Prioritized Clock Synchronization for Event Critical Applications in Wireless IoT Networks

Sabin Bhandari, *Student Member, IEEE*, and Xianbin Wang, *Fellow, IEEE*

***Abstract*— With the proliferation of the Internet of Thing (IoT) technologies in vertical industry applications, provisioning of accurate synchronization and low latency communication has become critical for dense wireless IoT networks to support distributed sensing and control. Due to the contention-based channel access, achieving accurate synchronization in most of unlicensed wireless IoT networks could be extremely challenging. Specifically, the critical challenge at medium access control (MAC) layer for dense IoT communication is how to eliminate random access delay while supporting a large number of heterogeneous nodes with diverse quality of service (QoS) requirements. In this paper, we propose an efficient MAC protocol for supporting distributed synchronization through guaranteed channel access for time-critical sensor nodes in dense wireless IoT networks. The proposed protocol assigns time slots to timestamp packets of the time-critical nodes using a prioritized channel access mechanism, and also guarantees channel access in event-based situations. In addition, the proposed protocol also provides deterministic scheduling for the scenarios where the delay bound of a certain priority traffic changes based on the circumstances of the emergency situation. Our results show that the proposed scheme significantly improves the synchronization precision of the event critical sensor nodes, and also enhances the reliability of overall IoT networks.**

***Index Terms*— Internet of Things (IoT), time synchronization, MAC protocol, access delay.**

I. Introduction

THE emerging Internet of Things (IoT) paradigm is expected to interconnect various objects and processes for massive information collection, analysis, and utilization [1]. The total number of connected sensors and machine type communication nodes has been rapidly increasing over recent years, and leading to dense wireless IoT networks. The interconnectivity, mutual interference, security, and synchronization are the open issues in dense wireless IoT networks.

Precise clock synchronization is one of the crucial issues in a wide range of distributed wireless IoT applications to perform event driven sensing and control. In distributed IoT networks, high synchronization precision is needed to maintain the common reference time for collaborative information exchange and data fusion [2], [3]. The density and closeness of sensor nodes make wireless IoT network more susceptible to packet congestion. The probability of packet loss also increases because of the collisions, the link constraints, and the retry limit [4]. Subsequently, the uncertainties arising in packet delays affect the overall clock synchronization process. To overcome the uncertainties that arise in the packet delivery process, a series of standards such as IEEE 802.15.4e [5], IEEE 802.11ah [6] and WirelessHart [7] have been proposed in recent years. Regardless of their extensive applications in industries, there is a lack of effective mechanisms in supporting the real time industrial applications. As presented in ISA 100, the data traffics in the automation and control applications are classified into safety, control, and monitoring based on the reliability and latency requirements [5]. The safety data traffic refers to emergency situations such as fire alarm and automatic shutdown. This kind of data traffic is more critical in nature and requires reliable synchronization to the controller within stringent deadlines. Failing to meet the synchronization bound for this kind of traffic may cause system instability and also pose a threat to human safety.

The major source of synchronization error in industrial IoT networks present in packet delivery is caused by the channel access process. The non-deterministic delays in packet delivery caused by the channel access time can be much larger than the required synchronization precision for the event critical applications [8]. The nature and magnitude of different delay sources in a packet transmission process are described in Section II. The Medium Access Control (MAC) layer is primarily responsible for channel access of nodes within a network that uses a shared medium. In the Time Division Multiple Access (TDMA), the node must wait for the specified time slot before transmitting the packet, whereas in a contention-based mechanism such as Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA), nodes must wait until the channel is clear before transmitting. In addition, the Request-To-Send (RTS)/ Clear-To-Send (CTS) mechanism is also required to exchange the packets.

The non-deterministic random back off delay caused by the channel access mechanism affects the accuracy and precision of the clock synchronization. The IEEE 802.15.4 is a widely used standard for wireless IoT networks, which adopts a hybrid protocol with CSMA/CA and TDMA for channel access [9]. In the Contention Access Period (CAP), CSMA/CA is used for the channel access, whereas in the Contention Free Period (CFP), TDMA scheme is used for data transfer by providing a number of Guaranteed Time Slots (GTS). The GTSs can provide the deterministic channel access for the time

Manuscript received February 14, 2019; revised April 8, 2019; accepted April 8, 2019. Date of publication April 23, 2019; date of current version July 17, 2019. This work was supported by the Natural Sciences and Engineering Research Council of Canada Discovery under Grant RGPIN-2018-06254. The associate editor coordinating the review of this paper and approving it for publication was Prof. Okyay Kaynak. *(Corresponding author: Xianbin Wang.)*

The authors are with the Department of Electrical and Computer Engineering, Western University, London, ON N6A 5B9, Canada (e-mail: sbhanda7@uwo.ca; xianbin.wang@uwo.ca).

Digital Object Identifier 10.1109/JSEN.2019.2912938

critical nodes, however it has following disadvantages: (i) the number of GTSs are limited to seven (i.e., GTS starvation), (ii) to use the GTS, the node has to send GTS request packet in CAP, and wait till the allocation confirmation in the next beacon period, and (iii) the GTS allocation based on the First Come First Served (FCFS) scheme does not provide a prioritized channel access for critical traffic, and also is not suitable for the event critical applications [10]. The range of access delays varies from 10 to 500 ms [11], which is significantly larger than other delay factors. To compensate the non-deterministic packet latency, the synchronization protocol designed for wireless IoT should fulfill the limited resources requirements of IoT applications.

The critical synchronization challenge at MAC layer for dense IoT communication lies in facilitating the guaranteed channel access to the extremely large number of nodes by supporting the unique traffic characteristics and diverse service requirements. In a dense IoT network, the series of timestamp messages transmission is required to estimate the relative clock offsets and skews between IoT nodes. In general, the time synchronization in an IoT network can be regarded as the process of estimation and mitigation of the random latencies during the message transmission in a wireless channel. In wireless IoT networks, several synchronization mechanisms use the MAC layer timestamping procedure to reduce the latency uncertainty. The timestamping at the MAC layer can be done just before the transmission of the packet or immediately after the packet has been received. As a consequence, the non-deterministic delay can be estimated.

In this paper, an efficient clock synchronization scheme is proposed for the irregularly deployed sensor nodes to maximize the synchronization precision of the event critical applications, and to further enhance the reliability of overall IoT networks. All the sensor nodes associated with the coordinator get the reference clock information during the network initialization and perform clock synchronization in a beacon interval. The critical nodes also perform the synchronization with every event detection to enhance the clock precision. In the proposed scheme, a priority-based deterministic channel access mechanism is employed to reduce the access latency by assigning different MAC layer attributes. Also, an Emergency Indication (EI) slot is used to prioritize the critical traffic over normal traffic. Finally, the performance of the proposed scheme is analyzed, and the synchronization precision of the event critical nodes is compared with normal nodes.

