bandwidth for other packets in Section 3.5. In Section 3.6, we discuss compatibility with ISA100.11a and overhead incurred in implementing our proposal before discussing our summary and future work in Section 3.7.

3.2 Challenges

The packet error rate (PER) is normally one of the most important parameters for evaluating reliability of a wireless network. In addition, guaranteed deadlines should be considered for IWSNs, because industrial applications require real time processing. Several wireless protocols have been already standardized and developed for industrial applications, including WirelessHART and ISA100.11a. To decrease the PER for collecting periodic data from sensors in dense and lossy wireless networks, these standards use TDMA-based MAC protocols. Such protocols overcome the problem of packet collisions in the network. In fact, the data collection ratio for WirelessHART reaches more than 99%, because SmartMesh WirelessHART devices basically perform retransmission twice at most [29]. This performance seems sufficiently high for remote monitoring purposes.

On the other hand, when an unpredictable on-demand packet has to be sent to or from a node, the packet either consumes assigned bandwidth or waits several seconds until the next assigned bandwidth comes available. This can cause random latency, which further depends on TDMA scheduling, retransmission timing, and wireless radio conditions. Most WSNs does not support real time communication [51] and it is difficult for even TDMA-based MAC protocols such as IEEE802.15.4e to support real time communication for large scale networks like AMI [30].

As described below in Section 3.3, when we use normal ISA100.11a for both remote monitoring and remote control of devices, it does not guarantee the latency of on-demand and multi-hop communication at any time, although it can transfer a higher-priority packet by applying the priority CSMA/CA scheme among single-hop neighbors. If such on-demand communication can be expected, then a system manager with an optimized scheduler may enable ISA100.11a to allocate communication timing for all nodes in order to transfer a higher-priority packet within a certain delay and with due consideration to maintain a high data collection ratio. Whenever the network topology is changed, because of the instability of the radio environment, however, the schedule must be updated, so this is an unrealistic solution.

As another solution to the problem, we could use multiple ISA100.11a network functions on different network interfaces, that is, one ISA100.11a function for remote monitoring, and another for remote control, for example. In this case, each function would concentrate on scheduling transmission of a single application packet with a high end-to-end path success probability. Unfortunately, this approach faces the same problem that a system manager must deliver an optimized schedule to all nodes whenever the network topology is changed. Moreover, each node would have to control multiple ISA100.11a network functions precisely, but the standard does not describe how an application can

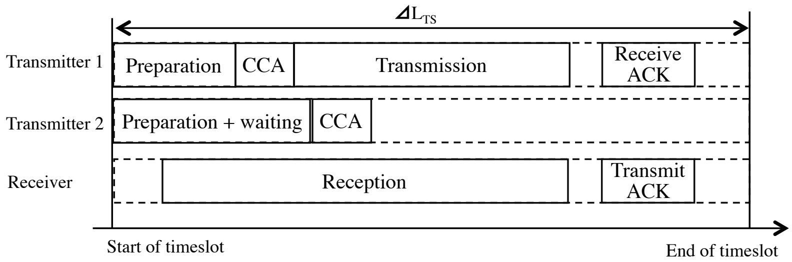

Figure 3.1: A shared time slot using the CSMA/CA technology of ISA100.11a.

manage multiple ISA100.11a networks.

Therefore, our challenges in this thesis are to mitigate unexpected latency for unpredictable and high-priority IWSN communications and to show how to meet system requirements for high end-to-end success probability of periodic communication.

3.3 Related Work

In wireless sensor networks, MAC is a key technology that in determines channel access delay and utilization. MAC protocols are roughly classified into three types: contention-based, contention-free, and hybrid.

First, contention-based schemes (using CSMA/CA) such as IEEE802.15.4 determine transmission timing by checking existence of carrier signals, i.e. carrier sense. When a network is large or dense, the PER is normally high and as such CSMA/CA-based MAC protocols cannot guarantee latency [52–54].

Second, contention-free MAC protocols using TDMA implement scheduled communication with a centralized coordinator, such as a network manager. In TDMA-based MAC protocols, a node transmits and forward a packet and to a neighbor according to an allocated time slot schedule. When packet transmission fails, a node should wait until the next assigned time slot to resend the packet. Therefore, the end-to-end delay depends on the whole schedule and its cycle length called superframe. To reduce latency in industrial networks, Saputra and Shin proposed a scheduling scheme for ISA100.11a superframes [55]. This scheme specifies how to build a superframe to guarantee the delay for periodic upward packets from sensors to a root node and how to check the schedulability of a superframe.

Finally, IWSNs often adopt hybrid schemes [24, 25, 56, 57]. While the hybrid standard schemes such as ISA100.11a and WirelessHART use a TDMA-based MAC protocol, they also provide periodic data communication at low PER. At the same time, these schemes adopt a CSMA/CA-based MAC protocol for unpredictable transmission requirements, such as network control packets, alert information, on-demand requests, and retransmission of data packets.

During a CSMA/CA period in a hybrid scheme, ISA100.11a nodes can use a priority CSMA/CA scheme, as shown in Figure 3.1. Waiting time proceeding to transmission of a high-priority packet is shorter than that of a low-priority packet as shown in Figure 3.1, where transmitter 1 has a high-priority packet and transmitter 2 has a low-priority one. Because of difference in waiting time, transmitter 2 can detect transmission of a high-priority packet during its CCA (Clear Channel Assessment) and stop the transmission attempt. This scheme enables priority control within single-hop communication and decreases the probability of collision among transmission of packets of different priority [33]. We also use this priority CSMA/CA like scheme in our approach.

3.4 Priority-based Transmission Scheme with Dynamic Channel Shift

In this section, we provide assumptions and terminologies at first. Then, we present an outline of our priority-based transmission scheme with dynamic channel shift. We also give detailed algorithms for priority-based channel selection and the transmission and reception mechanisms.

3.4.1 Assumptions and Requirements

In Chapter 2, we introduced common assumptions such as, "Industrial applications features", "Polling-based communication", and "Traffic patterns". In this section, we provide additional assumptions of our proposed scheme as bellow.

- **MAC protocols.** Recently, many new industrial wireless systems have been deployed. Most of them use TDMA-based protocols such as ISA100.11a, wirelessHART, and IEEE802.15.4e/g based protocols in order to avoid interference among internal nodes and keep communication success probability high. Those protocols provide multi-hop and time-synchronized networks that consist of a central manager and other nodes that are synchronized with the central manager. We assume that our targeted network is also multi-hop and time-synchronized, but our proposal operates over a MAC layer to decide what channel to use and when to transmit packets and does not rely on any specific MAC protocols.
- **Network topology and its condition.** Similarly to other TDMA-based WSN protocols, we also assume a tree topology whose root is a central manager. Our proposal does not specify any

routing protocols as far as a stable tree-based routing topology is established and maintained for a large-scale WSN. In simulation experiments, we consider a network of 500 nodes with 8 hops for 920 MHz and 16 hops for 2.4 GHz at maximum.

- **Priority Level of packets.** In an IWSN, multiple applications would simultaneously operate such as periodic data gathering and remote control. In addition, networking functions such as routing and time synchronization are also running. Among them, remote control is the most crucial and must be given the highest priority to guarantee real time communication. Furthermore, its responses from nodes to a root node should have the higher priority than those packets belonging to periodic data gathering. Although frequent loss of control packets affects stability and reliability of a WSN, a best-effort service is enough. We evaluate the lower bound of available bandwidth for lower priority packets in Section 3.6. Details of prioritization will be given in the next subsection.

- **Multiple communication channels.** TDMA-based MAC protocols for IWSNs have channel hopping functions to enable coexistence of multiple networks in the same area and dynamic bandwidth allocation. In this thesis, we assume that three channels are available to use.

3.4.2 Terminologies

We define our terminologies as follows;

- **Frame composition.** First, we define three types of frames over a MAC protocol.

  The first type is the **SlotFrame**, which consists of two timeslots as shown in Figure 3.2. Both ISA100.11a and IEEE802.15.4e technologies divide time into timeslots of configurable length, with typical durations ranging from 10 to 14 ms. These technologies do not, however, support MAC layer retransmission within a timeslot. The SlotFrame enables transmission of a packet within the $1st$ timeslot and retransmission within the $2nd$ timeslot. ΔL_{SF} denotes the length of a SlotFrame, e.g., 20 to 28 ms.

  The second type is the **ComFrame**, which consists of SlotFrames, as shown in Figure 3.3. The number of SlotFrames in a ComFrame is calculated as the maximum hop count in a multi-hop wireless network plus 3. In ISA100.11a, a root node knows the whole network topology. The number "3" is a key number that was chosen to avoid hidden terminal problems on a multi-hop route, as described in detail later. In this thesis, we assume that the number of SlotFrames in a ComFrame is $11(8 \text{ hops} + 3)$. In addition, through centralized administrative control, a

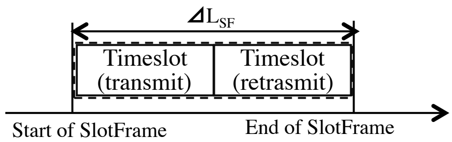

Figure 3.2: A composition of a SlotFrame

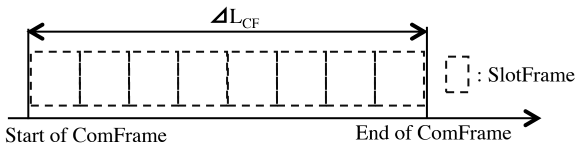

Figure 3.3: A composition of a ComFrame

root node assigns a ComFrame to a node when it joins the network. The assignment does not change even if the network topology changes.

