Zheng, T.; Gidlund, M.; Åkerberg, J., "WirArb: A New MAC Protocol for Time Critical Industrial Wireless Sensor Network Applications," in *IEEE Sensors Journal*, vol. 16, no. 7, pp. 2127-2139, April1, 2016.

https://doi.org/10.1109/JSEN.2015.2504948

WirArb: A New MAC Protocol for Time Critical Industrial Wireless Sensor Network Applications

Tao Zheng, *Member, IEEE*, Mikael Gidlund, *Member, IEEE*, and Johan Åkerberg, *Senior Member, IEEE*

***Abstract*— Wireless sensor networks are typically designed for condition monitoring applications and to conserve energy but not for time-critical applications with strict real-time constraints that can be found in the industrial automation and avionics domain. In this paper, we propose a novel medium access control (MAC) protocol defined as wireless arbitration (WirArb) which grants each user channel access based on their different priority levels. The proposed MAC protocol supports multiple users and each user is pre-assigned a specific arbitration frequency which decides the order of channel access. With this mechanism, we can ensure that the user with the highest priority will immediately gain channel access and we can guarantee a deterministic behavior. To evaluate the proposed MAC, we use a discrete-time Markov chain model to mathematically formulate the WirArb protocol. Our results show that the proposed protocol provides high performance to ensure deterministic real-time communication and bandwidth efficiency.**

***Index Terms*— Wireless arbitration, wireless sensor networks, deterministic, real-time, cross-layer.**

I. INTRODUCTION

WIRELESS technologies have become increasingly popular for emerging applications targeting the consumer market and the industrial automation domain [1]. Merging wireless communication and real-time systems is a non-trivial task, especially for industrial applications where the sensors and actuators are part of control loops, and predictable network performance in terms of message transfer delay and reliability is required [2]. This paper will address the problem of accessing the wireless channel and providing timeliness guarantees.

In wired networks, data packets can be efficiently scheduled using the Controller Area Network (CAN) bus, which has already been proven to be useful in industry. The medium access control (MAC) protocol in CAN is collision-free and uses a priority mechanism making it possible to schedule the bus if message characteristics (transmission times, jitter, etc.) are known, while also making it possible to compute the upper bound on message delay. The CAN protocol belongs to the family of dominance or binary countdown protocols [3].

For wireless networks, the Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) has mainly been used as a collision-free solution to compete for channel access. CSMA/CA is used as default media access mechanism in wireless local area networks (WLAN) based on the IEEE802.11 standard and in wireless sensor networks (WSNs) based on the IEEE802.15.4 standard. However, for time critical applications which require deterministic communication and have strict deadlines, it has been shown that the CSMA/CA protocol is not suitable due to unpredictable time delay being generated by random distribution of the backoff time [4]. CSMA/CA typically ensures minimum delay for low traffic loads, but as traffic load increases, the delay becomes unacceptable and throughput deteriorates. Since CSMA/CA is a random access scheme, it does not prioritize transmissions based on the physical processes monitored by the sensors and actuators [5], [6]. In addition to the above slotted contention-based CSMA/CA, a recent survey reviews and classifies asynchronous real-time MAC protocols in [7]. Compared with synchronous contention-based CSMA/CA, reduced real-time performance could be induced as it is necessary for the asynchronous random access mechanism to adopt certain strategies to overcome the decoupled situation between the transmitter and the receiver, which will increase transmission delay due to overhearing, over-emitting and even packet collisions. To improve the reliability of industrial wireless networks, current available standards such as WirelessHART [8], [9], ISA 100.11a [8], WIA-PA [10] and IEEE802.15.4e [11], uses a Time Division Multiple Access (TDMA) protocol combined with CSMA/CA to schedule the user channel access. One drawback with using TDMA in mission critical applications is that in the event of an emergency event with the highest priority, the transmission of the critical data packet needs to wait for its transmission time slot, which is unacceptable for applications with strict deadlines. Another drawback with TDMA-based WSNs for critical applications is that if a slot for emergency messages had to be reserved in every frame, the channel utilization will be reduced, especially if the refresh rate is high. This is one of the main reasons why TDMA-based WSNs are not a good choice for time-critical wireless sensor network applications. This is highlighted in the literature [4], although a series of real-time MAC protocols are designed to facilitate

Manuscript received July 10, 2015; revised September 30, 2015; accepted November 23, 2015. Date of publication December 2, 2015; date of current version February 10, 2016. This work was supported in part by the National Key Basic Research Program of China under Grant 2013CB329101 and in part by the Fundamental Research Funds for the Central Universities under Grant 2015JBM001. The associate editor coordinating the review of this paper and approving it for publication was Dr. Nitaigour P. Mahalik.

T. Zheng is with the School of Electronic and Information Engineering, Beijing Jiaotong University, Beijing 100044, China (e-mail: t.zheng@ieee.org).

M. Gidlund is with Mid Sweden University, Sundsvall 851 70, Sweden (e-mail: mikael.gidlund@miun.se).

J. Åkerberg is with ABB Corporate Research, Forskargränd 7, 721 78, Västerås, Sweden (e-mail: johan.akerberg@se.abb.com).

Digital Object Identifier 10.1109/JSEN.2015.2504948

the expected transmission latency for WSNs, they only manage to meet the real-time requirement by attempting to reduce the data processing time between transmitter and receiver rather than immediately reacting to different emergency events. Regardless of whether CSMA/CA or TDMA induces higher latency by avoiding additional channel conflict and contention, neither is suitable for the real-time control system where different triggered events are limited within different hard time-bounds.

In this paper, we address the problem of wireless channel access by proposing *WirArb* which has similar functions as the CAN bus. The proposal combines the physical layer and MAC layer to guarantee real-time performance. Before the period of data transmission, channel access arbitration will be performed first. Each user in the network is pre-assigned a dedicated arbitration frequency with the purpose to identify channel access priorities which is utilized to deterministically determine the channel access order. This procedure will ensure that in each arbitration cycle the most critical event can be the only one to obtain the highest priority, and immediately gain access to the wireless medium. Through the proposed scheme we can guarantee real-time performance, where all messages will meet their individual deadlines, for time critical industrial wireless applications. We provide a stochastic theoretical model that focuses on the performance evaluation of the *WirArb* scheme by employing a discrete time Markov chain (DTMC) analysis. As formulated analytical expressions for the theoretical analysis, it can be used to calculate the performance in terms of system throughput and communication delay. Furthermore, we compare the channel utilization between *WirArb* and TDMA-based WSNs such as WirelessHART and GinMAC for time-critical applications (for example safety) that require high refresh rate. The obtained results show that the proposed solution outperforms TDMA-based WSNs in terms of latency, throughput and channel utilization.

The remainder of the paper is organized as follows. Section II discusses related work and in Section III we describe the medium access mechanism for the *WirArb* in detail. In Section IV, we model *WirArb* into a discrete time Markov chain. In Section V, we validate system models from worst-case scenarios, which is followed by an analysis of performance evaluations with regard to throughput, delay and channel utilization. Finally, the results of the research are summarized and we propose future work on *WirArb*.

II. Related Work

In order to serve the stringent real-time requirements for time-critical event-based wireless applications, and overcome mentioned drawbacks on nondeterministic time delay as well as channel utilization, several MAC protocols supporting real-time have previously been presented [12].

Arbitrary contention is the main reason for the transmission delay caused by random backoff and retransmissions. Thus, CSMA/CA-based technologies are used to reduce the harmful impacts from unexpected data collision. In [13], Ye *et al.* presented the SMAC protocol for wireless sensor networks where the sensor nodes utilize the benefits of collision avoidance of the RTS/CTS handshake mechanism. However, because of the scheduling of the fixed duty cycle, this will cause an unacceptable delay which makes this MAC protocol unsuitable for time-critical applications, especially for applications in industrial automation. There are similar protocols based on SMAC in the literature, such as SMAC-AL [14] and DSMAC [15], which replace an adaptive duty cycle to reduce the forwarding delay. However, this class of protocols involves end-to-end delay due to the sleep period to avoid collision with others. In order to reduce the delay from unnecessary sleep, a wake-up scheduling method such as LEEMAC [16], DMAC [17], DW-MAC [18] or SPEED-MAC [19] is developed to decrease sleeping delay. Although the aforementioned protocols can reduce the delivery delay by scheduling the sleep slot, they are not designed for deterministic delay guarantee due to random backoff. All these MAC protocols (including but not limited to [13]–[19]) involves the CSMA/CA mechanism being more valuable to protect ordinary wireless sensor networks from packet collisions rather than time-critical Wireless Sensor and Actuator Networks (WSAN) with hard requirements for deterministic delay guarantees. HyMAC [20] is a class of MAC protocols merging TDMA and FDMA together. Although HyMAC guarantees a certain end-to-end delay, the main drawback is the lack of ability to adapt to the harsh wireless channel conditions of industrial automation or avionics. In [21], Suriyachai *et al.* presented the GinMAC protocol which they claim can give support for real-time guarantees for time-critical applications in industrial wireless sensor networks. However, their approach to TDMA scheduling will result in exclusive time slot usage which may prevent an emergency task from sending data immediately because it needs to wait for transmission slots.

III. Wireless Arbitration Medium Access Protocol

A. System Model

The centralized control model is commonly used in industrial applications. Peripheral networked devices form a complete system where different performance levels are required according to the industrial process needs. Therefore, the star network is firstly considered in this paper since it is a typical centralized network topology. It is shown in Fig. 1 which consists of the gateway as the central controller and several network users which include different kinds of industrial devices, such as sensor nodes and actuators.