The rest of this paper is organized as follows. Section II presents the sources of clock synchronization error in detail. Section III summarizes the related works and provides the literature review in the area of synchronization. Section IV describes the system model and proposed scheme in detail. Section V evaluates the performance of the proposed method via simulations. Section VI concludes this paper.

II. Sources of Clock Synchronization Error

In wireless IoT networks, the uncertainties in the packet delays cause clock estimation errors [12]. The delay sources in the packet transmissions that affect the synchronization process are as follows;

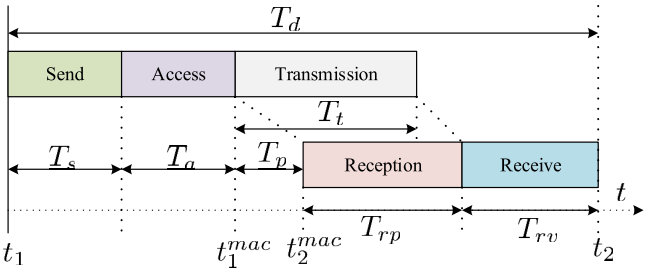

Fig. 1. Illustration of packet transmission and reception delays between two IoT nodes.

A. Send Time, T_s

The overall time required to prepare the packet at the application layer and send to the network layer, which may include the delay introduced by the operating system. The nature of this delay is non-deterministic.

B. Access Time, T_a

This delay is introduced at MAC layer during the channel access process. This is one of the major components in synchronization and also highly variable depending upon the application specific protocol.

C. Transmission, T_t/ Reception Time, T_{rp}

The time required for transmitting/ receiving a message at the physical (PHY) layer. This delay depends on the packet size and the rate of the wireless channel, and is deterministic in nature. The illustration of the packet delivery latency over the wireless link is presented in Fig. 1.

D. Propagation Time, T_p

The actual time required to transmit a message from sender to receiver through the wireless channel.

E. Receive Time, T_{rv}

The overall time spent to encode and transmit a received message to the application layer at the receiver side.

The total delivery delay, T_d can be calculated as

$$T_d = T_s + T_a + T_p + T_{rp} + T_{rv}. \quad (1)$$

The above delivery delay is directly related to the synchronization error. The delay can be compensated for as long as it can be computed accurately. The latency components can be classified into two categories; deterministic (i.e., fixed components) and non-deterministic/stochastic (i.e., random or variable components) [13]. The variable components depend on several network factors such as network traffic, channel conditions, and the number of nodes, and thus, there is no single mathematical model to estimate errors that fit every application scenario. The *Propagation time* is negligible in most of the IoT scenarios, as it is much smaller than the clock resolution. The *Reception Time* and *Transmission Time* are dependent on the message length and radio frequency. However, *Send time*, *Receive time*, and *Access time* are rather unpredictable, causing the delay uncertainty on the IoT networks. The values and the

TABLE I
DELAY SOURCES IN PACKET TRANSMISSIONS

Delay factor	Value	Nature
Send/Receive	0-100 ms	Non-deterministic, Depends on processor capacity
Access	10-500 ms	Non-deterministic, Depends on channel contention
Transmission/Reception	10-20 ms	Deterministic, Depends on packet size
Propagation	$<1\mu s$ (up to 300 meters)	Deterministic, Depends on distance
Interrupt Handling	$<5\mu s$ (in most cases)	Non-deterministic, Depends on interrupts
Encoding/Decoding	100-200μs	Deterministic, Depends on radio chipset
Byte Alignment	0-400μs	Deterministic, Can be calculated

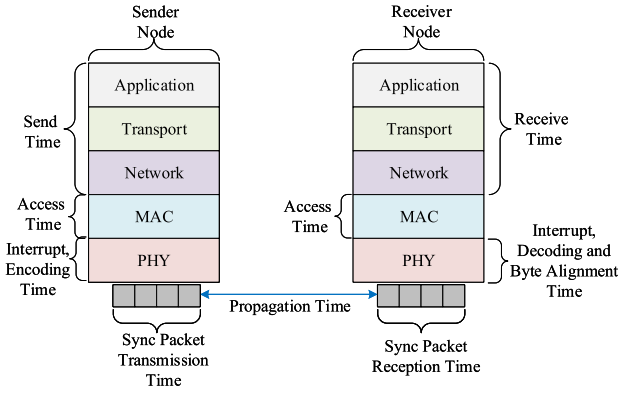

Fig. 2. Sources of delay uncertainties in packet delivery process.

nature of the various latencies are summarized in Table I [11]. The delay uncertainties in the packet delivery process are depicted in Fig. 2.

III. RELATED WORKS

In a distributed IoT network, each node has its own internal clock, and operates in different oscillation frequencies. In real time, the clock may drift in a micro-second scale with other nodes, and clock error can be accumulated over the time. As a consequence, IoT nodes may not operate in a synchronized state. In the traditional distributed wireless system, to overcome the clock synchronization issues, two basic schemes are adopted in the literature. The first scheme deals with the physical clock of the nodes, and the second scheme deals with the logical clock. In physical clock synchronization, each node agrees to tune to a common clock value, whereas, in logical clock synchronization, the knowledge of chronological ordering of real events is required [14]. Network Time Protocol (NTP) is widely used on the Internet for physical clock synchronization [15]. The NTP synchronizes the nodes to the UTC (Coordinated Universal Time), and follows the hierarchical client-server architecture. However, this protocol is designed for the traditional wired system, and does not suit for dense IoT due to the limited battery life and dynamic topology. The remote clock reading scheme is proposed to estimate the non-deterministic message latency by considering the client-server architecture [16]. The main drawbacks of this scheme are: (i) message complexity, and (ii) time uncertainty due to the network routing and traffic conditions.

In [17], the authors proposed Time-sync Protocol for Wireless Sensor Network (TPSN) based on two-way message exchange between the sender and receiver. The pairwise synchronization is performed at the edge of the hierarchical model. However, the TPSN suffers from non-deterministic message delay from the sender, and also is unable to perform the relative skew estimations and corrections. The Flood Time Synchronization Protocol (FTSP) is proposed for WSNs based on the one-way broadcast message from a sender to the multiple receivers [11]. At first, the global clock reference node is elected, and a spanning tree is built at the reference node. Afterwards, each node synchronizes with its parent node. The major limitations of FTSP are: (i) suffer from substantial overhead to elect the new root node if the root dies, and (ii) non-deterministic synchronization error due to the message route in the constructed tree. Based on the FTSP, authors in [18] presented the Ratio-based time Synchronization Protocol (RSP) to improve the estimation accuracy by eliminating the message delay. The RSP uses the two synchronization packets to estimate the clock drift of the receiver with the sender (i.e., root node).