The final frame type is the **AppFrame**, which consists of ComFrames on multiple channels, as shown in Figure 3.4. The number of ComFrames in an AppFrame is a system parameter that depends on the application requirements. ΔL_{AF} denotes the length of a AppFrame. For example, if an application remotely operates all devices in 30 min and collects data from all devices in 30 min, then $\Delta L_{AF} = 30\ min$. In Figure 3.4, the network manager divides the AppFrame to two blocks. In this example, block 1 is used for controlling all nodes ($L1$ and $L2$ packets), collecting data from all nodes ($L3$ packets), and transmitting network control packets ($L4$ packets). Other blocks, such as block 2 in Figure 3.4, are used for bidirectional communication ($L1$ and $L2$ packets) needed for repeat attempts at remote control or data collection from devices, and for transmitting network control packets ($L4$ packets). A system manager determines the number of blocks in an AppFrame.

- **Priority level and communication channels.** As mentioned above, we define 4 priority levels. The highest level ($L1$) is for downward packets from a root node to a sensor node (end device) that an application controls. The second ($L2$) is for upward packets in response to $L1$ packets. The third ($L3$) is for periodically collected data transferred from an end device to a root node, e.g., a network health report or sensing data. The lowest priority level ($L4$) is for network control packets, e.g., routing packets, time synchronization packets, or beacon packets. In our proposed scheme, $L1$ and $L2$ packets are transferred over communication channel 1 ($Ch1$), $L3$ packets are transferred over channel 2 ($Ch2$), and $L4$ packets are transferred over channel 3

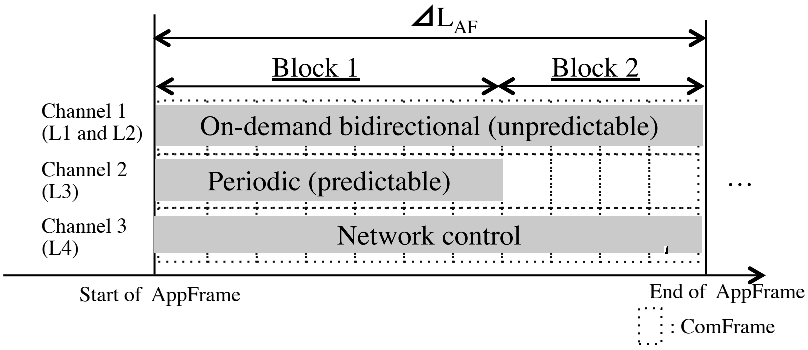

Figure 3.4: A composition of a AppFrame

($Ch3$). Here, $Ch1$ and $Ch2$ are contention-free channels like TDMA-based communication, whereas $Ch3$ is a contention-based channel like CSMA/CA-based communication.

3.4.3 Outline

We give an outline of how our scheme simultaneously fulfills several requirements of industrial wireless communications: a guaranteed deadline for on-demand communication, data collection at low PER, and communication of network control packets among neighbors.

In our scenario, there are three kinds of packets. The first kind is unpredictable packets for on-demand control. The second is periodic packets generated by sensors for periodic data collection. The third is network control packets that build multi-hop routes from sensors to a root node and exchange time information for synchronization among nodes.

We first rank packets according to industrial application requirements. To provide a guaranteed deadline, we define an on-demand downward packet from a root node to a sensor to have the highest priority ($L1$) and an on-demand upward reply packet from a sensor to a root node to have the second-highest priority ($L2$). The third priority ($L3$) is for periodic data collection packets from any sensor, and the lowest priority ($L4$) is for network control packets.

In addition, our priority-based dynamic multi-channel transmission scheme uses three communication channels. The $L1$ and $L2$ packets between a root node and sensor nodes share a communication channel ($Ch1$). The periodic $L3$ packets use another communication channel ($Ch2$) for a certain period of time, and the $L4$ packets use a third channel ($Ch3$). In other words, a root node sends an on-demand request packet with priority $L1$ while waiting to receive a reply packet ($L2$) for a previous request packet. During the same period, a sensor node sends a periodic data collection packet ($L3$) to a root node on $Ch2$. The timing for a sensor node to transfer such a periodic packet to a root node is

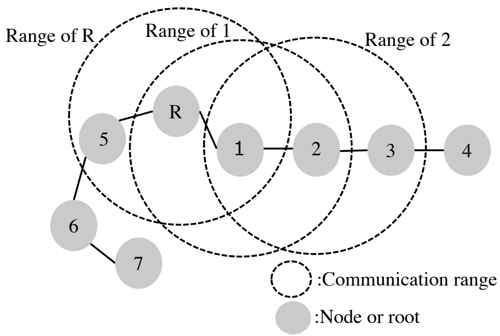

Figure 3.5: A simple example of network topology

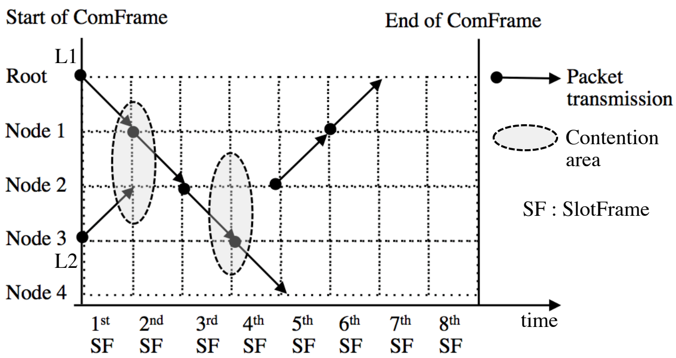

Figure 3.6: An example of packet flow over highest priority channel(Ch1).

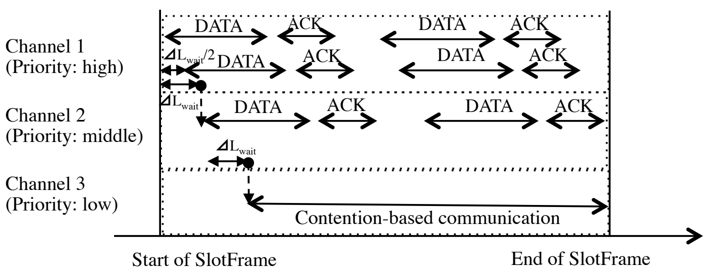

Figure 3.7: Outline of dynamic channel shift

decided by the root node when the sensor node joins the network. Network control packets ($L4$) should be transferred only when no neighbors have to transfer higher priority packets. In each SF, nodes scan channels in order of priority. When a node detects any packet is being transmitted in a higher priority channel, it stays at the channel and receives the packet. Otherwise, it moves to a lower priority channel and check the existence of packets. We describe details later in this section.

	1st SF	2nd SF	3rd SF	4th SF	5th SF	6th SF	7th SF
Root	Ch1(L1) R->1	Ch3	Ch2(L3) 5->R	Ch3	Ch3	Ch1(L2) 1->R	Ch3
Node 1	Ch1(L1) R->1	Ch1(L1) 1->2	Ch3	Ch3	Ch1(L2) 2->1	Ch1(L2) 1->R	Ch3
Node 2	Ch1(L2) 3->2	Ch1(L1) 1->2	Ch1(L1) 2->3	Ch3	Ch1(L2) 2->1	Ch3	Ch3
Node 3	Ch1(L2) 3->2	Ch3	Ch1(L1) 2->3	Ch1(L1) 3->4	Ch3	Ch3	Ch3
Node 4	Ch1	Ch3	Ch3	Ch1(L1) 3->4	Ch3	Ch3	Ch3
Node 5	Ch1	Ch2(L3) 6->5	Ch2(L3) 5->R	Ch3	Ch3	Ch3	Ch3
Node 6	Ch2(L3) 7->6	Ch2(L3) 6->5	Ch3	Ch3	Ch3	Ch3	Ch3
Node 7	Ch2(L3) 7->6	Ch3	Ch3	Ch3	Ch3	Ch3	Ch3

Figure 3.8: An example of SlotFrame usage in a ComFrame

3.4.4 Example of Priority-based Dynamic Multi-channel Transmission Mechanism

We next provide an example of how to ensure preferential communication of a downward packet from a root node to an end device ($L1$), and how to avoid contention between a downward packet and an upward packet ($L2$) in response to a previous downward packet. As noted above in 3.4.1 and 3.4.3, the order of the priority is pre-determined and all nodes share the information.

Figure 3.5 shows a simple example of a network topology. The network consists of 8 nodes and has a maximum of 4 hops. Figure 3.6 shows an example of packet flow, in which the root generates an $L1$ packet and node 3 generates an $L2$ packet. At the $2nd$ SlotFrame in the ComFrame, node 2 cancels forwarding of the $L2$ packet to node 1. Then, node 2 waits two SlotFrames to avoid collisions due to the hidden terminal problem. While node 2 waits, the $L1$ packet is delivered to node 4 without any delays. Then, the $L2$ packet is eventually transferred to the root node at the $6th$ SlotFrame. This crossed-transfer mechanism guarantees a maximum latency for transmission of the highest priority information.

