J Inf Process Syst, Vol.20, No.2, pp.252~262, April 2024
https://doi.org/10.3745/JIPS.03.0197

ISSN 1976-913X (Print)
ISSN 2092-805X (Electronic)

JOURNAL OF INFORMATION PROCESSING SYSTEMS JIPS

Confidential Convergecast Based on Random Linear Network Coding for the Multi-hop Wireless Sensor Network

Davaabayar Ganchimeg[1], Sanghyun Ahn[2,*], and Minyeong Gong[2]

Abstract

The multi-hop wireless sensor network (WSN) suffers from energy limitation and eavesdropping attacks. We propose a simple and energy-efficient convergecast mechanism using inter-flow random linear network coding that can provide confidentiality to the multi-hop WSN. Our scheme consists of two steps, constructing a logical tree of sensor nodes rooted at the sink node, with using the Bloom filter, and transmitting sensory data encoded by sensor nodes along the logical tree upward to the sink where the encoded data are decoded according to our proposed multi-hop network coding (MHNC) mechanism. We conducted simulations using OMNET++ CASTALIA-3.3 framework and validated that MHNC outperforms the conventional mechanism in terms of packet delivery ratio, data delivery time and energy efficiency.

Keywords

Confidentiality, Convergecast, Network Coding, Security, Wireless Sensor Network

1. Introduction

The wireless sensor network (WSN) has been widely applied to various areas like environmental monitoring, medical systems, etc. A multi-hop WSN consists of multiple sensor nodes (SNs) transmitting their sensory information to the sink node via multiple wireless links through sensors. This type of data transmission is referred to as convergecast or incast [1].

The random linear network coding (RLNC) scheme [2] has been used as a means to improve the reliability and the transmission efficiency of WSNs. Using RLNC, a SN encodes a set of packets by multiplying them with the randomly chosen linearly independent coefficients chosen from Galois field and transmits the encoded packets along with the coefficients. The receiver can recover the original packet by performing the Gaussian elimination.

Transmissions through wireless links are vulnerable to eavesdropping attacks. The confidentiality is one of the security goals that ensures data to be concealed from any eavesdropping attacks. The conventional almighty encryption and decryption procedures are too computation-intensive and costly to be implemented in resource-constrained WSNs.

Manuscript received January 9, 2024; accepted February 3, 2024.

* **Corresponding Author:** Sanghyun Ahn (ahn@uos.ac.kr)
[1] Buren Zuraglal, Ulaanbaatar, Mongolia (d.gana1024@gmail.com)
[2] School of Computer Science and Engineering, University of Seoul, Seoul, Korea (ahn@uos.ac.kr, gongmys@gmail.com)

Therefore, in this paper, we propose a simple and energy-efficient confidential convergecast mechanism that uses RLNC. To the best of our knowledge, this is the first RLNC-based confidentiality mechanism for WSNs. Our mechanism is suitable for the case where sensory data are periodically sent to the sink and the confidentiality of sensory data is a requisite. Our convergecast mechanism uses inter-flow RLNC for the multi-hop WSN with one sink and multiple source nodes (any SN can be a source).

Confidentiality is provided by having the sink select the linearly independent coefficients and securely distribute them to SNs at the initial stage, and by having each intermediate SN use the coefficients for encoding and send encoded packets without being accompanied by the coefficients used for encoding. Because the RLNC coefficients are not transmitted along with encoded packets, encoded packets cannot be decoded by attackers. That is, the RLNC coefficients play the role of encryption keys and the RLNC encoding plays the role of encryption. For our mechanism to work, a logical tree rooted at the sink has to be established first. Encoding is performed at each intermediate SN on the path from a source SN to the sink, along the logical tree, and decoding is performed only at the sink. By carrying out simulations using the OMNET++ CASTALIA-3.3 framework [3], we evaluate the performance of our mechanism in terms of packet delivery ratio, data delivery time and energy efficiency.

The rest of the paper is organized as follows: in Section 2, the related work on applying network coding (NC) to the multi-hop WSN is described. Section 3 describes our proposed mechanism composed of the logical tree configuration and the confidential convergecast, in detail. The performance evaluation of the proposed mechanism is given in Section 4 Finally, Section 5 concludes this paper.

2. Related Work

2.1 Random Linear Network Coding for WSNs

To date, RLNC has been applied to WSN to overcome the unreliability of wireless links. In RLNC, a sender encodes the original packets $p_1, p_2, \dots, p_q$ to be transmitted and generates r encoded packets $p'_1, p'_2, \dots, p'_r$, where $r \geq q$, according to the equation: $P' = C \times P$ where P' is the matrix of the encoded packets, C the matrix of the coefficients and P the matrix of the original packets.