A Reference Broadcast Synchronization (RBS) based on the receiver to receiver synchronization for WSNs is presented in [19]. The nodes calculate the relative offset and skew based on the reception times of the beacon message. Nevertheless, this scheme uses a series of message exchange to estimate both relative skew and offsets. The RBS is not appropriate for self-organized IoT networks due to the temporal data path failure and high probability of packet collisions. The Pairwise Broadcast Synchronization (PBS) based on the Receiver Only Synchronization (ROS) and Sender-Receiver Synchronization (SRS) is proposed in [20] to achieve network-wide synchronization. In [21], the authors presented the Relative Reference less Receiver-Receiver Synchronization (R^4Sync) based on the receiver-to-receiver synchronization scheme. To eliminate the single point failure problem in the RBS, this scheme allocates the reference clock function to all nodes instead of only one node. The authors in [22] introduced an extended and improved Emergent Broadcast Slot (EBS) scheme to guarantee robust communication between nodes. The EBS mechanism is fully decentralized, which facilitates collaboration between nodes within the duty-cycle to avoid packet collisions. Nevertheless, in most of the existing schemes, channels and time slots are randomly assigned for the packet transmissions, and less attention has been provided to the guaranteed synchronized channel access for the event critical sensor nodes.

IV. SYSTEM MODEL AND PROPOSED SCHEME

In this paper, we consider an industrial wireless IoT scenario composed of N number of sensor nodes. A high level of synchronization accuracy is required to maintain the time consistency of critical sensing information, and to ensure the system reliability to fulfill the censorious requirements of the industrial systems. We, therefore, have divided the data gathered by N nodes into two classes, i.e., time-critical data and normal data traffic. The sporadic time-critical packets are triggered by high priority P_h nodes, and is bounded by strict deadlines. The normal data packets are periodically generated by low priority P_l nodes for process-related sensing and measurement.

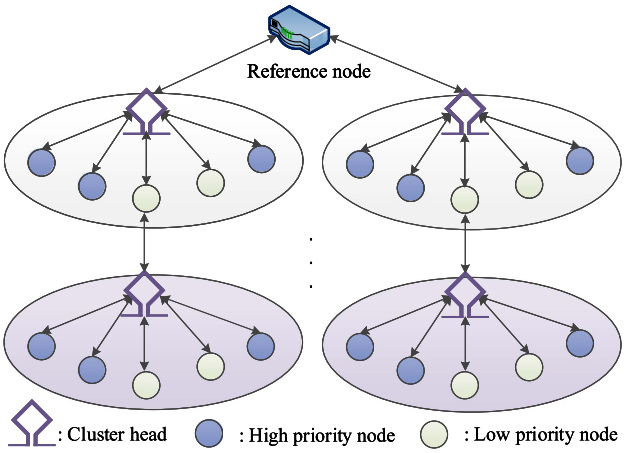

Fig. 3. Pictorial representation of the network scenario. The sensor nodes, including Cluster Head (CH) and the intermediate nodes are organized in a hierarchical model.

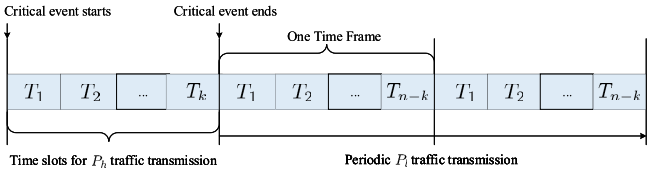

Fig. 4. Superframe structure of the proposed priority-based protocol.

In the proposed network, all the deployed nodes are associated with the Access Point (AP)/Cluster Head (CH). The sensor nodes, including CH and the intermediate nodes have child-parent relationships, i.e., organized in the hierarchical model. Moreover, the proposed topology is considered as a static network over the time. The pictorial representations of the network scenario is presented in Fig. 3. The CH is responsible for distributing the clock information to different sensors. Similarly, the intermediate nodes are responsible for transferring the timestamp information from one hop to another.

In the proposed scheme, the P_l traffic transmission follows the conventional channel access scheme, and has been allocated a time slot with a duration T_s to transmit its data. The proposed superframe structure is shown in Fig. 4. The P_l node will initiate its transmission if the channel is found to be idle. Otherwise, the node will defer its transmission and wait until its next time slot. The Emergency Indication (EI) slot with duration T_{ei} can be used to prioritize P_h traffic over P_l. The packet transmission between P_h nodes and the CH is carried out in the form of consecutive transmission cycles. Each transmission cycle is composed of a time slot with a duration T_s and an EI slot with duration T_{ei}. The EI slot in each transmission cycle is followed by three phases; Clear Channel Assessment (CCA), Packet Schedule Phase (PSP), and Packet Transmission Phase (PTP). The CCA is further divided into l dedicated subslots with duration T_l to send its channel access request. The transmission cycle of P_h is presented in Fig. 5. The deterministic channel access mechanism for the time critical nodes to complete its transmission within a delay bound, and clock synchronization and estimation process in the distributed network are detailed in subsections IV-A and IV-B, respectively.

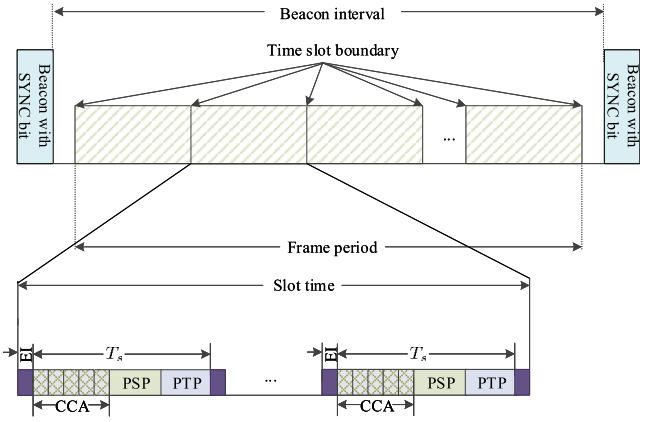

Fig. 5. Illustration of the transmission cycle of P_h traffic in the proposed protocol.

A. Deterministic Channel Access Mechanism

The non-deterministic random access delay is the major source of error in the clock synchronization process. The access delay mainly depends on the employed MAC layer protocol. The critical MAC layer challenge is to facilitate the deterministic channel access to a large number of sensor nodes with diverse service requirement and unique traffic characteristic. In this regard, we present the priority based deterministic channel access mechanism to reduce the channel access delay of the time-critical packets.

If node n_i has a P_h packet to send, it firstly transmits an indication signal within the EI time to take the slot from P_l nodes. After that, the node n_i asks for the guaranteed channel allocation for its P_h data transmission by sending a reservation packet to CH in the CCA subslot. The reservation request packet contains a binary payload, and corresponds to the decimal value d_i of the relative deadline (ms) of the P_h packet which needs to be delivered by n_i [23]. During the PSP, the activated P_h nodes are schedule based on the Earliest Due Date (EDD). Each node n_i will be scheduled for the channel access based on its d_i; such as the lowest d_i will gain the highest priority in the EDD schedule, and immediately access the channel [24]. This mechanism prioritizes the most urgent traffic for channel access to complete its transmission within a deadline bound.

