Industrial Paper

Firmware Distribution with Erasure Coding for IoT Devices

Takenori Sumi[†*], Yukimasa Nagai[†], Jianlin Guo[‡] and Hiroshi Mineno[*]

[†]Information Technology R&D Center, Mitsubishi Electric Corporation, Japan
[*]Graduate School of Science and Technology, Shizuoka University, Japan
[‡] Mitsubishi Electric Research Laboratories, USA
Sumi.Takenori@dc.MitsubishiElectric.co.jp

Abstract – Due to the increase in the number of IoT devices connected to the internet, 920 MHz frequency bands for wireless communication systems are attracting attention for various IoT applications, e.g., environmental monitoring, smart metering, process monitoring & control. With wireless communication systems on 920 MHz having the features of long distance, low rate and low power consumption, a huge number of IoT devices distributed in wide area can be connected to communication networks. When distributing the same data to IoT devices such as firmware distribution during operation, improving the efficiency of distribution method becomes an issue. We propose a new firmware distribution method with erasure coding for IoT devices. Our computer simulation result shows that proposed method improves the efficiency of distribution by 1.7 times for single-hop networks and by 1.5 times for multi-hop networks compared with conventional method and achieves higher spectrum efficiency.

Keywords: IoT, Firmware Distribution, Erasure Code

1 INTRODUCTION

In addition to smart phone, laptop and tablet connecting to the internet, IoT devices such as sensors are getting connect to the internet. Due to the increase in the number of IoT devices connected to the internet, 920 MHz frequency bands for wireless communication systems are attracting attention for various IoT applications, e.g., environmental monitoring application for location, temperature, humidity and water level, smart metering application, process monitoring & control application. Wireless communication systems on 920 MHz (Sub-1 GHz) have the features of long distance, low rate (several 10 kbps – 100 kbps) and low power consumption for conventional standards, e.g., IEEE 802.15.4g [1], LoRaWAN [2], SigFox [3], but the higher data rate up to several Mbps is also considered for applications such as infrastructure monitoring, surveillance camera in IEEE 802.11ah [4][5], marketed as Wi-Fi HaLow. For long life IoT devices, the firmware update for a large number of IoT devices is also considered as new IoT application. Since 920 MHz has the features of long distance from several 100 m to several km, a large number of IoT devices are deployed in the area. Thus, firmware distribution for IoT devices using 920 MHz with narrow band is a challenge. Efficient firmware distribution method for a large number of IoT devices should be considered. In this paper, we focus on 920MHz for IoT applications and propose the new firmware distribution method using erasure coding to achieve higher efficiency for limited radio frequency. Furthermore, since firmware distribution for IoT devices does not require real-time update, the hardware performance such as memory and CPU power for using erasure coding is not a major issue. Generally, updates are expected to occur within tens of minutes to hours for IoT devices.

The rest of this paper is organized as follows. Section II presents related work. Section III describes the proposed firmware distribution with erasure coding. Section IV shows the simulation architecture and results for various conditions. Finally, we conclude our paper in Section V.

2 RELATED WORK

There are existing researches for wireless communications using 920 MHz and firmware distribution. Since 920 MHz is narrow bands compared to 2.4 GHz and 5 GHz for ISM band, special regulation for “10 % transmission duty cycle” and “longer backoff mechanisms” are applied in Japan.

Throughput performance has been demonstrated in [6] and [7], which focus on the PHY and MAC protocol enhancement for higher-throughput, protocol efficiency and delay via simulation and measurement result using prototypes. For example, D. Hotta et al. introduce the performance of multi-hop routing construction using Wi-SUN FAN (Field Area Network) prototypes based on IEEE 802.15.4g FSK PHY [8].

Japanese standard ARIB STA-T108 (20 mW, unlicensed) defines the use of IEEE 802.15.4g system from 920.5 – 928.1MHz (7.6 MHz bandwidth), but the ARIB STA-T107 (250 mW, passive system) and the ARIB STD-T108 (250mW, licensed/registered) also define operation from 920.5 – 923.5 MHz (3.0 MHz). Therefore, 923.5 – 928.1 MHz (4.6 MHz bandwidth) is the only reasonable frequency band for IEEE 802.15.4g applications in the unlicensed spectrum. IEEE 802.15.4g is regulated to operate over 200 kHz bandwidth channel in the Sub-1 GHz band. Even low duty cycle constraint applied in the Sub-1 GHz band, e.g., Japanese and European standard allow up to 10% transmission duty cycle [9]-[12] for the number of IoT Devices increased with various standards. Therefore, ensuring higher efficiency for spectrum use in the Sub-1 GHz is clearly important.

In IoT applications, such as environmental monitoring applications for location, temperature, humidity, and water quantity, as well as smart meter applications, network topology is constructed around sensor information collection nodes. IEEE 802.15.4g / Wi-SUN uses RPL (IPv6 Routing Protocol for Low-Power and Lossy Networks) to construct network topology. In RPL, network topology is constructed by DODAG (Destination Oriented Directed Acyclic Graph). In DODAG, each node only has a route to the root, as shown in Figure 1. Routes are constructed by sending and receiving

DODAG Information Object (DIO) messages to and from neighboring nodes. Thus, each node can know the Rank and other information of its neighbors.

Considering firmware distribution, firmware is sent from Root in Figure 1 to each node. When distributing the same firmware data to a large number of nodes, it is more efficient to distribute the firmware data using multicast instead of sending it to each node via unicast. However, there is an issue of bandwidth congestion if the firmware data is transmitted by flooding, in which all nodes transfer the firmware data. Therefore, Multi-Point Relay (MPR) has been proposed as a more efficient transfer method [14]. According to [14], finding the smallest relay set is NP hard, but a heuristic relay node select is proposed. In the heuristic method, each node selects one-hop neighbors as a relay node to cover all two-hop neighbors. This allows for efficient distribution by limiting the number of nodes to which firmware data is transferred. Furthermore, to improve the reliability of delivery, a method using LT code, one of erasure codes, in MPR has been proposed [15]-[17]. These schemes improve delivery efficiency and reliability by network coding with LT codes on the MPR-constructed transfer path.

