IET Communications

Special issue Call for Papers

**Be Seen. Be Cited.
Submit your work to a new
IET special issue**

Connect with researchers and experts in your field and share knowledge.

Be part of the latest research trends, faster.

Read more

ORIGINAL RESEARCH

A novel artificial intelligence based wireless local area network channel access control scheme for low latency e-health applications

Zixin Liu[1] | Yunxin Lv[1] | Meihua Bi[1] | Yanrong Zhai[1,2]

[1]Key Laboratory of Data Storage and Transmission Technology of Zhejiang Province, School of Communication Engineering, Hangzhou Dianzi University, Hangzhou, Zhejiang, China

[2]School of Engineering, University of Kent, Canterbury, UK

Correspondence
Yunxin Lv, Key Laboratory of Data Storage and Transmission Technology of Zhejiang Province, School of Communication Engineering, Hangzhou Dianzi University, Hangzhou, Zhejiang, China.
Email: yunxinlyu@hdu.edu.cn

Abstract

To effectively support low-latency e-health applications, a novel artificial intelligence-based wireless local area network (WLAN) channel access control scheme named intelligent hybrid channel access (IHCA) is proposed and studied. In the IHCA scheme, each beacon interval comprises multiple cycles, with each cycle containing a contention-free period (CFP) and a contention period (CP). By adopting an artificial neural network (ANN) and utilizing its calculation outputs to (1) decide whether to poll a hub during the CFP and (2) determine the initial backoff count (IBC) of each hub, polling of empty hubs during CFP can be reduced, and collisions during CP can be relieved. The authors' simulation results show that the IHCA scheme can effectively reduce latency compared to the Hybrid coordination function Controlled Channel Access (HCCA) and the Request based Polling Access (RPA) reference designs.

1 | INTRODUCTION

With the continuous development of modern society, a series of information and communications technologies such as Internet of Things, big data, cloud computing, artificial intelligence, blockchain, are redefining the future of humanity [1, 2]. Smart transportation, smart healthcare, and smart education are moving from prophecy to reality and are affecting productions and human lives in profound ways [3].

Recently, with the emergence of tactile internet concept, real-time delivery of haptic/tactile/control data over telecommunication networks has attracted widespread interests [4]. By achieving real-time haptic/tactile/control data delivery, many advanced applications will be enabled over telecommunication networks, and human life as well as industrial/commercial fields will be revolutionized [5]. Healthcare is expected to be an important field that will be greatly revolutionized [6]. As shown in Figure 1, with a real-time haptic/tactile/control data delivery capable wireless local area network (WLAN) in a hospital intensive care unit (ICU), low-latency e-health applica-

tions such as telesurgery, telediagnosis, and real-time monitoring can be realized while patient and doctor/healthcare server are decoupled in geographic position (i.e. patient in ICU and doctor/healthcare server in office). Clearly, this can provide great convenience for patient and doctor/operator and also can protect each other while dealing with infectious diseases (e.g. Ebola, COVID-19). However, real-time haptic/tactile/control data delivery requires an end-to-end latency on the order of 1 ms to prevent users from experiencing phenomena such as 'cyber-sickness' (cyber-sickness: a symptom similar to motion-sickness), which means that to support low-latency e-health applications, a latency target of 0.5 ms is required for the underlying WLAN, and a 0.2-ms WLAN latency performance is preferred when considering the time budget for data processing [7, 8]. Clearly, such a strict latency target poses a huge challenge for WLANs.

To achieve the 0.5-ms latency target in WLANs, several investigations have been carried out so far by incorporating deep neural networks, setting strict channel access schedule, optimizing parameter selections, redesigning channel access

algorithms [9–12]. However, although these studies strive the realization of the 0.5-ms latency target in WLANs, their affordable load/users are very limited. Considering that for typical low-latency e-health applications the typical deployment environment of hospital ICU normally needs to cater for up to eight patients (i.e. hubs) and the traffic load can vary in a large range, to effectively support low-latency e-health applications, further investigation on reducing WLAN latency is still required [13].

In this paper, a novel artificial intelligence-based WLAN channel access control scheme is proposed and investigated to support low-latency e-health applications over a WLAN in a hospital ICU. We focus on the uplink direction as it is the bottleneck of network latency. In the proposed scheme, each beacon interval consists of multiple cycles, with each cycle including a contention-free period (CFP) and a contention period (CP). At the beginning of each cycle, by adopting an artificial neural network (ANN) and using its computational outputs to determine whether to poll a hub as well as the initial backoff count (IBC) of the hub, the scheme reduces the polling of empty hubs and collisions, thereby achieving low-latency performance.

The contributions of this paper are fourfold: (1) to support low-latency e-health applications, we propose a novel WLAN channel access control scheme that adopts an ANN to help determine the polling of hubs and the IBCs of hubs in each cycle; (2) to determine whether to poll a hub, we propose a threshold decision method using precision-recall (PR) curves, thereby properly solving this hub classification problem and avoiding the polling of empty hubs during the CFP; (3) a mapping rule is proposed to determine the IBC of each hub, thereby relieving the collision problem during the CP; and (4) we validate the low-latency performance of the intelligent hybrid channel access (IHCA) scheme by simulation.

The remainder of this paper is organized as follows. Section 2 reviews the related works and explains the background and the motivation. Section 3 introduces the proposed IHCA scheme in detail. Section 4 presents the simulation results and discusses the performance of the proposed IHCA scheme. Section 5 concludes this paper.

2 | RELATED WORKS AND HCCA FOR ACHIEVING LOW LATENCY IN WLAN

2.1 | Related works

Latency reduction is always a research hotspot for telecommunication networks. With the entering of 5G era, there are a lot of investigations reported to reduce the latency of WLANs. Lacalle et al. combined WLAN medium access control (MAC) with multi-link technology and revealed that with two links, it is possible to ensure 40% of the transmissions to reach their destination within 1 ms [14]. Al-Maqri et al. presented a dynamic Transmission Opportunities (TXOPs) assignment algorithm for WLANs, and achieved 1-ms latency by collecting the size of subsequent frame and adjusting TXOP allocation accordingly [15]. Fujiwara et al. introduced a selective CFP into the Hybrid coordination function Controlled Channel Access (HCCA)
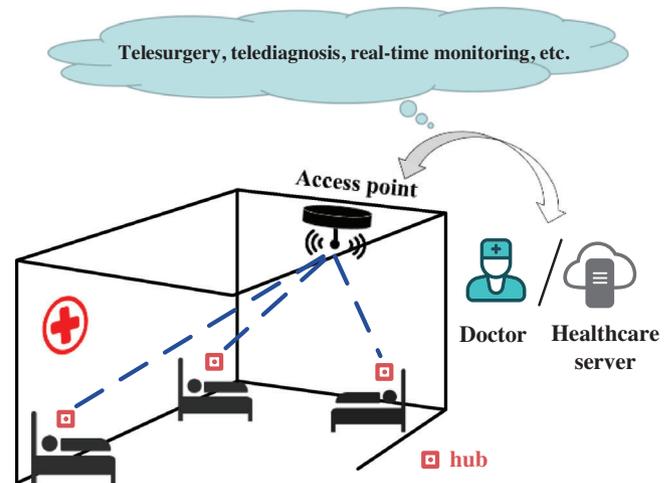

FIGURE 1 Low-latency e-health applications supported by a WLAN in a hospital ICU. (Each patient corresponds to a hub). ICU, intensive care unit; WLAN, wireless local area network.

scheme to enable a WLAN to respond to link condition changes for reducing latency and error-rate [16]. Chang et al. proposed a scheduling algorithm to assign exact TXOPs for variable bit rate traffic to reduce packet delay and packet loss rate [17].