Let n be the number of contending stations for the medium access in the given slot. In the case of saturation conditions, each station immediately transfers the packet after the completion of each successful transmission. However, due to the consecutive transmissions, each packet needs to wait for the random backoff interval before transmitting. The new arrival of the high priority packet P_h will immediately preempt lower priority packets currently being served on the queue, and get access to the services. During the channel access process, if the node detects a busy channel in present l subslots, then the node initiates a backoff stage. In the backoff stage, the node waits for a random number of time slots in the range of 0 to 2^{BE}, where BE is the backoff exponent and initial value is defined as $macMinBE$. For the delay estimation, we adapt the mathematical expression of CSMA/CA scheme of IEEE 802.15.4 presented in [25], [26]. Using this

mathematical model, the expected service time in the network can be presented as [25]

$$E[S] = E[D] + T_{Tx} + 2T_{turn} + T_{ACK}, \tag{2}$$

where $E[D]$ represents the time period from the epoch that the packet just arrives at the queue to the epoch just before packet transmission, or discarded. The T_{ACK} and T_{Tx} are the transmission time of the acknowledgment and data packet respectively, and T_{turn} represents the turnaround time. The parameter $E[D]$ in the above expression depends on the CSMA/CA mechanism and is affected by several MAC parameters such as $macMaxBE$, $macMinBE$, CW, and $macMaxCSMABackoffs$. The expected value $E[D]$ can be derived as [25]

$$E[D] = \sum_{\upsilon=0}^{m} \alpha^{\upsilon}(1-\alpha)\left\{\sum_{i=0}^{\upsilon}\frac{CW_i - 1}{2}\sigma + (\upsilon+1)T_{CCA}\right\} + \alpha^{m+1}\left\{\sum_{i=0}^{m}\frac{CW_i - 1}{2}\sigma + (m+1)T_{CCA}\right\}, \tag{3}$$

where T_{CCA} is the time interval for performing CCA, α is the channel busy probability, and σ is the length of the backoff slot. The contention window size for the ith retry is expressed by; $CW_i = min\left\{2^i macMinBE, macMaxBE\right\}$. The default values of $macMinBE$ and $macMaxBE$ are 3 and 5, respectively [25]. The data packets are dropped or discarded after $m+1$ attempts at CCA, and consequently the packet loss rate is expressed by [25]

$$P_{loss} = \alpha^{m+1}. \tag{4}$$

Then, the channel busy probability α can be expressed in term of P_{loss} as [25]

$$\alpha = \frac{(n-1)(1-P_{loss})E[n_\tau](T_{CCA}+T_{Tx}+2T_{turn}+T_{ACK})}{\frac{1}{\lambda}+E[n_\tau]E[D]}, \tag{5}$$

where n is the number of sensor nodes, n_τ is the number of data packets served in a busy period of the queuing system, and $E[n_\tau] = \frac{1}{1-\rho}$ [27]. Therefore, by solving the above equations, *i.e.*, (3), (4), and (5), we can get the corresponding values of α, P_{loss}, and $E[D]$.

B. Distributed Clock Synchronization and Estimation Process

In this subsection, we present the clock synchronization and estimation process in distributed IoT network in detail. Moreover, we also describe the effects of a number of nodes and hop counts on the overall precision of the clock synchronization.

1) Clock Synchronization: The clocks may drift due to the variations in the oscillators, and the duration of the time intervals of different events will not be noticed identical among the IoT nodes. In general, the clock function of the ith node can be expressed as

$$C_i(t) = f_i t + \theta_i, \tag{6}$$

where the parameters f_i and θ_i are clock skew (i.e., frequency difference) and clock offset (i.e., phase difference),

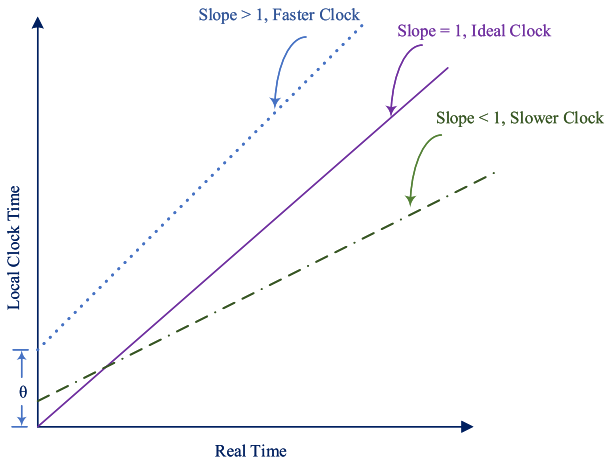

Fig. 6. Clock model of IoT nodes.

respectively. Similarly, the clock relationship between two nodes can be represented as

$$C_j(t) = f_{ij}C_i(t) + \theta_{ij}, \tag{7}$$

where f_{ij} and θ_{ij} are the relative clock skew and offset between node i and j, respectively. Let the clock of node i be the reference clock, then the function of the clock synchronization is to estimate the value of f_{ij} and θ_{ij} such that node j can adjust its timing information with reference node, when it is needed. The two nodes are perfectly synchronized when the value of $f_{ij} = 1$ and $\theta_{ij} = 0$. If the IoT network consists of n number nodes, then the global network-wide synchronization requires $C_i(t) = C_j(t)$ for all $i, j = 1, 2, 3, \ldots, n$ (i.e., all the relative clock skews and offsets should be estimated with respect to a reference clock). The graphical representation of the clock model of nodes is shown in Fig. 6.

The time synchronization error between clocks of node i and j at real time is given by $\mid C_i(t) - C_j(t) \mid$. Moreover, the average synchronization error of the overall network is the average of the time difference between every pair of nodes in the given network. For n number of nodes in the given network, the average synchronization error at real time t is given by

$$\zeta = \frac{2}{n(n-1)}\sum \mid C_i(t) - C_j(t) \mid, \quad \forall i,\ i \neq j. \tag{8}$$

The main objective of the time synchronization scheme is to ensure that the average synchronization error at time t should be less than the maximum acceptable error. In the IoT network, the maximum acceptable error is application dependent and is in the range of μs. In general, the clock parameters are changes due to several factors such as temperature, pressure, voltage fluctuation, and hardware aging, thus, the network should perform periodic resynchronization mechanisms to adjust the clock parameters. The clock precision/accuracy is a measurement of the deviation of the error from the mean, and the stability measures the mean variation with respect to the above-mentioned factors [28].