In addition, our scheme uses a dynamic channel shift mechanism to communicate information about other priority levels, as shown in Figure 3.7. As noted above in 3.4.1, our proposal uses three communication channels and nodes share the number of channels and the order of scanning channels as well as packet priority. All nodes choose $Ch1$ ($L1$ or $L2$) at first. Then, if they do not detect any packets over $Ch1$ during ΔL_{wait}, they move to $Ch2$ and scan the channel again. For example, in Figure 3.5, nodes 6 and 7 shift from $Ch1$ ($L1$ or $L2$) to $Ch2$ ($L3$), and node 4 does not detect any packets over $Ch2$ and so shifts to $Ch3$ ($L4$), while the other nodes stay at $Ch1$. For this example, Figure 3.8 summarizes the channel usage for all nodes and SlotFrames. Most TDMA-based protocols predetermine such a complete and detailed schedule as Figure 3.8 and send it to all nodes to follow the same schedule. On the other hand, emission of a request packet from a root node to node 4 and that of a data packet from node 7 to a root node are predetermined, but other detailed slot and channel usage are autonomously and dynamically decided by our dynamic channel shift mechanism. Since control packets belonging to $L4$ use remainder of network resources, we evaluate the available bandwidth for $L4$ in Section 3.5.

3.4.5 Detailed mechanism

- **Transmission policy.** Every node transmits a packet according to its SlotFrame usage policy and ComFrame usage policy.

- **SlotFrame usage policy.** All nodes select a communication channel for each SlotFrame by a dynamic multi-channel transmission mechanism. Then, nodes transmit $L1$, $L2$, and $L3$ packets over $Ch1$ or $Ch2$ as shown in Figure 3.7. They can also retransmit a packet once per SlotFrame according to the Retransmission policy below. Nodes transmit $L4$ packets by CSMA/CA over $Ch3$.

- **ComFrame usage policy.** A root node uses the $1st$ or $2nd$ SlotFrame to transmit an $L1$ packet to a node (final destination) in a ComFrame. It first checks the hop count to the final destination in the current network topology and the hop count of an $L1$ packet in a previous ComFrame. When the hop count of the previous $L1$ packet is even, the root node uses the $2nd$ SlotFrame to avoid packet collisions due to hidden terminal problems on a multi-hop route. A node transmits an $L2$ packet at the $1st$ SlotFrame when it received an $L1$ packet in a previous $ComFrame$. Thus, an $L1$ packet and an $L2$ packet are transmitted in the same ComFrame. Regarding $L3$ packets, the root node notifies a node of a ComFrame to use for $L3$ packet transmission when a node joins the network. Each node transmits an $L3$ packet at the $1st$ SlotFrame of its own ComFrame.

  Since the length of ComFrame is large enough for a packet sent by a node at any hop distance to reach a root node, ComFrame assignment can be maintained and fixed as far as the maximum hop count does not increase.

- **Retransmission policy.** The length of a time slot in a TDMA scheme like ISA100.11a is just

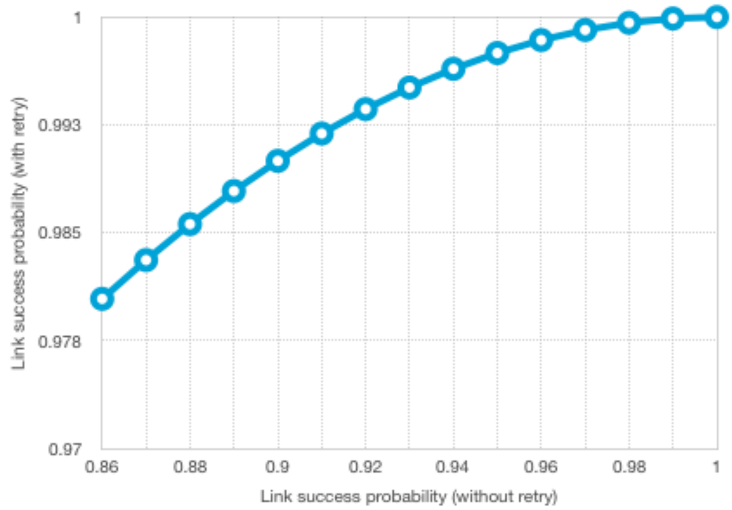

Figure 3.9: Communication success probability comparison in a SlotFrame between one hop communication with retry and without retry

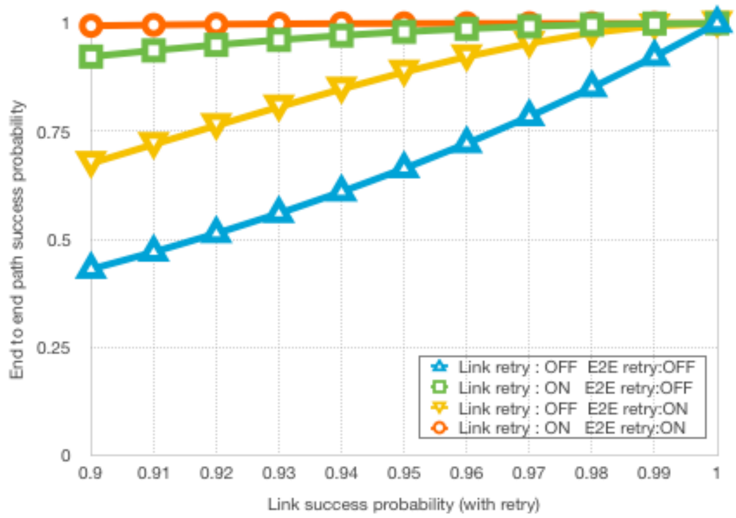

Figure 3.10: End to end path success probability.

long enough for a MAC frame of maximum size and its acknowledgement (ACK). Normally, TDMA schemes do not permit any retries in a time slot. For lossy networks, however, link quality (i.e., the communication success ratio) is significantly improved by permitting a node to send a retry packet, as shown in Figure 3.9. In our proposal, transmission of a retry packet is permitted for every one-hop communication of $L1$, $L2$, and $L3$ packets.

Figure 3.10 shows a comparison of successful path transmission probabilities among the

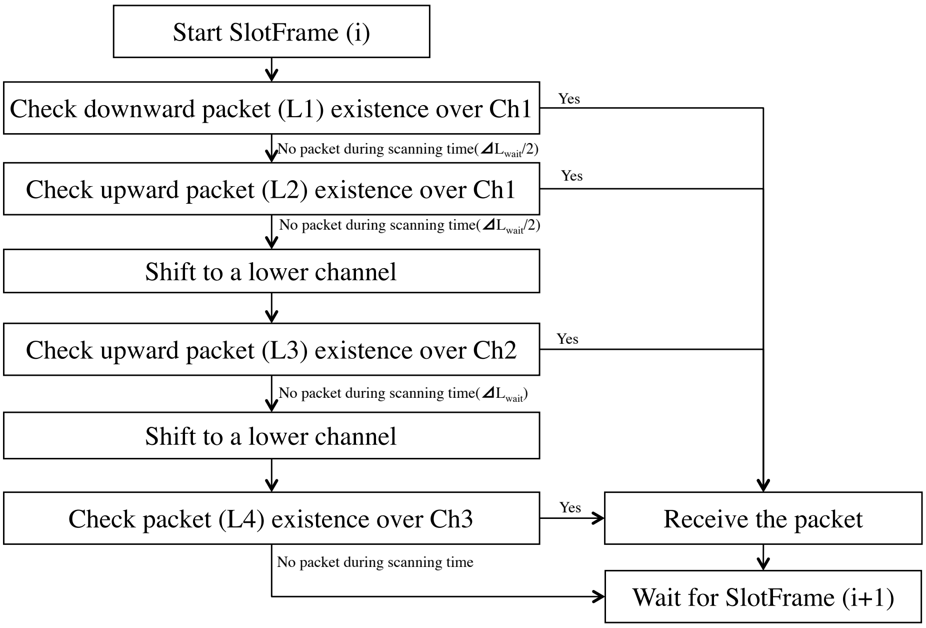

Figure 3.11: Outline of reception process with channel shift.

following four retransmission policies: The first policy does not support retry for either link communication or end-to-end communication; the second one supports link retry but not end-to-end retry; the third one supports end-to-end retry but not link retry; and the fourth policy supports both link and end-to-end retry. The figure shows that both retry types are effective even if the retry is only attempted once.

- **Packet forwarding policy.** As described above for the SlotFrame usage policy, our proposed scheme does not allocate the intermediate SlotFrames of all ComFrames. For example, only the 3 shaded SlotFrames are allocated in advance in Figure 3.8. Every node basically seeks to forward a packet received at a previous SlotFrame with a dynamic channel shift for the transmitting process rule when it does not detect any higher-priority packets.

- **Reception process with dynamic channel shift.**

A root node transmits an $L1$ packet to a node at the $1st$ or $2nd$ SlotFrame in each ComFrame. As shown in Figure 3.11, a non-root node first checks $Ch1$ over a period of ΔL_{wait}. If it detects an $L1$ packet, it maintains the channel to receive the packet. It determines the priority level of a packet by detecting the timing. If the timing is $0 \leq \Delta L_{wait}/2$, the packet is treated as $L1$; otherwise, it is treated as $L2$. After searching $Ch1$, the node checks $Ch2$ over another period of ΔL_{wait}. If it detects a packet, it maintains the channel to receive the packet as $L3$. Otherwise, it chooses $Ch3$ as the communication channel at the SlotFrame.