Although latency reduction is a research hotspot, how to realize 0.5-ms latency in WLANs has not been widely investigated yet. So far, there are only few works reported. Incorporating with a deep neural network, Gao et al. proposed an MAC design for WLANs and achieved 0.5-ms average uplink latency in a single-channel environment [9]. Wei et al. set a strict WLAN channel access schedule and realized 0.2-ms latency with high service quality [10]. Feng et al. investigated the possibility of achieving 0.1 ms latency in a WLAN and proposed a parameter selection method for the HCCA scheme to realize 0.1 ms latency [11]. Lv et al. proposed a request-based polling scheme named Request based Polling Access (RPA) for WLANs and achieved 0.2 ms latency by avoiding polling empty users [12]. Further, by theoretically analyzing the packet arrival probability, Lv et al. proposed a selective polling and controlled contention based WLAN MAC scheme to achieve 0.5 ms latency [18]. These works contribute to the realization of 0.5 ms latency in WLANs. However, as mentioned in Section 1, these proposals can only support limited load/users when achieving the target latency performance, and are therefore needed for further development.

2.2 | HCCA for achieving low latency in WLAN

In a WLAN, MAC plays a key role in affecting the latency performance. The original IEEE 802.11 standard defined two MAC schemes: distributed coordination function (DCF) and point coordination function (PCF). In 2005, IEEE introduced a Quality of Service (QoS) oriented amendment and provided a new coordination function known as the Hybrid Coordination Function (HCF). HCF includes two modes: enhanced
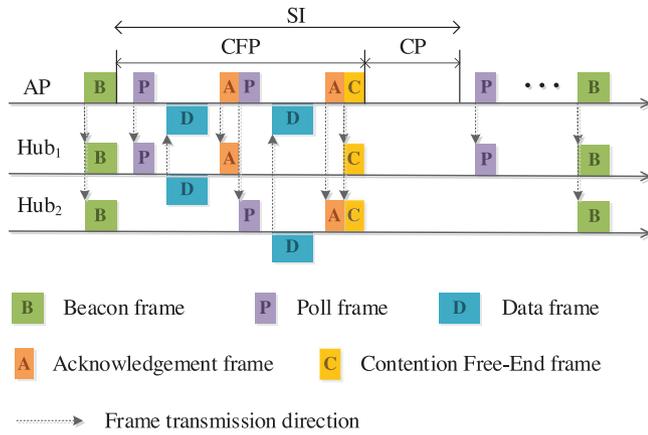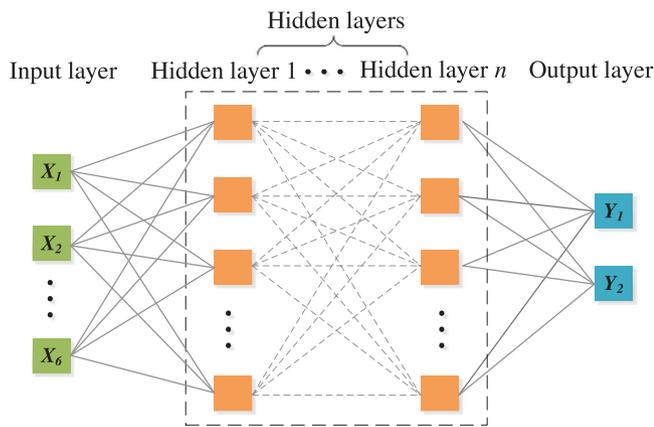
FIGURE 2 Basic operation of the HCCA scheme.

FIGURE 3 The structure of the ANN. ANN, artificial neural network.

FIGURE 4 Operations of collecting samples from the HCCA scheme. HCCA, hybrid coordination function controlled channel access.

distributed channel access (EDCA), which provides priority services, and HCCA, which satisfies parameterized QoS requirements. Currently, DCF, EDCA, and HCCA are the three main MAC schemes for WLANs [19].

Among the three MAC schemes, as collision is inevitable in the two contention-based MAC schemes (i.e. DCF and EDCA) and collisions in these two schemes can induce an extra packet delay of several milliseconds, the HCCA scheme has received increased attentions for achieving low latency in WLANs. The fundamental operation of the HCCA scheme is depicted in Figure 2. For illustration, here we adopt a one-AP-two-hub (hub: user) configuration. The HCCA scheme divides each beacon interval into multiple service intervals (SIs), and each SI consists of a CFP (also known as a controlled access phase) and a CP. During the CFP, the hybrid coordinator (HC) (i.e. AP) polls the hubs in turn by allocating TXOPs. During the CP, all connected hubs use the Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) mechanism to compete for channel access.

Clearly, by utilizing the HC to poll and allocate TXOPs to hubs, the HCCA scheme can ensure each hub to have an opportunity to access the channel in each SI, thereby reducing the impact of collisions and ensuring a low-latency performance.
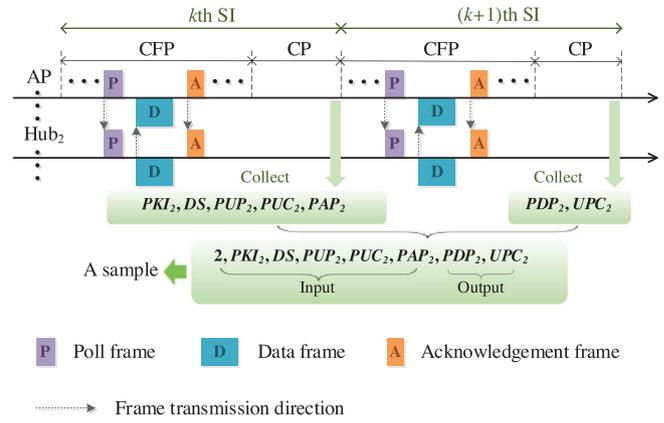
Although the HCCA scheme is attractive to achieve low latency in WLANs, it has some inherent problems: during the CFP, polling of empty hubs is inevitable; during the CP, collision is still existing. As these problems can result in significant resource waste and degrade the latency/affordable users of a WLAN, to effectively support low-latency e-health applications, further investigation and optimization are required. Recently, artificial intelligence has drawn intensive interests in the field of resource management, and it has been proven to be an effective tool for prediction [20, 21]. Considering that a prediction result can be used as a guidance to alleviate the problems of polling empty hubs and collisions, we propose a novel WLAN channel access control scheme called IHCA to achieve low latency in WLANs to support low-latency e-health applications.