However, MPR requires obtaining information on two-hop nodes in addition to one-hop nodes. In RPL, only the information of neighbor nodes is notified, and the information of two-hop nodes is not necessary to notify sensor data to the root. Furthermore, as mentioned earlier, the 920 MHz band is narrow and the 10% duty cycle in Japan and Europe makes it difficult to inform additional information among nodes for firmware distribution. As the propagation environment fluctuates, the connection relationships of the nodes also change, and the connection relationship information must be updated among all nodes.

Therefore, this paper proposes a scheme to determine the forwarding node for firmware distribution without prior downward route construction, and a scheme to improve distribution efficiency with erasure coding.

3 PROPOSED METHOD

3.1 Challenges in Firmware Distribution

Firmware distribution to IoT devices requires sending the same firmware data to many IoT devices via wireless communication. Firmware distribution using unicast is inefficient because the same firmware data is sent to each device individually. Therefore, it is considered more efficient to distribute to a large number of IoT devices using broadcast. However, it is difficult to deliver all firmware data to all IoT devices in a single communication because wireless communication suffers from packet errors due to low received signal power and/or interference from other wireless systems. As shown in Figure 2, packet errors occur randomly in IoT devices, requiring large number of packets to be retransmitted. Therefore, even when broadcasts are used for firmware distribution, there are issues with distribution efficiency. In this paper, we propose a method to reduce the number of transmitted packets in firmware distribution by using erasure coding.

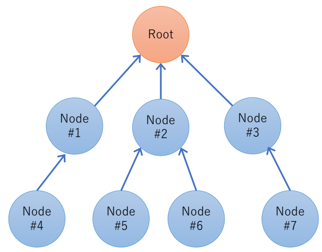

Figure 1: Routing Construction using RPL

3.2 Erasure Code

In this paper, RC QC-LDPC (Rate Compatible Quasi-Cyclic-LDPC) [18] is used as an erasure code for firmware distribution. In general, erasure encoding are used for distributed storage. In distributed storage, it is sufficient to be able to recover the loss of some data out of N, e.g., when some of the multiple storage devices fails. However, in wireless communication, it is necessary to cope with packet errors for some long time in the time axis direction because of bursty packet errors. RC QC-LDPC can handle bursty packet errors because it can take a large number of information packets and redundant packets, as described below. In the related studies described in Chapter 2, network coding was applied using LT code based on the forwarding path constructed by MPR. In this paper, however, standard erasure coding is used instead of network coding because the downward forwarding path is not constructed in advance. As shown in Figure 3, RC QC-LDPC generates redundant packets by erasure coding information packets that divide the data to be transmitted into packets of a certain data length. The data length of information packets and redundant packets are the same. The number of information packets, K, is a multiple of 36, and the number of redundant packets, M, is equal to K or twice as large. RC QC-LDPC can attempt to decode the transmitted data if the total number of received information packets and redundant packets is K or more. Even if some of the received information packets are missing, the transmitted data can be recovered using the redundant packets. Figure 4 shows the relationship between "coding redundancy rate" and decoding success rate. Here, the "coding redundancy rate" is defined as the sum of the number of information packets K_{rx} and the number of redundant packets M_{rx} at decoding and the original number of information packets K, using the following formula (1).

$$Coding\ Redundancy\ Rate = \frac{K_{rx}+M_{rx}}{K} \quad (1)$$

Figure 4 shows that RC QC-LDPC does not ensure successful decoding even when the redundancy is greater than 1. The higher the number of information packets K, the higher the decoding success rate for the coding redundancy rate. However, the larger the number of information packets K, the more memory and CPU resources are required for the decoding process.

3.3 Firmware Distribution with Erasure Code

Before describing firmware distribution in multi-hop networks, we describe how loss-correcting codes can be applied to firmware distribution in single-hop networks. In the proposed firmware distribution method, as shown in Figure 5, the transmitter first erasure codes the firmware data (K information packets in the figure) to generate M redundancy packets ((1) in the figure). If firmware data is large, the transmitter performs erasure coding multiple times.

Then, the Transmitter sends up to a total of K information packets and redundant packets ((2) in the figure). Figure 7 shows the format of information packets and redundant packets. The Information packet and the redundant packet contain packet type indicating whether it is an information packet or redundant packet, sequence number, packet index, and payload. The sequence number is incremented each time the transmitting station encodes firmware data and is used to identify which data the response is to. The packet index indicates the index of information packets or redundant packets. After K packets have been sent, depending on the number of packet errors in the IoT device, additional information packets and redundant packets are sent until the IoT device can decode the firmware data ((3) in the figure). Here, the additional packets to be sent are those not sent in (2) in the figure. The IoT device uses the received K or more packets to decode and recover the firmware data ((4) in the figure). As described in section 3.2, RC QC-LDPC may fail in decoding. If decoding fails, additional packets are sent from the transmitter and the IoT device performs the decoding process again.

To implement the proposed method, IoT devices need to inform the transmitter which packets are missing. In the proposed scheme, IoT devices notify the transmitter of a NACK (Negative Acknowledgment) packet and an ACK (Acknowledgment) packet in the sequence shown in Figure 6. The NACK packet is used to notify which packets are missing, the ACK packet is used to notify that decoding was successful. The packet formats of NACK and ACK packets are shown in Figure 8. Multicast UDP packets is used to send NACK and ACK packets, and Figure 8 shows only the UDP payload. The NACK packet contains packet type indicating whether it is NACK or ACK, destination node identifier, sequence number and a bitmap indicating packet loss. The length of the bitmap is the sum of the number of information packets K and the number of redundant packets M. The ACK packet contains packet type, destination node identifier and sequence number.