- **Transmission process with dynamic channel shift.** For transmission all nodes must check for packet existence over channels in order of priority until reaching the usage channel, as in the reception process with dynamic channel shift. The transmission process works as follows by packet priority level:

  $L1$**:** A root node (network manager) knows the current network topology and the hop count of a node that is the destination of a previous $L1$ packet. If the hop count is even, the root node cancels transmission of an $L1$ packet at the $1st$ SlotFrame and reserves the $2nd$ SlotFrame in order to avoid the hidden terminal problem on the path.

  $L2$**:** A node that transmit an $L2$ packet to a root node checks for packet existence over $Ch1$ for a period of $\Delta L_{wait}/2$. If it does not detect any packets, it transmits an $L2$ packet over $Ch1$.

  $L3$**:** A node checks for packet existence over all channels in order of priority until $Ch2$, as in the reception process with dynamic channel shift. If the node detects no higher-priority packets, it transmits an $L3$ packet over $Ch2$. Otherwise, it cancels transmission of the $L3$ packet at the current SlotFrame and reserves the next SlotFrame when the number of remaining SlotFrames in the ComFrame is greater than the hop counts.

  $L4$**:** A node checks for packet existence over all channels in order of priority until $Ch3$, as in the reception process with dynamic channel shift. If the node detects no higher-priority packets, it transmits an $L4$ packet over $Ch3$ by CSMA/CA.

- **Forwarding process with dynamic channel shift.** All nodes must check for packet existence, as in the transmission process with dynamic channel shift. The forwarding process works as follows by packet priority level:

  $L1$**:** A node forwards a packet received at a previous SlotFrame to the next-hop node.

  $L2$**:** A node checks for packet existence over $Ch1$ for a period of $\Delta L_{wait}/2$. If the node does not detect any packets, it forwards an $L2$ packet over $Ch1$. Otherwise, it cancels forwarding of the $L2$ packet at the current SlotFrame and reserves the next 3 SlotFrames.

  $L3$**:** A node follows the $L3$ behavior in the transmission process with dynamic channel shift.

  $L4$**:** A node follows the $L4$ behavior in the transmission process with dynamic channel shift.

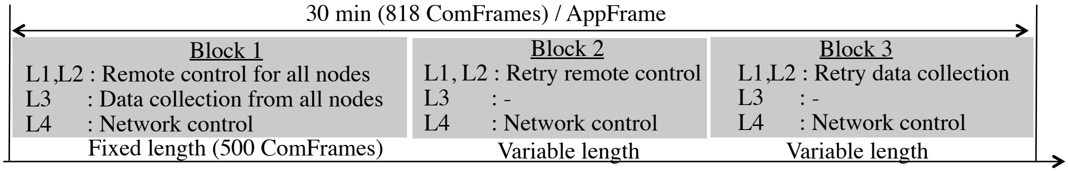

Figure 3.12: AppFrame composition for simulations

3.5 Evaluation

3.5.1 Simulation Settings

To evaluate the performance impact of our priority-based dynamic multi-channel transmission scheme, we performed a set of simulations with 501 nodes placed statically and randomly in a square field. A root node was placed at the lower left-hand corner of the field, and a routing protocol for low-power and lossy networks (LLNs) [58] was applied to create routes from all nodes to the root node with a shortest-path metric. Figure 3.13 shows an example of the resulting network topology.

The network topology was fixed during a simulation, and a total of 10 network topologies were tested to take into consideration localization of sensors. We also assume that packet loss among neighbors is caused by several factors such as propagation models, signal processing technology, transmitting power, antenna characteristics, and reception sensitivity, but except signal interference from other nodes because of TDMA-like transmission. We then determined the link PER at random as shown in Table 3.2. Although the link PER dynamically changes in reality, we assume it is stable and constant in this thesis. Evaluation under dynamic environment is left as future work.

The network is subject to three traffic: request/response type traffic from a root node as unpredictable packets, sensor-to-root traffic as periodic packets, and network control traffic that exchanges information among neighbors. In our proposal, transmission of $L1$ and $L2$ packets for remote control and response is scheduled by a root node to guarantee real time communication. On the other hand, $L3$ packets for periodic data gathering are emitted at predetermined intervals and $L4$ control packets are generated irregularly. Therefore, the worst case scenario is that all of those packets are generated in a certain short period.

In this thesis, we evaluate the worst case performance. More specifically, we define an AppFrame accommodating three traffic classes as shown Figure 3.12. An AppFrame consists of three blocks. The first block is used for a root node to send requests ($L1$) to all 500 nodes for remote control. Responses ($L2$) from nodes are also accommodated in the same block. Block 1 is also used for

Table 3.2: Simulation conditions.

Item	Notation	Value (920 MHz)	Value (2.4 GHz)
Number of nodes	N_{node}	500 nodes	500 nodes
L2 packet length	L_{data}	500 B	127 B
L2 ACK length	L_{ack}	100 B	40 B
Communication speed	v	100 kbps	250 kbps
Data collection cycle	Δ_{AF}	30 min	5 sec
Max hop count	H_{max}	8 hops	16 hops
Average hop count	H_{ave}	3 hops	8 hops
Link PER with retry	Per	0-10%	0-10%
Timeslot length	ΔL_{TS}	100 ms	10 ms
Wait time for channel shift	ΔL_{wait}	5 ms	1 ms

periodic packets ($L3$) and control packets ($L4$). The length of Block 1 is the same as the number of nodes in ComFrames. The second block is used for retransmission of requests to those nodes from which a root node does not receive any response in Block 1. On the contrary, polling-based retransmission of periodic packets is deferred to Block 3, because periodic data gathering is more delay tolerant than remote control. A root node sends a request to resend a data packet to each node from which it fails in receiving a report in Block 1. At this time, requests and responses are given priorities $L1$ and $L2$, respectively. Control packets ($L4$) are irregularly generated in both of Block 2 and 3 as in Block 1. Table 3.2 summarizes the details of the other parameter settings.

3.5.2 Simulation results

End-to-end delay of high-priority packets ($L1$ and $L2$)

All packets are transmitted by ComFrame. At a time when an application queues an $L1$ packet but ComFrame n is already in process, the L1 packet should stay in the queue until the head of the next ComFrame $n + 1$. The request is transmitted to the destination node at ComFrame $n + 1$, and the root node receives the reply packet from the destination node at ComFrame $n + 2$. Therefore, the following defines the range for the end-to-end delay time:

$$2 \times \Delta L_{CF} \leq Delay_{E2E} < 3 \times \Delta L_{CF}. \tag{3.1}$$

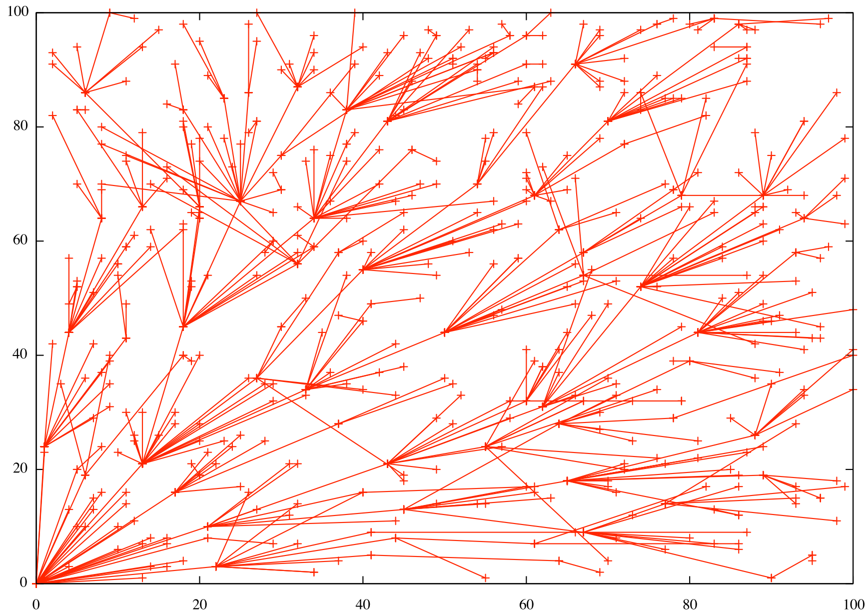

Figure 3.13: Simulated network topology (N_{node} = 500).

In the 920-MHz simulation case, the end-to-end delay was less than $6.6sec$ ($= 3 \times 11 SlotFrames \times 200ms$), while in the 2.4-GHz case, it was $1,140ms$ ($= 3 \times 19 SlotFrames \times 20ms$). Our proposal guarantees the deadline for remote operation, and these simulation results meet our target requirements, as listed in Table 3.1.