3 | THE PROPOSED METHOD

In this section, we explain the principles of the proposed IHCA scheme. This section begins with a specific description of our ANN structure and the sample collection procedure for the IHCA scheme, followed by a detailed explanation of the operation procedure of the IHCA scheme.

3.1 | ANN structure and sample collection

As shown in Figure 3, the ANN utilized in the IHCA scheme is designed with six neurons in the input layer and two neurons in the output layer. Multiple hidden layers reside between the input layer and the output layer. Each hidden layer contains multiple neuron units, which are non-linear functions mapping the associated inputs to the outputs. The input variables X_1, X_2, X_3, X_4, X_5, and X_6 are hub status-related parameters. The output variables Y_1 and Y_2 are a polling index and a competition index, respectively, which are used for polling decision (i.e. hub classification) and IBC decision in the IHCA scheme. The details of how the outputs are utilized will be introduced in Section 3.2.

The ANN needs to be trained by a set of training samples before being used in the IHCA scheme. These training

TABLE 1 The meaning of each notation used during sample collection.

Notation	Meaning
i	The index of hub$_i$
PKI_i	The average packet arrival interval of hub$_i$
DS	The duration of the kth SI
PUP_i	The number of packets uploaded by hub$_i$ during the CFP of the kth SI
PUC_i	The number of packets uploaded by hub$_i$ during the CP of the kth SI
PAP_i	The time passed since the arrival (arrive at hub$_i$) time of the last packet transmitted by hub$_i$
PDP_i	Whether hub$_i$ uploads packets during the CFP of the $(k+1)$th SI (PDP_i is set to 1 if hub$_i$ uploaded packets, and 0 otherwise)
UPC_i	Whether hub$_i$ uploads packets during the CP of the $(k+1)$th SI (UPC_i is set to 1 if hub$_i$ uploaded packets, and 0 otherwise)

samples are collected from the conventional HCCA scheme. The process for collecting these samples is depicted in Figure 4 and introduced as follows. In the HCCA scheme, once the kth ($k \in N+$) SI ends, the AP collects the following data from hub$_i$ ($i = 1, 2, \dots n$, where n is the number of hubs): PKI_i, DS, PUP_i, PUC_i, and PAP_i. The meaning of each notation is organized in Table 1. It needs to be mentioned here that in order to obtain PKI_i and PAP_i, in each data transmission, the transmitting hub is requested to upload the arrival time of the last packet of the current transmission and its average packet arrival interval (which is calculated by the hub itself) along with the data frame. This information can be included in six octets (four octets for covering the packet arrival time information and two octets for covering the average packet inter-arrival time information with 1-μs level accuracy) and can be easily delivered using the unused field in the data frame header (e.g. the address four field), and therefore will not induce any additional information overhead to the system. Then, when the $(k+1)$th SI ends, based on whether hub$_i$ has uploaded packets during the CFP and the CP of the $(k+1)$th SI, the AP determines a PDP_i value and a UPC_i value for hub$_i$. If hub$_i$ uploaded packets during the CFP of the $(k+1)$th SI, PDP_i is set to 1; otherwise, PDP_i is set to 0. If hub$_i$ uploaded packets during the CP of the $(k+1)$th SI, UPC_i is set to 1; otherwise, UPC_i is set to 0. The meanings of PDP_i and UPC_i as well as how their values are determined are also introduced in Table 1. At this point, a sample that is noted as $[i, PKI_i, DS, PUP_i, PUC_i, PAP_i, PDP_i, UPC_i]$ can be formed. Clearly, as i ranges from 1 to n, n samples can be collected at the end of each SI.

Note that in a sample, UPC_i indicates whether hub$_i$ has uploaded packets during the CP. Therefore, there is a problem that if hub$_i$ fails to compete for the channel, a sample with $UPC_i = 0$ will be collected even if hub$_i$ needs to upload packets during the CP. As it is easy to understand that a UPC_i representing whether hub$_i$ needs to upload packets during a CP is more meaningful, the UPC_i value is updated as follows. Considering

that a non-empty hub that fails to compete for the channel during the CP of the kth SI must uploads packets, when it is polled during the CFP of the $(k+1)$th SI, the UPC_i collected in the kth SI will be updated to 1 if hub$_i$ uploads packets during the CFP of the $(k+1)$th SI. After updating UPC_i, the modified sample can be used to train the ANN.

When training the ANN, the first six parameters in a sample (i.e. i, PKI_i, DS, PUP_i, PUC_i, and PAP_i) are the inputs, while the last two parameters (i.e. PDP_i and UPC_i) are the outputs. After training, the ANN can be used in the IHCA scheme.

3.2 | The IHCA scheme

This sub-section introduces the operation procedure of the IHCA scheme. The traffic flow diagram of the IHCA scheme is presented in Figure 5. In the IHCA scheme, each beacon interval contains multiple cycles, and each cycle contains a CFP and a CP. Before the start of the kth cycle, the AP first collects data and organizes an ANN input vector for each hub. The ANN input vector for hub$_i$ is organized as $[i, PKI_i, DC, NPP_i, NPC_i, PAP_i]$. The meaning of each notation is explained in Table 2. Similarly, to obtain PKI_i and PAP_i, in each data transmission, the transmitting hub is requested to upload the arrival time of the last arrived data packet of the current transmission and its average packet arrival interval (calculated by the hub itself) along with the data frame. Then, the ANN input vector of each hub is put into the trained ANN for calculation, and for hub$_i$, two output data, Y_{1i} and Y_{2i}, can be acquired. For hub$_i$, the output Y_{1i} will be compared with a threshold. If Y_{1i} is greater than the threshold, hub$_i$ will be classified as a hub that needs to be polled during the kth cycle, and the AP will add hub$_i$ to the polling list of the kth cycle. The method of selecting the threshold will be introduced in detail in Section 3.2.1. The other output Y_{2i} will be used to decide an IBC for hub$_i$, and the detailed rule of determining the IBC will be explained in Section 3.2.2. After finishing the threshold comparison and the IBC decision processes for all hubs, the AP starts the CFP of the kth cycle, and polls those hubs in the polling list of the kth cycle in turn. When the last hub in the polling list has finished uploading packets, the AP ends the CFP and uses a Contention Free-End frame to broadcast the IBCs to all hubs. Following this, the CP begins. During the CP, each hub competes for channel access using the CSMA/CA mechanism with its assigned IBC. The CP ends when a preset time restriction is reached. At the end of the CP, the AP collects the ANN input vector for each hub again, and prepares to enter the $(k+1)$th cycle. It should be mentioned here that calculating ANN outputs and processing the outputs (i.e. comparing with the threshold and determining IBCs) are both not complex tasks, and therefore we believe that in each cycle, the ANN calculation and outputs processing tasks can be finished within the PCF Interframe Space (PIFS) before sending out the first control frame (i.e. Poll frame, or Contention Free-End frame if no hub will be polled) of this cycle.