The authors of [4] proposed a RLNC-based encoding scheme for unicast flows in single-hop ad hoc networks and multi-hop wireless mesh networks to curtail the time spent on transmission. They made an implementation of a RLNC encoding scheme over finite file size of GF(2) with considering wireless medium access control (MAC) overhead cost. This mechanism is good for throughput, but not for confidentiality. If an attacker eavesdrops the coefficients in the encoded packets, the attacker can decode the encoded packets based on the obtained coefficients, which is a breach of confidentiality. The potential countermeasure for this type of attack is for relay nodes to encrypt the encoded packets and for the sink to decrypt the encrypted encoded packet. This causes tremendous overhead at both the sink and the relay nodes. Also, this is limited to 2-hop WSNs.

The authors of [5] studied a subspace coding-based error-correction scheme which is called the subspace coding strategy (SCS) consisting of permutation and data encoding. The scheme enhances the security with the expanded search space of a wiretapping attack. According to the evaluation, the scheme is more secure than conventional schemes in the real-world.

To the best of our knowledge, there is no work done on applying RLNC to WSN for the confidentiality purpose.

2.2 Confidentiality for WSNs

Confidentiality is the capability to keep data from being exposed to attackers during transmissions. A straightforward way of providing confidentiality is to encrypt data at their source SN with a symmetric key shared with the sink, which requires symmetric key distribution and management.

In WSNs, data must be transmitted and aggregated in encrypted form so that the compromised nodes cannot change the aggregated data or disclose the secrecy of the collected information [6]. In end-to-end encryption, a node cannot be able to know or to extract the information from the received packets beside the sink. Using this approach, it is possible to guarantee that it will be much difficult for an eavesdropper to gain access to the data. Encryption and decryption operations are computationally very expensive and time consuming. In link layer cryptography, the data is encrypted by the sender, and it is decrypted at intermediate nodes [7]. Then, the aggregation function is applied, and the result is encrypted again before being sent to the next hop. This can lead to overflowing queues.

Homomorphic encryption schemes are one possibility of ensuring secure aggregation [8], as they allow data aggregation to be performed on encrypted data. Encryption and decryption operations are computationally very expensive and time consuming. In homomorphic encryption, certain aggregation functions such as sum and average can be applied on the encrypted data, reducing the workload of the sensors in the network significantly. All sensors along the path apply the aggregation function on the encrypted data. The base station receives the encrypted aggregated result and decrypts it. A homomorphic encryption scheme allows arithmetic operations on ciphertexts.

3. Confidential Multi-hop Convergecast Mechanism

3.1 System Environments

We assume that a unique ID, chosen from 1 to the total number of SNs, is assigned to each SN and the sink knows all the assigned SN IDs. Each SN is assumed to generate a single equal-sized message with sensory data, called the native packet, per transmission period. The native packet can be encoded with coefficients, where the native packet is the original packet. An encoded packet E can be encoded again and, in this case, E is the original packet.

After the deployment of SNs, the SNs form a logical tree rooted at the sink and the sensing is performed periodically by each SN. Every intermediate SN encodes its own native packet and the native or encoded packet(s) from all of its child SN(s). That is, the number of packets to be encoded at a SN is the same as the number of its child SNs plus one (its own packet). The T-MAC protocol [9], which is contention-free, is used for the transmission scheduling of encoded packets. Once the sink receives all the encoded packets from its child SNs, it decodes them.

3.2 Logical Tree Construction

Because an RLNC-encoded packet has the aggregated information of all of its original packets, it is critical to keep encoded packets intact. Therefore, we decide to use the T-MAC protocol instead of contention-based MAC protocols like IEEE 802.15.4 [10]. For the transmission scheduling of SNs based on the T-MAC protocol, SNs are formed in a tree rooted at the sink by using Tree_Setup_Request

(TSReq) and Tree_Setup_Reply (TSRpl) messages.

At the initial stage, the sink broadcasts a TSReq message with Level = 0 and its ID to its 1-hop neighbors. On the receipt of the TSReq message for the first time, a SN increases the level in the received TSReq message by 1 and sets the sender of the TSReq message as its parent and broadcasts the modified TSReq message to its neighbors. After that, if the SN receives a TSReq message whose level is higher than its level by 1 and the parent ID is the same as its ID, it considers the sender of the TSReq message as one of its children.