Figure 3.14 shows end-to-end delay comparison. We conducted field experiments to obtain delay samples of WirelessHART. In the experiments, a root node of WirelessHART received 90.2% packets ($5,481\ packets(received)$ / $6,088\ packets(total)$) from nodes and the delay considerably fluctuates. The average delay was $1.309sec$. The theoretical maximum delay of our proposal in the similar condition is $1.14sec$ and smaller than the average delay of WirelessHART. To derive this, we substitute the average link success probability of 95% in the experiment to Equation (3.1)

Success rate of high-priority packets

Figure 3.15 shows the simulation results for the success rate of high-priority packets. The root node received the highest-priority packets ($L1/L2$) from all nodes when the link success probability (with retry in a SlotFrame) was greater than 93%. It also received all health report data ($L3$) from almost all nodes when the link success probability (with retry) was greater than 96%. The important points here are that our scheme guarantees the maximum delay for getting information from a node and

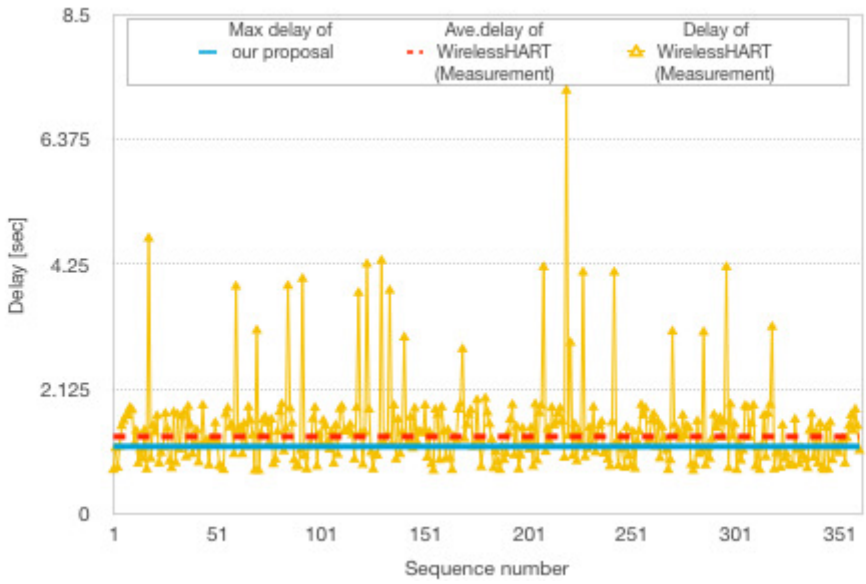

Figure 3.14: End-to-end delay comparison.

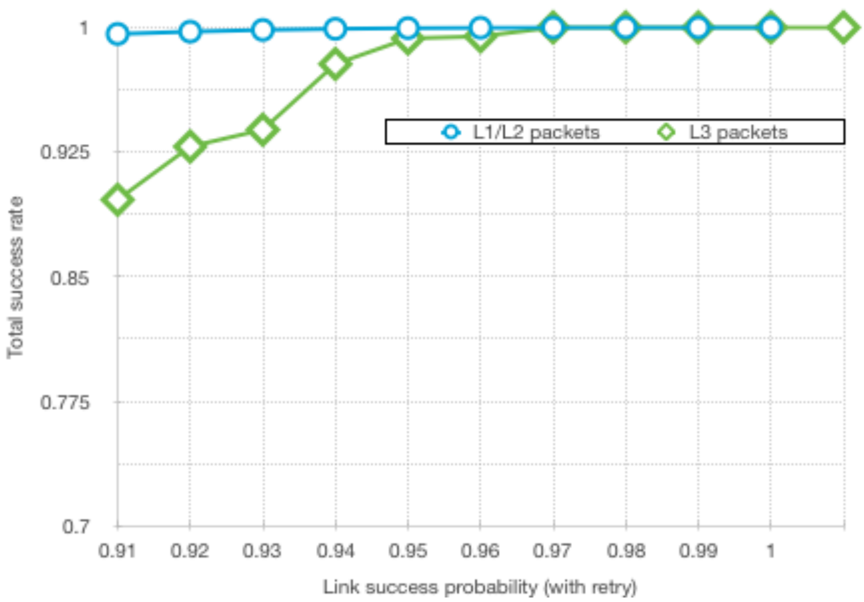

Figure 3.15: Total number of nodes from which the root node successfully received periodic data in an AppFrame.

provides a high success rate for getting packets of different priority at the same time.

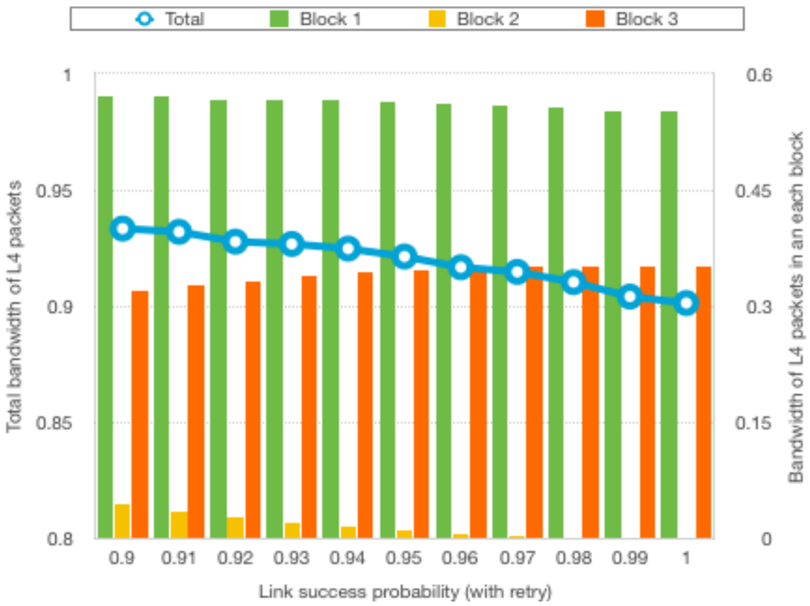

Figure 3.16: Total available bandwidth for $L4$ packets and breakdown by block.

Available bandwidth for $L4$ packets

Figure 3.16 shows the simulation results for the ratio of the available $L4$ channel usage time in which a node can totally transmit $L4$ packets in an AppFrame to the length of the AppFrame. According to the figure, the sum of the $L4$ packet bandwidth from Block 1 to Block 3 was almost 90%. Although up to three instances of higher-priority traffic was generated in Block 1, the impact of the traffic was very limited. Overall, our scheme provides sufficient bandwidth, because WirelessHART requires about 30% of the bandwidth for network control packets. In addition, when the link PER becomes high, more $L1$, $L2$, and $L3$ packets might drop on a multi-hop route. This means that the total usage for $L1$, $L2$, and $L3$ packets would drop, and the total available bandwidth for $L4$ packets would increase. When the network is not stable, more network control packets should be generated in order to repair routes than when the network is stable. Therefore, this approach is appropriate for an LLN.

3.6 Discussion

3.6.1 Lower bound of available bandwidth for $L3$ and $L4$ packets

As we show above, almost all $L3$ packets can delivered when the link success probability (with retry) is greater than 96%. To achieve that level, in Block 1, $L1$ or $L2$ packets and $L3$ packets are not

Table 3.3: Communication patterns for L1 and L2 packets with E2E retry.

No.	Appearance pattern				Required # of ComFrames
	L1 ($1st$)	L2 ($1st$)	L1 (retry)	L2 (retry)	
1	Pass	Pass	-	-	1.0
2	Pass	Fail	Pass	Pass	2.0
3	Pass	Fail	Pass	Fail	2.0
4	Pass	Fail	Fail	-	1.5
5	Fail	-	Pass	Pass	1.5
6	Fail	-	Pass	Fail	1.5
7	Fail	-	Fail	-	1.0

Table 3.4: Expected number of ComFrames for $L1/L2$ packets with an E2E retry

Index i	1	2	3	4	5	6	7
E2E success prob.	1.0	0.99	0.98	0.97	0.96	0.95	0.94
E_i	1.0	1.014	1.029	1.043	1.057	1.070	1.083

often generated on the same path. ISA100.11a can allocate all time slots for all nodes in order to reduce the traffic pattern. Our proposal, however, does not schedule time slots. Instead, the root node notifies each nodes of a sequence number in the ComFrame at which it can transmit $L3$ packets. The sequence number does not depend on either the traffic pattern or network topology but may purely be in order of nodes joining the network. Therefore, we evaluated the worst case scenario in which all $L1$, $L2$, and $L3$ packets are transmitted on the same path or nearby paths at the same time. In this case, the end-to-end path success probability for $L3$ packets in Block 1 is 0% because all nodes on the path are in use. Consequently, the number of data collection packets from sensors depends on the length of Block 3 (ΔL_{Block3}).

The length of Block 3 is calculated by Equation (3.2):

$$\Delta L_{Block3} = \Delta L_{AF} - \{\Delta L_{Block1} + \Delta L_{Block2}\}, \tag{3.2}$$

where ΔL_{Block1} is $N_{node} \times (H_{max} + 3)$, and $\Delta L_{Block1} + \Delta L_{Block2}$ is the number of devices successfully controlled with link retry and end-to-end path retry. Therefore, the length of Block 2 (ΔL_{Block2}) is calculated by Equation (3.3):

$$\Delta L_{Block2} = E_i \times N_{node} - \Delta L_{Block1}, \tag{3.3}$$

where E_i is the expected number of ComFrames per node for $L1/L2$ packets with both link retry

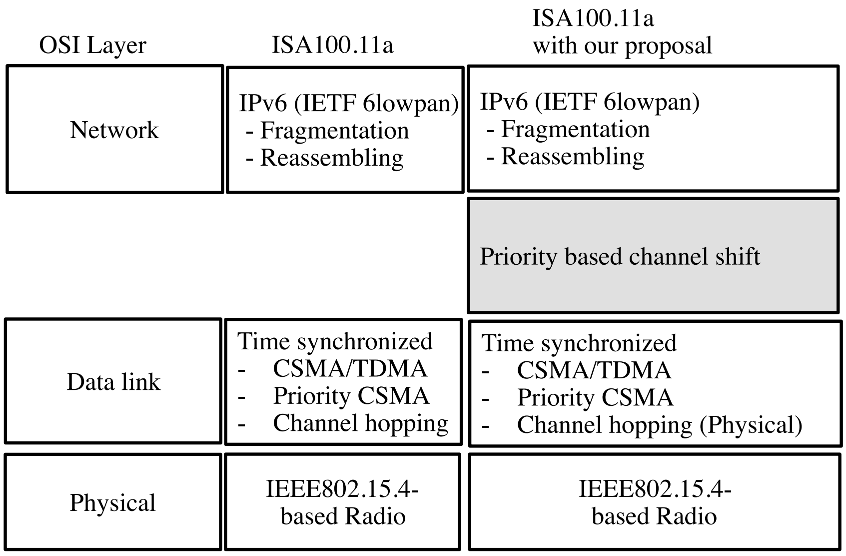

Figure 3.17: Comparison of functions between ISA100.11a and our proposal.

and end-to-end retry. Table 3.3 summarizes the communication patterns for $L1/L2$ packets with end-to-end retry. The expected number of ComFrames for $L1/L2$ packets with end-to-end retry is calculated as shown in Table 3.4. Finally, the number of successfully received $L3$ packets (N_{L3}) is

$$N_{L3} = \Delta L_{Block3} \times \frac{1}{E_i}, \tag{3.4}$$

where $1/E_i$ is the success probability of round-trip end-to-end communication with end-to-end retry. In the worst case, $L3$ packets can be collected at a rate of $45.52 - 63.63\%$ ($0.9 \leq 1/E_i \leq 1$).