Clearly, in the IHCA scheme, the accesses of hubs are central-coordinated by the AP during the CFP, and are distributed-coordinated by the hubs themselves during the CP.
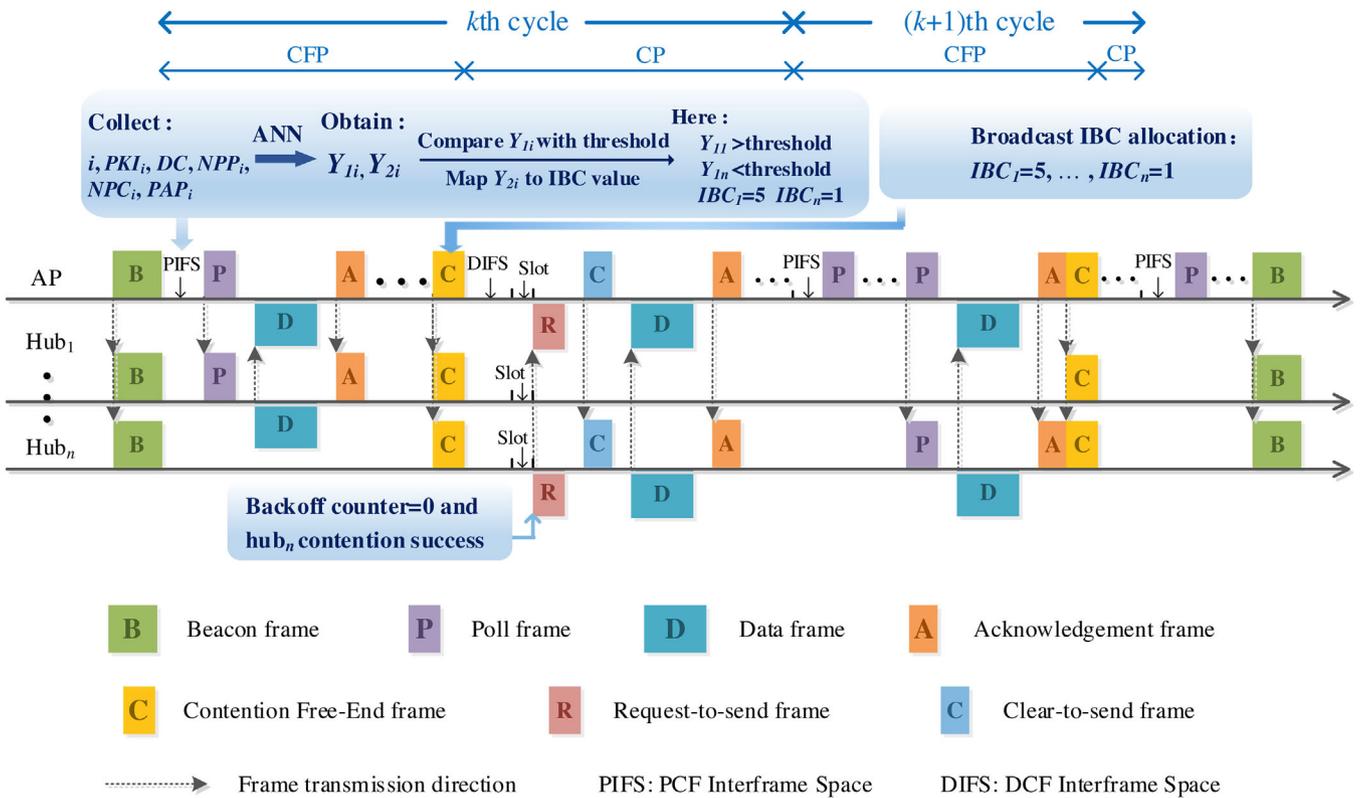
FIGURE 5 Traffic flow diagram of the IHCA scheme. IHCA, intelligent hybrid channel access.

TABLE 2 The meaning of each notation used in the IHCA scheme.

Notation	Meaning
i	the index of hub_i
PKI_i	the average packet arrival interval of hub_i
DC	the duration of the $(k\text{-}1)$th cycle
NPP_i	the number of packets uploaded by hub_i during the CFP of the $(k-1)$th cycle
NPC_i	the number of packets uploaded by hub_i during the CP of the $(k-1)$th cycle
PAP_i	the time passed since the arrival (arrive at hub_i) time of the last packet transmitted by hub_i
Y_{1i}	the first output of the ANN for hub_i
Y_{2i}	the second output of the ANN for hub_i

ANN, artificial neural network; IHCA, intelligent hybrid channel access.

3.2.1 | Threshold decision

In each cycle, for hub_i, the AP compares the first output of the ANN (i.e. Y_{1i}) with a threshold to determine whether to poll hub_i during the CFP. If Y_{1i} is greater than this threshold, the AP will poll hub_i; otherwise, it will not. Therefore, a proper selection of the threshold is important. We choose the threshold based on the performance of the threshold in classifying the training samples. To begin with, the training samples are divided into two categories: those with $PDP_i = 0$, and those

TABLE 3 Confusion matrix diagram.

		Observed value	
		1	**0**
Predicted value	1	True positive(TP)	False positive(FP)
	0	False negative(FN)	True negative(TN)

with $PDP_i = 1$. Then, the input parts of the two types of samples are fed into the trained ANN to obtain the corresponding Y_1 outputs. Considering that the quantity of $PDP_i = 0$ samples and the quantity of $PDP_i = 1$ samples skew under low/high traffic load, we use a PR curve to evaluate the classification performance of a threshold and derive the optimal threshold value as follows [22].

With an arbitrary threshold, the training samples can be classified by comparing the Y_1 outputs and the threshold, and the classification results can be organized in a confusion matrix, as shown in Table 3. The confusion matrix has four categories: true positive (TP) denotes samples correctly classified as positives (i.e. samples with $PDP_i = 1$ and $Y_1 \geq$ threshold); false positive (FP) refers to negative samples incorrectly classified as positives (i.e. samples with $PDP_i = 0$ and $Y_1 \geq$ threshold); true negative (TN) corresponds to negative samples correctly classified as negatives (i.e. samples with $PDP_i = 0$ and $Y_1 <$ threshold); and false negative (FN) refers to positive samples incorrectly classified as negatives (i.e. samples with $PDP_i = 1$ and $Y_1 <$ threshold).
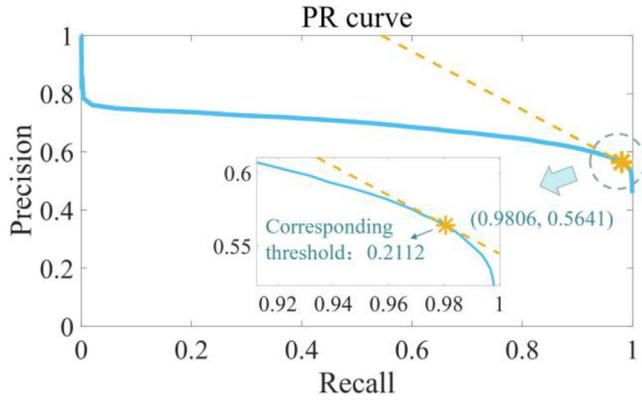
FIGURE 6 PR curve under the normalized traffic load of 0.05. PR, precision-recall.

Based on the number of samples in each category of the confusion matrix, the key parameters to derive the PR curve, that is, *recall* and *precision*, can be obtained. The *recall* and *precision* values are calculated as follows:

$$recall = \frac{TP}{TP + FN} \tag{1}$$

$$precision = \frac{TP}{TP + FP} \tag{2}$$

Different thresholds can result in different *recall* and *precision* values. Then, by varying the threshold from the minimal value of Y_1 to the maximum value of Y_1, a PR curve can be obtained with *recall* and *precision* as the horizontal and vertical axes, respectively. Theoretically, the larger the *recall* and *precision* values are, the better the classification performance is. Therefore, by selecting the threshold that can maximize (3), the optimal threshold can be obtained.

$$P = recall + precision \tag{3}$$

In Figure 6, we present the PR curve for the case that training samples are collected in a one-AP-eight-hub WLAN under the normalized traffic load of 0.05. It is shown that P has the maximum value when the *recall* = 0.9806 and the *precision* = 0.5641. Therefore, the corresponding threshold, which is 0.2112, is selected. Further, to show the rationality of the selected threshold, we show the Y_1 outputs distributions of $PDP_i = 0$ samples and $PDP_i = 1$ samples as well as the selected threshold (i.e. 0.2112) in Figure 7. Here, the Y_1 outputs distribution of $PDP_i = 0$ samples is represented by SO_0, while the Y_1 outputs distribution of $PDP_i = 1$ samples is represented by SO_1. Clearly, the selected threshold can divide the two types of samples well.

3.2.2 Mapping Rule

In each cycle, for hub_i, the second output of the ANN (i.e. Y_{2i}) is used to decide the IBC of hub_i. Suppose that there are n hubs connected to the AP. As Y_{2i} can represent the probability that hub_i requires to access the channel during the CP, the AP determines the IBC of each hub as follows. The AP first sorts
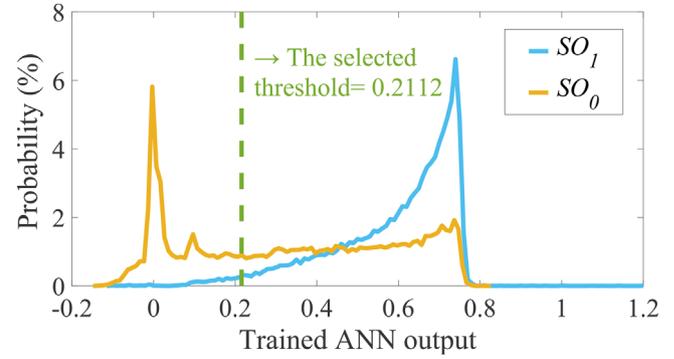

FIGURE 7 Y_1 outputs distributions of the two types of samples (for the samples collected under the normalized traffic load of 0.05).

TABLE 4 Simulation parameters.

Parameters	Value
Link rate	135 Mbps
Beacon frame	672 bits
Beacon interval	100 ms
Poll frame	160 bits
Acknowledgement frame	112 bits
Null frame	320 bits
Request-to-send frame	160 bits
Clear-to-send frame	112 bits
Contention free-end frame	160 bits
Short interframe space	16 μs
PCF interframe space	25 μs
DCF interframe space	34 μs
Frame check sequence	32 bits
MAC header	288 bits
PLCP header	48 bits
PLCP preamble	144 bits

DCF, distributed coordination function; MAC, medium access control; PCF, point coordination function; PLCP, physical layer convergence procedure.

all hubs according to their second outputs (i.e. Y_2) in ascending order. Then, following the order, (a) if n is an odd number, each hub is assigned an IBC from the sequence [0, 1, …, $(n-1)/2$, …, 1, 0] in order; (b) if n is an even number, each hub is assigned an IBC from the sequence [0, 1, …, $n/2-1$, $n/2-1$, …, 1, 0] in order. This IBC mapping rule has two features: (1) a hub having a relatively high probability of accessing the channel has a relatively high priority to access the channel; (2) a hub having a relatively high probability of accessing the channel and a hub having a relatively low probability of accessing the channel have the same IBC, thereby reducing the risk of collisions.

4 | SIMULATION AND ANALYSIS

In this section, we evaluate the latency performance of the IHCA scheme. The simulations are performed using MATLAB.
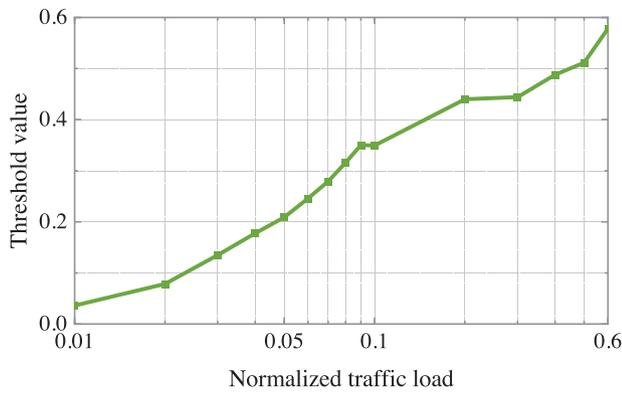
FIGURE 8 Threshold of the IHCA scheme. IHCA, intelligent hybrid channel access.
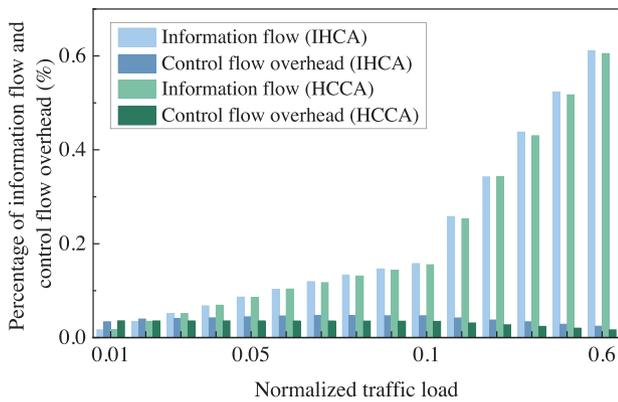

FIGURE 9 Channel occupancies of control flow overhead and information flow.

As there are typically eight patients in a hospital ICU, we consider a WLAN consisting of an AP and eight hubs. We focus on uplink direction, and we neglect propagation delay because of the short transmission distance. Mobility is not considered. Buffer length is infinite. Source traffic follows Pareto distribution, and the normalized traffic load ranges from 0.01 to 0.6. The conventional HCCA scheme is tested as well for comparison. Basic simulation parameters are organized and listed in Table 4 [19]. Packet size is set to 512 bits as low-latency e-health applications usually have small packet size [11]. The shape parameters for the ON and OFF intervals of the source traffic are set to 2.8 and 2.4, respectively [23]. Simulation time is set to 9 s. For the IHCA scheme, the CP duration is set to 80 μs. For the HCCA scheme, dynamic SI is adopted, and the CP duration is set to 80 μs as well. The maximum contention window (CW) ranges from 15 to 255. The average packet arrival interval is calculated based on the last 30 packets arrived. To have a good prediction performance and to prevent the overfitting issue, we empirically adopt a one-hidden-layer configuration for the ANN, and the hidden layer contains 12 neuron units. The transfer function of the hidden layer neuron is *tansig*, and that of the output layer neuron is *pureline*. Under each traffic load, we run the HCCA scheme for 5 s to collect samples, and we train the ANN for 1000 iterations.

4.1 | Computational complexity of the IHCA scheme

The adopted ANN has 6 neurons in the input layer, 12 neurons in the hidden layer, and 2 neurons in the output layer. Therefore, referring to [20], the time complexity of the ANN calculation process is $O(96)$. Given that the outputs comparison process (for Y_1) has a complexity of $O(n)$ and the outputs sorting process (for Y_2) has a complexity of $O(n^2)$ in the worst case, the total time complexity of the IHCA scheme can be expressed as $O(n^2 + 96n + n) = O(n^2)$.

4.2 | Threshold value analysis

In Figure 8, we present how threshold values vary as the traffic load increases. It can be observed that the threshold increases as the traffic load increases. This is due to the imbalance between the quantities of $PDP_i = 0$ and $PDP_i = 1$ samples. Under the low traffic loads, more $PDP_i = 0$ samples are collected. As training with more $PDP_i = 0$ samples can result in the trained ANN to produce a Y_1 output value closer to 0, the threshold value is low under the low traffic loads. When the traffic load is high, as more $PDP_i = 1$ samples are collected, the ANN is trained with more $PDP_i = 1$ samples. Therefore, the trained ANN is likely to produce a Y_1 output value closer to 1, thereby resulting in a high threshold value. As a result, the threshold increases with the traffic load.

4.3 | Channel occupancy comparison

In Figure 9, we present the channel occupancies of control flow overhead and information flow for both the IHCA and the HCCA schemes. As the traffic load increases, the channel occupancy rate of information flow increases in both schemes due to increased data packet arrivals. For the HCCA scheme, the channel occupancy rate of control flow overhead decreases as the traffic load increases. This is because when the traffic load increases, the time required for each transmission is extended as more data packets are needed to be transmitted. Therefore, the total number of transmissions during the simulation time is reduced, and hence reduces the total number of control flow frames such as Poll frame and Acknowledgement frame in the HCCA scheme. In contrast, for the IHCA scheme, the channel occupancy rate of control flow overhead first increases between the traffic loads of 0.01 to 0.07 and then decreases between the traffic loads of 0.07 to 0.6. This is because when the traffic load is between 0.01 and 0.07, with the traffic load increasing, the probability of polling a hub increases in the IHCA scheme. Therefore, more Poll frames and Acknowledgement frames will occur, and hence result in the channel occupancy rate of control flow overhead for the IHCA scheme to increase when the traffic load is increasing from 0.01 to 0.07. When the traffic load exceeds 0.07, as more data packets need to be transmitted, the time required for each transmission increases. Therefore, the total number of transmissions during the simulation time
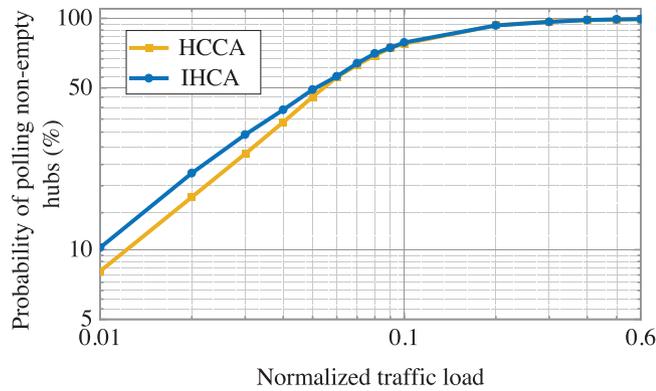
FIGURE 10 Probability of polling non-empty hubs during CFP. CFP, contention-free period.

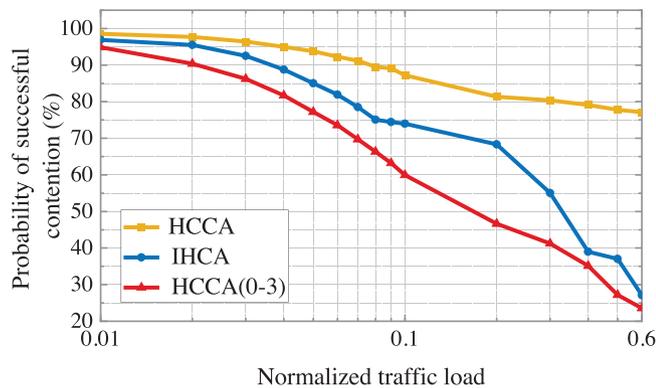
FIGURE 12 Average packet delay.

FIGURE 11 Probability of successful contention during CP. CP, contention-free period.

reduces, and hence results in the channel occupancy rate of control flow overhead for the IHCA scheme to decrease when the traffic load is higher than 0.07.

4.4 | Probability of polling non-empty hub and probability of successful contention comparisons

In Figure 10, we depict the probability of polling non-empty hubs for the IHCA and the HCCA schemes. As the normalized traffic load increases, the curves of both schemes increase accordingly. The IHCA scheme generally exhibits a higher probability of polling non-empty hubs, especially under the low traffic load scenarios. This implies that the IHCA scheme reduces the risk of polling empty hubs, and therefore can improve the channel utilization efficiency during the CFP.