2) Clock Estimation: The i reference node broadcasts ω_i number of packets for synchronization, and the nodes reply back with the η_j acknowledgment messages, where η_j is not

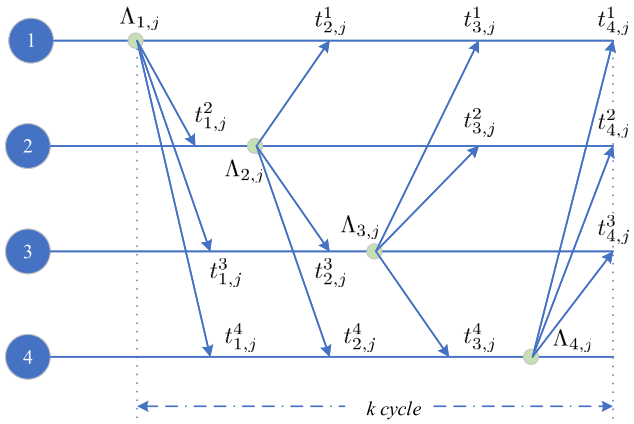

Fig. 7. Synchronization packet broadcast during one cycle.

necessarily equal to ω_i. The latency for line-of-sight (LOS) transmission between i reference node and the jth sensor node can be expressed as [29]

$$\tau_{i,j} = (f_j C^{(k)}_{i,j} + \theta_j) - (f_i t^{(k)}_i + \theta_i) + \varsigma^{(k)}_{i,j}, \quad k = 1, 2, \ldots, K_i, \tag{9}$$

where $t^{(k)}_i$, $C^{(k)}_{i,j}$ are the transmission and reception timestamps of i reference node and jth sensor node, respectively, and $\varsigma^{(k)}_{i,j}$ is the aggregate estimation error on the timestamps. The transmission and reception timestamps recorded in the i reference node and jth sensor node can be presented as $T_i = [t^{(1)}_i, t^{(2)}_i, \ldots, t^{(K_i)}_i]^T \in \mathbb{R}^{K_i \times 1}$ and $R_{i,j} = [C^{(1)}_{i,j}, C^{(2)}_{i,j}, \ldots, C^{(K_i)}_{i,j}]^T \in \mathbb{R}^{K_i \times 1}$, where K_i is the number of transmission made by the i reference node. Similarly, the error vector can be expressed as $\varsigma_{i,j} = [\varsigma^{(1)}_{i,j}, \varsigma^{(2)}_{i,j}, \ldots, \varsigma^{(K_i)}_{i,j}]^T \in \mathbb{R}^{K_i \times 1}$.

For the neighborhood of n nodes, each synchronization packet piggybacks $n-1$ previously received timestamps. The cycle of synchronization packet broadcasting is illustrated in Fig. 7, where $\Lambda_{i,j}$ denotes the jth synchronization packet of the node i, and $t^k_{i,j}$ denoted the reception timestamp at node k of the jth synchronization packet of node i. These timestamps are then used as samples to estimate relative skew and offset. The synchronization between node 1 and node 2 occur by gathering timestamp pairs $(t^1_{3,j}, t^2_{3,j})$ and $(t^1_{4,j}, t^2_{4,j})$. The timestamps obtained by each node can be arranged into the matrix of reception timestamps as [30]

$$T = \begin{bmatrix} * & t^1_{2,j} & t^1_{3,j} & t^1_{4,j} \\ t^2_{1,j} & * & t^2_{3,j} & t^2_{4,j} \\ t^3_{1,j} & t^3_{2,j} & * & t^3_{4,j} \\ t^4_{1,j} & t^4_{2,j} & t^4_{3,j} & * \end{bmatrix}, \tag{10}$$

where the * denotes the broadcasting node.

If u_k and v_k, $k \in \{1, \ldots, I\}$ denote the kth message reception timestamp of nodes 1 and 2, respectively, synchronization messages received by both nodes are used to construct timestamp samples. By using the maximum-likelihood (ML) estimation, we can obtained the resultant estimates for relative skew and offset using the timestamp values as

$$f_{ML} = \frac{\sum_{k=1}^{I} u_k \sum_{k=1}^{I} v_k - I \sum_{k=1}^{I} v_k u_k}{(\sum_{k=1}^{I} v_k)^2 - I \sum_{k=1}^{I} v_k^2}, \tag{11}$$

$$O_{ML} = \frac{1}{I}\Big(\sum_{k=1}^{I} u_k - \frac{\sum_{k=1}^{I} u_k \sum_{k=1}^{I} v_k - I \sum_{k=1}^{I} v_k u_k}{(\sum_{k=1}^{I} v_k)^2 - I \sum_{k=1}^{I} v_k^2} \sum_{k=1}^{I} v_k\Big). \tag{12}$$

In the synchronization process, sensor nodes estimate the relative skew and offset with respect to neighbor nodes locally and independently. However, in a multi-hop scenario, the synchronization between end points is also required. The intermediate nodes may have to forward the synchronization parameters to the next hop to calculate the relative parameters. Consider the node n_1 needs to synchronize with the remote node n_r with the established link is $\{n_1, n_2, n_3, \ldots, n_r\}$. In this case, the intermediate nodes n_i have to estimate the local relative parameters and forward to the n_{i+1}. Let t_{n_i} be the n_i's clock time at instant t, and $f_{n_i \to n_j}$ and $\theta_{n_i \to n_j}$ represent the relative skew and offset between nodes n_i and n_j, respectively. The clock reading of the intermediate nodes in the link can be expressed as [21]

$$t_{n_i} = f_{n_{i+1} \to n_i} t_{n_{i+1}} + \theta_{n_{i+1} \to n_i}, \quad i \in \{1, 2, \ldots, r-1\}. \tag{13}$$

Consequently, multi-hop estimators can be formulated as

$$f_{n_r \to n_1} = \prod_{i=1}^{r-1} f_{n_{i+1} \to n_i}, \tag{14}$$

$$\theta_{n_r \to n_1} = \sum_{i=2}^{r-1} \Big[\Big(\prod_{j=2}^{i} f_{n_j \to n_{j-1}}\Big)\theta_{n_{i+1} \to n_i}\Big] + \theta_{n_2 \to n_1}. \tag{15}$$

The ever-increasing number of connected nodes in ultra-dense IoT networks and the dynamic traffic patterns increase the channel access delay and packet collision rate. In this regard, we utilized a priority based fast and efficient channel access scheme to enhanced the synchronization precision of the event critical IoT nodes. For multi-hop environments, intermediate nodes forward local synchronization parameters to the communicating nodes to allow them to calculate multi-hop parameters. In the presented network scenario, we assumed that each node had its own internal clock and ran independently from other nodes. The synchronization is guaranteed by estimating clock relative parameters of the reference node. In spite of continuous synchronization message broadcasting, the duty cycling scheme is enabled. The nodes are synchronized at the beginning of the time slot. In each duty cycle, the reference node broadcasts the beacons with a synchronization message. The synchronization message contains a timestamp, reference node ID, and sequence number. By incorporating the timestamp information within the beacon frames, the communication overhead can be significantly reduced as compared to the RBS like protocols [21]. We have also assumed that each node has the capability to communicate with at least one remaining node and to discover its one-hop neighbors.