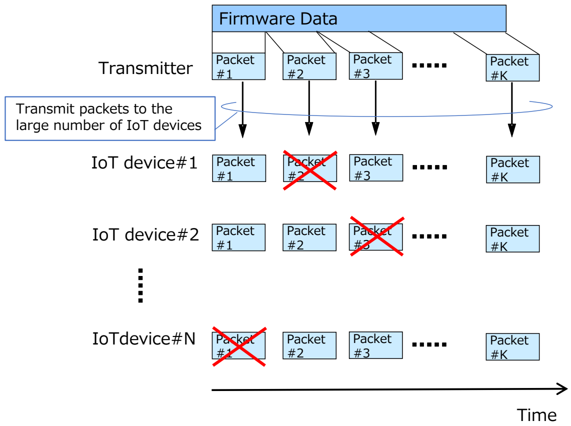

Figure 2: Firmware Distribution for IoT Devices

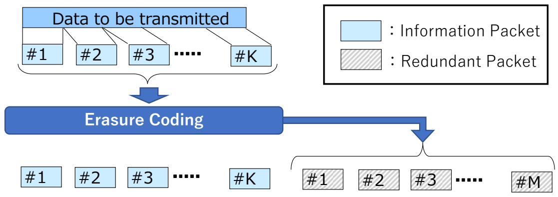

Figure 3: Erasure Coding

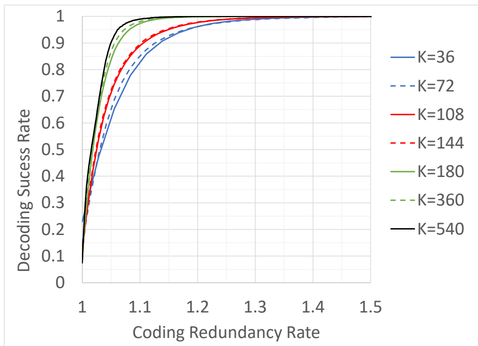

Figure 4: Redundancy Rate vs. Decoding Success Rate

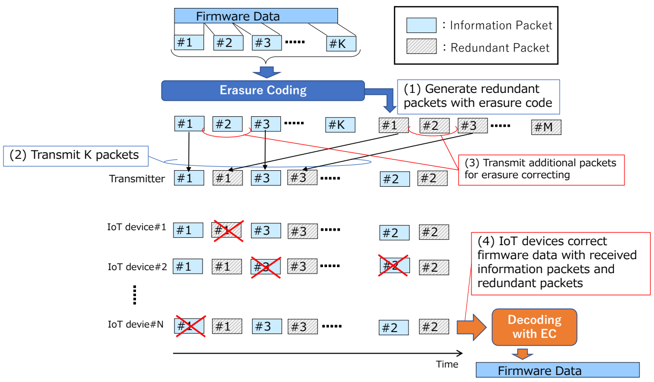

Figure 5: Proposed firmware distribution with EC

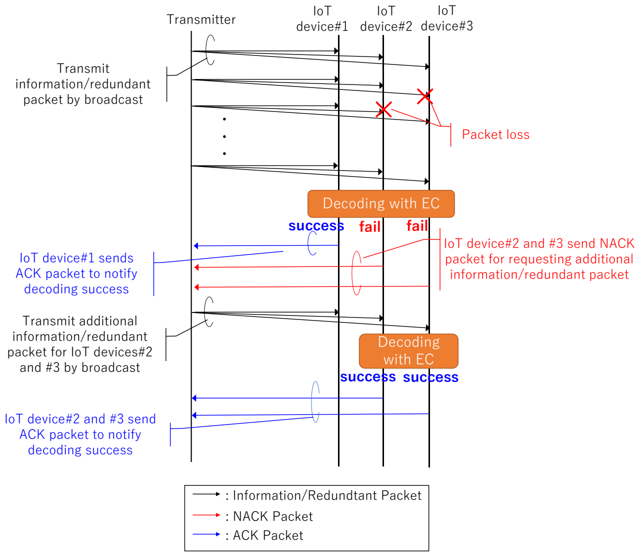

Figure 6: Packet sequence of the proposed method

1 Byte	1 Byte	2 Byte	2 Byte	
Packet type	Hop count	Sequence number	Packet index	Payload

Figure 7: Packet format of information packet and redundant packet

NACK Packet

1 Byte	2 Byte	2 Byte	(K + M)/8 Byte
Packet type	Destination	Sequence number	Packet loss bitmap 1: Received, 0: Packet lost

ACK Packet

1 Byte	2 Byte	2 Byte
Packet type	Destination	Sequence number

Figure 8: Packet formats of NACK and ACK packet

3.4 Firmware Distribution in Multi-hop Networks

Section 3.3 described the firmware distribution method in single-hop networks. In this section, we describe the firmware distribution method in multi-hop networks.

