Journal of Electrical Engineering, Vol. 77, No. 1, 2026, pp. 97-104
https://reference-global.com/journal/JEE

Computational optimality of combined sweeping-jumping algorithm for connectivity-based node insertion in Wi-Fi sensor networks

Woo-Yong Choi

The connectivity-based Medium Access Control (MAC) protocol leverages node connectivity in a wireless LAN to derive an optimal polling sequence, enabling the Access Point (AP) to utilize wireless bandwidth more efficiently when polling each node. However, when the Constraint Satisfaction Problem (CSP) algorithm is applied to generate the optimal sequence, the required computation time varies with the connectivity characteristics of the network; in some cases, such as a Wi-Fi sensor network with 1,000 sensors, it takes excessively long (over 14 seconds on average). To overcome this limitation, a new polling sequence derivation method, called the combined sweeping-jumping algorithm, has been developed. In this paper, we theoretically and experimentally demonstrate that the combined sweeping-jumping algorithm is the computationally optimal solution for the connectivity-based node insertion method.

Keywords: Wi-Fi network, MAC protocol, connectivity, node insertion method

1 Introduction

Since the development of Wi-Fi wireless networks in 1997, infrastructure-based Wi-Fi networks have made remarkable progress, being intensively deployed in hotspot areas with large crowds, and have greatly contributed to the advancement of an information-oriented society [1-3]. Furthermore, efforts have continued to extend Wi-Fi networks into wireless sensor networks in order to develop technologies that enable the application of Internet of Things (IoT) solutions in society [4-12].

In [4], an optimal power saving mechanism based on the Distributed Coordination Function (DCF) protocol was proposed. In [5, 6], a new method was proposed to divide nodes connected to an Access Point (AP) into multiple groups in order to reduce the degradation of wireless transmission efficiency caused by transmission collisions when many nodes are connected to the AP. In [7, 8], a Medium Access Control (MAC) protocol was proposed in which the AP efficiently selects pseudo-APs based on connectivity information among nodes, thereby allowing nodes with insufficient battery energy, which cannot directly transmit data to the AP, to transmit data via pseudo-APs. A new MAC protocol that enhances the DCF protocol was proposed to enable IoT-based services in [9]. A new Wireless Local Area Network (WLAN) MAC protocol was proposed for constructing underwater wireless sensor networks in [10]. Potential application services utilizing wireless sensor networks are classified, and the routing protocols, and the energy supply methods required for sensor networks were comprehensively reviewed in [11]. MAC, routing, and security protocols required in wireless sensor networks for medical application services were reviewed in [12]. In addition, various MAC protocols for wireless sensor networks based on Wireless Personal Area Networks (WPANs) such as Bluetooth and Zigbee have been proposed [13-17].

However, in the research on extending Wi-Fi networks to wireless sensor networks, relatively little work has been done on optimizing the Point Coordination Function (PCF) protocol – one of the fundamental MAC protocols developed in 1997 – for use in wireless sensor networks. The PCF protocol was originally developed as a MAC protocol for real-time communication. Compared to another fundamental MAC protocol, DCF, which was developed in 1997, it has the advantage of being unaffected by the hidden node problem. Furthermore, while the DCF protocol suffers from excessive wireless channel waste due to overly long backoff times when many sensors attempt to transmit data to the AP, the PCF protocol fundamentally prevents unnecessary transmission delays even as the number of sensor nodes attempting to transmit to the AP increases. Based on these advan-

Department of Industrial and Management Systems Engineering, Dong-A University,
Nakdong-daero, Saha-gu, Busan, Korea
wychoi77@dau.ac.kr

https://doi.org/10.2478/jee-2026-0010, Print (till 2015) ISSN 1335-3632, On-line ISSN 1339-309X

tages, research has been conducted on developing new MAC protocols based on the PCF protocol suitable for Wi-Fi-based wireless sensor networks, utilizing connectivity information between nodes [18-21].

According to [21], the combined sweeping-jumping algorithm derives a pseudo-optimal polling sequence within only a few milliseconds even when up to 1,000 sensors are connected to an AP and attempting to transmit data under the connectivity-based MAC protocol, which was developed as an enhancement of the conventional PCF protocol. In this paper, we aim to theoretically and experimentally demonstrate that the combined sweeping-jumping algorithm achieves the optimal computational complexity for the connectivity-based node insertion method in Wi-Fi sensor networks.