Each leaf node sends the TSRpl message with the Bloom filter, whose bit positions corresponding to the hashed value of its ID are marked, to its parent node. In the TSRpl message, the number of SNs marking the Bloom filter, the SN_Count (SNCnt) field, of the message is included; that is, when a leaf SN sends a TSRpl message, the SNCnt is set to 1. The SN receiving a TSRpl message adds its ID in the received Bloom filter and increases the SNCnt by 1 and sends the modified Bloom filter to its parent. Once the sink collects all the TSRpl messages (i.e., if all the SN IDs are collected from the Bloom filters of the TSRpl messages), it can completely figure out the logical tree network based on the pattern of the SN IDs included in the Bloom filters. The sink checks the Bloom filters $N_{BF} \times N_{sensor}$ times, where N_{BF} is the number of the received Bloom filters and N_{sensor} is the number of total SNs. Fig. 1 shows an example of how the sink deduces the logical tree topology from the received Bloom filters. For simplicity, we use the hash function dictating the bit position corresponding to the SN ID. In this example, there are 8 SNs each of which marks the bit position, corresponding to its ID, of the Bloom filter received from its child node(s). For example, s_2 marks the 6th bit position of the Bloom filter because its ID is 6. The sink

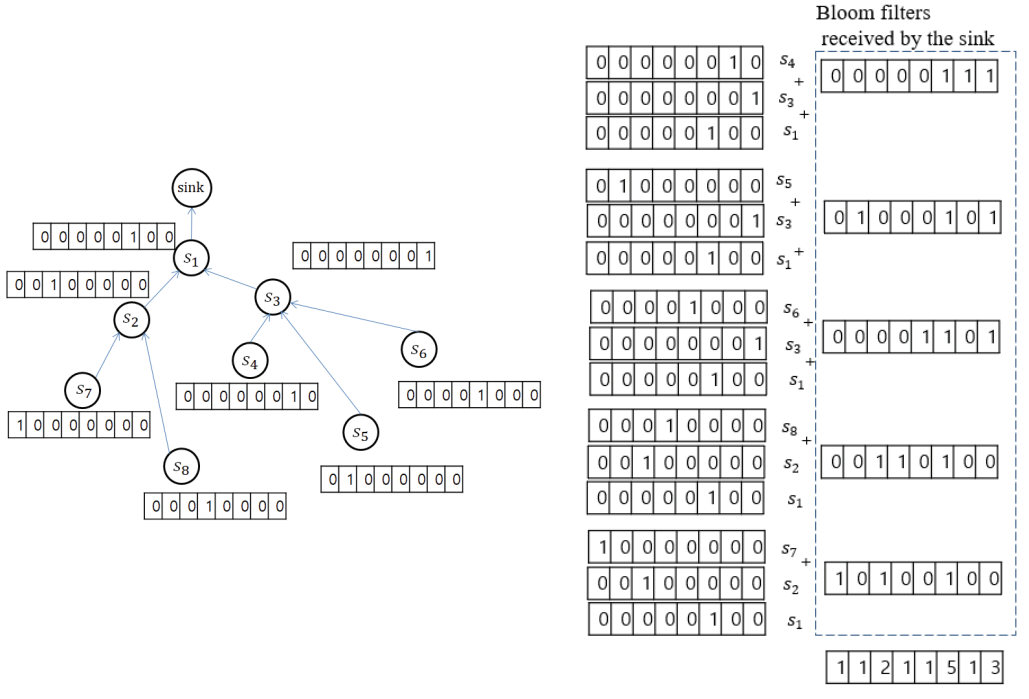


Fig. 1. An example of logical tree setup using the Bloom filter: (a) Bloom filter value of each SN and (b) Bloom filters received by the sink.

receives 5 Bloom filters originally generated by the leaf SNs s_4 to s_8. For example, the Bloom filter, delivered through the path from s_7 to the sink, is marked by s_7, s_2, and s_1 whose IDs are 8, 6, and 3, respectively. Once the sink collects all the Bloom filters from the leaf SNs, it counts how many times each SN ID is included in the Bloom filters. The sink knows that s_1 with ID = 3 is its only child because s_1 is included in all the 5 Bloom filters. And the sink knows that s_1 has two children s_2 with ID = 6 and s_3 with ID = 1 because they are included in 2 and 3 Bloom filters, respectively. Fig. 1 illustrates an example of logical tree setup using Bloom filters.

3.3 Confidential Multi-hop Convergecast based on Logical Tree

Once the logical tree is determined, the sink generates the ordered list, $\mathrm{K} =< k_1, k_2, \ldots, k_n >$, of the linearly independent encoding coefficients, and broadcasts the K_List (KLst) message with K′ to its children. The SN receiving the KLst message delivers the message to its non-leaf child node(s). This is because leaf SNs do not perform encoding (that is, leaf SNs do not require encoding coefficients). A SN can decide whether its child is a leaf node or not based on the TSRpl message received from the child. If the SNCnt value of the TSRpl message is 1, the sender of the TSRpl message is a leaf node. To make it more secure, coefficients may be updated periodically by the sink. Each leaf SN sends the native packet with its sensory data to its parent. A non-leaf SN encodes its own native packet and the native or encoded packets from all of its children using K. For the multi-hop encoding and decoding along the logical tree, we propose the multi-hop network coding (MHNC) mechanism.