The proposed scheme allows a network to determine a broadcast sequence by which nodes transmit and forward messages, and then conducts a network-wide synchronization based on received timestamp information. All the employed nodes are associated with coordinator/reference node. In the network initialization phase, all the nodes belonging to the

Algorithm 1 Clock Adjustment Procedure of Event Critical Applications

Event detection: Timer Started
  Set.timer(T)→ $C_i(t) = f_i t + \theta_i$;
  Send.msg();
Emergency Indication (EI): ON
Perform Priority Channel Access as presented in IV-A
Messages Received
  Gather Timestamps
  Perform Clock Estimation Procedure
Clock Adjustment
  Set.timer(T)→ $C_j(t) = f_{ij} C_i(t) + \theta_{ij}$;
  Send.msg();
  return;

TABLE II
SIMULATION PARAMETERS

Parameters	Value
Max Backoff Exponent	5
Min Backoff Exponent	3
Max CSMA Backoff	4
MAC Frame Payload	800 bits
Queue Size	51 frames
Data Rate	19.2 kbps
ACK Size	88 bits
MAC Overhead	48 bits
σ	0.32 ms
T_{ACK}	0.352 ms
T_{Tx}	1.12 ms
T_{CCA}	0.25 ms
T_{turn}	0.192 ms

same CH get the reference clock information during the synchronization period and contend to access the channel for data transfer. Moreover, to enhance the synchronization precision of the event critical nodes, the P_h nodes perform the clock estimation process with every event detection. The events are signaled by the hardware interrupts and depend on the specific platform [31]. The clock adjustment procedure of event critical applications is presented in Algorithm 1.

V. PERFORMANCE ANALYSIS

In this section, we analyze and evaluate the performance of the proposed scheme in terms of the synchronization precision. The simulation parameters are presented in Table II.

Figure 8 shows the access delay of packets with different priority level versus the number of sensor nodes in IoT network. The packet arrival pattern follows the Poisson distribution with the arrival rate of λ packets per second. The average access delay of both low and high priority packets increases as the number of node increases, due to the contention mechanism and higher service time. The low priority packets have the longer access delay as compared to the high priority packets because the access time must accommodate the interrupts of the high priority event critical packets. In addition, due to the guaranteed channel access and deterministic packet scheduling in event based situation, the critical packets do not get any interruptions from low priority normal packets and hence, enhance the overall synchronization precision of the critical nodes.

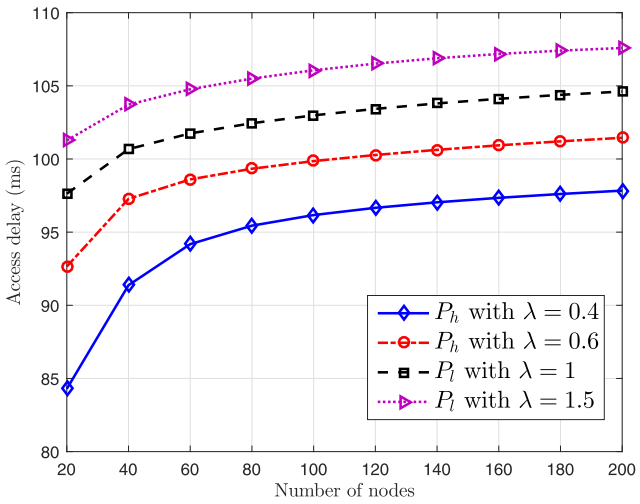

Fig. 8. Performance evaluation of the proposed priority-based channel access mechanism in terms of the access delay.

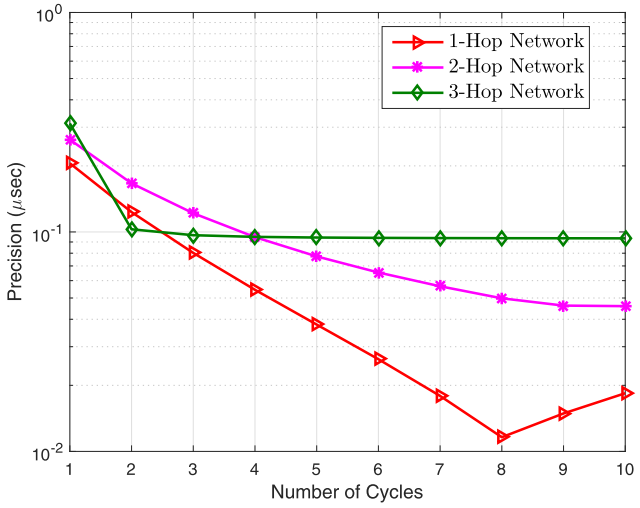

Fig. 9. Performance comparison on synchronization precision versus number of cycles in multi-hop scenarios.

Figure 9 presents the effect of the network hops in the synchronization process. The nodes in each hop should run the synchronization process locally and independently as described in Section IV. However, in the case of the multi-hop scenarios, the intermediate nodes have to forward the local synchronization parameters to the nodes in next hop, and allow them to estimate the relative parameters. From Fig. 9, it can be seen that the increase of synchronization samples/cycles considerably improves the precision with the number of hops. Also, the precision is found within the acceptable range even for the small number of synchronization samples/cycles. However, the synchronization precision decreases with the number of hops. The perceptive reason for the above mentioned trend is as following. The number of packet drops and collisions becomes more concurrent, and the probability of packet loss increases due to the retry limits and link constraints.

Figures 10 and 11 depict the synchronization precision versus number of cycles for a network with 100 and 200 nodes, respectively. In both scenarios, 30 percent of the network consists of the event critical high priority nodes (i.e., P_h nodes).

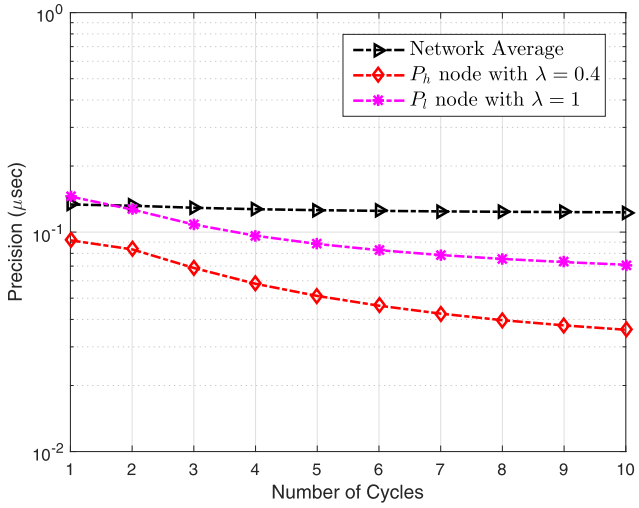

Fig. 10. Performance comparison of critical nodes with normal nodes in terms of synchronization precision (Total number of sensor nodes = 100, and 30 percent of nodes are P_h).