On the other hand, $L4$ packet traffic used almost 90% of the bandwidth in our simulation scenario. In the same worst case, we assume that the communication pattern in Block 1 is similar to those in Blocks 2 and 3. Consequently, the lower bound on the available bandwidth for $L3$ and $L4$ packets (P_{L4}) is 36.36%, as calculated by Equation (3.5):

$$\begin{aligned} P_{L4} &= \frac{\{(H_{max} + 3) - (H_{ave} + 3)\}}{\Delta_{L_{CF}}} \\ &= \frac{H_{max} - H_{ave}}{\Delta_{L_{CF}}}. \end{aligned} \tag{3.5}$$

3.6.2 Compatibility with ISA100.11a standard

Figure 3.17 shows how to adapt our proposal to the ISA100.11a standard protocol. Basically, our proposal is a technology between the network and data link layers, so that it does not directly affect processing in those layers. We do have to specify the operation mode and adjust some parameters of the data link layer to compose our own frames. We use priority CAMSA/CA, select slow-hopping mode as the channel hopping pattern, and bundle time slots defined by ISA100.11a to compose a SlotFrame, ComFrame, and AppFrame logically. Our scheme decides what channel ($Ch1$, $Ch2$, or $Ch3$) each node should used at each SlotFrame. The important point is that selecting a channel from among these three in our scheme is equivalent to deciding the operation mode in the data link layer: transmitting a packet, receiving a packet, forwarding a packet, or waiting to forward a packet. If we implement our scheme over one ISA100.11a data link function over one physical interface, the bandwidth for $L3$ and $L4$ packets will decrease. For example, in Figure 3.8, hidden terminal problems could result. An $L4$ packet from the root node to node 5 at the $2nd$ SlotFrame would collides with an $L3$ packet from node 6 to node 5. Also, an $L1$ packet from node 2 to node 3 at the $3rd$ SlotFrame would collide with an $L4$ packet from node 4. To avoid these collisions, we can define a longer length of SlotFrame so as not to overlap the times at which all levels of packets are transmitted. Or, more specifically, our proposal implements three ISA100.11a data link functions over one physical interface in order to guarantee the maximum delay for $L1$ and $L2$ packets and keep the bandwidth for $L3$ and $L4$ packets high.

3.6.3 Strengths and weaknesses of our proposal

Our proposal is very lightweight and much simpler than usual TDMA-based protocols like ISA100.11a. They typically have a scheduler for allocating time slots to meet application requirements. The network manager has to determine and deliver the schedule to all nodes whenever a new node joins the network or the network topology changes. It consumes considerable bandwidth and causes extra delay especially in a lossy and unstable network. Our proposal defines a ComFrame whose length is fixed during network operation. The length depends on maximum multi-hop count that is one of a predetermined system parameter. In a ComFrame, there are at most two high primal packets (an $L1$ packet and an $L2$ packet). Moreover, the $1st$ SF of a ComFrame is assigned to a node to transmit an $L3$ packet. Under these setting, all nodes autonomously determine when and what packets should be transmitted on which channels to avid packet collisions. Then, from scheduling point of view, our proposal is simple and does not need to reschedule and redeliver a schedule even if a network topology changes. Above features are strong points of our protocols.

On the other hand, as described above in 3.6.2, our proposal requires more hardware resources than a normal ISA100.11a, when our scheme operates over ISA100.11a. Our proposal needs at least three channels to avoid collisions among different priority packets. Then a node should have three physical interfaces each of which runs full functions of ISA100.11a or have virtual communication interfaces that operate independently over a physical interface to meet our requirements. In either case, hardware cost for a node becomes more expensive than a normal of ISA100.11a. It may hinder deployment of our proposal, but IWSN should be designed to meet real time requirement to guarantee interaction within a pre-determined deadline.

3.7 Summary

This chapter introduced a priority-based dynamic multi-channel transmission scheme for IWSNs. Our algorithm enables transmission of packets of different priority level in the same period without collisions. The highest-priority packets for remote control can be delivered within a guaranteed deadline through a hybrid control scheme that combines centralized control by a root node and autonomous decentralized radio channel shift by non-root nodes. At the same time, lower priority packets belonging to periodic data gathering and control can receive the satisfactory quality of service, where the collection ratio of periodic data packets is higher than 45% and the lower bound of bandwidth available to control packets is larger than 36% at the worst case scenario.

In this thesis, we do not address dynamic adaptation of our scheme to handle dynamic or unexpected changes in application and system requirements. For example, composition of AppFrame must be predetermined at the deployment phase under assumptions on system configurations, but it should be dynamically regulated to fit to actual traffic demand. In a case of an unstable network, control packets would be transmitted more frequently. Therefore, we need to organize an AppFrame to spare more bandwidth for L4 packets. We plan to tackle these issues as future work.

Chapter 4

Conclusion

IIoT incorporating big data analytics with machine learning has created a big stir in the global industrial world, for the purpose of improving production efficiency, ensuring optimal resource consumption, and operating systems more efficiently and economically. This thesis states such IIoT systems, applications, and wireless network technologies and their related problems regarding reliable and continuous connectivities among devices. IWSNs increase the communication reliability in a wireless network with frequent packet loss and big latency under the heavy network utilization. Thus, this thesis focused on polling-based communication schemes, which are suitable for IIoT applications, such as SCADA, over IWSNs in order to realize more reliable and continuous connectivities.

First, in Chapter 2, we introduce a new central control scheme and it does not depend on any specific IWSN protocols at all. For both periodical data collection with high success ratio and unpredictable on-demand data collection within a short guaranteed delay, our proposed scheme enables the uniform generation of network traffic load for the heterogeneous multi-cycle periodical data collection according to a pre-determined schedule. The uniform network traffic load distribution for periodical data collection enables all devices to have fair opportunities to communicate with a central device leading to high success ratio of communication. Moreover, the traffic distribution also leaves uniformly distributed time slots available to unpredictable on-demand communication.

In this thesis, we adopted a GA based algorithm to create a schedule, but our data traffic control scheme, of course, allows an algorithm that is not GA-based to decide a schedule. Polling request packets from a central node to end nodes can be uniform even if the polling cycles are multiple. This is achieved through a data traffic control scheme that generates a polling query at a fixed interval according to a schedule which is decided to make the occurrence of transmitting a polling request for periodical data collection, its retry in case of wireless communication failure or an interruption of higher priority request, and non operation. Through simulation experiments, we confirmed that our scheduler can give maximum opportunities for retransmitting polling requests and data collection ratio of periodic data packets achieves higher than 95.2%.

Second, to improve performance of polling-based communication over IWSNs in order to adapt polling-based schemes to more critical industrial applications, such as building automation and factory automation, our thesis moved to a hybrid control scheme, which is accomplished by incorporating central control by a central device and autonomous decentralized control by end devices in an standard IWSN, in Chapter 3.

Our main idea is that all devices in an IWSN autonomously select the best channel in a prioritized order of transmitting packets whenever communication is generated, and the neighbors also autonomously and temporarily follow the channel selection by the node. Thus, a higher priority packet can transferred to the final destination without collisions with lower priority packets in an IWSN. In this proposal, for more reliable and stable IWSNs, bandwidth for exchanging network control packets among devices is also considered in addition to application data traffic, such as periodical data collection and on-demand unpredictable communication. Through simulation experiments, we validated that the highest-priority packets for remote control can be delivered within a guaranteed deadline through a hybrid control scheme that combines centralized control by a central device and autonomous decentralized radio channel shift by non-central nodes. At the same time, lower priority packets belonging to periodic data gathering and control can receive the satisfactory quality of service, where the collection ratio of periodic data packets is higher than 45% and the lower bound of bandwidth available to control packets is larger than 36% at the worst case scenario.

In this thesis, we stated issues that standard IWSNs are not suitable for fundamental industrial applications, which use a polling-based communication scheme such as SCADA, and then we proposed a central control poling-based communication scheme and a central-autonomous decentralized hybrid scheme, which accommodates standard IWSNs to IIoT applications. Both schemes fulfill performance requirements of IIoT applications as described in Chapter 2 and 3. Furthermore, our proposals are compatible with standard technologies, which IIoT systems strongly require to use, at all. We believe that in the IIoT era, such polling based communication schemes are fundamental technologies to stably collect periodic and on-demand data, which is source of value in the industrial world. The digital information makes it possible to build smarter supply chains, and manufacturing processes.

