On the Behavior of Synchronous Data Transmission in WuR enabled IoT Networks: Protocol and Absorbing Markov Chain based Modeling

Debasish Ghose, Luis Tello-Oquendo, *Member, IEEE*, Vicent Pla, and Frank Y. Li

***Abstract*—In wake-up radio (WuR) enabled Internet of things (IoT) networks, a data communication occurs in a synchronous or asynchronous manner initiated either by a transmitter or receiver. A synchronous transmission is triggered when multiple devices report an event simultaneously, or by a common wake-up call. In this paper, we focus on synchronous transmissions and propose a *mu*lticast *tr*iggered *s*ynchronous *t*ransmission protocol, abbreviated as MURIST, which enables contention based and coordinated data transmissions among distributed devices in order to reduce transmissions latency and energy consumption. Furthermore, we develop a novel analytical model based on an absorbing Markov chain to evaluate the performance of MURIST in a network cluster. Unlike existing models that are merely targeted at the behavior of a single device, the novelty of our model resides in a generic framework to assess *the behavior of a cluster of devices* for synchronous data transmissions. Based on the analytical model, we obtain closed-form expressions for the distributions of successful and discarded transmissions, number of collisions, and delay, as well as for energy consumption. Extensive simulations are performed to validate the accuracy of the analytical model and evaluate the performance of our scheme versus that of two other schemes.**

***Index Terms*—IoT networks, wake-up radio, medium access control, absorbing Markov chain.**

I. Introduction

THE fifth-generation (5G) communication system is expected to facilitate human-type as well as machine-type communications (MTC) through a seamless and ubiquitous wireless network infrastructure. According to the third generation partnership project (3GPP), 5G targets at three major technology pillars based on cellular networks, i.e., enhanced mobile broadband (eMBB), ultra-reliable low latency communications (URLLC), and massive MTC (mMTC). These technologies will bring unprecedented innovations and fuel the growth of a variety of novel applications [1] [2].

Specifically, the avail of emerging MTC technologies will facilitate various Internet of things (IoT) applications such as industrial automation, smart cities and smart agriculture, critical infrastructure surveillance, vehicle-to-vehicle (V2V) or unmanned aerial vehicle (UAV) based communications, and smart grid applications. A common feature of these applications is to gather data from IoT or sensor devices, which are typically battery powered and deployed to measure and report different parameters of monitored environments.

Thanks to the rapid development of sensors, wireless communication technologies, and autonomous flying/moving capabilities, the potential applications of UAVs are rapidly increasing. For example, they have recently spread from asset inspection and image capturing to data collection from end IoT devices [3]. However, one of the major constraints of battery-powered UAVs is their short flight duration, which lasts typically for 15-30 minutes [4]. Such a short flight duration limits the data collection capability of UAVs when they need to collect data, especially across large areas. Thus, an efficient data communication protocol is of great importance for the operation of data collection with respect to both latency and energy consumption including not only data collectors like a UAV but also end devices.

Moreover, the lifetime of deployed *end devices* plays a vital role for the success of IoT applications. For instance, various 5G IoT/mMTC applications require that the lifetime of end devices lasts for 10 years or longer [1] [5]. To achieve long/ultra-long device lifetime, energy conservation through duty-cycled (DC) based medium access control (MAC) has been a legacy solution. However, the nature of DC based transmissions incurs a shortcoming of potentially long delay since a message cannot be transmitted when a device is in the sleep mode. In LoRaWAN [6], for instance, with a DC ratio of 1%, a device that has just transmitted a packet lasting for X seconds will have to wait for $99 \times X$ seconds for its next transmission. Consequently, the popularity of DC based data collection has diminished in many IoT applications, especially for scenarios that require short delay and low packet loss.

Wake-up radio (WuR), which allows an IoT device to sleep for most of the time without jeopardizing its capability to perform *instant data collection*, is a convincing technique that can

Manuscript received January 23, 2021; revised July 2, 2021 and September 17, 2021; accepted April 3, 2022. Date of publication MM DD, 2022; date of current version April 11, 2022. The research leading to these results has received funding from the NO Grants 2014-2021, under project contract no. 42/2021, RO-NO-2019-0499 - "A Massive MIMO Enabled IoT Platform with Networking Slicing for Beyond 5G IoV/V2X and Maritime Services (SOLID-B5G)". The work of V. Pla was supported in part by Grant PGC2018-094151-B-I00 funded by MCIN/AEI/10.13039/501100011033 and ERDF A way of making Europe, and in part by grant AICO/2021/138 funded by Generalitat Valenciana. The associate editor coordinating the review of this article and approving it for publication was P. Li. (Corresponding author: Frank Y. Li.)

D. Ghose and F. Y. Li are with the Department of Information and Communication Technology, University of Agder (UiA), N-4898 Grimstad, Norway (email: {debasish.ghose; frank.li}@uia.no).