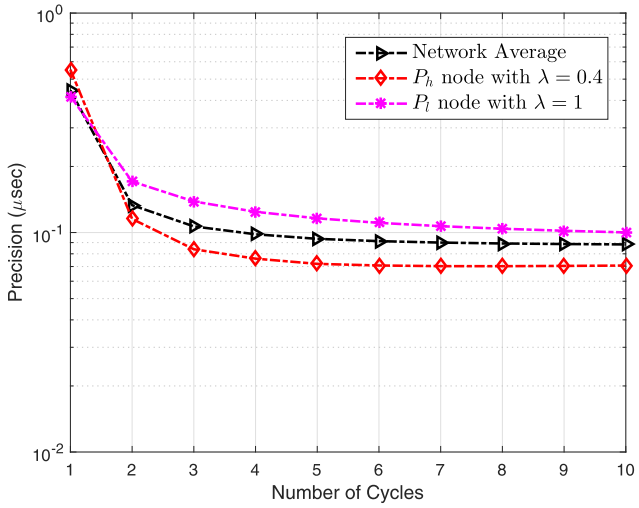

Fig. 11. Performance comparison of critical nodes with normal nodes in terms of synchronization precision (Total number of sensor nodes = 200, and 30 percent of nodes are P_h).

From both figures, it is observed that the proposed scheme maintains the synchronization accuracy of the critical nodes much precise as compared to the normal nodes. The synchronization accuracy of the first scenario is more precise than the second case because of a smaller number of nodes in the network. The average precision deteriorates as the number of nodes increases due to the channel congestion and packet collisions. The proposed scheme guarantees predictable and timely channel access for the time-critical sensor nodes in wireless IoT networks. Moreover, due to the improved channel access mechanism and queuing policy, the network reliability of the event critical nodes is noted to be higher than that of normal nodes.

Figure 12 shows the performance of the synchronization precision of proposed mechanism in multi-hop scenario. It is clearly observed that the synchronization precision deteriorates as the number of hops increases from 1 to 2. In the multi-hop scenario, the packet loss probability increases because of the

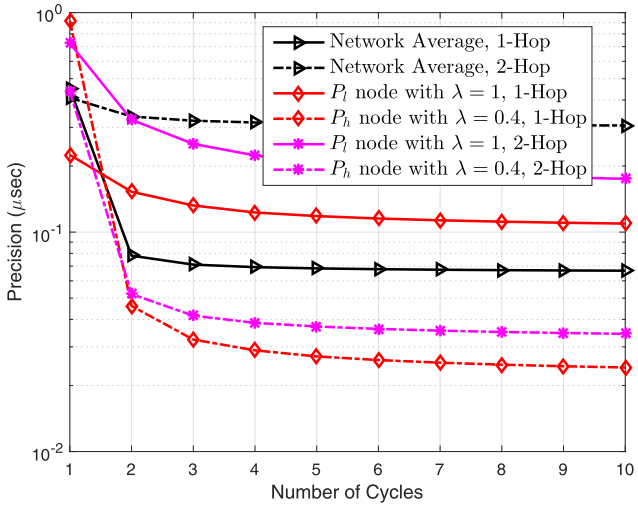

Fig. 12. Performance comparison of critical nodes with normal nodes in terms of synchronization precision in multi-hop scenario (Total number of sensor nodes = 100, and 30 percent of nodes are P_h).

link constraints, packet collisions, and retry limits. Moreover, the slot allocation decision coupled with predictable channel access mechanism contributes to bettering the synchronization accuracy of event critical sensor nodes as compared to normal nodes.

VI. Conclusions

In distributed IoT network, precise clock synchronization mechanism is essential to perform the event driven sensing and control. The resource constrained IoT nodes are affected by packet delays, packet loss, and packet collisions. The packet collision rate increases significantly with node density. The problem of network synchronization becomes worse in IoT networks due to the uncertainties arising in the packet delivery process. In this paper, we proposed an efficient clock synchronization scheme for the event critical applications in wireless IoT. The proposed scheme assigns time slots with high preference to the timestamp packets of critical nodes, and also guarantees the channel access in event based situations. Furthermore, the proposed scheme provides the deterministic packet scheduling, reduces the channel access delay, and enhances the synchronization precision. With the help of the simulation results, it has been shown that the proposed scheme substantially improves the synchronization precision of the event critical sensor nodes in comparison to the normal nodes. In our future work, we plan to implement the proposed algorithm in real hardware to analyze the performance of the proposed scheme in real-world IoT applications such as industrial automation and control sub-systems.

References

[1] J. Lin, W. Yu, N. Zhang, X. Yang, H. Zhang, and W. Zhao, "A survey on Internet of Things: Architecture, enabling technologies, security and privacy, and applications," *IEEE Internet Things J.*, vol. 4, no. 5, pp. 1125–1142, Oct. 2017.

[2] F. Lamonaca, A. Gasparri, E. Garone, and D. Grimaldi, "Clock synchronization in wireless sensor network with selective convergence rate for event driven measurement applications," *IEEE Trans. Instrum. Meas.*, vol. 63, no. 9, pp. 2279–2287, Sep. 2014.