IIoT systems with fundamental industrial applications has been drastically enhanced by other technologies: the cloud computing, Artificial Intelligence (AI), Big data, Digital Twin, and Augmented Reality (AR). For example, Digital Twin, which is a digital representation of a physical asset, can

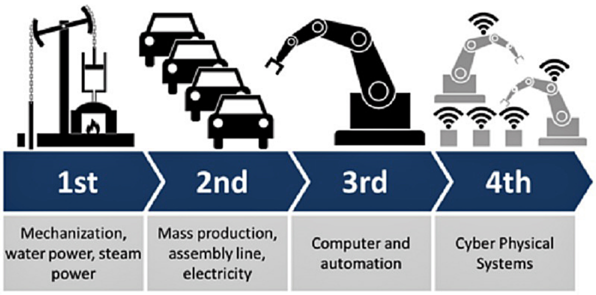

Figure 4.1: Industrial revolutions and Industrial 4.0 [59].

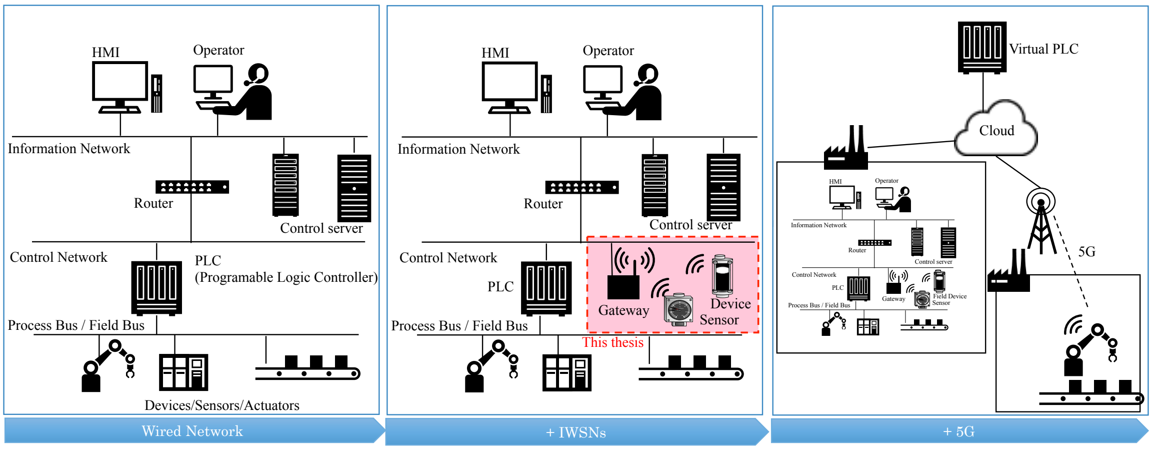

Figure 4.2: Progress and Changes in Industrial systems.

marry the digital and physical worlds to know what is happening in the physical world more exactly. Digital Twin would analyze and resolve incidents more quickly if any incidents are happen. Such the concept of IIoT systems is Industry 4.0 [60] as shown in Figure 4.1. The objects of Industry 4.0 are not simply connected, drawing physical information into information technology (IT) systems, but also communicate, analyze, and use the digital information to drive further intelligent action back in the physical world to execute a physical-to-digital-to-physical transition [59]. In such Industry 4.0 era, we believe that polling-based communication schemes, which provide periodical data collection at high success ratio and unpredictable on-demand communication within guaranteed short delay, will be needed as a way to gather sensing data, which is a basis for intelligent feed back actions to physical worlds.

Regarding wireless communication and network technologies, latency in wireless networks limits many IIoT applications. As described in Chapter 1, so far many IoT and IIoT systems have used cellular networks such as 3G and 4G LTE to connect to the cloud system but, end devices in the systems generate much data and it is hard to process quickly and to transmit the data to the cloud with short latency. In near future, 5th generation mobile networks (5G) could provide faster data transmission than 4G or IWSNs and lead to significant growth in IoT and IIoT systems. This enables multiple logical or virtual networks to operate concurrently on a shared physical infrastructure in order to realize Industry 4.0 as shown in Figure 4.2 [61]. In the 5G era, network architecture would be much more complicated but polling-based application protocols for IIoT applications would continue to use in the new generation industrial systems. Though using IWSNs for industrial applications is just a first phase to realize Industry 4.0, we think our polling-based communication scheme for IIoT contributes to advance industrial systems.

As our future work, we see the following challenges for IIoT systems over IWSNs. First, for our first proposal, research on deriving an optimal and flexible schedule needs to be continued. In this thesis, our work and study are just a first attempt to generate uniform network traffic load for periodical data collection in order to improve performance of simple polling-based communication, which all nodes reply a response packet corresponding to a polling request. For analyzing physical worlds deeply, an application such as error log monitoring on an end device may transmit more information than that which can be delivered in a reply packet. Therefore, we should consider how to provide a schedule management system which is flexibly adaptive to a schedule change and its cancellation in order to actual traffic demand. Second, it will also be necessary to consider unstable IWSNs. In a case of an unstable network, control packets would be transmitted more frequently. It may require more bandwidth for the network control than bandwidth for applications. This challenge is dynamic adaptation of our scheme to handle dynamic or unexpected changes. However these challenges are to give flexibility to our proposal in this thesis, polling-based communication schemes themselves can become base technologies for IIoT applications.

Bibliography

[1] K. Ashton, “That ‘Internet of Things’ Thing,” *RFiD Journal*, vol. 22, pp. 97–114, July 2009.

[2] D. Evans, “The Internet of Things: How the Next Evolution of the Internet Is Changing Everything,” *Cisco White Paper*, April 2011.

[3] B. Dorsemaine, J.-P. Gaulier, J.-P. Wary, N. Kheir, and P. Urien, “Internet of Things: A Definition & Taxonomy,” *IEEE*, 2015.

[4] J. Mocnej, A. Pekar, W. Seah, and I. Zolotova, “Network Traffic Characteristics of the IoT Application Use Cases.” Available online : https://ecs.victoria.ac.nz/foswiki/pub/Main/TechnicalReportSeries/IoT_network_technologies_embfonts.pdf (acceсssd on 28 January 2019).

[5] G. Fortino, C. Savaglio, and M. Zhou, “Toward opportunistic services for the industrial Internet of Things.,” *In Proceedings of the 13th IEEE Conference on Automation Science and Engineering (CASE)*, pp. 825–830, August 2017.

[6] G. Andrei, L. Madhusanka, and K. Dmitry, “Secure Communication and Data Processing Challenges in the Industrial Internet,” *Baltic Journal of Modern Computing*, vol. 4, pp. 1058–1073, 2016.

[7] A. Hansson, M. Khodari, and A. Gurtov, “Analyzing Internet-connected industrial equipment.,” *In Proceedings of ICSigSys’18, IEEE*, 2018.

[8] SmartGrid.gov, “advanced metering infrastructure.” Available online : https://www.smartgrid.gov/files/NIST_SG_Interop_Report_Postcommentperiod_version_200808.pdf (accessed on 28 January 2019).

[9] D. Olsen, “Grid Integration of Aggregated Demand Response, Part 1: Load Availability Profiles and Constraints for the Western Interconnection.” Available online : http://escholarship.org/uc/item/6ps4r3xp (accecssd on 28 January 2019), 2014.

[10] SmartGrid.gov, “Distribution Automation.” Available online : https://www.energy.gov/sites/prod/files/2016/11/f34/Distribution(accessed on 28 January 2019).

[11] Mitsubishi electric systems & service co. Ltd., “Factory Automation, Building Automation, Process Automation.” Available online : http://www.melsc.co.jp/business/catalog/pdf/english/x901103-236a.pdf (accecssd on 28 January 2019).

[12] G. Kalani, *Industrial process control: advances and applications*. Gulf Professional Publishing, 2002.

[13] “Predictive Maintenance Definition - What does Predictive Maintenance mean?.” Available online : https://www.techopedia.com/definition/32027/predictive-maintenance (accessed on 28 January 2019).

[14] D. Gaushell and H. Darlington, “Supervisory control and data acquisition.,” *Proc. IEEE*, vol. 12, pp. 1645–1658, 1987.

[15] DLMS, “DLMS User Association: COSEM Identification System and Interface Classes (2010).” Available online : https://dlms.com/information/whatisdlmscosem/index.html (accecssd on 28 January 2019).

[16] “Modobus..” Available online : http://www.modbus.org/(accecssd on 28 January 2019).

[17] ZigBee Alliance, “ZigBee.” Available online : http://www.zigbee.org (accessed on 28 January 2019).

[18] The Z-Wave Alliance, “Z-Wave.” Available online : https://www.z-wave.com/ (accessed on 28 January 2019).

[19] C. Gomez, J. Oller, and J. Paradells, “Overview and Evaluation of Bluetooth Low Energy: An Emerging Low-Power Wireless Technology,” *Sensors*, vol. 12, no. 9, pp. 11734–11753, 2012.

[20] The LoRa Alliance, “White Paper: A Technical Overview of Lora and Lorawan.” Available online : https://lora-alliance.org/resource-hub/what-lorawantm (accessed on 28 January 2019), 2015.