As discussed in Chapter 2, in IoT applications, each node only needs to know the path to the root node in order to collect sensor data to the root node. However, firmware distribution is distributed from the root node via a downward path. To improve the efficiency of distribution, the number of forwarding nodes needs to be reduced, as in MPR. The proposed method reduces the number of forwarding nodes by using ACK or NACK packets used to confirm the delivery of firmware distribution. In the multi-hop network shown in Figure 9, firmware data can be delivered to all nodes if only Node#1 and Node#3 perform forwarding. The red dotted lines in the figure indicate possible communication paths. In the proposed scheme, firmware data is divided into packets of a certain size and transmitted with erasure coding for each K packets. After receiving the first K packets, each node sends ACK or NACK to the source node by multicast. Here, if Node#4 sends an ACK or NACK packet to Node#1 first, Node#5 can receive the ACK or NACK packet sent by Node#4. If Node#5 has not yet sent an ACK or NACK packet, Node#5 can make Node#1 the destination for ACK or NACK packets, so that the destination for ACK or NACK notification is the same as Node#4. After each node sends its first K packet, only the node that receives an ACK or NACK packet from the next hop node will perform the transfer, thereby reducing the number of nodes transferring firmware data in a multi-hop network and improving delivery efficiency. Furthermore, the number of hops in the information packet or redundant packet of each node and the reception power of the packet can be used to determine the candidate destination for ACK or NACK packet notification, so that the network topology can be maintained and a node with stable reception power can be selected as the ACK or NACK packet destination. In addition, the timing of ACK or NACK notification is shifted for each node by a random number to consolidate the ACK or NACK notification destinations.

4 SIMULATION EVALUATION

We evaluate the proposed method using computer simulation. Table 1 shows simulation parameters. In the computer simulation, IEEE 802.15.4g, which is used in IoT devices in the 920 MHz band, was used as the wireless communication method and evaluated in an environment where IoT devices are connected in a star network or mesh network from a transmitter. IEEE 802.15.4g is a communication method used for smart utility networks [19], and according to TTC JJ-300.10 [20], it is also used for smart meters in Japan. In this simulation, the parameter values of MAC and PHY of IEEE 802.15.4g and JJ-300.10 are used. Assuming environmental monitoring using smart sensors, the number of IoT devices is assumed to range from 1 to 100 in consideration of installation on office (indoor use case) and university campus (outdoor use case) for single-hop network, and from 100 to 1000 in consideration of installation on suburban area for multi-hop network. The firmware data is divided into 248 Bytes each and multicast to the IoT device as the payload of information packets. RC QC-LDPC was used as erasure code, and the number of information packets K and redundancy packets M were set to 36 - 540. In the computer simulation, the conventional firmware distribution without erasure code also broadcasts the firmware data to the IoT devices by dividing it into 248 Bytes each, as in the proposed method. Then, after the K-packet transmission is finished, the IoT devices send ACK or NACK to the transmitter to notify whether it needs to retransmit data. This is to ensure that the opportunities to send ACK and NACK packets are the same for the conventional and proposed methods. In the conventional method, the number of redundancy packets M is 0 because erasure coding is not performed for the firmware data.

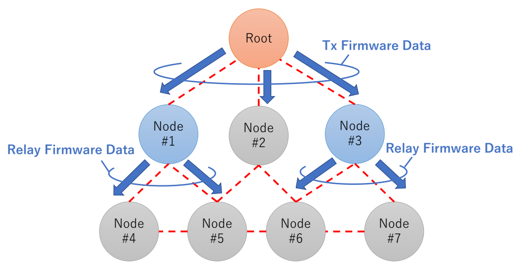

Figure 9: Firmware distribution in multi-hop networks

Table 1: Simulation parameters

Firmware distribution parameters	Value
The number of IoT devices	Single-hop: 1 – 100 Multi-hop : 100 - 1000
Erasure code	Rate-Compatible QC-LDPC
K, the number of information packets	36, 72, 108, 144, 180, 360, 540
M, the number of redundant packets	Conventional method: 0 Proposed method: same as K
Firmware distribution packet length	248 Byte
Firmware distribution ACK packet	1 Byte
Firmware distribution NACK packet	1+(K+M)/8 Byte
6LowPAN and UDP header	53 Byte
PHY/MAC	IEEE 802.15.4g
MAC parameters	Value
macMinBE	5
macMaxBE	8
LIFS	1000 us
aUnitBackoffPeriod	1130 us
phyCcaDuration	130 us
aTurnaroundTime	1000 us
tack	1000 us
MAC header size	11 Byte
FCS	2 Byte
PHY parameters	Value
Data rate	100 kbps
Modulation	2-FSK
Modulation index	1.0
Frequency	923.7 MHz
Channel spacing	400 kHz
Propagation Model	SEAMCAT Extended Hata Model (Suburban)
Antenna height	1.5 m

4.1 Single-hop Network Evaluation

First, the relationship between the number of information packets K and effective throughput is shown in Figure 10, which shows the simulation results when firmware is distributed to 20 IoT devices and PER is 10%. Since a PER of 10% or less is often set for wireless communication systems to take operations into account [19], the simulations for single-hop network in this paper are based on an evaluation at a PER of 10%. According to the figure, the throughput of the proposed method with erasure coding is higher than that of the conventional method without erasure coding, regardless of the value of K. As the value of K increases, the effective throughput of the proposed method increases. As mentioned in section 3.2, this is because the larger K is, the higher decoding success rate at the same redundancy rate. However, the memory and decoding processing load increases as K increases. The relationship between K ant the memory size required for erasure coding is shown in Figure 11. The required memory size increases linearly with K. On the other hand, Figure 10 shows that throughput improvement slows down when K is greater than 180. Furthermore, if K is greater than 180, the required memory size exceeds 100 KB. Therefore, for embedded devices with limited memory size, it is better to set K to 180. The subsequent simulation evaluation will be performed for the case where the number of information packets K is 180, where the effective throughput increase rate by the proposed method is relatively high.

Next, Figure 12 shows the simulation result on the relationship between PER and effective throughput when firmware is distributed to 20 IoT devices with the number of information packets K set to 180. The figure shows that if the PER is 1% or higher, the effective throughput is higher with the proposed method than with the conventional method. However, the effective throughput at a PER of 0% with no packet errors is approximately 40 kbps for the conventional method, but 33 kbps for the proposed method, which is lower than the conventional method. As mentioned in Section 3.2, this is because RC QC-LDPC codes may not succeed in decoding even if a total of K or more information packets and redundancy packets are received. Additional packets need to be sent from the transmitter to the IoT devices. Since packet errors occur in wireless communication systems, the proposed method can be applied to improve the effective throughput in an environment with a PER < 10%, where wireless communication systems are normally operated.

Finally, the relationship between the number of IoT devices and effective throughput is shown in Figure 13, which shows simulation results when the number of information packets K is 180 and PER is 10%. The figure shows that the effective throughput of the proposed method is higher than that of the conventional method when firmware distribution is performed to two or more IoT devices. In the case of a single IoT device, the effective throughput is lower than the conventional method due to the significant impact of the possibility of unsuccessful decoding even if the number of received packets is K or more. For 20 IoT devices, it is 1.60 times the effective throughput of the conventional method; for 50 devices, it is 1.73 times; and for 100 devices, it is 1.76 times. However, as the number of IoT devices increases, the effective throughput decreases: 29.7 kbps for 20 IoT devices, 24.8 kbps for 50 devices, and 19.8 kbps for 100 devices. This is likely due to an increase in NACK packets.

The transmission duty cycle of the transmitter in the simulation of Figure 13 is shown in Figure 14. As mentioned in Chapter 2, Japanese and European standard allow up to 10% transmission duty cycle. Figure 14 shows that regardless of the number of IoT devices, the transmission duty cycle of the transmitter exceeds 10% for both the proposed and conventional methods. The transmission duty cycle of the proposed method is lower than that of the conventional method, and it tends to be lower when the number of IoT devices is larger. Therefore, in Japan and Europe, when the transmission duty cycle is 10% or less, the difference between the effective throughput of the proposed method and the conventional method is expected to widen. The firmware delivery time to 100 IoT devices is shown in Figure 15 considering that IoT devices are communicating in normal operation, i.e., notifying sensor data, the firmware distribution time was evaluated with the firmware delivery bandwidth usage rate as 1%, 5%, and 10%. The difference in firmware delivery time when erasure coding is applied or not applied is about 8.5 hours when the firmware data size is 256 KB, 1.7 hours when the bandwidth utilization is 1%, 1.7 hours when 5%, and 0.85 hours when 10%. However, when the firmware data size increases, for example, to 1024 KB, the difference is 34 hours at 1%, 13.6 hours at 5%, and 6.8 hours at 10%. If the normal state communication of the IoT device is bandwidth-hungry and the firmware data size is 1024 MB or larger, the proposed method can improve the firmware delivery time by half a day to a day or more.

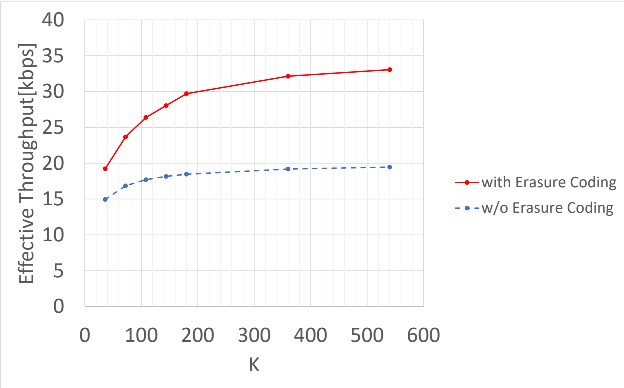

Figure 10: K vs. Throughput (PER: 10%, 20 devices)

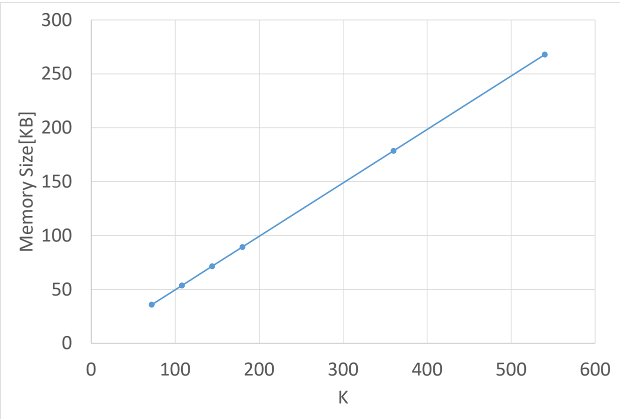

Figure 11: K vs. Memory Size

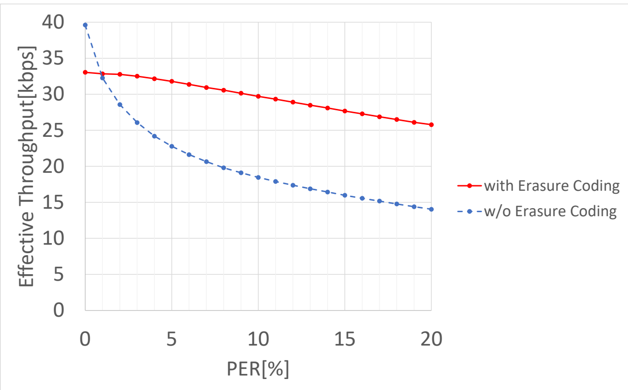

Figure 12: PER vs. Throughput (K: 180, 20 devices)

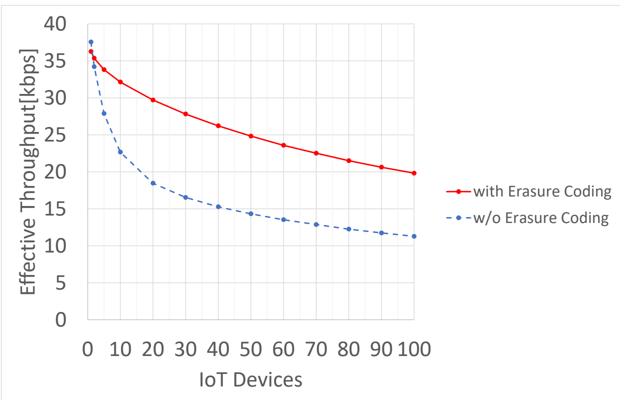

Figure 13: The number of IoT devices vs. Throughput (K: 180, PER: 10%)

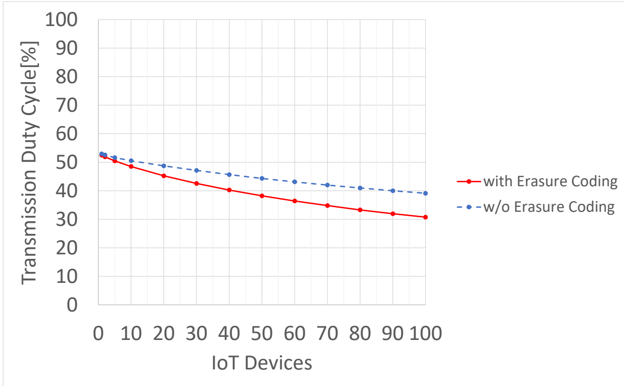

Figure 14: The number of IoT devices vs. Duty Cycle (K: 180, PER: 10%)

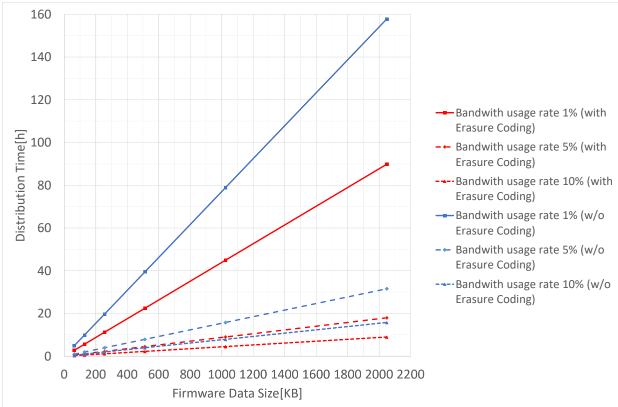

Figure 15: Firmware Data Size vs. Distribution Time (K: 180, PER: 10%, 100 devices)

4.2 Multi-hop Network Evaluation

For the evaluation of the multi-hop network, firmware data transmitter is placed at the center of a circle with a radius of 500 m, and 100 to 1000 nodes are randomly placed within the circle. Suburban of SEAMCAT Extended Hata Model was used as the propagation model and the received power was calculated. Figure 16 shows the relationship between distance and received power. With 2-FSK, the modulation scheme used in the simulation, reception is stable down to a minimum of -90 dBm, which is about 89 m from the figure. In MPR, 1-hop and 2-hop nodes were calculated with an upper limit of 80 m distance to account for packet errors due to interference. For the number of information packets K of erasure coding, 180 was used based on the single-hop evaluation. In the single-hop evaluation, since only the transmitter sends firmware data packets, frame collisions did not occur, so the packet error rate was set and packet errors were generated randomly, but in multi-hop, multiple nodes transfer firmware data packets, so frame collisions occur. Therefore, the packet error rate was not set, and the evaluation was based on packet loss due to frame collisions.

First, Figure 17 shows an example of terminal placement under the MPR method and the proposed method in which the relay node is determined by ACK or NACK packets for firmware distribution. The figure shows an example with 500 nodes. The black dotted line connecting each node indicates that firmware data is being transferred by relay. In the proposed method, 244 out of 500 nodes are set as relay nodes, while in MPR, 237 nodes are set as relay nodes, which are almost equal in number. It can be seen that the proposed method can set up the same number of relay nodes as MPR without having any routing information other than the uplink route by RPL.

Next, Effective throughput is shown in Figure 18 for the proposed method, MPR, and flooding with and without Erasure coding, respectively. The figure shows that the proposed method and MPR achieve almost the same effective throughput. Compared to the flooding case, the proposed method improves effective throughput by about 20-25%. In addition, the effective throughput is improved by 52-57% when Erasure coding is used in the proposed method compared to the case where Erasure coding is not used in flooding. The average and maximum number of hops are shown in Figure 19. The figure shows that for 400 or more IoT devices, the average and maximum hop counts are almost equal. This means that the decrease in effective throughput is not caused by an increase in the number of hops, but by an increase in the number of terminals.

Finally, Figure 20 shows the packet transmission redundancy. The redundancy of packet transmission is expressed by the following formula (2).

$$Packet\ Transmission\ Redundancy = \frac{\sum_{n=1}^{N} P_{FWn}+P_{ACKn}+P_{NACKn}}{P_{FW}KN} \quad (2)$$

Here, P_{FWn} is the total size of firmware data packets (information packets or redundant packets) sent by node#n, P_{ACKn} is the total size of ACK packets sent by node#n, P_{NACKn} is the total size of NACK packets sent by node#n, P_{FW} is the size of a firmware data packet, K is the number of information packets for erasure coding, and N is the number of nodes. In other words, the redundancy of packet transmission is calculated based on the size of the transmitted data when all nodes except the firmware data transmission node transmit the firmware data once. The figure shows that the proposed method and MPR have almost the same level of redundancy, ranging from 0.5 to 0.6. In the case of flooding, the redundancy is about 1.1 to 1.2, indicating that the proposed method can deliver firmware data in about half the transmission time compared to flooding.