In MHNC, at each period, each intermediate SN n encodes the packet(s) from its child node(s) and its own packet. If the number of packets to be encoded is j, the coefficient matrix $[\widehat{\mathrm{K_j}}]$ is the $j \times j$ diagonal matrix whose main diagonal is $< k_1, \ldots, k_j >$; i.e., the main diagonal of $[\widehat{\mathrm{K_j}}]$ is the ordered list of the first j elements of K. n multiplies $[\widehat{\mathrm{K_j}}]$ by the to-be-encoded packet matrix $[\mathrm{P}_j]^T$, which yields the encoded packet matrix $\left[\breve{P}_j\right]^T$. Then, n sends the encoded packet $\breve{P}_j$, corresponding to $\left[\breve{P}_j\right]^T$, to its parent SN. This encoding procedure is repeated at each SN along the path to the sink. When the sink receives the encoded packets from all of its children, it decodes them based on the topology and schedule information. That is, the sink decodes the received $\breve{P}_j$, by multiplying it with the inverse matrix, $\left[\mathrm{M_j}\right]^{-1}$, of $[\widehat{\mathrm{K_j}}]$. And this decoding procedure is repeated at the sink until all the original packets are obtained.

For the ease of understanding, we explain the operation of MHNC by using the example in Fig. 2. s_5 encodes the original packets, a_5, a_6, and a_7, by multiplying them with the coefficients k_1, k_2, and k_3 of K in that order, respectively, and sends the encoded packet $\breve{P}_3$ to its parent s_2, where $\breve{P}_3$ is composed of $k_1 \times a_5$, $k_2 \times a_6$, and $k_3 \times a_7$. Let $k_1 \times a_5 = a_5^1$, $k_2 \times a_6 = a_6^1$, and $k_3 \times a_7 = a_7^1$ (here, a_i^j implies the original packet a_i encoded j times). Then, s_2 encodes a_2, a_3, a_4, and $\breve{P}_3$ by multiplying them with the coefficients k_1, k_2, k_3, k_4, k_5, and k_6, respectively, and sends the encoded packet $\breve{P}_6$ to s_1. Here, $\breve{P}_6$ means that it is encoded by s_2, and a_5^2 means that a_5 is encoded twice. s_1 encodes a_1 and $\breve{P}_6$ by multiplying them with the coefficients k_1, k_2, k_3, k_4, k_5, k_6, and k_7, respectively, and sends the encoded packet $\breve{P}_7$ to the sink. Then, the sink decodes $\breve{P}_7$ based on the logical tree configuration information. That is, $\breve{P}_7$ is multiplied by the inverse matrix of $[\widehat{\mathrm{K}_7}]$, $\breve{P}_6$ is multiplied by the inverse matrix of $[\widehat{\mathrm{K}_6}]$, and $\breve{P}_3$ is multiplied by the inverse matrix of $[\widehat{\mathrm{K}_3}]$, until all the original packets are obtained. Fig. 2 shows an example of the encoding and decoding procedure of the MHNC mechanism.