In the following section, we describe the connectivity-based node insertion method for Wi-Fi sensor networks and the combined sweeping-jumping algorithm. In Section 3, we theoretically prove that the combined sweeping-jumping algorithm is the optimal algorithm for the connectivity-based node insertion method in terms of computational complexity. Section 4 presents the simulation results of the number of computations required by the combined sweeping-jumping algorithm as the number of nodes in the Wi-Fi sensor network increases, along with experiments related to the computational requirements of the combined sweeping-jumping algorithm. Finally, in Section 5, we draw conclusions and present directions for future research.

2 Combined sweeping-jumping algorithm for node insertion method

In a Wi-Fi sensor network, an AP and a set of sensors are deployed. In accordance with the specifications of wireless LAN standards, each sensor is permitted to attempt uplink channel access for the transmission of data to the AP by adhering to a MAC protocol by which the Wi-Fi sensor network governed. Data exchange between the AP and the sensors within such network may be realized under either contention-based or polling-based MAC protocols. In the present study, it is assumed that both the AP and the sensors operate under a polling-based MAC protocol for data transmission.

As indicated in the introduction, the DCF protocol is subject to a critical limitation: when the number of sensors attempting transmission to the AP is large, a high frequency of packet collisions occurs, resulting in substantial wireless channel inefficiency due to repeated backoff processes. Conversely, in the polling-based MAC protocol, such inefficiency is not incurred, even under conditions of heavy transmission demand from numerous sensors. Moreover, the polling-based approach eliminates the wireless channel degradation caused by the hidden node problem, which constitutes a further drawback inherent to the DCF protocol.

In polling-based MAC protocols, when the AP transmits the next polling frame via the downlink, it may either piggyback the corresponding data within the polling frame or deliver the downlink data frame directly to the target sensor. Each sensor is permitted to transmit data to the AP only upon reception of a polling frame; thus, sensors that do not receive such a frame are denied transmission opportunities. Consequently, the efficiency of the AP's polling frame transmission strategy plays a critical role in improving MAC performance by ensuring the effective utilization of the wireless transmission channel, particularly in environments where a large number of sensors exist within a Wi-Fi sensor network. The transmission of data from the AP to individual sensors is characterized by a relatively straightforward operational procedure. Upon the termination of an ongoing transmission sequence, data may be conveyed, at the discretion of the AP, either by being incorporated into a polling frame, as delineated previously, or by being delivered through a dedicated data frame. Given the inherent simplicity of this transmission mechanism, the principal emphasis of the present study is placed upon the downlink data transmission procedure.

In this context, this section introduces the PCF protocol, which was the earliest polling-based protocol proposed in [3], as well as the connectivity-based MAC protocol described in [18-20]. Furthermore, this section discusses the underlying mechanisms that enable the connectivity-based MAC protocol, including the Constraint Satisfaction Problem (CSP), the node insertion technique, and the combined sweeping-jumping algorithm.

According to the definition of the PCF protocol provided in [3], when data transmission between each sensor and the AP is conducted under the PCF protocol, the AP dispatches polling frames to the sensors sequentially, following a round-robin scheduling mechanism. Upon reception of a polling frame, each sensor is permitted to transmit a response frame. This response frame is classified into two categories: a data frame, transmitted when the sensor has data to forward to the AP, and a null frame, transmitted when the sensor possesses no data for transmission. If a sensor fails to deliver a response frame to the AP within the designated duration of one Short Inter-Frame Space (SIFS), the AP assumes that the polling procedure has failed. In such an event, the AP proceeds with the error-recovery process by transmitting a polling frame to the subsequent sensor in the polling sequence.

According to the connectivity-based MAC protocol described in [18, 19], the AP initializes sensor connectivity information through the conventional PCF protocol. This information is subsequently updated via the connectivity-based MAC protocol to reflect network characteristics. Connectivity information indicates whether a sensor can successfully receive and decode another sensor's transmission. Based on this protocol, the AP optimizes the polling procedure for uplink transmission opportunities by employing a multi-polling scheme. To implement multi-polling, the AP constructs a polling sequence in which each sensor, whenever possible, follows another whose transmission it can overhear. This design minimizes the number of multi-polling frames required to ensure transmission opportunities for all sensors. Specifically, when a multi-polling frame contains a sequence of multiple sensor MAC addresses, and each sensor can overhear the transmission of its predecessor, a single frame transmission suffices to efficiently grant uplink access to all sensors in the sequence.