In Figure 11, we present the probability of successful channel contention (i.e. the probability of success in a channel contention attempt) for the IHCA and the HCCA schemes. For a fair comparison, considering that eight hubs can result in the IBC of the IHCA scheme to range between 0 and 3, we also include another HCCA scheme that uses an initial maximum CW of 3 (noted as HCCA(0-3) scheme). As shown in Figure 11, despite adopting an ANN for IBC allocation, the
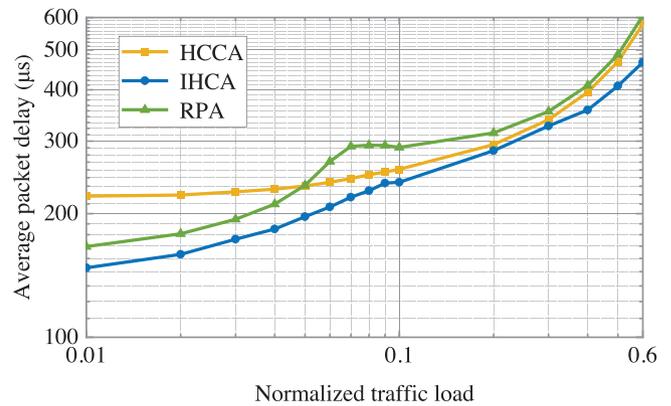
IHCA scheme consistently has a lower probability of successful contention than the HCCA scheme. This is because the IBC range for the HCCA scheme is from 0 to 15, whereas the IBC range for the IHCA scheme is from 0 to 3, thereby resulting in the IHCA scheme to have a higher collision probability. However, with the same IBC range, the IHCA scheme outperforms the HCCA(0-3) scheme in terms of successful contention probability, validating the rationality of IBC allocation in the IHCA scheme for collision reduction.

4.5 | Average packet delay comparison

In this sub-section, we assess the latency performance of the proposed IHCA scheme by comparing it with the traditional HCCA scheme and the RPA scheme proposed in reference [12]. We introduce the RPA scheme into the latency performance comparison because the RPA scheme achieves low latency performance by avoiding polling empty hubs as well, and therefore can be a useful benchmark for assessing the performance of the IHCA scheme. The parameter settings of all the three schemes are the same. In Figure 12, we can observe that generally all the three schemes show an upward trend as the normalized traffic load increases. This is expected because the increment in traffic load results in more frequent arrivals of data packets, thereby increasing the queue length and the packet delay. It is evident from Figure 12 that the IHCA scheme outperforms the HCCA and the RPA schemes in terms of low average packet delay. This is because compared to the HCCA scheme, the IHCA scheme can reduce the polling of empty hubs to improve the channel utilization efficiency during the CFP, and can allocate rational IBC for each hub to allow a hub having high channel accessing probability to contend the channel with high accessing priority and low collision probability. Compared to the RPA scheme, the IHCA scheme can reduce the polling of empty hubs without inducing any additional costs for request uploading. Therefore, the proposed IHCA scheme can show a lower average packet delay than the HCCA scheme and the RPA scheme. It is observed that the IHCA scheme can reduce the average packet delay by 3% to 33% and 8% to 25% compared to the HCCA scheme and the RPA scheme, respectively,
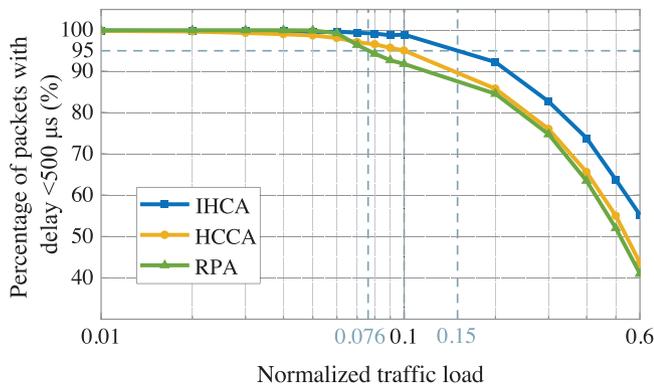
FIGURE 13 Percentage of packets with delay < 500 μs.

FIGURE 14 Average packet delay under different numbers of hubs.

and can achieve an average packet delay of less than 200 μs as long as the normalized traffic load is below 0.05. These results demonstrate the capability and advantage of the IHCA scheme in serving low-latency e-health applications.

4.6 | Comparison of percentage of packets with delay < 500 μs

To evaluate the service quality performance, in Figure 13, we show the percentage of packets that satisfy the 500 μs delay constraint for the IHCA scheme, the HCCA scheme, and the RPA scheme. It can be observed that the IHCA scheme generally shows a higher percentage than both the HCCA and the RPA schemes, which means that the IHCA scheme can provide a higher service quality when delivering low-latency e-health applications. When considering a 95% QoS constraint (i.e. ≥ 95% packets satisfying the 500 μs delay constraint), the IHCA scheme, the HCCA scheme, and the RPA scheme can afford normalized traffic loads of 0.15, 0.1, and 0.076, respectively.

4.7 | Performance under different numbers of hubs

In this sub-section, we show the effectiveness of the IHCA scheme under different numbers of hubs by evaluating the average packet delay performances of the three schemes (i.e. the IHCA scheme, the HCCA scheme, and the RPA scheme) with the number of hubs ranging from 2 to 8. We consider two normalized traffic load scenarios: low (0.05) and high (0.25). As shown in Figure 14, under both traffic load scenarios and under all hub quantity settings, the IHCA scheme always has a better average packet delay performance than the HCCA scheme and the RPA scheme. Furthermore, the average packet delay advantage of the IHCA scheme over the HCCA scheme and the RPA scheme will increase as the number of hubs increases. With the number of hubs increasing from 2 to 8, it can be observed that (1) the delay optimization of the IHCA scheme with respect to the HCCA scheme increases
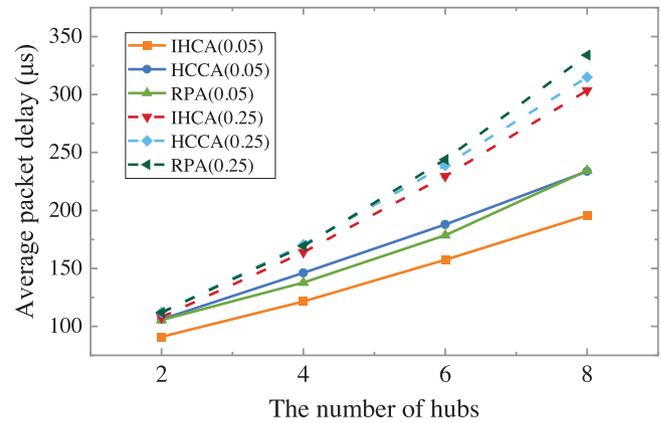
from 16 to 38 μs and from 5 to 11 μs for the low and high traffic load scenarios, respectively, and (2) the delay optimization of the IHCA scheme with respect to the RPA scheme increases from 15 to 39 μs and from 4 to 31 μs for the low and high traffic load scenarios, respectively. In summary, the IHCA scheme is effective in optimizing the delay performance under the considered hub quantity scenarios, showing the ability of the IHCA scheme to support low-latency e-health applications under typical patient quantity settings in a hospital ICU.