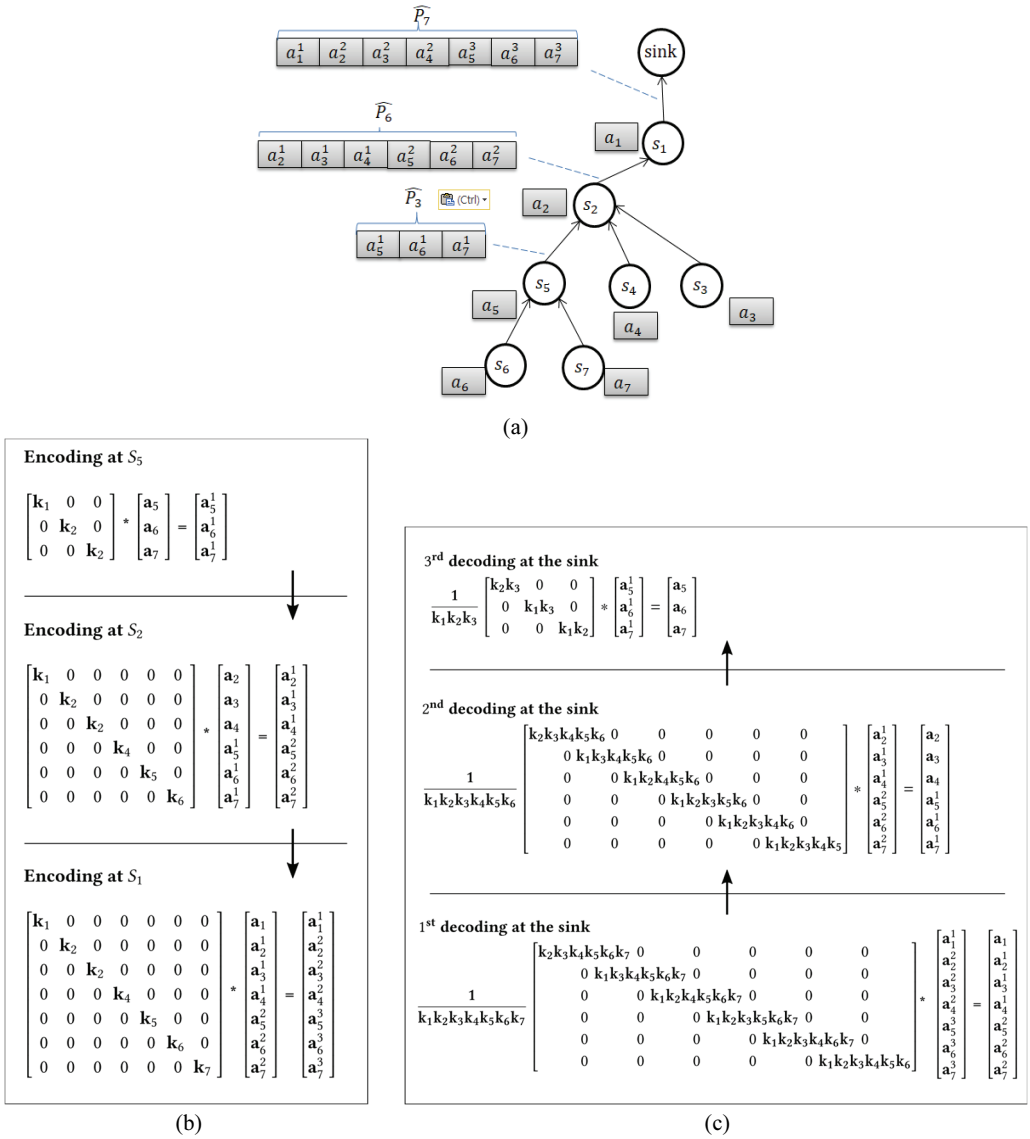
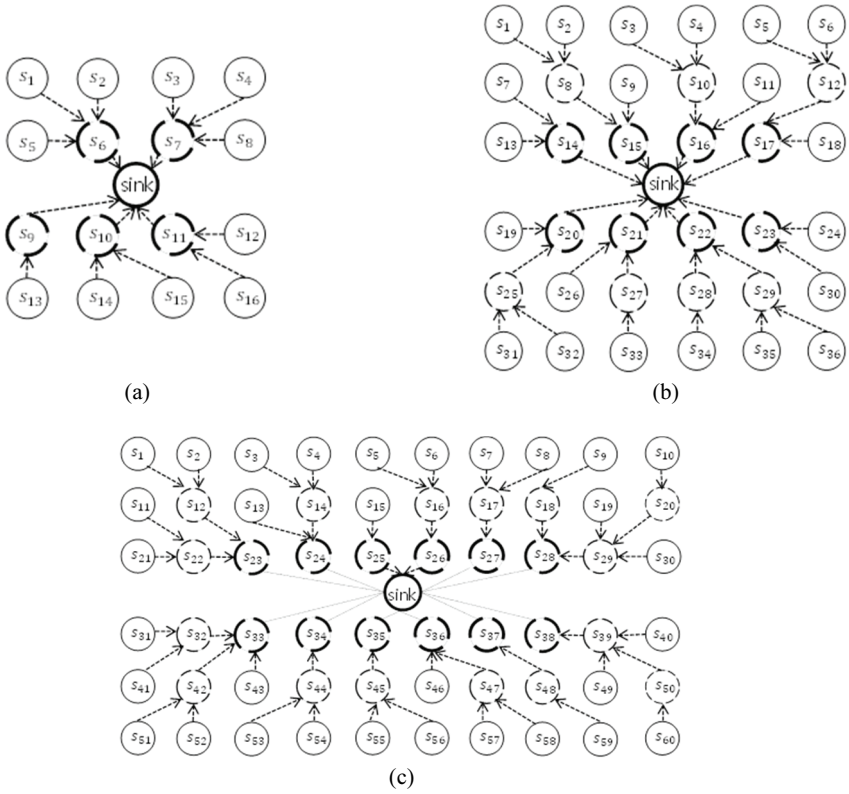


Fig. 2. The MHNC mechanism: (a) an example of applying MHNC, (b) the encoding at s_5, s_2 and s_1, and (c) the decoding at the sink.

4. Performance Evaluation

4.1 Simulation Environments

The OMNET++ CASTALIA-3.3 framework [3] is used for the simulations. The wireless multi-hop sensor network is composed of one sink node and up to 60 SNs. The sink node is placed at the center of the network and SNs are deployed in a grid pattern as shown in Fig. 3, where the parent SN is pointed by the head of an arrow. The T-MAC protocol [9] is used for transmission scheduling among SNs, and only the sink node is kept active during the whole simulation time of 1,000 seconds. Each sensory data is set to the size of 4 bytes. Fig. 3 represents the configuration of the sensor networks used in the simulations.



(a) (b)

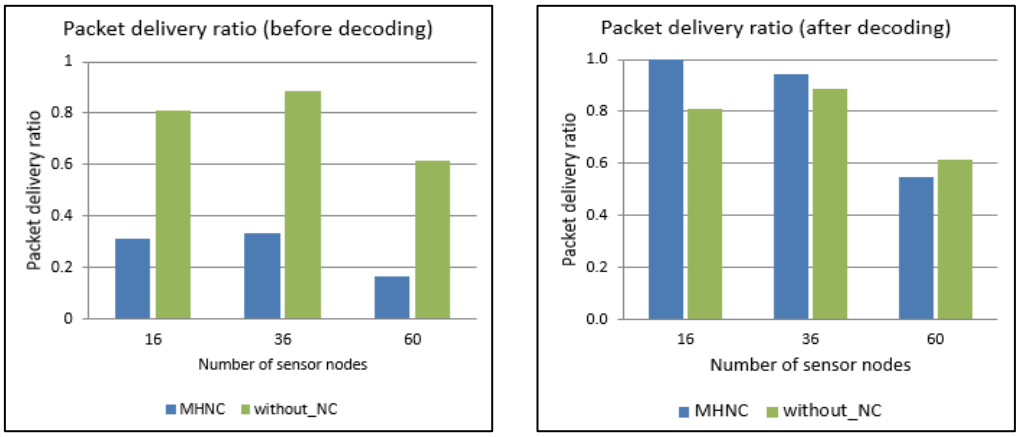
(c)

Fig. 3. The simulation sensor network: (a) with 16 sensor nodes, (b) with 36 sensor nodes, (c) with 60 sensor nodes.

4.2 Simulation Results

The performance of MHNC is compared with that of the conventional transmission mechanism (i.e., transmission with no network coding; we call this the without_NC mechanism). For the analysis of delivery performance, the packet delivery ratios before and after decoding are measured. The packet delivery ratio before decoding (PDRbD) is computed by dividing the number of packets received at the sink by the number of packets generated at the SNs. And the packet delivery ratio after decoding (PDRaD) is obtained by dividing the number of decoded original packets at the sink by the number of packets generated at the SNs. Fig. 4(a) and 4(b) show PDRbD and PDRaD, respectively, for the 16-, 36- and 60-SN networks. For the 16-SN network, MHNC does not suffer from packet losses. As for the 36-SN network, with MHNC, the sink node recovers 33 original packets from 11 encoded packets, and this results in a higher PDRaD than the PDR of without_NC. However, for the 60-SN network, the PDRaD of MHNC is lower than the PDR of without_NC. We can deduce, from these results, the tendency of the sensor network with more SNs (with more levels) lowering PDRaD. Thus, it is recommended to use MHNC in small-scale WSNs.

From the perspective of time-wise data delivery performance, data delivery time, which is the time required to send all the sensory data packets from all the SNs to the sink during one transmission period, is measured. As shown in Fig. 5, data delivery time increases as the number of nodes increases in both of the mechanisms. In all the cases, without_NC suffers from longer data delivery time than MHNC because without_NC transmits more packets than MHNC.



(a) (b)

Fig. 4. Packet delivery ratio vs. the number of sensor nodes: (a) packet delivery ratio before decoding and (b) packet delivery ratio after decoding.

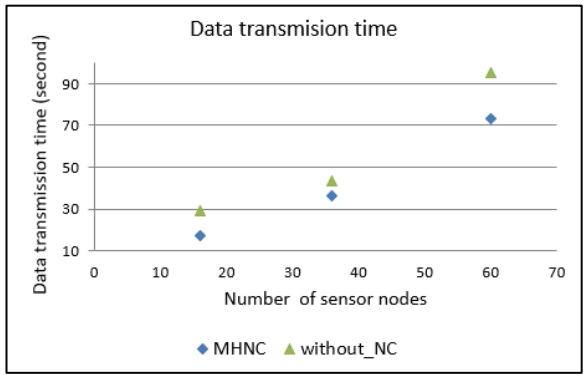
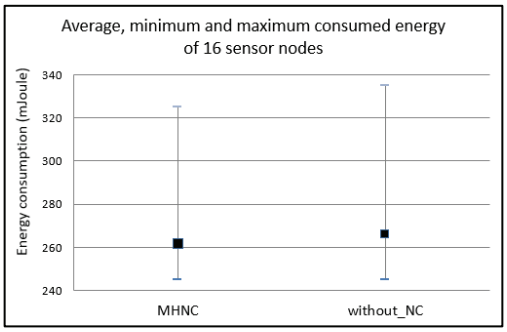


Fig. 5. Data delivery time vs. the number of sensor nodes.