To derive the optimal polling sequence for the n sensors deployed in a Wi-Fi sensor network, the CSP algorithm based on the Traveling Salesman Problem (TSP) can be employed. However, depending on the characteristics of the connectivity information among sensors within the network, the CSP algorithm has the drawback that it may require an excessively long computation time to obtain the optimal polling sequence. As a result, experimental evaluations with 1,000 sensors demonstrated that the CSP algorithm required, on average, more than 14 seconds to determine the optimal polling sequence [20]. The fundamental cause of this inefficiency lies in the reliance on backtracking, a process in which the most recently added sensor in the currently constructed polling sequence is removed to further refine the solution. Since the CSP algorithm constructs a polling sequence by considering only the connectivity information between the most recently added sensor and the unselected sensors at each step of the search, backtracking – namely, deleting the most recently added sensor to revise the sequence – is an unavoidable step in the search process. Therefore, unless this backtracking procedure is eliminated, it is inherently difficult to resolve the problem efficiently.

To overcome the limitations of the CSP algorithm, a novel node insertion method was proposed that utilizes connectivity information between recently added sensors in the polling order and the remaining nodes during the search phase, together with an implementation algorithm [20]. In this method, the polling order is constructed by selecting as the next node the one that can receive signals from the largest number of recently added nodes, thereby eliminating the need for backtracking to remove previously selected nodes from the polling sequence [20].

The node addition process according to the node insertion method can be described more precisely as follows. Suppose that at the stage of inserting a new node, the number of nodes already configured is r, denoted as $d_1, d_2, \ldots, d_r$, and that these nodes have been added in the order of $d_1, d_2, \ldots, d_r$. For any node j not yet inserted, the value D_j can be determined as follows. If node j can receive the transmission signals from all of the nodes $d_1, d_2, \ldots, d_r$, then $D_j = 0$. Otherwise, D_j for node j is defined as an index i that satisfies the following condition:

- Node j is unable to detect the signal from node d_i, while it can successfully receive the signals from nodes $d_{i+1}, d_{i+2}, \ldots, d_r$.

According to the node insertion method, assuming that nodes $d_1, d_2, \ldots, d_r$ have been inserted, the next node j to be added is selected from the set of nodes not yet inserted, such that node j has the minimum value of D_j.

The new combined sweeping-jumping algorithm, which further improves the node insertion method proposed in [20], was introduced in [21]. According to the proposed combined sweeping-jumping algorithm, as illustrated in Fig. 1, the values of D_j for nodes not yet inserted are searched, and whenever a new node represented by a circle is added, the D_j values are continuously updated to reflect the connectivity status between recently inserted nodes in the current polling sequence and each not-yet-inserted node j. As explained earlier, D_j represents the insertion order of the most recently added node in the polling sequence from which node j cannot receive transmission signals. Therefore, a smaller value of D_j indicates that node j has stronger connectivity with recently inserted nodes in the current polling sequence. In particular, if $D_j = 0$, it implies that node j can receive signals from all nodes already included in the current polling sequence.

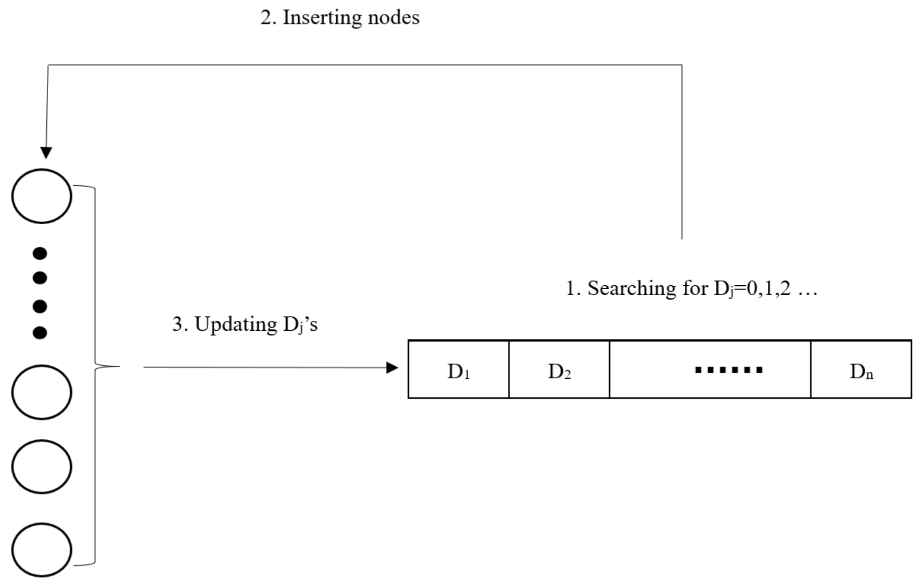

Fig. 1. Flow of sweeping algorithm

According to the combined sweeping-jumping algorithm, all nodes with $D_j = 0$ are sequentially added to the polling order, followed by all remaining nodes with $D_j = 1, 2, \ldots$, which are also sequentially included in the order as the value of D_j increases by one. Each time a node is added to the polling order, the D_j values of all nodes j that have not yet been included must be continuously updated. Based on the updated D_j values, new nodes are subsequently added to the polling order. This process of iteratively adding nodes is referred to as the *sweeping process*. To further accelerate node insertion during this sweeping process, the *jumping process* was proposed in [21]. However, whether the algorithm incorporates the jumping process does not affect the optimality of the combined sweeping-jumping algorithm, as its optimality is guaranteed solely by the sweeping process.

3 Optimality of combined sweeping-jumping algorithm

In this section, we aim to prove that the combined sweeping-jumping algorithm, introduced as an implementation approach for the node insertion method described in Section 2, constitutes the optimal algorithm for the node insertion method.

Consider a Wi-Fi sensor network consisting of n nodes, where an AP is responsible for collecting the connectivity information among the nodes. Based on the acquired connectivity data, the AP seeks to execute the node insertion method by employing a designated algorithm. The only assumption imposed in this model is that each node and the AP remain within each other's transmission range to ensure mutual connectivity. No additional constraints are presumed; for example, it is not assumed that the AP is located at the geometric center of the service region, nor is any specific spatial distribution of nodes within the service area required. Accordingly, the connectivity information among the nodes is regarded as being obtained under highly general conditions.

In applying the proposed algorithm to perform the node insertion method for n nodes, the procedure begins by inserting the first node into the multipolling sequence. Subsequently, nodes $d_1, d_2, \ldots, d_{m1}$ are sequentially inserted up to the m_1-th node, where each newly inserted node is assumed to be capable of receiving the transmission signals from all previously inserted nodes. Beyond this point, none of the remaining uninserted nodes can receive the complete set of transmission signals from the already inserted m_1 nodes. In the special case where $m_1 = 1$, this condition indicates that, after insertion of the first node, there exists no other node that can receive its transmission signal. Accordingly, the minimum number of operations, denoted as p_1, required by the algorithm up to the insertion of the m_1 nodes is given by

$$p_1 = 1 + 2 + \ldots + (m_1 - 1). \quad (1)$$

This is because each newly added node is required to perform the operations that verify the connectivity information between itself and all previously added nodes.

After the insertion of m_1 nodes, the remaining nodes that have not yet been inserted cannot receive the transmission signals from all of the already inserted m_1 nodes. In order to verify this situation by means of the node insertion method, each of the remaining $n - m_1$ nodes must at least perform a connectivity check with the m_1-th inserted node. Therefore, the minimum number of operations required to insert the (m_1+1)-th node using the given algorithm via the node insertion method is $n - m_1$.

Furthermore, let us assume that, under the given algorithm, up to the point when the m_2-th node is inserted, each inserted node can receive the transmission signals from the (m_1+1)-th node as well as from all subsequently inserted nodes that precede it. Then, similar to Eqn. (1), the minimum number of operations required to insert the m_2-th node, starting from the (m_1+1)-th node, can be computed as follows:

$$p_2 = (n - m_1) + 1 + 2 + \ldots + (m_2 - m_1 - 1) \quad (2)$$

Let the added nodes up to the m_1-th node be grouped as the first cluster, and the added nodes from the (m_1+1)-th node to the m_2-th node be grouped as the second cluster. In this manner, by performing the node insertion method using the given algorithm until all n nodes are added, assume that a total of c clusters can be formed. The minimum number of operations required to construct the i-th cluster can then be derived as follows:

$$p_i = (n - m_{i-1}) + 1 + 2 + \ldots + (m_i - m_{i-1} - 1). \quad (3)$$

The minimum number of operations required to add all n nodes can thus be derived in a modified form as follows:

$$p = p_1 + p_2 + \ldots + p_c = 1 + 2 + \ldots + (m_1 - 1) + 1 + 2 + \ldots + (m_2 - m_1) + 2 + 3 + \ldots + (m_3 - m_2 + 1) + 3 + 4 + \ldots + (m_4 - m_3 + 2) + \ldots + (c - 1) + c + (c + 1) + \ldots + (c + m_c - m_{c-1} - 2) \quad (4)$$

For example, when the algorithm is applied to 18 nodes, the first cluster consists of 5 nodes, the second cluster consists of 6 nodes, and the third cluster consists of 7 nodes. Accordingly, we have m_1=5, m_2=11, and m_3 = 18. By applying Eqns. (1), (2), and (3), the values of p_1, p_2, p_3, and $p = p_1 + p_2 + p_3$ are obtained as follows:

$$p_1 = 1 + 2 + 3 + 4 = 10,$$
$$p_2 = (18 - 5) + 1 + 2 + \ldots + 5 = 28,$$
$$p_3 = (18 - 11) + 1 + 2 + \ldots + 6 = 28,$$
$$p = 10 + 28 + 28 = 66.$$

However, even if the minimum number of operations required to add all 18 nodes is obtained through Eqn. (4), the same value can also be derived as follows:

$$p = 1 + 2 + 3 + 4 + 1 + 2 + 3 + \ldots + 6 + 2 + 3 + \ldots + 8 = 66$$

When dividing n nodes into c clusters, how many nodes should be included in each cluster in order to minimize the minimum number of operations p defined in Eqn. (4)? The motivation for this consideration is to determine the minimum number of operations required when inserting n nodes into the multipolling sequence using the node insertion method. For example, in the case of 18 nodes, $p_1 = 1 + 2 + 3 + 4$ represents the minimum number of operations needed to form the first cluster, p_2= 1 + 2 + 3 + ⋯+ 6 represents the minimum number of operations for the second cluster, and p_3 = 2 + 3 + 4 + ⋯+ 8 represents the minimum number of operations for the third cluster. To minimize the total minimum number of operations, $p = p_1 + p_2 + p_3$, it is advantageous to make the final terms in the calculations of p_1, p_2, and p_3 as close as possible. Accordingly, assigning 7 nodes to the first cluster, 6 nodes to the second cluster, and 5 nodes to the third cluster achieves this goal. A graphical representation of this clustering for the 18 nodes is shown in Fig. 2, where each circle denotes an inserted node; the first number inside the circle indicates the order of occurrence of the node, and the second number represents the minimum number of operations required to add that node.

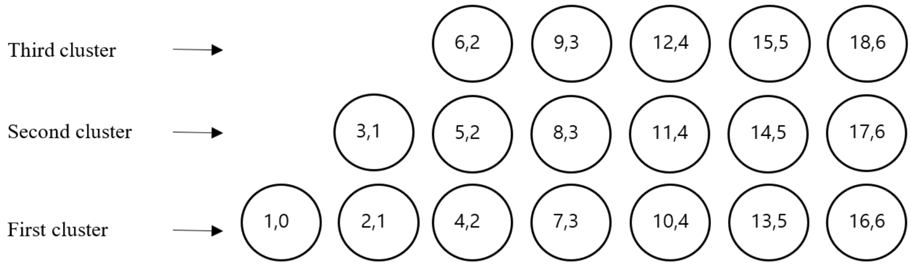

Fig. 2. Minimized clustering of 18 nodes into three clusters

Accordingly, $m_1 = 7$, $m_2 = 13$, and $m_3 = 18$. Thus, by applying Eqn. (4), the required minimum number of operations $p_{\min}$ is recalculated as follows:

$$p_{\min} = p_1 + p_2 + p_3 = 1 + 2 + \ldots + 6 + 1 + 2 + \ldots + 6 + 2 + 3 + \ldots + 6 = 0 + (2 - 1)2 + (3 - 1)3 + (3)3 + (3 + 1)3 + (3 + 2)3 + (3 + 3)3 = 62.$$

If the number of nodes n that exist in a Wi-Fi sensor network increases continuously according to the following equation,

$$n = 1 + 2 + 3 + 4 + \ldots + c + c + c + c + \ldots \quad (5)$$

we can always match the final number for the calculation of the minimum number of operations p_i for c clusters, and accordingly, the required minimum number of operations $p_{\min}$ can be derived as follows:

$$p_{\min} = 0 + (2 - 1)2 + (3 - 1)3 + (4 - 1)4 + \ldots + (c - 1)c + (c)c + (c + 1)c + (c + 2)c + \ldots . \quad (6)$$

According to Eqns. (5) and (6), for example, when the number of clusters is $c = 5$ and $n = 1 + 2 + 3 + 4 + 5 + 5 + 5 + 5 = 30$, the minimum required number of operations is given by

$$p_{\min} = 0 + (2-1)2 + (3-1)3 + (4-1)4 + (5-1)5 + (5)5 + (5+1)5 + (5+2)5 = 130.$$

Starting from the (c+1)-th term in Eqns. (5) and (6), as n increses, the minimum required number of operations $p_{\min}$ also increases. It can be observed that for some natural number k, when n increases by $n^+ = k \cdot c$ from the (c+1)-th term according to Eqn. (5), the corresponding increment in $p_{\min}$ is given by

$$p_{\min}{}^+ = k \cdot c \cdot \mathrm{c} + c \cdot k(k + 1) / 2 = c \cdot n^+ + (n^+ \cdot n^+ + c \cdot n^+) / (2c) = 0.5\, n^+ + c \cdot n^+ + 0.5(n^+ \cdot n^+) / \mathrm{c} \quad (7)$$

which represents the growth pattern of $p_{\min}$. It can be observed that for a finite number of clusters c, $p_{\min} = O(n^2)$. As discussed in [21], the combined sweeping-jumping algorithm has a computational complexity of $O(n^2)$. Therefore, the combined sweeping-jumping algorithm is the optimal algorithm for the node insertion method.

In this section, assuming that in a Wi-Fi sensor network with n nodes, regardless of how efficiently the node insertion method proposed in [20] is implemented, the number of required clusters c generated by the node insertion method increases only up to a finite upper bound as n grows and does not increase further, it is proven that the required minimum computational cost becomes $O(n^2)$. Furthermore, as discussed in [21], since the computational complexity of the combined sweeping-jumping algorithm for implementing the node insertion method is $O(n^2)$, it is thereby proven that when the number of required clusters c remains finite even as the number of nodes n increases, the combined sweeping-jumping algorithm constitutes the optimal algorithm for implementing the node insertion method.

If the number of required clusters c for the node insertion method diverges to infinity as the number of nodes n increases, then the combined sweeping-jumping algorithm cannot be regarded as a proven optimal algorithm for implementing the node insertion method, and therefore further research is necessary. However, since the problem that the node insertion method aims to solve also includes Wi-Fi networks in which the number of required clusters c remains finite even as the number of nodes increases, the combined sweeping-jumping algorithm can be considered the optimal algorithm for implementing the node insertion method from the perspective of solving all such problems.

The concept of a cluster defined in this section differs from that of the connected polling sequence defined in [18, 19]. A single multipolling sequence can be divided into multiple clusters and connected polling sequences. Within a cluster, any newly added node must be able to receive the transmission signals of all previously added nodes in the cluster, whereas within a connected polling sequence, a newly added node is only required to receive the transmission signal from the immediately preceding node. Therefore, for *n* nodes, the number of clusters derived using the node insertion method is greater than the number of connected polling sequences derived.

In a Wi-Fi sensor network, it is straightforward to construct an example where the number of nodes *n* diverges to infinity while the number of clusters *c*, derived through the node insertion method, remains finite. For instance, assume that a finite number of fixed positions are designated within a given service area, and nodes are continuously placed at these fixed positions so that the total number of nodes *n* grows without bound. Furthermore, suppose that all nodes share the same data transmission radius. When the node insertion method is applied to such a Wi-Fi sensor network, the number of induced clusters can be shown to remain less than or equal to the number of fixed positions, regardless of the growth of *n*. This is because, according to the node insertion method, when adding a node to the multipolling sequence, although nodes located at different fixed positions within the same cluster may be inserted, the method first prioritizes nodes that fall within the transmission range of previously inserted nodes. Consequently, once a node from a given fixed position is included in a cluster, all nodes at that position will inevitably be incorporated into the same cluster. Moreover, as long as the number of possible locations remains constant, the number of clusters induced by the node insertion method will not exceed the number of fixed positions, regardless of how nodes are added or whether their exact occurrence positions shift among those fixed positions.

4 Simulation results

In this section, we present simulation results for the number of iterations required, the number of derived connected polling sequences, and the number of resulting clusters when applying the connectivity-based MAC protocol proposed in [18, 19], to a Wi-Fi sensor network, under the assumption that the combined sweeping-jumping algorithm is employed to derive the necessary multipolling sequence. To this end, we consider a Wi-Fi sensor network with a circular service area of radius r, where the AP is located at the center of the area and the number of nodes is given as $n = 100$, 200, 300, …, 1000. As in [21], each node is assumed to have a circular transmission range, with its transmission radius set to one of the six values r, $r + 0.1r$, $r + 0.2r$, $r + 0.3r$, $r + 0.4r$, or $r + 0.5r$. These six transmission ranges are uniformly assigned to the nodes. Furthermore, each node is assumed to be randomly located within the Wi-Fi sensor network.

For each case of $n = 100$, 200, 300, …, 1000, one hundred random networks were generated. For each network, a near-optimal multipolling sequence was derived using the combined sweeping-jumping algorithm. During this process, the number of operations required to obtain the multipolling sequence was recorded, and the resulting multipolling sequence was decomposed into the corresponding connected polling sequences and clusters. Figures 3 and 4 illustrate, for a representative Wi-Fi sensor network, the average number of required operations, the average number of induced connected polling sequences, and the average number of induced clusters across different values of *n*.

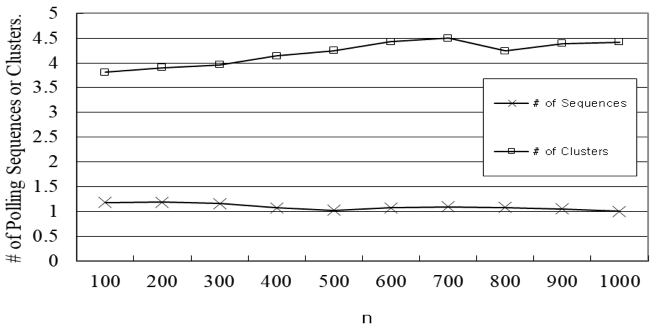

Fig. 3. Average number of connected polling sequences and clusters

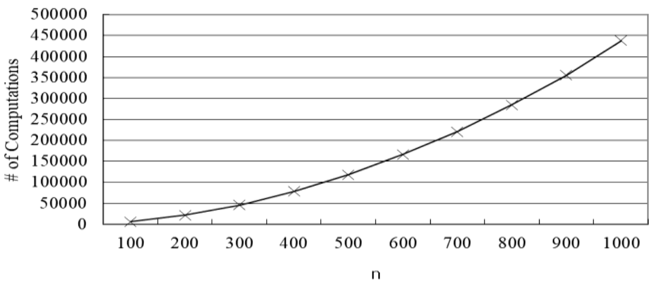

Fig. 4. Average number of computations

Examining Fig. 3, it can be observed that the number of induced clusters increases slightly as *n* increases. However, the number of clusters does not increase indefinitely with larger *n*, but rather converges to a certain limit beyond which no further growth occurs. Furthermore, as discussed in Section 3, for a given multipolling sequence, the number of induced clusters is greater than or equal to that of the connected polling sequence. Turning to Fig. 4, it is shown that as *n* increases, the number of operations required by the combined sweeping-jumping algorithm grows in the form of the function $f(n) = a \cdot n^2$ for a certain constant a. This result provides experimental support for the discussion in [21], which established that the computational complexity of the combined sweeping-jumping algorithm is $O(n^2)$.

5 Conclusions

The connectivity-based MAC protocol is a multi-polling-based MAC protocol designed to minimize the polling overhead of the access point (AP) in Wi-Fi sensor networks while efficiently granting uplink transmission opportunities to each sensor. However, as the number of sensors in a Wi-Fi sensor network increases, optimizing the multipolling sequence becomes a time-consuming and challenging problem. To address this issue effectively, the node insertion method has been proposed, and the combined sweeping-jumping algorithm has been introduced as an implementation of the node insertion method.

In this paper, we prove that for a Wi-Fi sensor network with n sensors, any algorithm implementing the node insertion method must have at least $O(n^2)$ computational complexity. Based on this result, we show that the combined sweeping-jumping algorithm, which has $O(n^2)$ computational complexity, is the optimal algorithm for implementing the node insertion method. Furthermore, we present simulation results to support this claim.

References

[1] K. Pahlavan and P. Krishnamurthy, "Evolution and Impact of Wi-Fi Technology and Applications: A Historical Perspective", International Journal of Wireless Information Networks, vol. 28, pp. 3-19, 2021.

[2] C. Wu, B. Wang, O.C. Au and K.J.R. Liu, "Wi-Fi Can Do More: Toward Ubiquitous Wireless Sensing", IEEE Communications Standards Magazine, vol. 6, pp. 42-49, 2022.

[3] IEEE Std 802.11, "IEEE Standard for Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications", IEEE Press, 1997.

[4] E.-S. Jung and N.H. Vaidya, "An Energy Efficient MAC Protocol for Wireless LANs", Proc. of Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies, 2002.

[5] IEEE Std 802.11ah, "Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 2: Sub 1 GHz License Exempt Operation", IEEE Press, 2017.

[6] L. Tian, S. Santi, A. Seferagic, J. Lan and J. Famaey, "Wi-Fi HaLow for the Internet of Things: An Up-to-date Survey on IEEE 802.11ah Research", Journal of Network and Computer Applications, vol. 182, 2021.

[7] Woo-Yong Choi, "Energy-efficient MAC protocol for wireless LANs with WiFi sensors", Journal of Electrical Engineering, vol. 72, pp. 532-535, 2021.

[8] Woo-Yong Choi, "Novel MAC routing protocol for wireless sensor networks based on IEEE 802.11 ad hoc networks", Journal of Electrical Engineering, vol. 73, pp. 355-358, 2022.

[9] A. Alnazir, R.A. Mokhtar, H. Alhumyani, E.S. Ali, R.A. Saeed and S. Abdel-khal, "Quality of Services Based on Intelligent IoT WLAN MAC Protocol Dynamic Real-Time Applications in Smart Cities", Computational Intelligence and Neuroscience, vol. 2021, 2287531, 2021.

[10] R. Cv and K. Sathish, "Performance Analysis of Clustered Based Underwater Wireless Sensor Network by Deploying Application as CBR", Proc. Third International Conference on Intelligent Computing Instrumentation and Control Technologies, 2022.

[11] N. Heydarishahreza, S. Ebadollahi, R. Vahidnia and F.J. Dian, "Wireless Sensor Networks Fundamentals: A Review", Proc. IEEE Annual Information Technology, Electronics and Mobile Communication Conference, 2020.

[12] A. Mathur and T. Newe, "Comparison and Overview of Wireless Sensor Network Systems for Medical Applications", International Journal on Smart Sensing and Intelligent Systems, vol. 7, pp. 1–6, 2020.

[13] R. Ahmad, R. Wazirali and T. Abu-Ain, "Machine Learning for Wireless Sensor Networks Security: An Overview of Challenges and Issues", Sensors, vol. 22, 4730, 2022.

[14] K.D. Mathew and A. Jones, "CSRS-MAC: Cluster Based Synchronous Radio Scheduling MAC Protocol Using Carrier Sense Multiple Access for Wireless Sensor Network", Wireless Personal Communications, vol. 126, pp. 209-229, 2022.

[15] L. Goudjil and F. Semchedine, "COSFI-RIMAC: A Cooperative Short Frame Identifier Receiver Initiated MAC Protocol for Wireless Sensor Network", Wireless Personal Communications, vol. 128, pp. 2379-2395, 2023.

[16] J. Bhar and I. Bouazzi, "Elastic Hybrid MAC Protocol for Wireless Sensor Networks", International Journal of Electrical & Computer Engineering, vol. 11, pp. 4174-4182, 2021.

[17] M. Tolani, A. Bajpai, A. Balodi, S. Shama and R.K. Singh, "Energy-Efficient Duty-Cycle Hybrid Medium Access Control Protocol for Wireless Sensor Network", Wireless Personal Communications, vol 131, pp. 609-621, 2023.

[18] Woo-Yong Choi, "Hybrid Polling Method for Direct Link Communication for IEEE 802.11Wireless LANs", EURASIP Journal on Wireless Communications and Networking, vol. 2008, Article ID 598038, 2008.

[19] Woo-Yong Choi, "Connectivity-Based Reliable Multicast MAC Protocol for IEEE802.11Wireless LANs", EURASIP Journal on Wireless Communications and Networking, vol. 2009, Article ID 968408, 2009.

[20] Woo-Yong Choi, "Efficient Node Insertion Algorithm for Connectivity-Based Multipolling MAC Protocol in Wi-Fi Sensor Networks", Applied Sciences, vol. 13, 11974, 2023.

[21] Woo-Yong Choi, "Combined Sweeping and Jumping Method to Enhance Node Insertion Algorithm for Wi-Fi Sensor Networks", Applied Sciences, vol. 14, 7762, 2024.

Woo-Yong Choi was born in Busan, Korea in 1970. He received the B.S., M.S. and Ph.D. degrees in industrial engineering from POSTECH in 1992, 1994 and 1997, respectively. From 1997 to 2001 he was a senior member of technical staff at Hyundai Electronics Industries Co., Ltd.. From 2001 to 2005 he was a senior member of technical staff at ETRI. Since 2005 he has been with Department of Industrial & Management Systems Engineering at Dong-A University, where he is currently a full tenured professor.

Received 15 January 2026