The *WirArb* medium access protocol uses an *arbitration cycle* (also called the arbitration phase). An arbitration phase consists of two parts, which are arbitration decision period and arbitration execution period. The arbitration decision period is the first step in an arbitration phase. This is to process channel access requests and determine a deterministic channel access order. Then, the arbitration execution period occurs when the actual data is transmitted. Specifically, depending on its priority, the user with the highest priority can immediately access a channel to transfer data, while the remaining users with lower priority cannot access the channel. The percent that an arbitration phase spends in the decision period is a

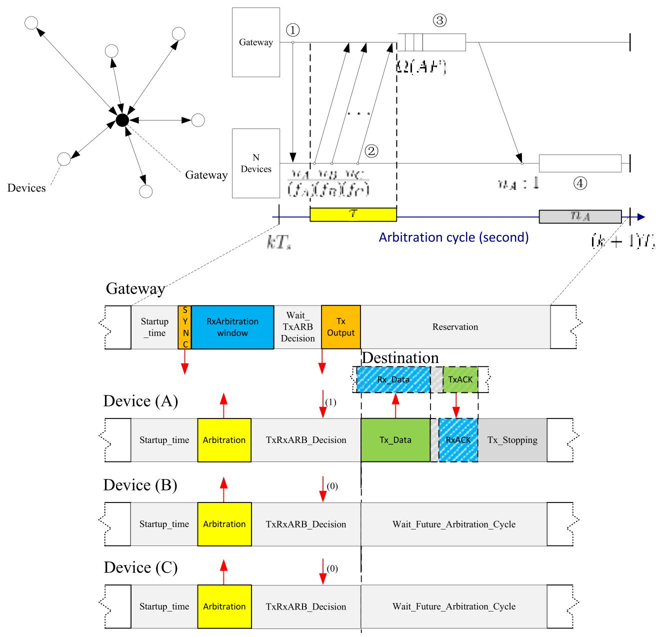


Fig. 1. The network considered for the *WirArb* and the media access control mechanism.

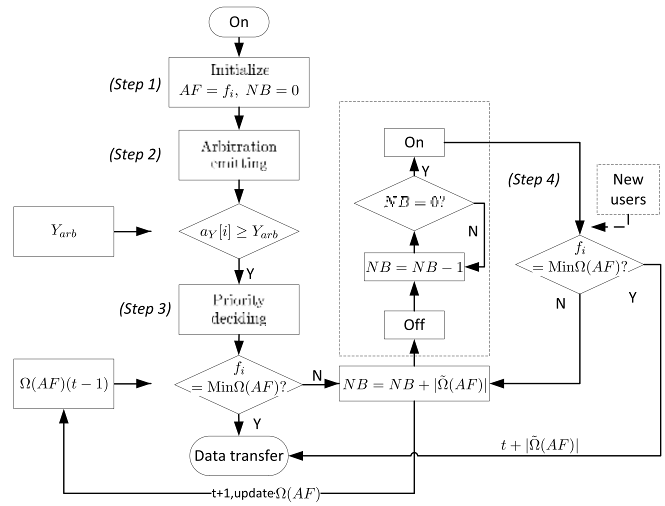


Fig. 2. The flow diagram for the *WirArb*.

fraction of the total arbitration cycle, and should be small enough to ensure that the device with the highest priority will have enough time to complete its data transmission. This can be achieved because it is possible for a device to use a signal short enough to deliver the arbitration request to the gateway. We define a repeating period T_s as the fixed arbitration interval.

All the network users need to send their channel access request signals to the gateway before data transmission. Therefore, the users should support the following functions: synchronization with the gateway; transmission of channel access request signals before attempting to access the channel; transmission of data packets. The gateway handles two key functions including network synchronization and channel access request arbitration. In order to let the gateway identify different channel access request signals, some new research works on PHY layer should be carried out for both the gateway's receiver and the network user's transmitter. In this paper, we let the *Arbitration Frequency* (AF) to represent event priority, which is designed with equivalence to the subcarrier frequency. Each user $n_i, i \in \mathcal{N} = \{1, 2, \ldots, N\}$ is only pre-assigned one subcarrier frequency f_i, which implies that the designated arbitration frequencies should remain orthogonal with each other, i.e., $\{f_i \cap f_j = \varnothing, i, j \in \mathcal{N}$ and $i \neq j\}$. Regarding the gateway, because several channel access request signals modulated over different subcarriers arrive at its receiver randomly, the gateway should have the capability to continuously keep sensing the arbitration signals from all subcarriers in a random manner. Therefore, a new receiver model is introduced briefly since PHY layer modeling is not the focus of this paper. The new receiver of the gateway should consist of a plurality of modules which includes a bandpass filter, a sampler, a Fourier transformer, as well as a comparison and decision block. Specifically, the bandpass filter is used to allow channel access request signals within a selected range of arbitration frequencies to be sensed and decoded, while preventing interfering signals at unwanted frequencies from getting through. After spectral sampling of the filtered signals, these channel access request signals captured by the receiver are still mixed. Therefore, the Fast Fourier Transform (FFT) is used to extract the signals' information of different frequencies. After FFT processing, the peak magnitude values over different frequencies are given. In order to eliminate interferences and noises from the harsh industrial environment, an arbitration decision threshold Y_{arb} is predefined to compare with the amplitudes of the FFT signals over different arbitration frequencies with purposed to accurately identify the users. Finally, according to the pre-assigned subcarrier frequencies, a channel access order can be arbitrated.

A series of interactive steps are executed in each arbitration cycle (see Fig. 1 for 1,…,4). For easier implementation, *WirArb* defines a beacon-enabled channel access mode where periods can be announced in the beacon issued by the gateway.

B. Arbitration Decision Process

One of the core parts of *WirArb* is the arbitration decision process in Fig. 2. Four variables are introduced in the arbitration process: (1) AF; (2) $\Omega(AF)$; (3) NB; (4) S. AF represents the orthogonal arbitration frequency which is pre-assigned to a dedicated user for transmitting its data; $\Omega(AF)$ represents the subcarrier frequency set which is dynamically updated in the gateway. This set is used to save arbitration frequencies identified from the comparison block by the gateway in each arbitration cycle. After the final arbitration decision, this set will be represented as a queue, where all the remaining users with lower priorities are queuing in the ascending order of arbitration frequencies according to their priorities; NB represents the number of delayed arbitration cycles before the user tried to access the channel for its data transmission, which is defined as the waiting stage; S represents the On-Off state of the user in each wireless arbitration cycle.

In the initialization stage when random users join a network, AF, NB and S respectively are defaulted to f_i, 0 and 0, where $i \in \{1, 2, \ldots, N\}$ represents the numeral ID of the pre-assigned arbitration frequency. $\Omega(AF)$ stored in the gateway that is used to represent a queue including all users in the previous arbitration phase. The boundary of the next period of the arbitration phase is assigned in this initialization stage (marked as step 1). After initialization, users begin to send their channel access request signals (also called the arbitration signals) to the gateway. Before arbitration signals receiving time expires, the gateway continually scans and receives incoming arbitration signals. If the received signal spectrum amplitude $a_Y[i]$ over the arbitration frequency f_i is not smaller than the predefined arbitration threshold Y_{arb} stored in the gateway, the user's activation status for data transmission can be detected by the gateway in step 2. Step 3 creates a strategic decision based on the priority which determines how the wireless channels should be scheduled without contention. In each arbitration cycle, if an activated user n_i intends to join the network and prepares to transmit its data, the MAC layer of the gateway will increment the frequency set $\Omega(AF)$ with a new arbitration frequency f_i, which corresponds to this activated user, and the cardinality of frequency set $\Omega(AF)$ is increased by 1. We define $\text{Min}\Omega(AF)$ as the minimum element of the frequency set $\Omega(AF)$ which could equal to f_i. If the arbitration frequency f_i is the minimum of the frequency set $\Omega(AF)$, data transmission will start in the following time period. This is because the wireless channel must be allocated to the user with the highest priority only; otherwise, NB will be incremented by an integer $|\tilde{\Omega}(AF)|$, where we define the cardinality of the subset $\tilde{\Omega}(AF)$ as $|\tilde{\Omega}(AF)|$, with the purpose of ensuring that the user carrying the most time-critical event will access a channel immediately and complete its data transmission within the required hard time-bound. As a subset of the frequency set $\Omega(AF)$, $\tilde{\Omega}(AF)$ means that all included members are smaller than arbitration frequency f_i. As a result, a user with lower priority will have to wait at least $|\tilde{\Omega}(AF)|$ waiting stages until NB is decreased to 0 (marked as step 4). If there is no other user with higher priority to access the channel in the next arbitration phase, user n_i can access the channel immediately; otherwise, step 4 will be repeated. During the periods of waiting, users with lower priority should remain in off state to save energy consumption.

As an example, we consider a network having three users n_A, n_B, n_C which are granted arbitration frequency f_A, f_B, f_C, respectively. For simplicity, arbitration frequencies are in ascending order. In the current arbitration phase $[kT_s, (k+1)T_s]$, after time-synchronization, the gateway captures three arbitration signals $x_A(t)$, $x_B(t)$, $x_C(t)$ which are respectively modulated on the sub-carrier frequency f_A, f_B, f_C; after arbitration comparison and decision, output 1 is granted to user n_A, while the remaining two users get output 0. The final channel access is thereby given to user n_A first, since two necessary conditions are satisfied in that (1) the spectrum amplitude of the received arbitration signal from user n_A is higher than the predefined *arbitration level* as well as (2) arbitration frequency f_A is the lowest. Once user n_A completes data transmission in the current arbitration phase, the channel resource will be immediately released. Following this, the next arbitration phase $[(k+1)T_s, (k+2)T_s]$ will start. If no new active user with lower arbitration frequency than that of user n_B applies for the channel, or if there are no other users in queue before user n_B, the second channel access will be assigned to user n_B because the arbitration frequency of user n_B is the lowest of all the current participants. Next, the channel control passes from n_B to n_C, and then to the next active user with the highest arbitration frequency.

IV. Mathematical Formulation

In order to evaluate the performance of *WirArb* we model the stochastic behavior of each device as a discrete time Markov chain (DTMC). We assume that the probability to start sensing the wireless channel is constant and independent across all users.

A. Stochastic Markov Decision Process

Consider that a sequence of random state variables X_t with the time epoch t is assumed to integrate a discrete time state space $\mathcal{S} = \{X_t, t = 0, 1, 2, \ldots\}$, the Markov property should be satisfied as follows

$$\begin{aligned} P(X_t = x_t | X_0 = x_0, X_1 = x_1, \ldots, X_{t-1} = x_{t-1}) \\ = P(X_t = x_t | X_{t-1} = x_{t-1}) \end{aligned} \quad (1)$$

where $x_k \in \mathcal{S}$ for all $k = \{0, 1, 2, \ldots, t\}$ are realized states of the stochastic process. This condition can be described in words as the future state only depends on the present state, and is independent on the full history state in the discrete time stochastic process $\{X_t, t = 0, 1, 2, \ldots\}$. Here, we assume that $X_{t-1}, t \geq 1$ is denoted as the current state, the future state following time $t-1$ is listed in set $\{X_t, X_{t+1}, \ldots\}$, while the past state is collected in set $\{X_0, \ldots, X_{t-3}, X_{t-2}\}$. If the value of the current state x_{t-1} is known, then the evolution value of the future state x_t only relies upon x_{t-1}, which indicates that it is stochastically independent of the known value of the past state $\{x_0, \ldots, x_{t-3}, x_{t-2}\}$.

The initialization state of any user intending to join the network can be constructed by using a two-state Markov decision process which is of the form $\{s_{t_1}, s_{t_2}\}$. Power control is assumed to be used in order to minimize energy consumption. If the wireless channel is not idle in the present time slot, users need to remain in sleep mode by temporarily preventing transmission before the next round of attempting to access the wireless channel. Hence, in each initialization period, the user is in either off state or on state which represents the inactive (or sleep) and active (or awake) state, respectively. At each decision epoch (instant times), the off state is identified by default as the last historical state of the on state. For ease of description, the off and on states are marked as -1 and 1 respectively. $s_{t_1} \in \{-1\}$ represents the inactive state at time t_1 and $s_{t_2} \in \{1\}$ represents the active state at time t_2.

Fig. 3 shows the symbolic representation of the two-state Markov decision process and possible transitions for any newly activated user in each decision epoch. In state s_{t_2}, the decision maker selects an action a_{t_2,t_1} provides the user with a state change after an event trigger, which refers to the user

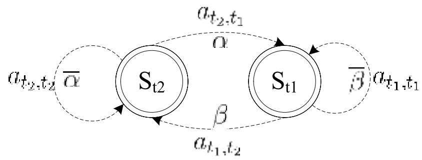


Fig. 3. Symbolic representation of the two-state Markov decision process.

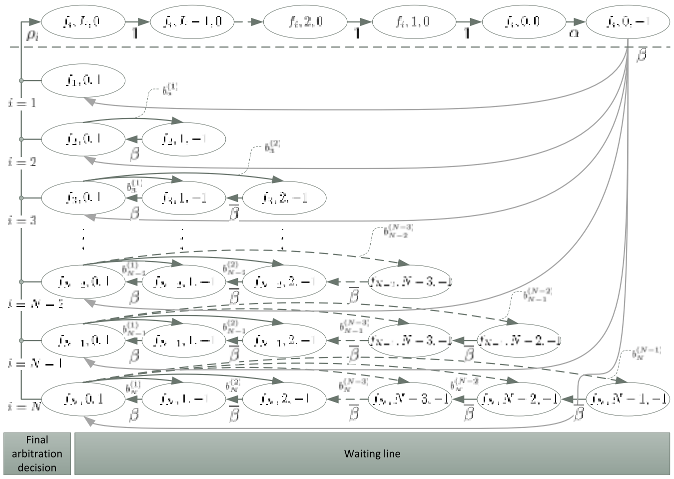


Fig. 4. Discrete time Markov chain model for the *WirArb*.

being in state s_{t_1} with probability α at the next time slot; if instead an action a_{t_2,t_2} is chosen in state s_{t_2}, the user maintains this original state with probability $\bar{\alpha}$ at the next time slot. In state s_{t_1}, the decision maker can likewise choose either action a_{t_1,t_2} or action a_{t_1,t_1}. Choosing action a_{t_1,t_2} in state s_{t_1} makes the user move to state s_{t_2} with probability β at the next time slot; state s_{t_1} remains unchanged with probability $\bar{\beta}$, if action a_{t_1,t_1} is chosen at the next time slot.

B. Modeling Discrete Time Markov Chain

We assume a network of a fixed number N of users and each user always has a packet available for transmission. In this way we can model the behavior of users using the DTMC shown in Fig. 4. The DTMC describes the state transition diagram of different users, where $\{f_i(t), w_i(t), s_i(t)\}$ is the three-dimensional state. We use a discrete and integer time scale where t and $t+1$ correspond to the starting point of two consecutive time slots. Let $f_i(t)$ be the arbitration frequency assigned to the user n_i at time t and $w_i(t)$ is the stochastic process representing the waiting stage of the user n_i at time t. If m is defined as the maximum waiting stage, regarding the user owning arbitration frequency f_i, it is reasonable to estimate that the value of the waiting stage should be less than or equal to the result of sequence number i minus 1. Because the user with arbitration frequency f_{i+1} will not obtain a higher priority from the gateway to access the channel than the user with arbitration frequency f_i. If user n_i and user n_{i+1} both have a data packet to transfer in the same arbitration phase, user n_i will send the data first due to its higher priority. Therefore, $w_i(t)$ falls into the set $[0, 1, 2, \ldots, i-1], i = 0, 1, 2, \ldots, N$. Let $s_i(t)$ be the stochastic process representing the active-inactive state of user n_i at time t. Next, as a result of any user changes to the state of the data transmission, $l \in [1, L]$ represents data transmission phase, where L is the maximum packet transmission duration measured in the integer number of time slots that are not allowed to be interrupted. The idle state $\{f_i, 0, -1\}$ stands for the inactive period when the user n_i has no data to transfer.

The stationary distribution of the DTMC can be described as

$$b\{f_i, w, s\} = \lim_{t\to\infty} P\{f_i(t) = i, w_i(t) = w, s_i(t) = s\},$$
$$i \in [1, N], \quad w \in [0, i-1], \; s \in \{-1, 0\} \qquad (2)$$

Given the DTMC in Fig. 4 the one-step transition probabilities can be written as

$$\begin{aligned} b\{f_i, 0, 1 | f_i, 0, -1\} &= \beta, \quad i \in [1, N] \\ b\{f_i, i-2, -1 | f_i, i-1, -1\} &= \bar{\beta}, \quad i \in [3, N] \\ b\{f_i, 0, 1 | f_i, 1, -1\} &= \beta, \quad i \in [2, N] \\ b\{f_i, L, 0 | f_i, 0, 1\} &= \rho_i, \end{aligned} \qquad (3)$$

The first equation in (3) describes the probability of starting a new data transmission request at the beginning of each arbitration phase. It represents each arbitration user that transfers from inactive state to active state. The second equation describes a situation where the user with a smaller arbitration frequency needs to wait for a number of cycles of arbitration phase to allow the high-priority users to transfer data first, meanwhile the corresponding waiting stage is reduced one by one. The third equation states the probability of transiting from the last waiting stage to the next arbitration phase since the counter for the waiting stage is cleared. The last equation implies that since the data transmission request from user n_i has been detected in the earlier arbitration phase, the gateway pre-assigned a priority to this user; if no higher-priority users are detected by the gateway in the current arbitration phase, user n_i obtains the highest priority; we define the following data transmission period will be used by user n_i with probability ρ_i.

We define the DTMC's m-step transition probability as the probability of transiting from state $\{f_i, 0, 1\}$ to state $\{f_i, m, -1\}$ $(i \in [2, N], m \in [1, i-1])$ in m steps as

$$b_i^{(m)} = P\{f_i, m, -1 | f_i, 0, 1\}, \quad i \in [2, N], \; m \in [1, i-1] \qquad (4)$$

then, the state transition probability matrix for the DTMC can be expressed as

$$\begin{pmatrix} b_2^{(1)} & 0 & 0 & \cdots & 0 & 0 \\ b_3^{(1)} & b_3^{(2)} & 0 & \cdots & 0 & 0 \\ b_4^{(1)} & b_4^{(2)} & b_4^{(3)} & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ b_{N-2}^{(1)} & b_{N-2}^{(2)} & b_{N-2}^{(3)} & \cdots & 0 & 0 \\ b_{N-1}^{(1)} & b_{N-1}^{(2)} & b_{N-1}^{(3)} & \cdots & b_{N-1}^{(N-2)} & 0 \\ b_N^{(1)} & b_N^{(2)} & b_N^{(3)} & \cdots & b_N^{(N)} & b_N^{(N-1)} \end{pmatrix}$$

$$\begin{cases} b\{f_i,0,1\} = \beta b\{f_i,0,-1\} + \bar{\alpha} b\{f_i,0,1\} & i=1 \\ \left.\begin{array}{l} b\{f_i,0,1\} = \beta b\{f_i,1,-1\} + \beta b\{f_i,0,-1\} + \bar{\alpha} b\{f_i,0,1\} \\ b\{f_i,1,-1\} = b_i^{(1)} b\{f_i,0,1\} + \bar{\beta} b\{f_i,2,-1\} + \bar{\beta} b\{f_i,1,-1\} \\ b\{f_i,2,-1\} = b_i^{(2)} b\{f_i,0,1\} + \bar{\beta} b\{f_i,3,-1\} + \bar{\beta} b\{f_i,2,-1\} \\ \qquad \vdots \qquad\qquad \vdots \qquad\qquad \vdots \\ b\{f_i,i-2,-1\} = b_i^{(i-2)} b\{f_i,0,1\} + \bar{\beta} b\{f_i,i-1,-1\} + \bar{\beta} b\{f_i,i-2,-1\} \\ b\{f_i,i-1,-1\} = b_i^{(i-1)} b\{f_i,0,1\} + \qquad + \bar{\beta} b\{f_i,i-1,-1\} \end{array}\right\} & 2 \le i \le N \end{cases} \tag{5}$$

From the DTMC, we can note that the matrix is a $(N-1)$-step lower triangular matrix.

C. Transition Probabilities

For any user with arbitration frequency f_i, we can create a set of steady-state linear functions as in Equation (5), as shown at the top of this page.

In general, owing to the chain regularities, we can determine the steady-state probability by

$$b\{f_i,1,-1\} = b\{f_i,0,1\} \sum_{j=2}^{i} \frac{1}{\beta^{j-1}} (1-\beta)^{j-2} b_i^{(j-1)} \tag{6}$$

For condition $i = 1$, we obtain

$$b\{f_i,0,-1\} = \frac{\alpha}{\beta} b\{f_i,0,1\}, \quad i=1 \tag{7}$$

According to the second relationship in (5), we obtain the steady-state probability for condition $2 \le i \le N$ which can be determined by

$$b\{f_i,0,-1\} = \frac{1}{\beta}\left[\alpha - \sum_{j=2}^{i}\left(\frac{1-\beta}{\beta}\right)^{j-2} b_i^{(j-1)}\right] b\{f_i,0,1\} \tag{8}$$

From the given relations, all the values $b\{f_i,0,-1\}, i \in N$ are expressed as functions of the value $b\{f_i,0,1\}$ and of the conditional transition probabilities between on and off state (α and β). For the case of $i \ge 2$, the m-step transition probability $b_i^m, (1 \le m \le N-1)$ should be considered to be a condition to determine the value $b\{f_i,0,-1\}, 2 \le i \le N$. By using normalization the steady-state probability $b\{f_i,0,1\}$ can be simplified and it is given by Equation (9), as shown at the bottom of this page.

The steady-state probability in queue shown in Fig. 4 can be calculated by

$$b\{f_i,w,-1\} = b\{f_i,0,1\} \sum_{j=1}^{i-w} \frac{1}{\beta^j} (1-\beta)^{j-1} b_i^{(w+j-1)} \tag{10}$$

where $w \in [1, i-1]$ is equivalent to the count of the waiting stage.

Regarding the timeline for data transmission, the following equations for any user $i \in [1, N]$ are satisfied.

$$\alpha b\{f_i,l-1,0\} = b\{f_i,l,0\}, \quad l=1 \tag{11}$$

$$b\{f_i,l,0\} = b\{f_i,l-1,0\}, \quad l \in [2,L] \tag{12}$$

$$\alpha b\{f_i,l-1,0\} = \beta b\{f_i,0,-1\}, \quad l=1 \tag{13}$$

$$b\{f_i,L,0\} = \rho_i b\{f_i,0,1\} \tag{14}$$

According to (7), (9), (11)–(14), for $i = 1$, we obtain

$$b\{f_1,0,1\} = \frac{1}{1+\left(L+\frac{1}{\alpha}\right)\rho_1 + \frac{\alpha}{\beta}} \tag{15}$$

Furthermore, according to (8), (9), (11)–(14), for $2 \le i \le N$, we obtain (16), as shown at the bottom of this page.

Then, substituting the expression in Equation (15) and (16) leads to a new expression for the total probability

$$b\{f_i,0,1\} = \begin{cases} \dfrac{\alpha\beta}{\alpha\beta + (\alpha\beta L + \beta)\rho_1 + \alpha^2}, & i=1 \\ \dfrac{\alpha\beta}{\alpha\beta + (\alpha\beta L + \beta)\rho_2 + \alpha^2}, & i=2 \\ \dfrac{\alpha\beta}{\alpha\beta + (\alpha\beta L + \beta)\rho_i + \alpha^2 + \alpha\phi_i}, & 3 \le i \le N \end{cases} \tag{17}$$

$$1 = \begin{cases} b\{f_i,0,1\} + \sum_{l=0}^{L} b\{f_i,l,0\} + b\{f_i,0,-1\}, & i=1 \\ b\{f_i,0,1\} + \sum_{w=1}^{i-1} b\{f_i,w,-1\} + \sum_{l=0}^{L} b\{f_i,l,0\} + b\{f_i,0,-1\}, & 2 \le i \le N \end{cases} \tag{9}$$

$$b\{f_i,0,1\} = \frac{1}{1+\left(L+\frac{1}{\alpha}\right)\rho_i + \sum_{w=1}^{i-1}\sum_{j=1}^{i-w}\frac{1}{\beta^j}(1-\beta)^{j-1} b_i^{(w+j-1)} + \frac{1}{\beta}\left[\alpha - \sum_{j=2}^{i}\left(\frac{1-\beta}{\beta}\right)^{j-2} b_i^{(j-1)}\right]} \tag{16}$$

where

$$\phi_i = \sum_{w=2}^{i-1} \left(\frac{1-\beta}{\beta}\right)^{-w} \sum_{j=w}^{i-1} \left(\frac{1-\beta}{\beta}\right)^{j} b_i^{(j)} \tag{18}$$

which can be simplified to

$$\phi_i = \sum_{m=2}^{i-1} b_i^{(m)} \sum_{k=0}^{m-2} \left(\frac{1-\beta}{\beta}\right)^{k} \tag{19}$$

In addition, for $i = 1$, we note $\rho_i = \alpha$, and for $2 \leq i \leq N$, we note $\rho_i = \alpha - \sum_{j=1}^{i-1} b_i^{(j)}$.

A new data transmission will occur when all users with higher priority have completed their tasks. When the waiting stage counter is equal to zero (illustrated in Fig. 2), we can measure the probability of the user beginning to access the channel in a random arbitration phase by

$$\tau_i = \rho_i b\{f_i, 0, 1\} = b\{f_i, L, 0\} \tag{20}$$

Considering the best-case scenario where there is only one active communication link between a unique user and the gateway, the probability of allocating the following data transmission slot time to this user should be constant $\rho_1 = 1$. Hence, the DTMC can be simplified as a two-state Markov chain $b\{f_1, 0, 1\} \leftrightarrows b\{f_1, 0, -1\}$.

Regarding the worst-case scenario, τ_i depends on the m-step state transition probability b_i^m which is derived from both the probability ρ_i and the steady-state probability $b\{f_i, 0, 1\}$ which is non-independent of the state transition probability in queue. According to the above description of the media access mechanism, the m-step state transition probability b_i^m is related with the probability p^* which implies a new transmission attempt occasionally encounters a collision with the current user's channel access, which is due to the presence of higher-priority users in the current arbitration phase. m active users are considered waiting for data transmission in their independent steady state, this probability p^* can be defined as $p^* = \sum \left(\binom{i-1}{m} \prod \tau_i\right)$. We know that the total time delay is mainly caused by the waiting stage which is formed by several of the inactive states $\{f_i, w, -1\}$.

The busy mode of the wireless channel for data transmission can be modeled to be a series of state transfer processes which consist of L steady-state probability which means that a user transmits a data packet with constant length L. $b\{f_i, l, 0\}$ is defined as the probability of an event whereas the user n_i is transmitting in the lth time slot, while $b\{f_i, 0, 1\}$ is the probability of the case that certain events require continuous channel occupancy in the following arbitration phase. We have

$$\tau = \sum_{l=0}^{L} b\{f_i, l, 0\} = \left(L + \frac{1}{\alpha}\right) \rho_i b\{f_i, 0, 1\} \tag{21}$$

In order to evaluate mutual effects between users, when they access a channel in a random manner, we provide the expressions for the remaining unknown m-step transition probabilities $b_i^{(m)}$. First, we define two new events, one is denoted by $S(w) = -1$ as the event when the user is queued for a future arbitration phase; another event is denoted by $S^j(n) = 0, j \geq 2$ as the event when the user with arbitration frequency f_j is preparing to access the medium as the waiting stage counter is equal to zero. According to the analysis of the DTMC, $S^j(w) = -1$ is equivalent to state $\{f_j, w, -1\}$ while $S^j(n) = 0$ is equivalent to state $\{f_j, 0, 1\}$. The variable $S(w) = -1$ can be mapped as an event-based action, which implies that in this decision-making process the expected reward given to the user n_j can be expressed as time delay (here, the reward works as a kind of penalty to the system performance). The maximum waiting stage under this action $S(w) = -1$ should be $m = |\widetilde{\Omega}(AF)|$, where m is the maximum transition steps from state $\{f_j, 0, 1\}$ to state $\{f_j, w, -1\}$. Then, the probability θ_j is defined as the conditional probability that an active user n_j meets a random non-empty waiting queue set $\widetilde{\Omega}(AF)$ because there are users with lower arbitration frequencies that can be defined as $P(S^j(w) = -1 | S^j(n) = 0 \cap S(w) = -1)$. Given the channel state transition probabilities, we can calculate θ_j as

$$\begin{aligned}\theta_j &= P(S^j(w) = -1 | S^j(n) = 0 \cap S(w) = -1) \\ &= P(S^j(w) = -1 | (S^j(n) = 0 \cap (\cup_{k=1}^{j-1} S^k(w) = -1)))\end{aligned} \tag{22}$$

where the superscript k represents the arbitration frequency number. Accordance with provisions of the DTMC, and only when the conditions that user n_k with a lower arbitration frequency than f_k is still queuing in the waiting stage at current time t while user n_j with arbitration frequency f_j is attempting to access the channel are both satisfied, a state transition process for user n_j from $S^j(n) = 0$ to $S^j(w) = -1$ will happen the next time $t+1$. At this moment, $f_k \in \widetilde{\Omega}(AF)$, $1 \leq k \leq j-1$ is notable. With regard to the transition steps m mentioned in the state transition process, it only depends on the number of users who meet the above mentioned conditions, and it implies that the maximum transition step will not exceed this limit $|\widetilde{\Omega}(AF)|$, where $m \leq |\widetilde{\Omega}(AF)|$ is true.

Since a key assumption in this paper is that the probability of sensing the channel is independent across all users, taking a 1-step transition into consideration, we have

$$\theta_j^{(1)} = P\left(S^j(n) = 0\right) \times \left(\sum_{k=1}^{j-1} P\left(S^k(w) = -1\right)\right) \tag{23}$$

Regarding the 2-step transition, we have

$$\begin{aligned}\theta_j^{(2)} = P&\left(S^j(n) = 0\right) \\ &\times \{P\left(S^1(w) = -1\right) \times P\left(S^2(w) = -1\right) \\ &+ P\left(S^1(w) = -1\right) \times P\left(S^3(w) = -1\right) \\ &+ P\left(S^2(w) = -1\right) \times P\left(S^3(w) = -1\right) \\ &+ P\left(S^2(w) = -1\right) \times P\left(S^4(w) = -1\right) \\ &+ \cdots P\left(S^{j-2}(w) = -1\right) \times P\left(S^{j-1}(w) = -1\right)\}\end{aligned} \tag{24}$$

which implies that in the current time when user n_j is attempting to access the channel, there are 2 random users with lower arbitration frequency in queue. As a result, the user n_j has to be subsequently transited for two waiting stages. To express this transmission pattern more intuitively, we define a new random matrix as $\mathbf{C}_{j-1}^2 = [\zeta_{j-1}^2]_{\binom{j-1}{2}\times 2}$ which means that $|j-1|$ participants that have lower arbitration frequencies than user n_j contains 2 randomly activated users who are queuing in the current arbitration phase. In (24), $\mathbf{C}_{j-1}^2$ can be expressed by $[(f_1, f_2), (f_1, f_3), \ldots, (f_2, f_3), (f_2, f_4), \ldots, (f_{j-2}, f_{j-1})]^T$. Through mathematical induction, the general form of condition probability θ_j in terms of m-step transition can be rewritten as

$$\theta_j^{(m)} = P\left(S^j(n) = 0\right) \times \left\{ \sum_{q=1}^{\binom{j-1}{m}} \left(\prod_{f_k \in r(q)} P\left(S^k(w) = -1\right)\right)\right\} \quad (25)$$

where $r(q)$ represents the row elements of the matrix $\mathbf{C}_{j-1}^m = [\zeta_{j-1}^m]$, q is the row number. $\binom{j-1}{m}$ returns a binomial coefficient containing all possible combinations of $j-1$ items taken m at a time. Matrix $\binom{j-1}{m}$ has m columns and $(j-1)!/((j-1-m)!m!)$ rows. When there are random m users with lower arbitration frequencies in queue while user n_j attempts to access the channel in the same arbitration phase. The formula variables $P(S^k(w) = -1)$ are defined as the existence probability of the random users with lower arbitration frequency f_k than f_j in the current arbitration phase.

For $j = 2$, the existence probability is

$$P(S^k(w) = -1) = b\{f_1, 0, 1\} \quad (26)$$

For $3 \leq j \leq N$, the existence probability can be derived from (5) and is given by

$$P(S^k(w) = -1) = b\{f_k, 0, 1\} + \sum_{w=1}^{k-1} (b\{f_k, w, -1\}) \quad (27)$$

we get

$$\begin{aligned} &P(S^k(w) = -1) \\ &\quad = b\{f_k, 0, 1\} + \sum_{w=1}^{k-1} b_k^{(w)} (b\{f_k, 0, 1\}) \\ &\qquad + \bar{\beta} \left\{ b\{f_k, 0, 1\} \cdot \sum_{j=2}^{k} \frac{1}{\beta^{j-1}} (1-\beta)^{j-2} b_k^{(j-1)} \right\} \end{aligned} \quad (28)$$

then,

$$P(S^k(w) = -1) = b\{f_k, 0, 1\} \left\{ 1 + \sum_{w=1}^{k-1} \left(1 + \left(\frac{1-\beta}{\beta} \right)^w \right) b_k^{(w)} \right\}. \quad (29)$$

V. Performance Analysis

To validate the performance of the proposed *WirArb* medium access protocol we consider a star network used for an application in the industrial automation domain which in general has hard real-time requirements [22]. In this article we assume that if the star network approaches complexity with more than one user, τ_i is reduced to less than 1 due to positive transition probability, $0 \leq P_{tr} < 1 \cup 0 < P_s \leq 1$ is notable. As a consequence, the MAC throughput will decrease due to signal collisions with data packets and/or channel errors caused by interference [23]. To validate the performance of the proposed *WirArb* protocol we use the values and parameters that can be found in the specifications of the MAC sub-layer for IEEE802.15.4. The maximum expected data transmission rate is equal to $250kbps$ under the QPSK physical layer. The data format, such as the PHY header and MAC header, is defined by the standard specifications. Specifically, the MAC address issue is not considered in this analysis, and due to the limitation of the MAC protocol data unit length, the maximum length appropriated by the MAC payload is $960bits$. The value of the parameters used to obtain the analysis results are listed in Table I.

TABLE I
System Parameter Used to Obtain Numerical Results

Date transmission rate	250 $kbps$
Packet payload	960 $bits$
Arbitration phase	10 ms
Per time slot	320 μs
Non-transmission time	320 μs
Startup time	352 μs
Turn around time	192 μs
Transmit arbitration signal	352 μs
Set output	640 μs
PHY header	192 μs
MAC header	224 μs
Payload transmission time	3.84 ms

A. System Delay Analysis

The flow chart for *WirArb* shown in Fig. 2 shows that the total duration of a data transmission should be considered for the following two cases: *need no waiting* and *need waiting*. The first case is based on a situation where the highest priority user (the so-called master) has the lowest arbitration frequency and can access the channel directly in the present arbitration phase as there are no other users queueing for the channel. The second case to consider is when users with lower priority are forced to let higher priority users with lower arbitration frequency access the channel first, which results in low priority users having to queue for random waiting stages. Therefore, the total transmission time delay of the highest priority user consists of three main parts, i.e., 1) time for arbitration signal transmission T_{arb}; 2) time for final arbitration decision receiving $T_{TxRxarb}$; 3) time for data transmission T_{data}. For other low priority users, the total transmission time delay needs to be added to a random delay $T_{waiting}(i)$ in addition to the previous three parts. This random time delay should be a function of the index variable i of the user's arbitration frequency f_i.

The maximum total time delay can be expressed as

$$T_{max}(i) = \begin{cases} T_{arb} + T_{TxRxarb} + T_{data}, & i = 1 \\ T_{arb} + T_{TxRxarb} + T_{data} + T_{waiting}(i), & 2 \le i \le N \end{cases} \tag{30}$$

For a user with arbitration frequency f_1, which is supposed to be the lowest of the entire arbitration frequency set, or for the network as a special type of the star network where there is only one activated user intending to transfer data, the possible total transmission delay is statistically deterministic. We consider the above case the *best-case scenario*. However, in the general case when the packet is transmitted in a considered time slot, we can calculate the maximum waiting duration for any user as

$$T_{waiting}(i) = (i-1)T_s \tag{31}$$

As T_{arb}, $T_{TxRxarb}$ and T_{data} are constants, when all users with lower arbitration frequency than that of user n_i are willing to access the channel for data transmission in the same considered time slot, the total communication delay $T_{max}(i)$ of user n_i depends on the value $T_{waiting}(i)$, which determines the upper limit of the queue, which is determined by the period of each arbitration phase T_s.

B. Normalized Throughput Analysis

In this section we will analyze the system saturation throughput for the proposed *WirArb* protocol. We consider the proposed DTMC model to analyze the throughput performance of the *WirArb* protocol. To calculate the average of the normalized system throughput (S) we follow the procedure in [5] and S is defined as the fraction of time the channel is used to successfully transmit payload bits and this can be expressed as

$$S = \frac{E[\text{payload information transmitted in a slot time}]}{E[\text{length of a slot time}]} \tag{32}$$

Consider that $E[T_P]$ is the average duration used for a packet payload transmission and $E[T_{suc}]$ is the average duration of a successful transmission, $E[T_{wait}]$ is the average duration of a transmission failure because of non-empty set $\widetilde{\Omega}(AF)$, which means there is at least one user in queue with a lower arbitration frequency. In addition, $E[T_{null}]$ is the average duration of a non-transmission time slot. Hence, the normalized system throughput can be expressed as

$$S = \frac{P_{tr}P_s r_{data} E[T_P]}{P_{tr}P_s E[T_{suc}] + P_{tr}\overline{P}_s E[T_{wait}] + \overline{P}_{tr} E[T_{null}]} \tag{33}$$

where the probability P_{tr} is used to statistically estimate how the shared channel can be randomly occupied by users from the system point of channel state, and describe how only one transmission in a randomly considered time slot is expected to occur. Accordingly, P_s is the probability that a transmission occurring on the channel is successful. When it comes to empty time slots, the average length of this non-transmission time slot is obtained with probability $\overline{P}_{tr}$. Then, we obtain

$$\overline{P}_{tr} = 1 - P_{tr} = \prod_{i=1}^{N} (1 - \rho_i b\{f_i, 0, 1\}) \tag{34}$$

If exactly one user with the highest priority transmits on the channel in the current arbitration phase, we have

$$P_s = 1 - \overline{P}_s = \frac{\sum_{j=1}^{N} \tau_j \left(\prod_{i=j+1}^{N}(1 - \rho_i b\{f_i, 0, 1\})\right)}{P_{tr}} \tag{35}$$

where $\overline{P}_s$ represents the current user encountering a transmission failure due to the presence of other higher-priority users in queue and has to wait.

If we consider the case where all data packets have the same fixed packet size, $E[P] = r_{data}$, $E[T_P] = P$. We define $E[P]$ as the average packet payload size, the average amount of payload information successfully transmitted in a slot time should be $P_{tr}P_s E[P]$, since a successful transmission occurs in a slot time with probability $P_{tr}P_s$. In the general case, it is thus necessary to assume a suitable probability distribution function $f(\cdot)$ for the payload size. To calculate the normalized system throughput for this case it is necessary to specify the corresponding values $E[T_{suc}]$ and $E[T_{wait}]$. From the flow chart shown in Fig. 2, we obtain

$$\begin{cases} E[T_{suc}] = T_{setup} + \delta + T_{arb} + \delta + T_{TxRxarb} + \delta + T_{data} \\ E[T_{wait}] = T_{wait} \end{cases} \tag{36}$$

where T_{setup} is used to receive the notification from the gateway to align time boundaries for network synchronization. Each user has a period of time T_{arb} used to transmit its arbitration signal. Then, $T_{TxRxarb}$ is used to set the output bit according to the final arbitration decision from gateway. It is reasonable to add δ as the turnaround time (i.e., transmitting/receiving switches). Finally, T_{data} is a period of time when the channel remains busy because of successful channel access and the master completes data transmission. Then, we obtain

$$T_{data} = T_{phy} + T_{mac} + E[T_P] \tag{37}$$

where T_{phy} and T_{mac} are defined as the time delay for the physical layer and MAC layer header, respectively. Furthermore, T_{wait} is a period of time during which the channel is allocated to another user with higher priority, i.e., $T_{wait} = T_s$.

C. Worst-Case Scenario

Without loss of the generality, we assume that the network is composed of at least two users, which are equipped with default functions to switch to active state from inactive state with a random transition probability $\beta \in (0, 1]$ which represents different events, i.e., wake-up from sleep with a certain probability, and migrated into the current network from other networks.

Fig. 5 shows the performance evaluation in terms of the normalized system throughput with different numbers of users.

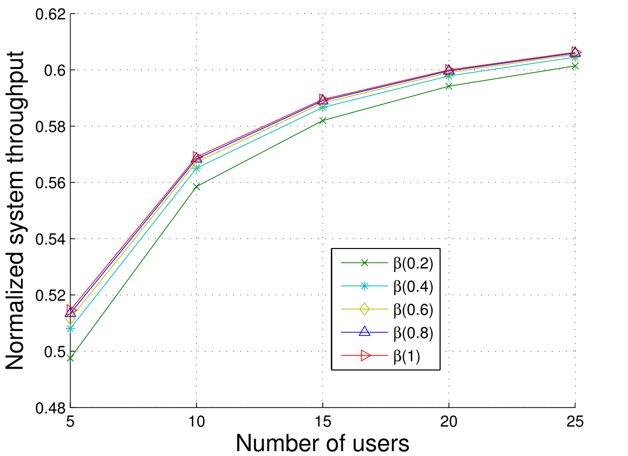


Fig. 5. Normalized saturation system throughput versus different numbers of users.

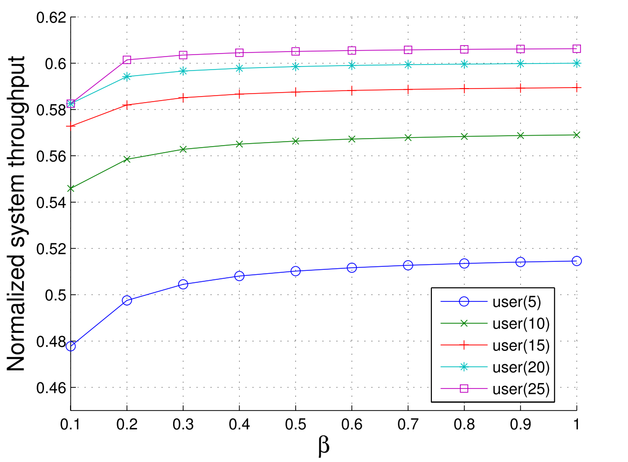


Fig. 6. Normalized system throughput versus different probability β.

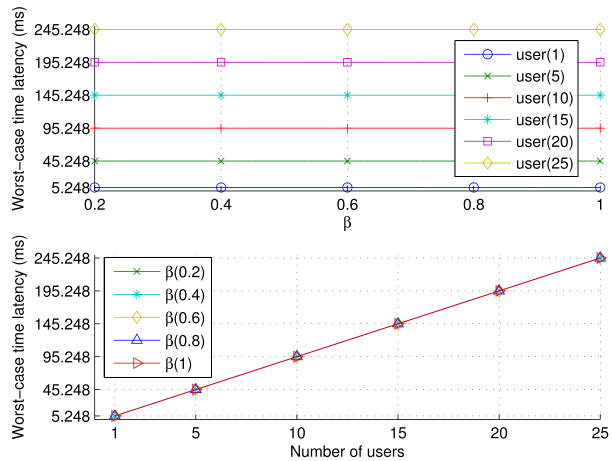


Fig. 7. Time delay versus different numbers of users.

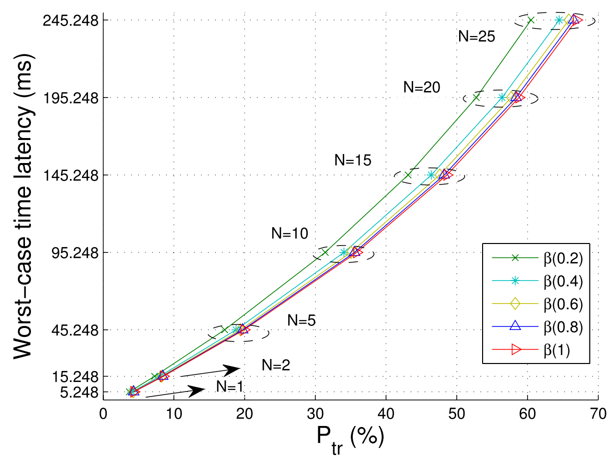


Fig. 8. Time delay versus probability P_{tr} under different numbers of users.

From this figure, it can be noted that when the user is activated from the inactive state at a greater probability, *WirArb* can enhance the system throughput performance. This distribution is derived from the ability to successfully transmit more MAC payload bits in each arbitration phase. As explained above, *WirArb* implies that more activated users with data transmission requests will results in a larger probability which shows how the wireless channel is expected to be randomly occupied by users; and also, how the channel can be ensured to be successfully assigned to the user (the so-called master) with the highest-priority without any collisions. Therefore, more activated users in the network yield higher system throughput performance. This fact will enhance the probability of channel occupancy which results in higher channel utilization efficiency.

Fig. 6 and Fig. 7 show the performance of the *WirArb* focusing on the normalized saturation system throughput and the transmission delay, respectively, versus different transition probabilities from inactive state to active state and different numbers of activated users. It shows that the saturation throughput of the system can be increased when the user can switch to be activated with a larger transition probability, or when the network size increases. The multiple access scheme and numerical results above describe that, the probability of occupying the channel in one arbitration phase will increase with the number of users, as shown in Fig. 7. Meanwhile in Fig. 7, it can be noted that for any non-highest priority user the total transmission duration will be inevitably extended as discussed in DTMC of *WirArb*, which is a result of queuing for arbitration in order to avoid interference from other users with higher priority than the current user. In the same figure, regarding the user with the lowest arbitration frequency in any worst-case scenario, it can be noted that the time delay is always maintained at a value of $5.248ms$. Another distribution we can obtain from the same figure is that if any new user with a lower AF is activated, it will affect other users of data transmission, i.e, the data transmission of other users with greater arbitration frequencies will be affected, which is reflected in the total transmission delay being magnified by several multiples of the period of arbitration phase. The maximum total transmission duration is deterministic and predictable rather than indistinct, due to different AFs representing different time criticality boundaries.

From Fig. 8, it is clear that the transmission duration increases with the total number of users in the network. A larger network size leads to increasing the probability P_{tr} by increasing the transition probability from inactive state to active state. The longest transmission duration is generated

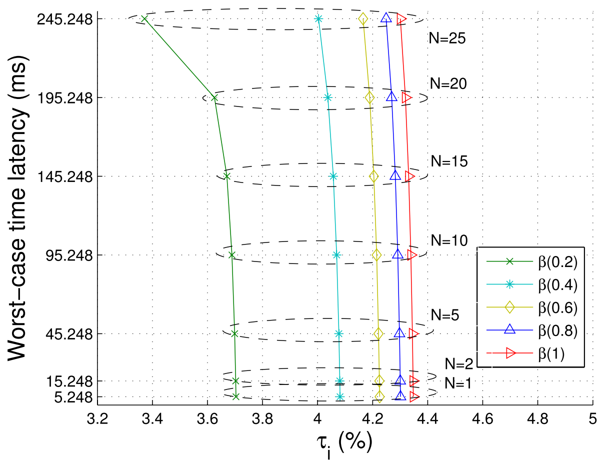


Fig. 9. Time delay versus probability τ_i under different numbers of users.

by the user who owns the highest arbitration frequency. The minimum total transmission delay is definitely guaranteed in $5.248ms$, which is the lowest limit for data transfer used by the user who owns the lowest arbitration frequency. As the number of users increases, the maximum transmission time delay of the user with the highest arbitration frequency is superimposed by integer multiple of $10ms$. Since user n_i owning the arbitration frequency f_i should avoid to interfere with the highest-priority user's data transmission, it has to queue up to $i-1$ times the period of the arbitration cycle. Like Fig. 8, Fig. 9 describes a distinct group of curves, which concentrates on the relationship between time delay and user state. Here, we take the probability τ_i into consideration which represents a statistical evaluation on the channel access and transmission in a random arbitration phase. Focusing on the user who owns the highest arbitration frequency, it has to avoid a non-competition and non-conflict idle channel in the current time slot for data transmission of other higher-priority users, the probability of accessing a channel in the current arbitration phase increases as the number of higher-priority users increases. As explained above, *WirArb* implies that more activated users having data transmission requests will result in a larger probability which shows how the wireless channel is expected to be randomly occupied by users; and also, how the channel can be ensured to be successfully assigned to the user (the so-called master) with the highest-priority without any collisions.

D. Channel Utilization

In this section we analyze the channel utilization by comparing *WirArb* with a TDMA-based WSN, such as WirelessHART. We define the channel utilization in the average number of transmissions per time slot in a superframe that have been received successfully by the gateway. The channel utilization can be calculated as

$$U = \frac{E[\sum_{i=1}^{N} N_{data}(i)]}{E[N_{slot}]} \tag{38}$$

where $N_{data}(i)$ represents the total number of data transmissions in a superframe by user n_i, and N_{slot} represents the total number of time slots in a superframe.

Figure 10 shows a typical TDMA structure which is scheduled with several independent time slots. The superframe is divided into n periodic frames of equivalent period, which is determined by the users' update rate (refresh rate or sampling rate). Considering WSAN where the regular data is forced to be exchanged periodically, regarding uplink service, the sensor should report environmental information or mechanical operation to the gateway periodically. For the downlink service, actuators should execute output instructions from the gateway periodically, and the typical period time can vary from $250ms$ to $1s$ depending on the applications in the industrial control system. In this paper we compare *WirArb* with WirelessHART, which is based on TDMA. In WirelessHART each event is at most allocated one fixed length time slot which is limited to $10ms$. As can be seen from Fig. 10, event $e_1, e_2, e_3, \ldots, e_i, \ldots, e_N$ can be correspondingly acquired by the $1^{st}, 2^{nd}, 3^{rd}, \ldots, i^{th}, N^{th}$ time slot within each frame periodically. As a result, the total number of time slots allocated to any active user $n_i, \forall i \in \{1, 2, \ldots, N\}$ is n. In addition, consider the time critical applications where an emergency event could be triggered when a specific condition is satisfied, such as in safety and security services. In the event of emergency, actuators must react within the required deadline. For instance, if the response time for the emergency event is limited to $250ms$, after the frame of TDMA is rescheduled, the $(N+1)^{th}$ time slot within each frame is assigned to this emergency e_e. Although a total of n time slots are scheduled to e_e, only the $(N+1)^{th}$ slot in the current frame when the emergency was triggered is effectively used while other $n-1$ slots will remain null. That is because the emergency event could not happen periodically.

Let us consider the delay boundary for regular data transmission which is fixed to $250ms$, and the time slot is limited to $10ms$. Therefore, in the case of TDMA-based network, we can calculate that the maximum number of network users in one frame that can be effectively scheduled is $N = 25$ under the constraint that no more than one user can use a channel in a given time slot. We assume that regardless of periodic regular data or aperiodic emergency data, the links are scheduled to communicate continuously. Fig. 11 and 12 illustrate a comparison result of the channel utilization of *WirArb* with TDMA. It can be seen that full channel utilization can always be achieved by the *WirArb* protocol regardless of the number of network users or the length of superframe. The maximum number of network users is determined by the priority of emergency events, which implies that any new user can access a channel at any time, once it obtains the highest priority (assuming that the receiver has enough capacity to distinguish arbitration frequencies). It is not necessary to reserve any time slots to use the non-periodic scheduling strategy of *WirArb*. The drawback of TDMA is poor channel utilization mainly due to handling a temporary emergency event; the TDMA scheduling reserves periodic time slots for each event. Our proposed *WirArb* is an on-demand scheduling MAC procotol, which means that the current time slot is only assigned to the user with the highest priority, while those following time slots are sequentially assigned to the remaining users in accordance with the predefined priority order. Before the period of data

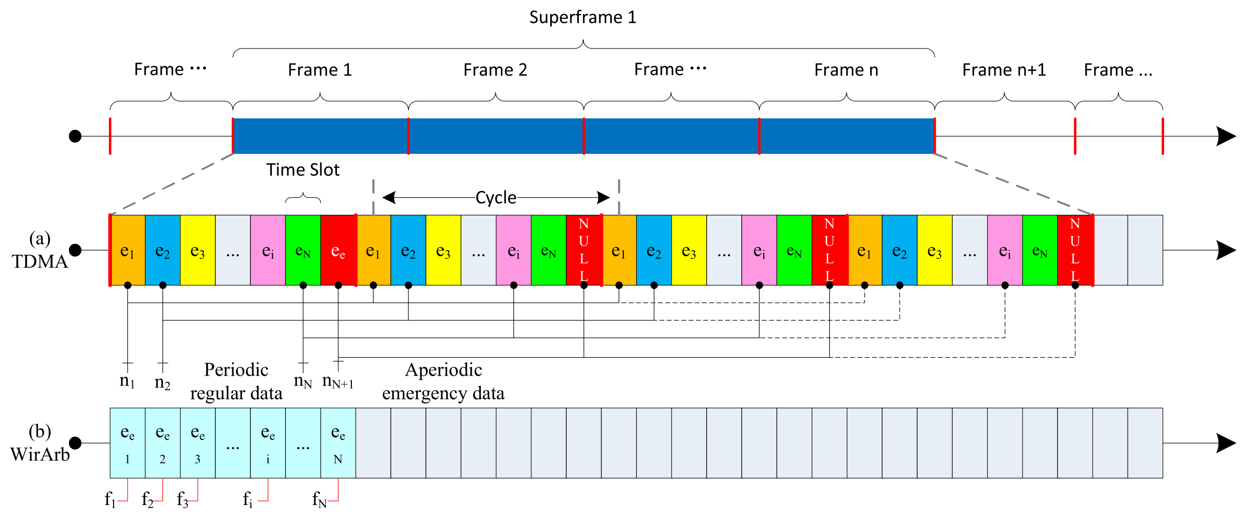


Fig. 10. The time slots scheduling for regular data and emergency data with TDMA and *WirArb*.

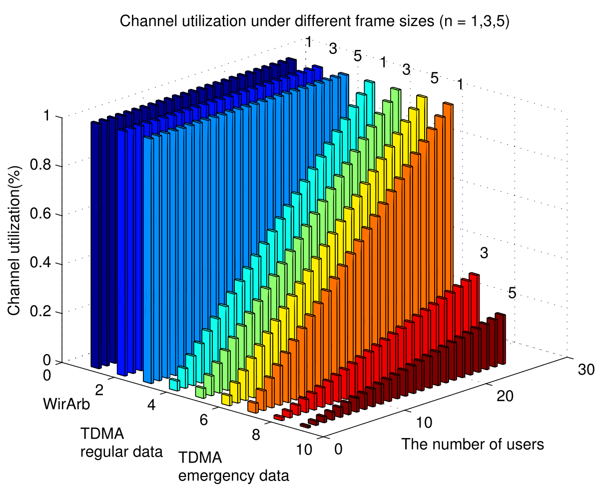


Fig. 11. The channel utilization of various protocols versus different numbers of users under delay boundary $250ms$.

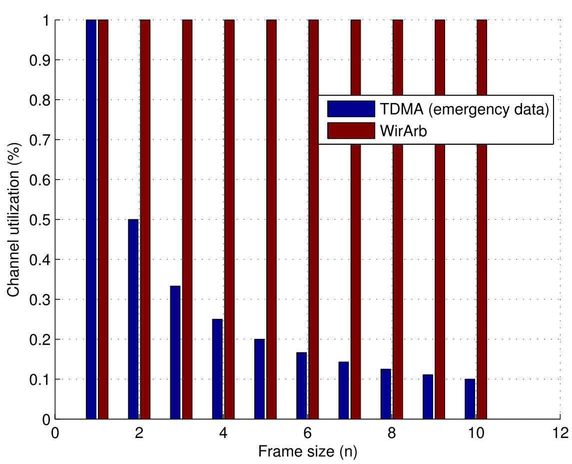


Fig. 12. The upper boundary of channel utilization of various protocols versus different superframe sizes.

transmission, a channel access arbitration procedure should be processed first. Each user in the network is pre-assigned a dedicated arbitration frequency with the purposed to identify channel access priorities which is utilized to determine channel access order. It ensures that in each data transmission stage the most critical event is the only one to immediately gain access to the channel, while the remaining users have to sleep. There is no channel contention and interference for the connections between the high priority node and the gateway. Therefore, the channel utilization by *WirArb* is 100%. This means that *WirArb* can give us a hard real-time guarantee with deterministic and predictable performance as well as better channel utilization than TDMA.

VI. Conclusion

Wireless sensor networks have gained acceptance in a number of domains, and in the future WSNs might be considered for time-critical real-time applications that can be found within industrial automation and avionics. Control applications in industrial automation typically have a hard deadline on data packet delivery and require deterministic communication. Unfortunately, today's WSNs based on CSMA/CA random backoff and TDMA periodic scheduling cannot offer that fully. Therefore, in this article we propose a new medium access protocol called *WirArb* which achieves time-critical data delivery in wireless sensor networks with low energy expenditure.

In *WirArb* each user is pre-assigned a dedicated arbitration frequency to determine the order for users to gain channel access. In this way we can offer a collision-free and deterministic communication over the wireless medium. To evaluate the performance of the proposed protocol we use a discrete time Markov chain. The results obtained show that the proposed *WirArb* performs well in terms of delay since the main bottlenecks in current WSNs are removed. The protocol is intended for short-range communication, and for this purpose, our protocol is reliable, even in harsh industrial environments. Since the proposed protocol is in the early stages of development there are several challenges that need to be investigated before a full-scale implementation can be achieved. For instance, more work is needed on setting the arbitration threshold, evaluation of response time and investigation of how the proposed protocol could work in multihop scenarios.

REFERENCES

[1] V. C. Gungor and G. P. Hancke, "Industrial wireless sensor networks: Challenges, design principles, and technical approaches," *IEEE Trans. Ind. Electron.*, vol. 56, no. 10, pp. 4258–4265, Oct. 2009.

[2] J. Åkerberg, M. Gidlund, T. Lennvall, J. Neander, and M. Björkman, "Efficient integration of secure and safety critical industrial wireless sensor networks," *EURASIP J. Wireless Commun. Netw.*, Nov. 2011.

[3] A. K. Mok and S. A. Ward, "Distributed broadcast channel access," *Comput. Netw.*, vol. 3, no. 5, pp. 327–335, 1979.

[4] M. Khanafer, M. Guennoun, and H. T. Mouftah, "A survey of beacon-enabled IEEE 802.15.4 MAC protocols in wireless sensor networks," *IEEE Commun. Surveys Tuts.*, vol. 16, no. 2, pp. 856–876, Second Quarter 2014.

[5] G. Bianchi, "Performance analysis of the IEEE 802.11 distributed coordination function," *IEEE J. Sel. Areas Commun.*, vol. 18, no. 3, pp. 535–547, Mar. 2000.

[6] S. Pollin *et al.*, "Performance analysis of slotted carrier sense IEEE 802.15.4 medium access layer," *IEEE Trans. Wireless Commun.*, vol. 7, no. 9, pp. 3359–3371, Sep. 2008.

[7] M. Doudou, D. Djenouri, and N. Badache, "Survey on latency issues of asynchronous MAC protocols in delay-sensitive wireless sensor networks," *IEEE Commun. Surveys Tuts.*, vol. 15, no. 2, pp. 528–550, Second Quarter 2013.

[8] 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.

[9] J. Åkerberg, F. Reichenbach, M. Gidlund, and M. Bjorkman, "Measurements on an industrial wireless HART network supporting PROFIsafe: A case study," in *Proc. IEEE 16th Conf. Emerg. Technol. Factory Autom. (ETFA)*, Sep. 2011, pp. 1–8.

[10] M. Wei and K. Kim, "Intrusion detection scheme using traffic prediction for wireless industrial networks," *J. Commun. Netw.*, vol. 14, no. 3, pp. 310–318, Jun. 2012.

[11] *IEEE Standard for Local and Metropolitan Area Networks–Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs) Amendment 1: MAC Sublayer*, IEEE Standard 802.15.4e-2012, 2012, pp. 1–225.

[12] P. Suriyachai, U. Roedig, and A. Scott, "A survey of MAC protocols for mission-critical applications in wireless sensor networks," *IEEE Commun. Surveys Tuts.*, vol. 14, no. 2, pp. 240–264, Second Quarter 2012.

[13] W. Ye, J. Heidemann, and D. Estrin, "An energy-efficient MAC protocol for wireless sensor networks," in *Proc. IEEE 21st Annu. Joint Conf. Comput. Commun. Soc. INFOCOM*, vol. 3. Jun. 23–27, 2002, pp. 1567–1576.

[14] W. Ye, J. Heidemann, and D. Estrin, "Medium access control with coordinated adaptive sleeping for wireless sensor networks," *IEEE/ACM Trans. Netw.*, vol. 12, no. 3, pp. 493–506, Jun. 2004.

[15] P. Lin, C. Qiao, and X. Wang, "Medium access control with a dynamic duty cycle for sensor networks," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, vol. 3, Mar. 2004, pp. 1534–1539.

[16] S. W. Hussain, T. Khan, and S. M. H. Zaidi, "Latency and energy efficient MAC (LEEMAC) protocol for event critical applications in WSNs," in *Proc. IEEE Int. Symp. Collaborative Technol. Syst. (CTS)*, May 2006, pp. 370–378.

[17] G. Lu, B. Krishnamachari, and C. S. Raghavendra, "An adaptive energy-efficient and low-latency MAC for tree-based data gathering in sensor networks," *Wireless Commun. Mobile Comput.*, vol. 7, no. 7, pp. 863–875, 2007.

[18] Y. Sun, S. Du, O. Gurewitz, and D. B. Johnson, "DW-MAC: A low latency, energy efficient demand-wakeup MAC protocol for wireless sensor networks," in *Proc. 9th ACM Int. Symp. Mobile Ad Hoc Netw. Comput.*, 2008, pp. 53–62.

[19] L. Choi, S. H. Lee, and J.-A. Jun, "SPEED-MAC: Speedy and energy efficient data delivery MAC protocol for real-time sensor network applications," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2010, pp. 1–6.

[20] M. Salajegheh, H. Soroush, and A. Kalis, "HYMAC: Hybrid TDMA/FDMA medium access control protocol for wireless sensor networks," in *Proc. IEEE 18th Int. Symp. Pers., Indoor Mobile Radio Commun. (PIMRC)*, Sep. 2007, pp. 1–5.

[21] P. Suriyachai, J. Brown, and U. Roedig, "Time-critical data delivery in wireless sensor networks," in *Distributed Computing in Sensor Systems* (Lecture Notes in Computer Science). Heidelberg, Germany: Springer-Verlag, 2010, pp. 216–229.

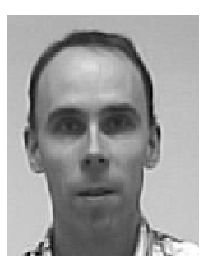
[22] J. Åkerberg, M. Gidlund, and M. Björkman, "Future research challenges in wireless sensor and actuator networks targeting industrial automation," in *Proc. 9th IEEE Int. Conf. Ind. Inform. (INDIN)*, Jul. 2011, pp. 410–415.

[23] L. Angrisani, M. Bertocco, D. Fortin, and A. Sona, "Experimental study of coexistence issues between IEEE 802.11b and IEEE 802.15.4 wireless networks," *IEEE Trans. Instrum. Meas.*, vol. 57, no. 8, pp. 1514–1523, Aug. 2008.

Tao Zheng (S'09–M'15) received the B.S. degree in communications engineering and the Ph.D. degree in communication and information systems from Beijing Jiaotong University, Beijing, China, in 2006 and 2014, respectively.

He was a Visiting Researcher with the ABB Corporate Research, Sweden, from 2012 to 2013, where he was involved in the Industrial Communication and Embedded Systems. He is currently a Faculty Member with Beijing Jiaotong University. His specific areas of research interest mainly focus on physical layer and MAC layer communication technology in industrial network and Internet of Things, interference avoidance technology and coexistence network optimization, and hardware implementation of wireless sensor networks.

Mikael Gidlund (M'99) received the M.Sc. and Ph.D. degrees in electrical engineering from Mid Sweden University, Sweden, in 2000 and 2005, respectively. He is currently a Full Professor of Computer Engineering with Mid Sweden University, and since 2014, he is also working as a Scientific Advisor with ABB Corporate Research. In 2005, he was a Visiting Researcher with the Department of Informatics, University of Bergen, Norway. From 2006 to 2007, he was a Research Engineer and Project Manager, responsible for wireless broadband communication at Acreo AB, Sweden. From 2007 to 2008, he was a Senior Specialist and Project Manager with responsibility for next-generation IP-based radio solutions at Nera Networks AS, Bergen, Norway. From 2008 to 2013, he was a Senior Principal Scientist and Global Research Area Coordinator of Wireless Technologies with ABB Corporate Research. He holds more than 20 patents (granted and pending applications) in the area of wireless communications, and has authored or co-authored over 100 scientific publications in refereed fora. His research interest are wireless communication and networks, wireless sensor networks, access protocols, and security. He won the Best Paper Award at the IEEE International Conference on Industrial IT in 2014. He is an Associate Editor of the IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS.



Johan Åkerberg (M'08–SM'11) received the M.Sc. and Ph.D. degrees in computer science and engineering from Mälardalen University, Sweden. He has close to 20 years' experience within ABB in various positions, such as Global Research Area Coordinator, Research and Development Project Manager, Industrial Communication Specialist, and Product Manager. He is currently a Principal Scientist with ABB Corporate Research, Sweden. He is also a Vice Chair in the IEEE-IES Technical Committee on Factory Automation. He is mainly involved in communication for embedded real-time systems in industrial automation and is frequently invited to give talks to governmental bodies, international universities, and automation fairs. He holds more than ten patents (granted and pending applications) in the area of wired/wireless industrial automation, and has authored or co-authored numerous scientific publications in refereed conferences and journals.