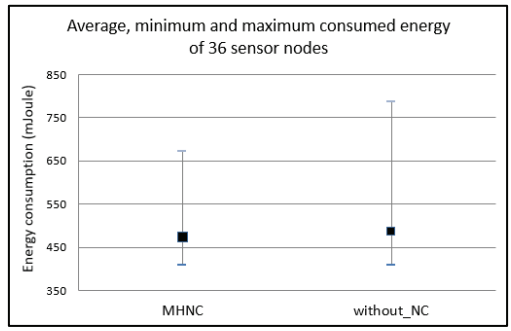
In harsh environments with SNs operating with batteries, which is common situations, it is impossible or very hard to replace or recharge batteries. Therefore, energy efficiency is very critical in WSNs. Fig. 6 shows the average, the minimum and the maximum consumed energy of a SN. Energy is consumed more in a larger sensor network because a SN in a larger network transmits more data, resulting in more energy consumption.

Fig. 7 is the graph illustrating the energy consumption pattern of each node in the 16-SN network by arranging the nodes on the x-axis. The sink node is denoted by 0 on the x-axis and each SN is denoted by its subscript in Fig. 3(a) on the x-axis. As we can easily anticipate, the sink consumes energy the most, because it is kept active during the transmission period, and the leaf nodes consume energy the least, because they do not have data packets from their children to forward.

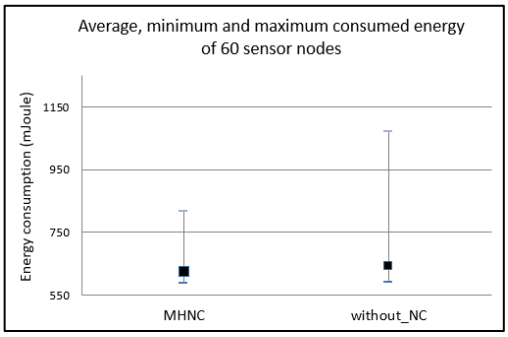
From the simulation results, we have observed that MHNC performs well in terms of packet delivery ratio, data delivery time and energy consumption, in all aspects. Because of its tendency of degraded packet delivery ratio in large-scale WSNs, it is recommended that MHNC is suitable for small-scale WSNs. However, since MHNC has the advantage of providing confidentiality, MHNC can be used in large-scale WSNs where sensory data are periodically generated and slight degradation of packet delivery ratio is insignificant or tolerable.



(a)



(b)



(c)

Fig. 6. Energy consumption of a sensor node: (a) the sensor network with 16 sensor nodes, (b) the sensor network with 36 sensor nodes, and (c) then sensor network with 60 sensor nodes.

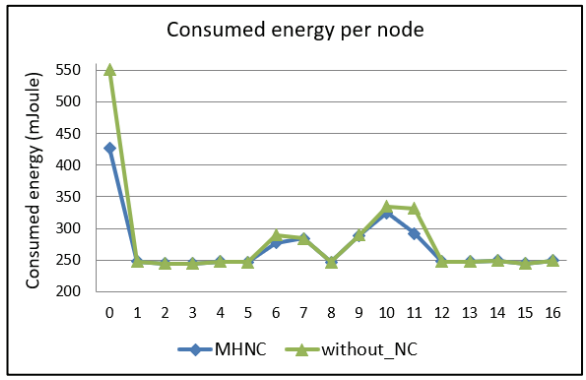


Fig. 7. Energy consumed at each node in the sensor network with 16 sensor nodes.

5. Conclusion

In this paper, MHNC, an energy-efficient confidential convergecast mechanism based on inter-flow RLNC, is proposed for multi-hop WSNs. MHNC is designed for periodic sensory data transmissions from SNs to the sink node with ensuring the confidentiality of sensory data. Confidentiality is provided by not transmitting the RLNC coefficients along with encoded packets. According to the simulation results, the sound operation of MHNC is validated from the perspective of packet delivery ratio, data

delivery time and energy efficiency. Overall, MHNC outperforms the naïve mechanism without using RLNC, except for slight degradation of packet delivery ratio for the 60-SN network case. Thus, we recommend to use MHNC in WSNs that can endure very few packet losses due to periodic sensory data transmissions. Even though the decoding procedure of MHNC is simple, the encoding procedure generates larger encoded packets as it is performed at SNs closer to the sink. The issue of the encoded packet size of MHNC is left for future research.