5 | CONCLUSIONS

In this paper, an artificial intelligence-based wireless channel access control scheme called IHCA is proposed and investigated to facilitate the delivery of low-latency e-health applications over a WLAN in a hospital ICU. This scheme adopts an ANN to benefit the access control of wireless channel. By utilizing the outputs of the ANN, the IHCA scheme reduces the polling of empty hubs during the CFP and relieves collisions during the CP, thereby achieving low-latency performance. We validate the low-latency advantage of the IHCA scheme by comparing it with the conventional HCCA scheme and the RPA scheme. Our simulation results show that the IHCA scheme can reduce the average packet delay by 3% to 33% and 8% to 25% compared to the HCCA and RPA schemes, respectively, which highlights the capability of our IHCA scheme to serve low-latency e-health applications.

CONFLICT OF INTEREST STATEMENT

The authors declare no conflict of interest.

DATA AVAILABILITY STATEMENT

The data that support the findings of this study are available from the corresponding author upon reasonable request.

ORCID

Zixin Liu https://orcid.org/0000-0001-8633-1182

REFERENCES

1. Celik, A., Romdhane, I., Kaddoum, G., et al.: A top-down survey on optical wireless communications for the internet of things. IEEE Commun. Surv. Tutor. 25(1), 1–45 (2023)
2. Wang, X., Han, Y., Leung, V.C.M., et al.: Convergence of edge computing and deep learning: A comprehensive survey. IEEE Commun. Surv. Tutor. 22(2), 869–904 (2020)
3. Saini, A., Wijaya, D., Kaur, N., et al.: LSP: Lightweight smart-contract-based transaction prioritization scheme for smart healthcare. IEEE IoT J. 9(15), 14005–14017 (2022)
4. Kim, K.S., Kim, D.K., Chae, C.-B., et al.: Ultrareliable and low-latency communication techniques for tactile internet services. Proc. IEEE. 107(2), 376–393 (2019)
5. Aijaz, A., Sooriyabandara, M.: The tactile internet for industries: A review. Proc. IEEE 107(2), 414–435 (2019)
6. Qadri, Y.A., Nauman, A., Zikria, Y.B., et al.: The future of healthcare internet of things: A survey of emerging technologies. IEEE Commun. Surv. Tutor. 22(2), 1121–1167 (2020)
7. Martirosov, S., Kopecek, P.: Cyber sickness in virtual reality—Literature review. In: Annals of DAAAM & Proceedings, Zadar, Croatia, pp. 718–726 (2017)
8. Porambage, P., Okwuibe, J., Liyanage, M., et al.: Survey on multi-access edge computing for internet of things realization. IEEE Commun. Surv. Tutor. 20(4), 2961–2991 (2018)
9. Gao, J., Li, M., Zhuang, W., et al.: MAC for machine-type communications in industrial IoT—part II: Scheduling and numerical results. IEEE IoT J. 8(12), 9958–9969 (2021)
10. Wei, Y.-H., Leng, Q., Han, S., et al.: RT-WiFi: Real-time high-speed communication protocol for wireless cyber-physical control applications. In: Real-Time Systems Symposium (RTSS), Vancouver, BC, Canada, pp. 140–149 (2013)
11. Feng, Y., Jayasundara, C., Nirmalathas, A., et al.: A feasibility study of IEEE 802.11 HCCA for low-latency applications. IEEE Trans. Commun. 67(7), 4928–4938 (2019)
12. Lv, Y., Dias, M.P.I., Ruan, L., et al.: Request-based polling access: Investigation of novel wireless LAN MAC scheme for low-latency E-health applications. IEEE Commun. Lett. 23(5), 896–899 (2019)
13. Baker, S.D., Hoglund, D.H.: Medical-grade, mission-critical wireless networks [Designing an enterprise mobility solution in the healthcare environment]. IEEE Eng. Med. Biol. Mag. 27(2), 86–95 (2008)
14. Lacalle, G., Val, I., Seijo, O., et al.: Analysis of latency and reliability improvement with multi-link operation over 802.11. In: IEEE International Conference on Industrial Informatics (INDIN), Palma de Mallorca, Spain, pp. 1–7 (2021)
15. Al-Maqri, M.A., Othman, M., Ali, B.M., et al.: Adaptive multi-polling scheduler for QoS support of video transmission in IEEE 802.11e WLANs. Telecommun. Syst. 61(4), 773–791 (2016)
16. Fujiwara, R., Miyazaki, M., Katagishi, M.: Low-latency wireless LAN system using polling-based MAC. In: IEEE International Symposium on Circuits and Systems (ISCAS), Melbourne, Australia, pp. 1504–1507 (2014)
17. Chang, C.-Y., Yen, H.-C., Benslimane, A., et al.: A pragmatic VBR stream scheduling policy for IEEE 802.11e HCCA access method. IEEE Trans. Emerg. Top. Comput. 3(4), 514–523 (2015)
18. Lv, Y., Liu, Z., Bi, M., et al.: Selective polling and controlled contention based WLAN MAC scheme for low-latency applications. IEEE Commun. Lett. 27(3), 1050–1054 (2023)
19. IEEE Standard for Information Technology–Telecommunications and Information Exchange between Systems—Local and Metropolitan Area Networks–Specific Requirements—Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. IEEE Std. 802.11-2020 (Revision of IEEE Std 802.11-2016), 1-4379 (2021)
20. Ruan, L., Dias, M.P.I., Wong, E.: Enhancing latency performance through intelligent bandwidth allocation decisions: A survey and comparative study of machine learning techniques. J. Opt. Commun. Netw. 12(4), 20-B32 (2020)
21. Ruan, L., Dias, M.P.I., Wong, E.: Machine learning-based bandwidth prediction for low-latency H2M applications. IEEE IoT J. 6(2), 3743–3752 (2019)
22. Lin, G.-M., Nagamine, M., Yang, S.-N., et al.: Machine learning based suicide ideation prediction for military personnel. IEEE J. Biomed. Health. Inf. 24(7), 1907–1916 (2020)
23. Wong, E., Dias, M.P.I., Ruan, L.: Predictive resource allocation for tactile internet capable passive optical LANs. J. Lightw. Technol. 35(13), 2629–2641 (2017)