5 CONCLUSION

We proposed the new firmware distribution method using erasure coding to achieve higher efficiency for limited radio frequency and the method for selecting relay nodes in a multi-hop network without additional downward routing information. The performance of the proposed method was evaluated by computer simulation and compared to the effective throughput of conventional firmware distribution methods. For single-hop network, by applying the proposed method to firmware distribution, the effective throughput was found to be 1.60 times higher than that with the conventional method when there are 20 IoT devices receiving firmware data, 1.73 times higher when there are 50 devices, and 1.76 times higher when there are 100 devices. For multi-hop network, by applying the proposed method to firmware distribution, effective throughput was found to be about 1.5 times. The memory size required for encoding is about 100 KB for the erasure encoding scheme used in this paper, which is considered to be operable on mid- to high-performance models of embedded CPUs. However, we plan to investigate memory-saving erasure coding schemes that can operate on even lower-performance models in the future. Furthermore, in this paper, we evaluated the performance with computer simulation. Our future work is conduct evaluations using actual equipment taking into account the amount of memory and other factors.

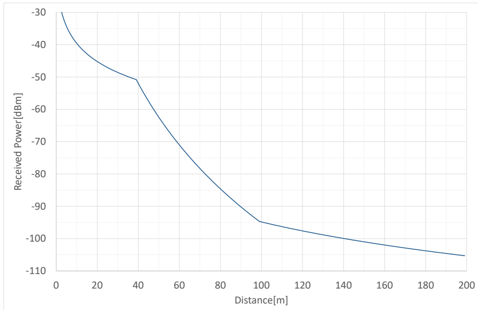

Figure 16: Distance vs. Received power

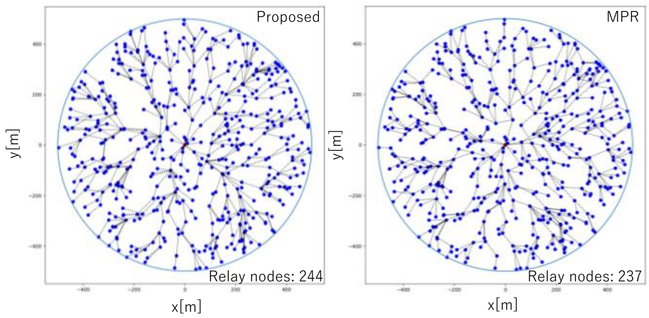

Figure 17: Example of relay node selection (500 nodes)

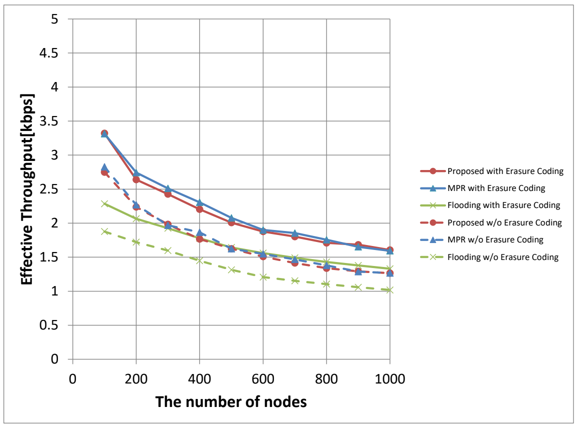

Figure 18: Effective throughput (Multi-hop)

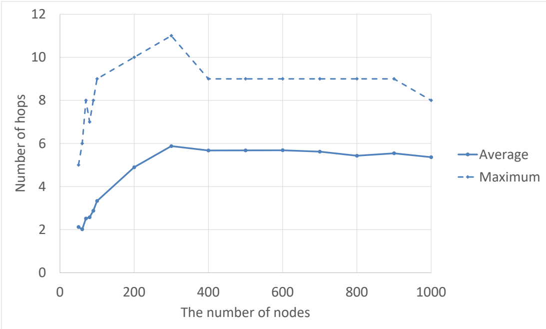

Figure 19: Average and maximum hops

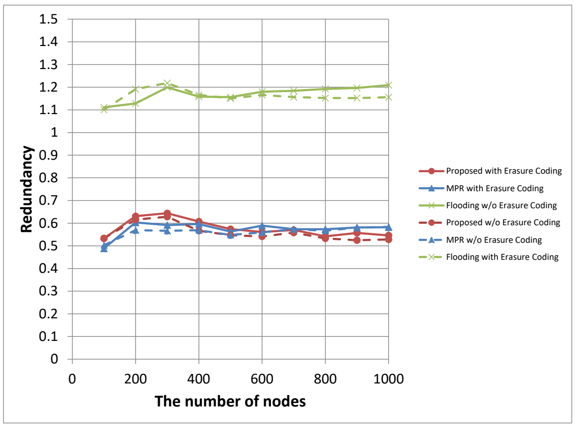

Figure 20: Packet transmission redundancy (Multi-hop)

REFERENCES

[1] IEEE 802.15.4g-2012, "IEEE Standard for Local and metropolitan area networks – Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs) Amendment 3: Physical Layer (PHY) Specifications for Low-Data-Rate, Wireless, Smart Metering Utility Networks," IEEE 802.15.4g-2012, IEEE, (2012).

[2] LoRa Alliance, "LoRaWAN 1.1 Specification," (2017).

[3] Sigfox, "Sigfox connected objects: Radio specifications Rev. 1.6," (2022).

[4] IEEE 802.11ah-2016, "IEEE Standard for Information technology-Telecommunications and information exchange between systems – Local and metropolitan area networks – Specific requirements – Part11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 2: Sub 1 GHz License Exempt Operation," IEEE 802.11ah-2016, IEEE, (2016).

[5] 802.11ah Promotion Councill, "IEEE802.11ah Use case," https://www.11ahpc.org/pdf/usecase-eng.pdf, (2019).

[6] C.-S. Sum, F. Kojima, and H. Harada, ''Performance analysis of a multiPHY coexistence mechanism for IEEE 802.15.4g FSK network,'' in Proc. IEEE Wireless Communication Network. Conf. (WCNC), pp. 41–46, (2013).

[7] F. Righetti, C. Vallati, D. Comola, and G. Anastasi, ''Performance measurements of IEEE 802.15.4g wireless networks,'' in Proc. IEEE 20th Int. Symposium World Wireless, Mobile Multimedia Network (WoWMoM), pp. 1–6, (2019).

[8] D. Hotta, R. Okumura, K. Mizutani, and H. Harada, ''Stabilization of multi-hop routing construction in Wi-SUN FAN systems,'' in Proc. IEEE 17th Annual Consumer Communication and Network Conference (CCNC), pp. 1–6, (2020).

[9] ARIB STD-T107 version 1.1, "920MHz-Band RFID Equipment for Specified Low Power Radio Station," (2017).

[10] ARIB STD-T108 version 1.4, "920MHz-Band Telemeter, Telecontrol and Data Transmission Radio Equipment," (2021).

[11] Y. Nagai, J. Guo, P. Orlik, B. A. Rolfe, and T. Sumi, "Proposal for Section 6.1 Japan on Recommended Practice for Local and Metropolitan Area Networks - Part 19: Coexistence Methods for 802.11 and 802.15.4 based systems operating in the Sub-1 GHz Frequency Bands," IEEE 802.19-19/0049r0, https://mentor.ieee.org/802.19/2019/, IEEE 802.19, (2019).

[12] T. Sumi, Y. Nagai, J. Guo, and P. Orlik, "920 MHz Status Update in Japan," IEEE 802.19-18/0084r0, https://mentor.ieee.org/802.19/2018/, IEEE 802.19, (2018).

[13] RFC 6550, "RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks," IETF, (2012).

[14] A. Qayyum, L. Viennot, and A. Laouiti, "Multipoint Relaying: An Efficient Technique for Flooding in Mobile Wireless Networks," in The 35th Annual Hawaii International Conference on System Sciences, (2001).

[15] N. Kadi and K. A. Agha, "Optimized MPR-based flooding in wireless ad hoc network using network coding," in 1st IFIP Wireless Days, (2008).

[16] K. A. Agha, N. Kadi, and I. Stojmenovic, "Fountain Codes with XOR of Encoded Packets for Broadcasting and source independent backbone in Multi-hop Networks using Network Coding," in IEEE VTC Spring 2009, (2009).

[17] N. Kadi and K. A. Agha, "MPR-based flooding with distributed fountain network coding," in 2010 The 9th IEIP Annual Mediterranean Ad Hoc Networking Workshop (Med-Hoc-Net), (2010).

[18] W. Matsumoto, R. Sakai, H. Yoshida, "Rate Compatible QC-LDPC codes," SITA 2006, pp.387-390, (2006).

[19] IEEE 802.15.4-2020, "IEEE Standard for Low - RateWireless Networks," IEEE 802.15.4-2020, IEEE, (2020).

[20] TTC Standard JJ-300.10 Version 2.2, "Home Network Communication Interface for ECHONET Lite (IEEE802.15.4/4g/4e 920 MHz-band Wireless)," TTC, Version 2.2, (2015).

(Received: March 1, 2023)
(Accepted: June 29, 2023)

Takenori Sumi received B.E. degree and M.E. degree in Faculty of Engineering of Doshisha University, Kyoto, Japan in 2005 and 2007, respectively. He is currently a Ph.D student in Graduate School of Science and Technology, Shizuoka University, Japan. He has been working as a senior researcher at Information Technology R & D Center of Mitsubishi Electric Corporation from 2007. His current research interests include spectrum sharing, wireless coexistence and V2X. He is also engaged in the Wi-Fi architect and network architect for embedded system. IEEE Senior member, IEICE member and IPSJ member.

Yukimasa Nagai received B.E. and M.E. degrees from The University of Electro-Communications, Japan in 1998 and 2000, respectively. In 2022, he received his Ph.D. degree in informatics from Graduate School of Science and Technology, Shizuoka University, Japan. He has been working at Mitsubishi Electric Corporation in Japan (2000-2016, 2020-) and Mitsubishi Electric Research Laboratories in USA (2016-2020). His research work has covered various topics on communication systems, IoT, CN and he has acted as key parts of getting standardization groups formed in Wi-Fi Alliance and IEEE 802.11/15/19. Wi-Fi Alliance Automotive MSTG Vice Chair, IEEE Senior member, and IPSJ member.

Jianlin Guo is a Senior Principal Research Scientist at Mitsubishi Electric Research Laboratories in Cambridge, Massachusetts, USA. He received his Ph.D. in 1995 from University of Windsor, Canada. His research interests include coexistence of the heterogeneous wireless networks, routing and resource management in wireless IoT networks, distributed machine learning, time sensitive networking, edge computing for vehicular networks, networked control systems, and fault detection in industrial automation. He served as the Technical Editor for IEEE 802.19.3 Task Group. He is IEEE Senior Member.

Hiroshi Mineno received his B.E. and M.E. degrees from Shizuoka University, Japan in 1997 and 1999, respectively. In 2006, he received his Ph.D. degree in information science and electrical engineering from Kyushu University, Japan. Between 1999 and 2002, he was a researcher in the NTT Service Integration Laboratories. In 2002, he joined the Department of Computer Science of Shizuoka University as an Assistant Professor. He is currently a Professor. His research interests include Intelligent IoT systems as well as heterogeneous network convergence. He is a senior member of IEEE, IEICE and IPSJ, a member of ACM and the Informatics Society.