Acknowledgement

This work was supported by 2020 Research Fund of the University of Seoul.

References

[1] L. Cheng, L. Kong, Y. Gu, J. Niu, T. Zhu, C. Liu, S. Mumtaz, and T. He, "Collision-free dynamic convergecast in low-duty-cycle wireless sensor networks," *IEEE Transactions on Wireless Communications*, vol. 21, no. 3, pp. 1665-1680, 2022. http://dio.org/10.1109/TWC.2021.3105983

[2] X. Zhou, X. Yang, J. Ma, I. Kevin, and K. I. K. Wang, "Energy-efficient smart routing based on link correlation mining for wireless edge computing in IoT," *IEEE Internet of Things Journal*, vol. 9, no. 16, pp. 14988-14997, 2022. https://doi.org/10.1109/jiot.2021.3077937

[3] M. Kumar and S. Z. Hussain, "Simulation model for wireless body area network using Castalia," in *Proceedings of 2022 1st International Conference on Informatics (ICI)*, Noida, India, 2022, pp. 204-207. http://dio.org/10.1109/ICI53355.2022.9786924

[4] A. Akbar, M. B. Malik, J. Qureshi, and M. Awais, "MAC aware random network coding for wireless unicast flows," in *Proceedings of 2020 14th International Conference on Open Source Systems and Technologies (ICOSST)*, Lahore, Pakistan, 2020, pp. 1-6. https://doi.org/10.1109/icosst51357.2020.9332803

[5] M. A. Brahimi, F. Merazka, and G. K. Kurt, "Secure network coding for data encoded using subspace codes," *Physical Communication*, vol. 48, article no. 101408, 2021. https://doi.org/10.1016/j.phycom.2021.101408

[6] O. A. Khashan, R. Ahmad, and N. M. Khafajah, "An automated lightweight encryption scheme for secure and energy-efficient communication in wireless sensor networks," *Ad Hoc Networks*, vol. 115, article no. 102448, 2021. https://doi.org/10.1016/j.adhoc.2021.102448

[7] P. Prakasam, M. Madheswaran, K. P. Sujith, and M. S. Sayeed, "An enhanced energy efficient lightweight cryptography method for various IoT devices," *ICT Express*, vol. 7, no. 4, pp. 487-492, 2021. https://doi.org/10.1016/j.icte.2021.03.007

[8] C. Marcolla, V. Sucasas, M. Manzano, R. Bassoli, F. H. Fitzek, and N. Aaraj, "Survey on fully homomorphic encryption, theory, and applications," *Proceedings of the IEEE*, vol. 110, no. 10, pp. 1572-1609, 2022. https://doi.org/10.1109/jproc.2022.3205665

[9] R. S. Cotrim, J. M. L. P. Caldeira, V. N. Soares, and Y. Azzoug, "Power saving MAC protocols in wireless sensor networks: a survey," *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, vol. 19, no. 6, pp. 1778-1786, 2021. http://doi.org/10.12928/telkomnika.v19i6.19148

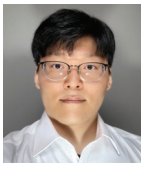
[10] L. Alkama and L. Bouallouche-Medjkoune, "IEEE 802.15.4 historical revolution versions: a survey," *Computing*, vol. 103, pp. 99-131, 2021. https://doi.org/10.1007/s00607-020-00844-3

Davaabayar Ganchimeg **https://orcid.org/0009-0007-1819-7015**

She received the B.S. degree in Computer Science in Hardware Engineering from Mongolian University of Science and Technology in 2014, and the M.S. degree in computer science from University of Seoul, Korea, in 2020. She is currently a network engineer at Buren Zuraglal, Ulaanbaatar, Mongolia.

Sanghyun Ahn **https://orcid.org/0000-0001-7640-4480**

She received the B.S. and M.S. degrees in computer engineering from Seoul National University, Seoul, Korea, in 1986 and 1988, respectively, and received the Ph.D. degree in computer science from University of Minnesota in 1993. She is currently a professor in the School of Computer Science and Engineering, University of Seoul, Seoul, Korea. Her research interests include wireless ad hoc, sensor and vehicular networks, Internet protocols, routing protocols, IoT, etc.



Minyeong Gong **https://orcid.org/0000-0002-9777-9895**

He received the B.S. and Ph.D in electrical and computer engineering from the University of Seoul, Korea, in 2016 and 2023, respectively. He has been engaged in the research of Department of Computer Science and Engineering. His research interests include vehicle networks, IoT, and cloud computing.