[21] SIGFOX S.A., “Sigfox Technical Overview.” Available online : https://www.disk91.com/wp-content/uploads/2017/05/4967675830228422064.pdf (accessed on 28 January 2019).

[22] B. Molina, C. Palau, G. Fortino, A. Guerrieri, and C. Savaglio, “Empowering smart cities through interoperable Sensor Network Enablers.,” *In Proceedings of the 2014 IEEE International Conference on Systms, Man and Cybernetics,*, pp. 5–8, October 2014.

[23] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, “Internet of Things (IoT): A vision, architectural elements, and future directions.,” *Future Gener. Comput. Syst.*, vol. 29, pp. 1645–1660, 2013.

[24] IEC, “IEC 62591. Ed 1.0: Industrial Communication Networks—Wireless Communication Network and Communication profiles–WirelessHART,” *IEC 62591*, 2010.

[25] ISA, “ISA100, Wireless Systems for Automation.” Available online : http://www.isa.org/Community/SP100WirelessSystemsforAutomation (Accessed on 28 January 2019).

[26] Institute of Electrical and Electronics Engineers (IEEE), “802. 15.4e-2012: IEEE Standard for Local and Metropolitan Area Networks-Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs) Amendment 1: MAC Sublayer IEEE Std,” *IEEE*, 2012.

[27] T. Watteyne, J. Weiss, L. Doherty, and J. Simon, “Industrial IEEE802.15.4e networks Performance and trade-offs,” *2015 IEEE International Conference on Communications (ICC)*, 2015.

[28] M. Nixon, “A comparison of WirelessHART and ISA100.11a.” Available online : https://www.emerson.com/documents/automation/white-paper-a-comparison-of-wirelesshart-isa100-11a-en-42598.pdf (accessed on 10 March 2019).

[29] “SmartMesh WirelessHART Application Notes.” Available online : http://cds.linear.com/docs/en/application-note/SmartMesh_WirelessHART_Application_Notes.pdf (accessed on 28 January 2019).

[30] H. Kurunathan, R. Severino, A. Koubaa, and E. Tovar, “Worst-case bound analysis for the time-critical MAC behaviors of IEEE 802.15.4e,” *2017 IEEE 13th International Workshop on Factory Communication Systems (WFCS)*, 2017.

[31] T. Lennvall, S. Svensson, and F. Hekland, “A comparison of WirelessHART and ZigBee for industrial applications,” *2008 IEEE International Workshop on Factory Communication Systems*, October 2008.

[32] TEPCO, “Basic Concept for Smart Meter Specification based on RFC.” Available online : http://www.tepco.co.jp/en/corpinfo/procure/invited/pdf/2_concept-e.pdf (accessed on 28 January 2019).

[33] N. Q. Dinh and D.-S. Kim, “Performance evaluation of priority CSMA-CA mechanism on ISA100. 11a wireless network,” *Computer Standards & Interfaces*, vol. 34, no. 1, pp. 117–123, 2012.

[34] Y. Igarashi, Y. Matsuura, M. Koizumi, and N. Wakamiya, “Priority-Based Dynamic Multichannel Transmission Scheme for Industrial Wireless Networks.,” *Wirel. Commun. Mob. Comput. 2017*, 2017.

[35] Z. Zhiming, P. Martin, A. Jones, I. Taylor, V. Stankovski, G. Salado, G. Suciu, A. Ulisses, and C. de Laat, “Developing, Provisioning and Controlling Time Critical Applications in Cloud.,” *In Proceedings of the European Conference on Service-Oriented and Cloud Computing*, pp. 169–174, 2017.

[36] H. Jetmir, E. Poorter, I. Moerman, and J. Hoebeke, “A Survey of LoRaWAN for IoT: From Technology to Application.,” *Sensors*, 2018.

[37] S. Watt, H. Loehner, S. Achanta, A. Kivi, and B. Rowland, “Extending SCADA Networks Using Wireless Communications.,” *In Proceedings of the Power Energy Automation Conference*, 2015.

[38] D. F. Ramírez and S. Céspedes, “Routing in Neighborhood Area Networks: A survey in the context of AMI communications,” *Journal of Network and Computer Applications*, vol. 55, pp. 68–80, 2015.

[39] M. Mitchell, “An Introduction to Genetic Algorithms,” *MIT Press: Cambridge, MA, USA*, 2016.

[40] “SmartMesh IP.” Available online : http://www.analog.com/en/products/rf-microwave/wireless-sensor-networks/smartmesh-ip.html (accecssd on 28 January 2019).

[41] P. Raja and G. Noubir, "Static and dynamic polling mechanisms for fieldbus networks.," *ACM SIGOPS Oper. Syst. Rev.*, vol. 27, pp. 34–45, 1993.

[42] S. Cavalieri, S. Monforte, A. Corsaro, and G. Scapellato, "Multicycle Polling Scheduling Algorithms for FieldBus Networks.," *Real-Time Syst*, vol. 25, pp. 157–185, 2003.

[43] W. Wang, Y. Wang, X. Li, W. Song, and O. Frieder, "Efficient interference-aware TDMA link scheduling for static wireless networks.," *In Proceedings of the 12th Annual International Conference on Mobile Computing and Networking*, pp. 262–273, 2006.

[44] A. Saifullah, Y. Xu, C. Lu, and Y. Chen, "Real-Time Scheduling for WirelessHART Networks.," *In Proceedings of RTSS '10 Proceedings of the 2010 31st IEEE Real-Time Systems Symposium*, pp. 150–159, 2010.

[45] O. Chipara, C. Lu, and G.-C. Roman, "Real-Time Query Scheduling for Wireless Sensor Networks.," *In Proceedings of the 28th IEEE International Real-Time Systems Symposium*, 2007.

[46] Y. Zhijun, L. Zhu, H. Ding, and Z. Guan, "A priority-based parallel schedule polling mac for wireless sensor networks.," *J. Commun.*, vol. 11, pp. 792–797, 2016.

[47] S. Pogorilyy and R. Bilous, "Genetic Algorithm for Network Performance Optimization.," *Proc. IAM*, pp. 121–128, 2012.

[48] X. Sun, J. Wang, W. Wu, and W. Liu, "Genetic Algorithm for Optimizing Routing Design and Fleet Allocation of Freeway Service Overlapping Patrol.," *Sustainability*, vol. 10, no. 4120, 2018.

[49] J. Yi, T. Clausen, and Y. Igarashi, "Evaluation of routing protocol for low power and lossy networks: LOADng and RPL.," *In Proceedings of the IEEE Conference on Wireless Sensors*, 2013.

[50] A. Abuagoub, "An Overview of Industrial Wireless Sensor Networks.," *Int. J. Comput. Sci. Inf. Technol. Res.*, vol. 4, pp. 68–80, 2016.

[51] Y. Li, C. S. Chen, Y. qiong Song, and Z. Wang, "Real-time QoS support in wireless sensor networks: a survey," *7th IFAC International Conference on Fieldbuses and Networks in Industrial & Embedded Systems*, 2007.

[52] D. Vassis and G. Kormentza, "Throughput Analysis for IEEE 802.11 Ad Hoc Networks Under the Hidden Terminal Problem," *Consumer Communications and Networking Conference*, 2006.

[53] G. Bianch, "Performance Analysis of the IEEE 802.11 Distributed Coordination Function," *IEEE Journal on Selected Areas of Communication*, vol. 18(3), pp. 535–547, 2000.

[54] B. Shrestha, E. Hossain, and K. W. Choi, "Distributed and Centralized Hybrid CSMA/CA-TDMA Schemes for Single-Hop Wireless Networks," *IEEE Trans. Wireless Commun*, vol. vol. 13, pp. pp. 4050–4065, Jul 2014.

[55] O. D. Saputra and S. Y. Shin, "Real-time based Superframe for ISA100. 11a in Wireless Industrial Network," *Journal of Communication and Computer*, vol. 12, pp. 28–32, 2015.

[56] I. Rhee, A. Warrier, M. Aia, J. Min, and M. L. Sichitiu, "Z-MAC: A Hybrid MAC for Wireless Sensor Networks," *IEEE/ACM Transactions on Networking*, 2008.

[57] R. Costa, P. Portugal, F. Vasques, and R. Moraes, "A TDMA-based mechanism for real-time communication in IEEE 802.11e networks," *Emerging Technologies and Factory Automation (ETFA), 2010 IEEE Conference*, 2010.

[58] J. Yi, T. Clausen, and Y. Igarashi, "Evaluation of Routing Protocol for Low Power and Lossy Networks: LOADng and RPL," *2013 IEEE Conference on Wireless Sensors*, 2013.

[59] B. Sniderman, M. Mahto, and M. J. Cotteleer, "Industry 4.0 and manufacturing ecosystems - Exploring the world of connected enterprises." Available online : https://www2.deloitte.com/content/dam/insights/us/articles/manufacturing-ecosystems-exploring-world-connected-enterprises/DUP_2898_Industry4.0ManufacturingEcosystems.pdf(accessed on 16 February 2019), 2016.

[60] L. D. Xu, E. L. Xu, and L. Li, "Industry 4.0: state of the art and future trends," *Intrenational Journal of Publication Research*, vol. 56, no. 8, 2018.

[61] "Exploring new real-time communication technologies for industrial automation." Available online : https://www.pc-control.net/pdf/032018/technology/pcc_0318_real-time-communication_e.pdf (accessed on 10 March 2019).