[3] K. S. Kim, S. Lee, and E. G. Lim, "Energy-efficient time synchronization based on asynchronous source clock frequency recovery and reverse two-way message exchanges in wireless sensor networks," *IEEE Trans. Commun.*, vol. 65, no. 1, pp. 347–359, Jan. 2017.
[4] M.-H. Zayani, V. Gauthier, and D. Zeghlache, "A joint model for IEEE 802.15.4 physical and medium access control layers," in *Proc. IEEE IWCMC*, Jul. 2011, pp. 814–819.
[5] Q. Wang and J. Jiang, "Comparative examination on architecture and protocol of industrial wireless sensor network standards," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 3, pp. 2197–2219, 3rd Quard., 2016.
[6] S. Bhandari, S. K. Sharma, and X. Wang, "Device grouping for fast and efficient channel access in IEEE 802.11ah based IoT networks," in *Proc. IEEE Int. Conf. Commun. Workshops (ICC)*, May 2018, pp. 1–6.
[7] S. Petersen and S. Carlsen, "Wirelesshart versus ISA100.11a: The format war hits the factory floor," *IEEE Ind. Electron. Mag.*, vol. 5, no. 4, pp. 23–34, Dec. 2011.
[8] M. K. Maggs, S. G. O'Keefe, and D. V. Thiel, "Consensus clock synchronization for wireless sensor networks," *IEEE Sensors J.*, vol. 12, no. 6, pp. 2269–2277, Jun. 2012.
[9] T. N. Dinh and P. H. Ha, "Advanced GTS scheduling in IEEE 802.15.4 networks for industrial application," in *Proc. 16th IEEE Annu. Consum. Commun. Netw. Conf. (CCNC)*, Jan. 2019, pp. 1–4.
[10] H. Farag, A. Mahmood, M. Gidlund, and P. Oesterberg, "PR-CCA MAC: A prioritized random CCA MAC protocol for mission-critical IoT applications," in *Proc. IEEE ICC*, May 2018, pp. 1–6.
[11] M. Maróti, B. Kusy, G. Simon, and Á. Lédeczi, "The flooding time synchronization protocol," in *Proc. 2nd Int. Conf. Embedded Netw. Sensor Syst.* New York, NY, USA: ACM, Nov. 2004, pp. 39–49.
[12] B. J. Choi, H. Liang, X. Shen, and W. Zhuang, "DCS: Distributed asynchronous clock synchronization in delay tolerant networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 23, no. 3, pp. 491–504, Mar. 2012.
[13] D. Shrestha, Z. Pang, and D. Dzung, "Precise clock synchronization in high performance wireless communication for time sensitive networking," *IEEE Access*, vol. 6, pp. 8944–8953, Feb. 2018.
[14] S. Youn, "A comparison of clock synchronization in wireless sensor networks," *Int. J. Distrib. Sensor Netw.*, vol. 9, no. 12, Dec. 2013, Art. no. 532986.
[15] D. L. Mills, "Internet time synchronization: The network time protocol," *IEEE Trans. Commun.*, vol. 39, no. 10, pp. 1482–1493, Oct. 1991.
[16] F. Cristian, "Probabilistic clock synchronization," *Distrib. Comput.*, vol. 3, no. 3, pp. 146–158, Sep. 1989.
[17] S. Ganeriwal, R. Kumar, and M. B. Srivastava, "Timing-sync protocol for sensor networks," in *Proc. 1st Int. Conf. Embedded Netw. Sensor Syst.*, New York, NY, USA: ACM, Nov. 2003, pp. 138–149.
[18] J.-P. Sheu, W.-K. Hu, and J.-C. Lin, "Ratio-based time synchronization protocol in wireless sensor networks," *Telecommun. Syst.*, vol. 39, no. 1, pp. 25–35, Sep. 2008.
[19] J. Elson, L. Girod, and D. Estrin, "Fine-grained network time synchronization using reference broadcasts," *SIGOPS Oper. Syst. Rev.*, vol. 36, no. SI, pp. 147–163, Dec. 2002.
[20] K.-L. Noh, E. Serpedin, and K. Qaraqe, "A new approach for time synchronization in wireless sensor networks: Pairwise broadcast synchronization," *IEEE Trans. Wireless Commun.*, vol. 7, no. 9, pp. 3318–3322, Sep. 2008.
[21] D. Djenouri, "R^4Syn: Relative referenceless receiver/receiver time synchronization in wireless sensor networks," *IEEE Signal Process. Lett.*, vol. 19, no. 4, pp. 175–178, Apr. 2012.
[22] P. Yadav, J. A. McCann, and T. Pereira, "Self-synchronization in duty-cycled Internet of things (IoT) applications," *IEEE Internet Things J.*, vol. 4, no. 6, pp. 2058–2069, Dec. 2017.
[23] J. Åkerberg, M. Gidlund, and M. BjÃ¶rkman, "Future research challenges in wireless sensor and actuator networks targeting industrial automation," in *Proc. 9th IEEE Int. Conf. Ind. Inform.*, Jul. 2011, pp. 410–415.
[24] H. Farag, M. Gidlund, and P. Österberg, "A delay-bounded MAC protocol for mission- and time-critical applications in industrial wireless sensor networks," *IEEE Sensors J.*, vol. 18, no. 6, pp. 2607–2616, Mar. 2018.
[25] T. O. Kim, J. S. Park, H. J. Chong, K. J. Kim, and B. D. Choi, "Performance analysis of IEEE 802.15.4 non-beacon mode with the unslotted CSMA/CA," *IEEE Commun. Lett.*, vol. 12, no. 4, pp. 238–240, Apr. 2008.
[26] P. Park, P. Di Marco, P. Soldati, C. Fischione, and K. H. Johansson, "A generalized Markov chain model for effective analysis of slotted IEEE 802.15.4," in *Proc. IEEE 6th Int. Conf. Mobile Adhoc Sensor Syst.*, Oct. 2009, pp. 130–139.
[27] S. Bhandari, S. K. Sharma, and X. Wang, "Latency minimization in wireless IoT using prioritized channel access and data aggregation," in *Proc. IEEE Global Commun. Conf.*, Dec. 2017, pp. 1–6.
[28] *IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems*, IEEE Standard 1588-2008 (Revision of IEEE Std 1588-2002), Jul. 2008, pp. 1–300.
[29] S. P. Chepuri, R. T. Rajan, G. Leus, and A.-J. van der Veen, "Joint clock synchronization and ranging: Asymmetrical time-stamping and passive listening," *IEEE Signal Process. Lett.*, vol. 20, no. 1, pp. 51–54, Jan. 2013.
[30] S. C. Park, M. Tummala, and J. C. McEachen, "Method and apparatus for hybrid time synchronization based on broadcast sequencing for wireless ad hoc networks," U.S. Patent 9 693 325, Jun. 27, 2017.
[31] R. Pagliari and A. Scaglione, "Scalable network synchronization with pulse-coupled oscillators," *IEEE Trans. Mobile Comput.*, vol. 10, no. 3, pp. 392–405, Mar. 2011.

Sabin Bhandari (S'17) received the bachelor's degree in electronics and communication engineering from Tribhuvan University, Nepal, in 2010, and the M.E. degree in computer engineering from Chosun University, Gwangju, South Korea, in 2016. He is currently pursuing the Ph.D. degree with the Department of Electrical and Computer Engineering, Western University, London, ON, Canada.

He was a Lecturer and Project Assistant with the Kantipur Engineering College, Nepal, from 2011 to 2014. His current research interests include Internet of Things (IoT), medium access control (MAC) protocol, IEEE 802.15.4, and IEEE 802.11.

Xianbin Wang (S'98–M'99–SM'06–F'17) received the Ph.D. degree in electrical and computer engineering from the National University of Singapore in 2001.

From 2001 to 2002, he was a System Designer at STMicroelectronics. He was with the Communications Research Centre Canada (CRC) as a Research Scientist/Senior Research Scientist from 2002 to 2007. He is a Professor and Tier 1 Canada Research Chair with Western University, Canada. He has over 350 peer-reviewed journal and conference papers, in addition to 29 granted and pending patents and several standard contributions. His current research interests include 5G technologies, Internet-of-Things, communications security, machine learning, and locationing technologies.

Dr. Wang is a Fellow of the Canadian Academy of Engineering and an IEEE Distinguished Lecturer. He has received many awards and recognitions, including Canada Research Chair, the CRC President's Excellence Award, the Canadian Federal Government Public Service Award, the Ontario Early Researcher Award, and six IEEE Best Paper Awards. He was involved in many IEEE conferences, including GLOBECOM, ICC, VTC, PIMRC, WCNC, and CWIT, in different roles such as symposium chair, tutorial instructor, track chair, session chair, and TPC co-chair. He was also an Associate Editor of the IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS from 2007 to 2011, and the IEEE WIRELESS COMMUNICATIONS LETTERS from 2011 to 2016. He currently serves as an Editor/Associate Editor for the IEEE TRANSACTIONS ON COMMUNICATIONS, the IEEE TRANSACTIONS ON BROADCASTING, and the IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY.