

## Compact wireless control network protocol with fast path switching

Yasutaka Kawamoto<sup>\*1</sup>, Makoto Kubomi<sup>2</sup>, Yuichi Kado<sup>3</sup>

<sup>1</sup>Smart Network Technologies R&D Department, Corporate R&D Center, OKI Electric Industry Co., Ltd., 541-0051, Japan

<sup>2</sup>Industrial Network Architecture Group Embedded Network Technology Department, Information Technology R&D Center, Mitsubishi Electric Corporation, 247-8501, Japan

<sup>3</sup>Department of Electronics, Kyoto Institute of Technology, 606-8585, Japan

### ARTICLE INFO

#### Article history:

Received: 24 May, 2017

Accepted: 20 July, 2017

Online: 01 August, 2017

#### Keywords:

Sensor network protocol

Communication failure

Wireless control networks

### ABSTRACT

Sensor network protocol stacks require the addition or adjustment of functions based on customer requirements. Sensor network protocols that require low delay and low packet error rate (PER), such as wireless control networks, often adopt time division multiple access (TDMA). However, it is difficult to add or adjust functions in protocol stacks that use TDMA methods. Therefore, to add or adjust functions easily, we propose NES-SOURCE, a compact wireless control network protocol with a fast path-switching function. NES-SOURCE is implemented using carrier sense multiple access/collision avoidance (CSMA/CA) rather than TDMA. Wireless control networks that use TDMA prevent communication failure by duplicating the communication path. If CSMA/CA networks use duplicate paths, collisions occur frequently, and communication will fail. NES-SOURCE switches paths quickly when communication fails, which reduces the effect of communication failures. Since NES-SOURCE is implemented using CSMA/CA rather than TDMA, the implementation scale is less than one-half that of existing network stacks. Furthermore, since NES-SOURCE's code complexity is low, functions can be added or adjusted easily and quickly. Communication failures occur owing to changes in the communication environment and collisions. Experimental results demonstrate that the proposed NES-SOURCE's path-switching function reduces the amount of communication failures when the communication environment changes owing to human movement and others. Furthermore, we clarify the relationships among the probability of a changing communication environment, the collision occurrence rate, and the PER of NES-SOURCE.

## 1. Introduction

This paper is an extension of a work originally presented at the 2016 TRON Symposium [1].

Recently, sensor network technology has been applied to smart networks and Internet of Things networks, such as wireless control networks (WCN) for factory and plant automation. Such practical applications require a low delay, a low packet error rate (PER), and delay guarantees [2, 3].

In general sensor network protocols, e.g., the ZigBee PRO specification [4], nodes in the network determine the communication path autonomously. Therefore, it is difficult to consider communication delay when designing a system because the communication route is unknown. Therefore, many WCN protocols that require delay guarantees adopt a fixed routing system, i.e., external routing information is the input.

Some WCN protocols employ multiple routes simultaneously during data transmission to eliminate the delay caused by packet loss. For example, WirelessHART [5] increases communication reachability by duplicating all communication paths from each node to the base station. However, duplicating communication paths can cause collisions.

<sup>\*</sup>Corresponding Author: Yasutaka Kawamoto, Smart Network Technologies R&D Department, Oki Electric Industry Co., Ltd., +81662600700, Japan  
Email: kawamoto728@oki.com

Thus, most existing WCN protocols adopt TDMA [5–9] because it is contention-free.

However, TDMA has the following problems.

(1) Handling difficulty

TDMA requires high synchronization accuracy. Generally, it is difficult to implement programs that can maintain synchronization accuracy in less than 10 ms because, in embedded software environments, unexpected delays of several milliseconds occur randomly, such as OS task switches and interrupts.

For example, the flooding time synchronization protocol [10], which is a general time synchronization technology, handles such random delays by including time information in the send frame under the RFLSI driver sequence. Software that crosses network layers is difficult to develop without errors. The WCN stack must operate with various types of hardware wherein processing capacity and how connected sensors are handled vary. In addition, customer requirements must be considered for each application. When using a stack with TDMA, we must adjust the stack each time the hardware changes. In particular, when using sensors that generate various interrupts or require time-consuming operations, it is difficult to maintain accurate TDMA synchronization.

As described above, TDMA systems are difficult to implement, i.e., it is difficult to add or modify functions to such systems.

(2) High procurement cost

Note that rigid hardware specifications must be satisfied to maintain high TDMA synchronization accuracy.

For example, Linear Technology's WirelessHART module maintains high synchronization accuracy by measuring temperature changes and clock error changes during the manufacturing process and writing this information to memory.

However, fabricating such hardware is difficult, and procurement costs are high.

Some WCN studies have employed CSMA/CA protocols [11, 12]. However, CSMA/CA protocols are designed for large networks and are ineffective in typical WCNs (approximately 25 nodes) [3, 5].

In this paper, we propose the NES-SOURCE protocol stack for a WCN of practical size.

NES-SOURCE has compact implementation. Its code size is approximately one-quarter that of a general sensor network protocol stack and approximately one-half that of the TDMA-based WCN protocol stack. The cyclomatic complexity, which is a measure of source code complexity, of NES-SOURCE's functions is at most 10, which is less than other WSN [19] and WCN [20] protocol stacks. This indicates that NES-SOURCE stacks can be customized and adjusted quickly.

NES-SOURCE has a high-speed path-switching function, i.e., paths are switched quickly when communication failures are detected. With this function, NES-SOURCE achieves a low delay and a low PER without duplicating the communication path, which is the case with TDMA-based technology. Compared to a simple

fixed-path protocol, NES-SOURCE has the following features that allow routes to be changed quickly.

(1) Reduced back-off time

Sensor data and commands used in a WCN are at most several dozen bytes. By limiting the frame size to approximately 100 bytes, the random back-off time with CSMA/CA can be reduced from a maximum of approximately 15 ms to a maximum of 2.8 ms, as specified in IEEE 802.15.4g.

(2) ACK omission

If the communication path comprises a single hop, when NES-SOURCE detects packet loss in the MAC layer, it resends the frame using a backup route without waiting for the network layer ACK timeout.

With the above features, NES-SOURCE's communication delay can be reduced to approximately 60 ms for a communication path using a normal single-hop route and a backup two-hop route, which has been confirmed experimentally.

Packet loss occurs owing to packet collision or changes in the communication environment. We demonstrate experimentally that the PER and communication delay can be improved by implementing NES-SOURCE's route change function in a changing communication environment, e.g., in a factory environment where people and equipment interfere with wireless communications. Furthermore, we clarify the relationships between the probability of interference from obstacles in a communication environment and the collision occurrence rate and the PER of NES-SOURCE. The NES-SOURCE path-switching method can be considered an optimal method during system design.

## 2. Problems with Standardized Technologies and Related Work

### 2.1. Standardized Technologies

Several technologies have been standardized for WCNs.

IEEE 802.15.4 compliant technologies [8] are most frequently used in short-range wireless sensor networks (WSN), e.g., IEEE 802.15.4 is used as the MAC layer protocol in ZigBee IP [13].

IEEE 802.15.4 has two operational modes, i.e., non-beacon and beacon modes. The non-beacon mode is a CSMA/CA-based communication mode, and the beacon mode is a hybrid of TDMA and CSMA/CA. If nodes are in beacon mode, they send packets during a contention-free period (CFP). During the CFP, nodes communicate using TDMA. If the delay request for packets is short, nodes send packets during a contention access period (CAP). Note that nodes communicate using CSMA/CA during the CAP. IEEE 802.15.4's beacon mode is a well-thought-out protocol; however, it has not been implemented broadly because the specification is complicated and difficult to implement.

Time Slotted Channel Hopping (TSCH) is a TDMA-based protocol standardized in IEEE 802.15.4e [15]. TSCH nodes change communication channels each time the nodes change their TDMA timeslot. TSCH network systems are highly noise resistant. Owing to these characteristics, TSCH is used in WirelessHART [5]. TSCH is a suitable MAC layer protocol for WCNs. However, it requires improved clock accuracy, which is expensive. For example, in WirelessHART, clock correction information for

nodes is measured and written during the manufacturing process; however, such processes cannot be performed in all factories.

WirelessHART [5] uses TSCH to enhance communication reliability by incorporating a redundant communication path. However, this protocol control network information resides on the base station side; therefore, the system will fail if nodes cannot communicate with the base station. In addition, these protocols use TDMA, and it is difficult to guarantee delay on the order of 50–100 ms [3].

Sub-GHz bands are suitable for WCNs because they allow communication over greater distances compared to the 2.4-GHz band, which is used in IEEE 802.15.4 [8]. For example, IEEE 802.15.4g [14] is a wireless communication standard that uses a sub-GHz band. The theoretical communication distance of IEEE 802.15.4g is greater than 1 km at 20 mW. This is greater than the IEEE 802.15.4 communication distance (approximately 250 m at best). In addition, compared to 2.4-GHz bands, which are very noisy ISM bands, sub-GHz bands are not noisy.

The maximum frame size of IEEE 802.15.4g is 2048 bytes. Therefore, back-off time during CSMA/CA communication is approximately 15 ms at maximum. This back-off time is too large for a WCN, which requires a communication delay of less than approximately 50–100 ms.

## 2.2. Related Work

Researchers have improved the reliability of and delays in WCNs. WCN protocols can be classified as TDMA contention-free and CSMA/CA contention-based methods.

WirelessHART [5], PEDAMACS [7], and GinMAC [6] are representative contention-free schemes.

In WirelessHART [5], the base station regularly updates the routing and channel information for the network nodes. Therefore, the system does not function if nodes cannot communicate with the base station [15].

PEDAMACS [7] solves this problem by increasing the base station's transmission power to implement single-hop transmission from the base station. However, single-hop transmission often cannot be implemented in a WCN application environment.

GinMAC [6] avoids this problem by pre-installing node routing information. However, GinMAC uses TDMA and is difficult to implement.

MMSPEED [11] and Dwarf [12] are representative contention-based schemes.

MMSPEED [11] guarantees delay by controlling data stream QoS attributes, and this method is effective for large-scale data streams and networks. The network scale in [11] is approximately 150 nodes, which is greater than a typical WCN (approximately 25 nodes)

Dwarf [12] improves reliability by utilizing unicast-based flooding. This is effective when node density is high in a large-scale network. Therefore, Dwarf is unsuitable for WCNs.

Table 1 Practical WCN requirements

Network size	Up to 25 nodes
Packet occurrence frequency	0.1-1.0Hz (for monitoring) $3 \times 10^{-3}$ Hz (for FA or PA)
Packet loss rate	$10^{-4}$ order
Delay guarantees	1s (for monitoring) 50-100msec(for FA or PA)
Network topology	Most nodes will be located 1 hop from the station node and at most a distance of 3 hops will be observed.

## 3. NES-SOURCE Design

### 3.1. WCN Requirements

According to Zandra et al. [3], control applications can be categorized into three levels depending on the importance of message timing. Wired network communication is reasonable for applications with high-level requirements (e.g., control of life-threatening emergency stop devices). On the other hand, applications with low-level requirements (e.g., applications that safely allow manual intervention and/or monitoring) can use general WSNs.

WCNs can be used for applications with medium-level requirements. Table 1 lists the requirements generally assumed for WCNs. Note that the information given in Table 1 is based on the requirements provided by Samarasinghe et al. [6] and Kumar et al. [2]. The main application of a WCN that satisfies the requirements in Table 1 is assumed to be “a network for automatic control requiring responsiveness on the order of one second from measurement to response.” [2, 15]. Note that “automated valve control at the factory” is often given as an example [15].

NES-SOURCE is assumed for use in environments wherein people and equipment move frequently, such as manufacturing plants. When a person or piece of equipment enters the communication path, the communication environment changes, and packet loss can occur. A WCN node cannot use an obstructed path until the obstruction is removed. Therefore, a WCN protocol requires a route-changing function.

As mentioned previously, in the general sensor network protocol, each node changes the communication path autonomously. In contrast, in the general WCN protocol, nodes do not construct routes autonomously. For example, in WirelessHART, nodes use duplicate communication paths simultaneously to transmit packets rather than changing routes.

In WirelessHART and comparable technologies that adopt TDMA, collisions do not occur even if the communication path is duplicated because TDMA is collision-free. However, when using CSMA/CA, collisions tend to occur if communication paths are duplicated because packets are transmitted to the same destination at essentially the same time. In this case, transmission and ACK packet collisions cause contention.

Therefore, when considering a WCN stack using CSMA, we must also consider a method to detect communication failure and switch communication paths quickly.

### 3.2. NES-SOURCE Features

We designed and developed NES-SOURCE to satisfy the conditions listed in Table 1 [16]. NES-SOURCE has the following features.

- PHY and MAC layers

The PHY/MAC layer of NES-SOURCE is based on IEEE 802.15.4g. The usable frequency band is the 920-MHz band, the modulation method is GFSK, and the output power is 20 mW.

Generally, if the single-hop communication distance is long, the number of hops in a multi-hop network decreases, and as a result, communication delay decreases. The communication distance of IEEE 802.15.4g is four times that of IEEE 802.15.4 at 2.4 GHz. Therefore, with IEEE 802.15.4g, it is highly probable that the effective communication delay will be less than that of IEEE 802.15.4. Furthermore, the diffraction characteristic of the 920-MHz band is better than that of the 2.4-GHz band. Therefore, IEEE 802.15.4g is suitable for the application environment assumed by NES-SOURCE (e.g., a factory with complicated piping).

The CSMA random back-off time of IEEE 802.15.4g is up to approximately 60 ms, which is greater than the maximum random back-off time of IEEE 802.15.4d [17] (11.2 ms) because the maximum frame size of IEEE 802.15.4g is 2048 bytes. This maximum frame size is larger than the maximum frame size of IEEE 802.15.4d (128 bytes). Note that a WCN does not need to communicate large volumes of data; thus, 128 bytes is a suitable maximum frame size.

Therefore, NES-SOURCE adopts the random back-off time of IEEE 802.15.4d. Our experimental results demonstrate that communication delay is improved by 6.5 ms (an average improvement of 27%) when transmitting 20 bytes by reducing the random back-off time [18].

- Network layer

Similar to GinMAC, NES-SOURCE routing is a source routing method that uses an input fixed route. With NES-SOURCE routing, it is possible to set multiple paths to a single destination node, and these paths can have predetermined priorities.

When NES-SOURCE receives transmission requests from the upper layer, it first transmits the packet using the highest priority path. When a packet transmission failure is detected at the network layer, NES-SOURCE retransmits the packet using the next priority path.

NES-SOURCE uses ACK communication in the MAC and network layers to improve communication reliability. ACK communication in the network layer is required to implement path switching in the network layer. However, when an ACK from the network layer is used in a single-hop route, two transmissions (i.e., network layer data and ACK communications) are performed to confirm the transmission. However, such communication is useless, and reducing the frequency of such useless communication benefits the system.

NES-SOURCE only uses ACK communications from the MAC layer, i.e., it does not use ACK communications from the network layer when the communication path to the destination node is a single hop. In this case, NES-SOURCE retransmits using the next priority path when the ACK communication of the MAC layer fails. As a result, an average communication delay improvement of approximately 9 ms is confirmed for a single-hop communication path [16].

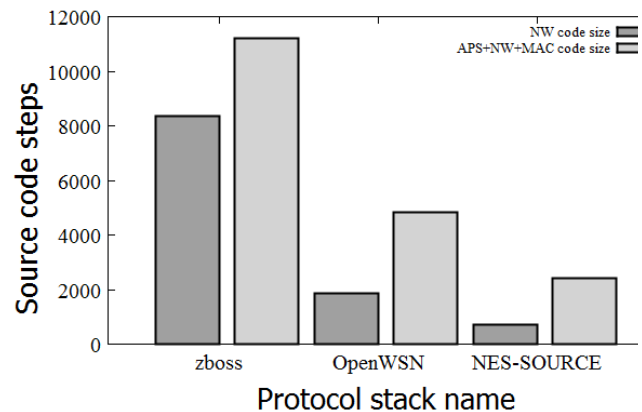


Figure 1 Protocol stack code size comparison

- Software

The NES-SOURCE source code is more compact compared to the protocol stacks for WSNs and WCNs. A source code size comparison is shown in Figure 1.

As an implementation of a general sensor network protocol, we refer to zboss [19], which is an open-source implementation of ZigBee PRO. The zboss code size (excluding the PHY driver) is 11217 steps in total (the NW layer code is 8366 steps).

As an implementation of the control radio protocol stack, we refer to OpenWSN [20], which is an open-source implementation of TSCH. The OpenWSN code size (excluding the PHY driver) is 4833 steps (the NW layer code is 1859 steps).

For NES-SOURCE, the total code size (excluding the PHY driver) is 2403 steps (the NW layer code is 727 steps). Even if only the number of code steps is compared, NES-SOURCE is a very compact implementation compared to other implementations.



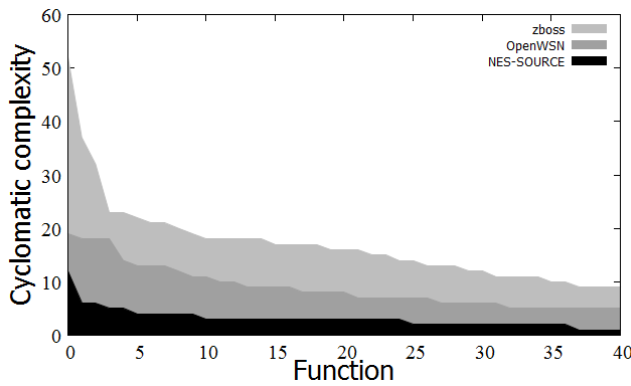


Figure 2 Program complexity

Figure 2 compares the function complexity of each protocol stack. The complexity of a function can be measured by its cyclomatic complexity [21]. Note that adding new code to a function is difficult if the cyclomatic complexity is large. Generally, there is no problem if the cyclomatic complexity is 10 or less. Figure 2 compares the cyclomatic complexity of the top 40 functions in the source code of each protocol stack. For example, zboss has 35 complex functions (the cyclomatic complexity is greater than or equal to 10), and OpenWSN has 10 complex functions. In contrast, NES-SOURCE has only one complex function. Thus, we consider that NES-SOURCE stacks can be customized and adjusted faster than general WSN and WCN protocol stacks.

A size comparison of the highly complex functions in each protocol stack is shown in Figure 3. Generally, fewer steps per function result in better code readability, which makes the code easier to maintain. As can be seen, NES-SOURCE has fewer steps per function than the other stacks; thus, NES-SOURCE is easy to maintain.

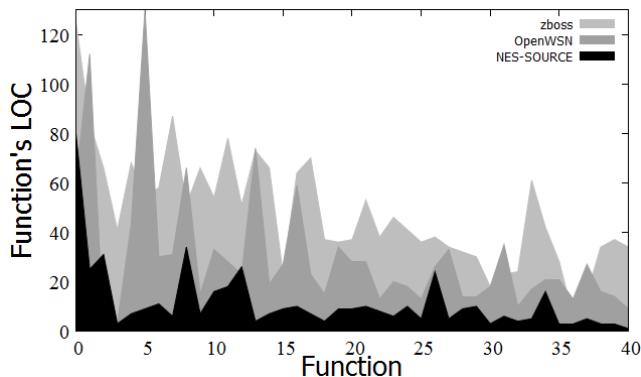


Figure 3 Size comparisons of complex functions

#### 4. Evaluation on Actual Machine

To evaluate and verify NES-SOURCE, we implemented it on an OKI MH 920-Node-232. The MH 920-Node-232 is a sensor node that supports IEEE 802.15.4g and is equipped with a Cortex-M3 processor. Here we adopted FreeRTOS [22] as the embedded OS, and we developed the stack using the C programming language. The other experimental variables are shown in Table 3.

In this experiment, receiving nodes were arranged to form a hexagon, and the transmitting node was installed at the center of the hexagon. The transmission packet was 53 bytes (including overhead), the distance between nodes was 2 m on average, the communication system's transmission frequency was 1 Hz, the number of people in the room was always approximately 10 people, and the number of trials was 10000 for each node (approximately 16 hours).

The experiment evaluated the following packet loss management methods.

- Method 1: Retransmission by the same route  
The transmission node retransmits using the same single-hop route if communication fails. Here, the maximum number of retransmissions is four (Figure 4).
- Method 2: Retransmission by another route  
The transmission node retransmits using a two-hop route if single-hop communication fails. Here, the maximum number of retransmissions is one (Figure 5).
- Method 3: Retransmission by the same and another route

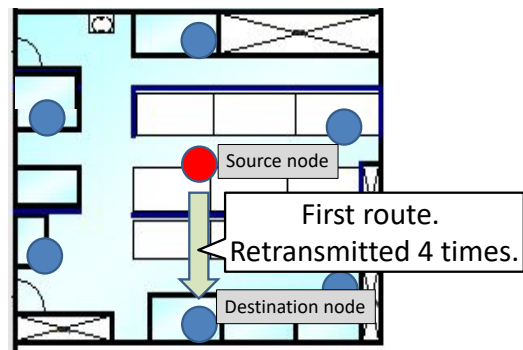


Figure 4 Method 1: first route only

The transmission node retransmits using a two-hop route if single-hop communication fails five times. Here, the maximum number of retransmissions for two-hop communication is one (Figure 6).

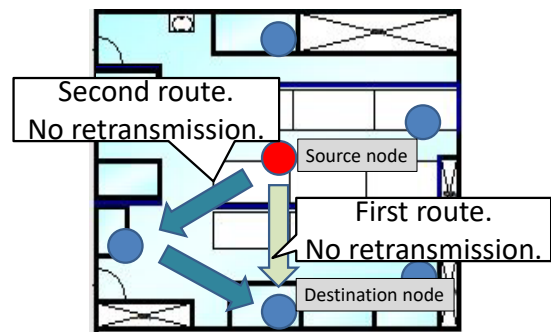


Figure 5 Method 2: two routes without retransmission

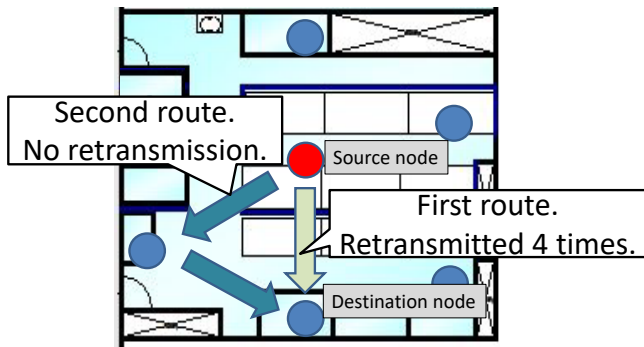


Figure 6 Method 3: two routes with retransmission

Figures 7 and 8 show the experimental results. Figure 7 shows the maximum communication delay for each method (i.e., 55.37, 51.84, and 95.93 ms for Methods 1, 2, and 3, respectively). Figure 8 shows the PER for each method ( $8.71 \times 10^{-3}$ ,  $5.83 \times 10^{-4}$ , and  $5.33 \times 10^{-4}$  for Methods 1, 2, and 3, respectively).

The number of single- and two-hop communication routes for Method 2 was 58138 and 1827, respectively. The number of single- and two-hop communication routes for Method 3 was 59422 and 546, respectively.

## 5. Discussion

The experimental results indicate that the route change function in the proposed NES-SOURCE stack works effectively.

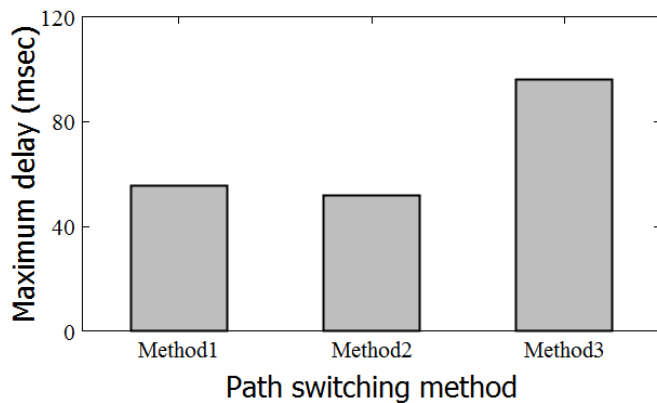


Figure 7 Maximum communication delay

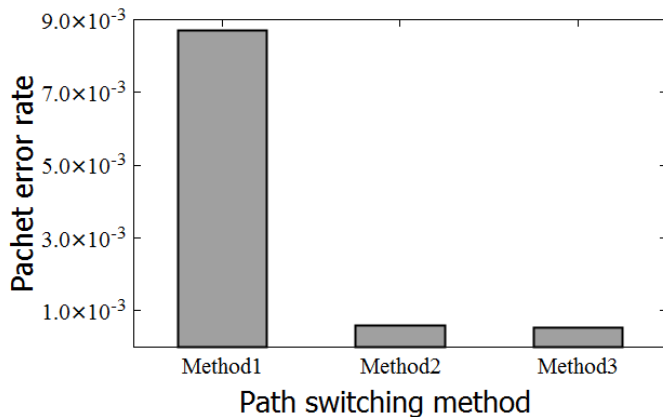


Figure 8 Packet error rate

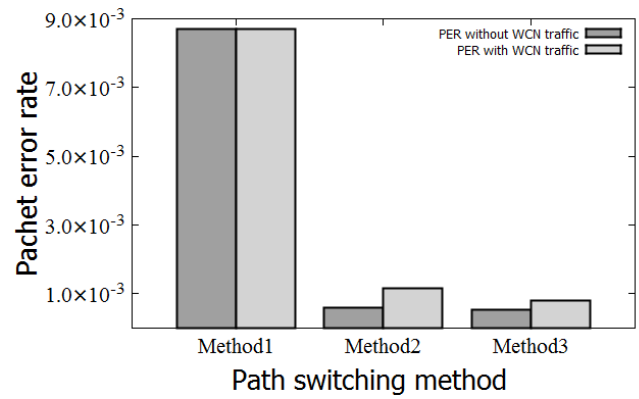


Figure 9 Packet error rate with/without WCN traffic

Table 2 Simulation parameter values

aUnitBackoffPeriod [8]	200 usec
aCCATime [8]	100 usec
macMinBE [8]	3
macMaxBE [8]	4
macMaxCSMABackoffs [8]	5
The number of peripheral nodes	25
Frame send period	3.5 msec
Frame send interval	1 sec
Measurement time	30000 sec × 50

Table 3 Variable for calculating the PER

Number of successful communications method 1 by 1 hop	$N_{m1\_1hop}$
Number of failed communications method 1	$N_{m1\_failed}$
Number of successful communications method 2 by 1 hop	$N_{m2\_1hop}$
Number of successful communications method 2 by 2 hops	$N_{m2\_2hop}$
Number of failed communications method 2	$N_{m2\_failed}$
Number of successful communications method 3 by 1 hop	$N_{m3\_1hop}$
Number of successful communications method 3 by 2 hops	$N_{m3\_2hop}$
Number of successful communications method 3 by 2 hops	$N_{m3\_failed}$
Number of trials	$N_{trial}$
PER of 1 hop without retransmission under the WCN traffic	$PER_{1hop\_noretry\_c}$
PER of 2 hops without retransmission under the WCN traffic	$PER_{2hop\_c}$
PER of 1 hop with 4 retransmissions under the WCN traffic	$PER_{1hop\_4retry\_nc}$

In the experimental environment, Method 2 was more advantageous relative to PER (Figure 5) than Method 1 as a measure against packet loss.

Furthermore, Method 2 was clearly superior to Method 3 relative to maximum delay (Figure 4). However, when compared relative to PER (Figure 5), the difference between Methods 2 and 3 is small.

These results demonstrate that the effect of link retransmission on packet loss was low in our experimental environment.

Thus, in a changing communication environment (e.g., a person enters the communication path), changing the path is more

effective relative to PER and maximum delay compared to link retransmission.

As stated previously, changes in a communication environment cause packet loss and collision. Note that our experiment did not consider packet collisions. Thus, to calculate a more realistic PER, the collision occurrence rate in WCN traffic was calculated in a simulation.

The simulation was created based on the communication frequency assumed by a WCN (Table 1) and the communication device settings given in Section 4. The simulation parameters are shown in Table 2.

The simulation results show that, for 25 nodes in the vicinity and a packet transmission frequency of 1 Hz for each node (i.e., a WCN general data transmission environment; Table 1), the PER of one-to-one communication without retransmission ( $PER_{1hop\_noretry\_c}$ ) was 0.8%. In addition, if four retransmissions were allowed ( $PER_{1hop\_4retry\_c}$ ), the PER was 0%, and the PER of two-hop communication without retransmission ( $PER_{2hop\_noretry\_c}$ ) was 1.59%.

On the basis of the simulation and experimental results given in Section 4 and Table 3, respectively, we can calculate the PER of NES-SOURCE in WCN traffic. The relationships among the values shown in Table 3 are expressed as follows.

$$N_{trial} = N_{m1\_1hop} + N_{m1\_failed} \quad (1)$$

$$N_{trial} = N_{m2\_1hop} + N_{m2\_2hop} + N_{m2\_failed} \quad (2)$$

$$N_{trial} = N_{m3\_1hop} + N_{m3\_2hop} + N_{m1\_failed} \quad (3)$$

The PER of each method in a collision-free environment ( $PER_{m1\_nc}$ ,  $PER_{m2\_nc}$ ,  $PER_{m3\_nc}$ ) can be expressed as follows.

$$PER_{m1\_nc} = N_{m1\_failed} / N_{trial} \quad (4)$$

$$PER_{m2\_nc} = N_{m2\_failed} / N_{trial} \quad (5)$$

$$PER_{m3\_nc} = N_{m3\_failed} / N_{trial} \quad (6)$$

On the basis of Equations 4–6, the PER of NES-SOURCE under WCN traffic conditions can be expressed as follows.

$$PER_{m1\_nc} = \frac{N_{m1\_failed} + N_{m1\_1hop} \times PER_{1hop\_4retry\_c}}{N_{trial}} \quad (7)$$

$$PER_{m2\_nc} = \frac{N_{m2\_failed} + (N_{m2\_1hop} \times PER_{1hop\_4retry\_c} + N_{m2\_2hop} \times PER_{2hop\_noretry\_c})}{N_{trial}} \quad (8)$$

$$PER_{m3\_nc} = \frac{N_{m3\_failed} + (N_{m3\_1hop} \times PER_{1hop\_4retry\_c} + N_{m3\_2hop} \times PER_{2hop\_noretry\_c})}{N_{trial}} \quad (9)$$

On the basis of the measurement results (Section 4), simulation results, and Equations 7–9, the PER values of NES-SOURCE under WCN traffic conditions for Methods 1 to 3 were  $8.71 \times 10^{-3}$ ,  $1.16 \times 10^{-3}$ , and  $8.05 \times 10^{-4}$ , respectively (Figure 9).

According to Zandra et al. [3], the PER request of a WCN is on the order of  $10^{-4}$ , and the delay request is on the order of 50–100 ms. Therefore, NES-SOURCE can be used in an experimental

environment if the given WCN application has a slightly lower PER requirement.

When we construct WCN systems, we cannot predetermine traffic conditions or the frequency of communication problems. To mitigate this uncertainty, NES-SOURCE can flexibly change the packet loss countermeasure depending on the application's requests and the environment. For example, if the network traffic is assumed to be high, Method 1 is reasonable. On the other hand, when the network is constructed in an environment where people frequently enter the communication path, Method 2 is reasonable. Method 3 is appropriate if the goal is to increase the PER, even at the expense of the maximum delay.

“When a person enters the communication path, packet loss occurs.”

“Packet loss due to a person can always be avoided by changing the communication path.”

Assuming the above points are correct, the relationship between the probability that a person is within the communication path ( $P_{ps}$ ) and the PER of each method can be expressed as follows.

$$PER_{Method1} = (1 - P_{ps}) \times PER_{1hop\_4retry\_c} \quad (10)$$

$$PER_{Method2} = (1 - P_{ps}) \times PER_{1hop\_noretry\_c} + P_{ps} \times PER_{2hop\_noretry\_c} \quad (11)$$

$$PER_{Method3} = (1 - P_{ps}) \times PER_{1hop\_4retry\_c} + P_{ps} \times PER_{2hop\_noretry\_c} \quad (12)$$

For example, Figure 10 shows the relationship between  $P_{ps}$  and PER when  $PER_{Method1} = 0.8 \times 10^{-3}$ ,  $PER_{Method2} = 1.59 \times 10^{-3}$ , and  $PER_{Method3} = 0$  (this is the same as the simulation results). In this case, when a person's appearance rate in the communication path exceeds 0.8%, it is better to adopt Method 2 relative to PER.

## 6. Conclusions

In this paper, we have proposed, implemented, and clarified the effectiveness of NES-SOURCE as a WCN protocol that adopts CSMA/CA, which is easy to handle.

WSN protocol stacks must be customized and adjusted for each application, and the same is true for a WCN protocol stack, which is a WSN protocol stack with strict delay and PER requirements.

Many WCN stacks are implemented using TDMA. However, TDMA is difficult to handle; thus, customization and adjustment are difficult.

Therefore, we have developed the compact NES-SOURCE WCN protocol stack. The proposed NES-SOURCE employs the CSMA/CA method rather than the TDMA method. As a result, NES-SOURCE's code size is approximately 2700 steps, which is less than one-half the size of the TDMA-based WCN stack. In addition, NES-SOURCE has a lower code complexity than conventional WCN stacks and can be customized and adjusted quickly.

The WCN protocol with TDMA ensures communication reliability by duplicating the communication path. In contrast, the proposed NES-SOURCE ensures communication reliability by fast path switching when communication fails.

We conducted experiments in an environment where people entered and exited the communication path, and we confirmed that NES-SOURCE's fast path-switching function works effectively. The experimental results demonstrate that NES-SOURCE's communication failure detection and path-switching mechanism are fast and that communication delay was within general WCN requirements.

We have also clarified the relationships between the probability of a person entering the communication path and the PER due to the occurrence of collisions relative to each of the PER of the path-switching methods that the proposed NES-SOURCE employs. Thus, when designing systems that use NES-SOURCE, it will be possible to predetermine the optimum path-switching method relative to communication frequency and the varying degrees of obstructions in the physical environment.

Conventionally, TDMA must be adopted in WCN systems; however, this is difficult to customize and incurs high costs. In contrast, according to our experimental and simulation results, we can construct less expensive CSMA/CA-based WCN systems that can be customized quickly.

### Acknowledgment

The names of products or services are trademarks or registered trademarks of their respective companies.

### References

- [1] Y. Kawamoto, Y. Kado, "NES-SOURCE: Indoor small-scale wireless control network protocol that has a communication failure point avoidance function" in TRON Symposium (TRONSHOW), Tokyo Japan, 2016.
- [2] Ovsthus, K., Kristensen, L. M., "An industrial perspective on wireless sensor networks—a survey of requirements, protocols, and challenges" *IEEE Communications Surveys & Tutorials*, 16(3), 1391-1412, 2014
- [3] Zand, P., Chatterjea, S., Das, K., & Havinga, P., "Wireless industrial monitoring and control networks: The journey so far and the road ahead." *Journal of sensor and actuator networks* 1.2, 123-152, 2012.
- [4] ZigBee Alliance, ZigBee PRO specification, Standard, 2007.
- [5] J. Song, S. Han, A. Mok, D. Chen, M. Lucas, M. Nixon, W. Pratt, "WirelessHART: Applying wireless technology in real-time industrial process control" in *Real-Time and Embedded Technology and Applications Symposium*, IEEE RTAS'08, 377-386, 2008
- [6] K. Samarasinghe, T. Voigt, L. Mottola, U. Roedig, "Network coding with limited overhearing" in *8th European Conference on Wireless Sensor Networks*, Bonn Germany, 2011
- [7] Ergen, S. C., Varaiya, P., "PEDAMACS: Power efficient and delay aware medium access protocol for sensor networks", *IEEE Transactions on Mobile Computing*, 5(7), 920-930, 2006
- [8] IEEE Computer Society, "IEEE Standard for Local and metropolitan area networks—Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs)", IEEE Std, 2011
- [9] IEEE Computer Society, "IEEE Standard for Local and metropolitan area networks—Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs) - Amendment 1: MAC sub-layer" IEEE Std, 2012
- [10] M. Maróti, B. Kusy, G. Simon, A. Lédeczi, "The flooding time synchronization protocol" in *Proceedings of the 2nd international conference on Embedded networked sensor systems*, Baltimore Maryland, 2004
- [11] E. Felemban, C. Chang-Gun, Eylem. EKICI, "MMSPEED: multipath Multi-SPEED protocol for QoS guarantee of reliability and Timeliness in wireless sensor networks", *IEEE transactions on mobile computing*, 5(6), 738-754, 2006
- [12] M. Strasser, A. Meier, K. Langendoen, P. Blum, "Dwarf: Delay-aware robust forwarding for energy-constrained wireless sensor networks" in *International Conference on Distributed Computing in Sensor Systems*, Springer Berlin Heidelberg, 2007
- [13] ZigBee Alliance, ZigBee IP specification. ZigBee Standard, 2011
- [14] IEEE Computer Society, "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 Std, 2012
- [15] T. O'Donovan, J. Brown, U. Roedig, C.J. Sreenan, A. Dunkels, A. Klein, L. Wolf, "Ginseng: Performance control in wireless sensor networks", in *Sensor Mesh and Ad Hoc Communications and Networks (SECON)*, Boston United States, 2010
- [16] M. Kubomi, Y. Kado, Y. Kawamoto, "Improving reliability of 920 MHz-band wireless networks using advanced source routing function to guarantee real-time property.", in *Internet of Things and Applications (IOTA)*, Pune India, 2016.
- [17] IEEE Computer Society, "IEEE Standard for Local and metropolitan area networks—Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs) Amendment 3: Alternative Physical Layer Extension to support the Japanese 950 MHz bands", IEEE Std, 2009
- [18] Y. Kado, A. Kobayashi, M. Kubomi, Y. Nogami, Y. Kawamoto, "Application of Economical 920-MHz Band Wireless Communication to Power Routing in HVDC Networks", in *Proceedings of 2014 IEEE International Conference on Smart Grid Communications*, Venice Italy, 2014
- [19] A. Cunha, A. Koubaa, R. Severino, M. Alves, "Open-ZB: an open-source implementation of the IEEE 802.15. 4/ZigBee protocol stack on TinyOS", in *Mobile Adhoc and Sensor Systems*, Pisa Italy, 2007
- [20] T. Watteyne, X. Vilajosana, B. Kerkez, F. Chraim, K. Weekly, Q. Wang, K. Pister, "OpenWSN: a standards-based low-power wireless development environment.", *Transactions on Emerging Telecommunications Technologies*, 23(5), 480-493, 2012
- [21] T.J. McCabe, C. W. Butler, "Design complexity measurement and testing.", *Communications of the ACM*, 32(12), 1415-1425, 1989
- [22] R. Barry, FreeRTOS. Internet, 2008.