L. Tello-Oquendo is with the College of Engineering, Universidad Nacional de Chimborazo, Riobamba 060108, Ecuador, and the Facultad de Ingeniería en Electricidad y Computación, Escuela Superior Politécnica del Litoral, Guayaquil, Ecuador (email: lptelloq@ieee.org).

V. Pla is with the Department of Communications, Universitat Politècnica de València (UPV), València 46022, Spain (email: vpla@upv.es).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TWC.2022.....

Digital Object Identifier 10.1109/TWC.2022...

meet such lifetime and latency requirements [7] [8]. However, one of the inherent shortcomings of WuRs is that wake-up calls (WuCs), which are needed to trigger a data transmission procedure, are transmitted at a low data rate. This is due to the fact that wake-up receivers (WuRxs) are typically designed with low receiver sensitivity since the primary purpose of WuR is to minimize energy consumption. Such a long WuC duration induces extra delay in data communication, especially if a centralized polling approach is adopted, e.g., when a unicast WuC is adopted for data collection from multiple devices in a time-division and point-to-point manner. Although a centralized polling protocol is preferable for *periodic data report*, its benefit is achieved at a cost of extra protocol overhead for handshake between a transmitter and a receiver. To operate a network utilizing a time-division based mechanism, coordination or synchronization between a controller and its covered devices is required. A centralized polling or time-division based mechanism also faces another drawback of low slot utilization for sporadic MTC traffic since some devices may not have any data to transmit during their allocated slots. To overcome these limitations, distributed access protocols are preferable for *irregular on-demand data collection* in WuR enabled IoT networks, especially under sporadic/low traffic load conditions. Following a distributed access protocol, only nodes that have data to transmit will compete for channel access. Such a procedure can be triggered through a one-to-many broadcast/multicast WuC, resulting in shorter delay and higher channel utilization.

However, a shortcoming of distributed access based data transmission is its vulnerability to potential collisions since a device in a distributed network is not aware of the transmission(s) of other devices in the same network. A collision could occur either when two or more devices initiate their WuCs simultaneously or when the transmissions of data packets by more than one device overlap with each other. In the literature, many protocols that deal with WuC collisions have been proposed [9]–[11].

To evaluate the performance of these protocols, a few analytical models have been proposed based on steady state analysis [12] [13]. However, a main drawback of these models is the common assumption that there does not exist any dependency among devices. Moreover, important performance metrics such as delay distribution, distribution of the number of collisions, and distribution of the number of backoff (BO) cycles are not straightforwardly obtainable from these existing analytical models.

The aforementioned observations triggered our motivation to propose a *mu*lticast t*ri*ggered *s*ynchronous *t*ransmission protocol, abbreviated hereafter as MURIST, to reduce data transmission latency. By *synchronous* transmission, it is meant that multiple devices attempt to transmit their packets at the same time in a distributed manner, triggered either by a common WuC or an event observed by multiple devices. Furthermore, we develop a generic analytical framework that models the behavior of a set of devices when performing synchronous data transmissions.

The main contributions and the novelty of this paper are summarized as follows.

- A WuR based MAC protocol, which initiates a bunch of synchronous packet transmissions from a cluster of devices through a *common multicast* WuC is proposed.
- To model the behavior of the proposed protocol, a generic analytical framework based on an absorbing Markov chain is devised. In contrast to steady state based analyses, our framework focuses on analyzing the transient regime upon the occurrence of a synchronizing event.
- Closed-form expressions for the distribution of BO, successful or discarded transmissions, number of collisions, and access delay distribution are deduced. Energy consumption is also calculated. The accuracy of the model is validated through extensive discrete-event simulations and the performance of the protocol is evaluated in comparison with that of two reference protocols.

To the best of our knowledge, this is the first effort to develop an absorbing Markov chain based framework to model the behavior of a set of WuR devices that act in a synchronized and distributed manner. Moreover, our model focuses on the analysis of the transient regime upon the occurrence of a synchronizing event, in contrast to the other steady state based analyses. We believe that this type of protocol and its performance modeling are more appealing in the context of WuR enabled IoT networks with irregular traffic generation where transmission epochs do not happen often nor periodically.

The remainder of this paper is structured as follows. In Sec. II, we provide preliminaries on WuR, summarize the related network, and highlight the differences between our work and the existing ones. The network scenario is presented in Sec. III. Afterwards, we explain the proposed protocol in Sec. IV. In Sec. V, we develop a novel analytical model and derive expressions for calculating performance metrics. Then, numerical results and performance evaluation are presented in Sec. VI. Finally, the paper is concluded in Sec. VII.

II. Preliminaries and Related Work

In this section, we first provide some preliminaries and background information on WuR. Then, the trend of WuR research and related work are summarized.

A. Preliminaries on WuR

Traditionally, DC based MAC protocols have been the most popular solution for improving energy efficiency and extending lifetime of sensor networks. However, DC MAC protocols suffer from idle listening and overhearing when a device idly listens to a channel for receiving control messages and overhears the transmission to other devices, respectively.

In recent years, a paradigm shift from DC MAC to WuR has been envisaged [7]. A WuR enabled IoT device is operated in an on-demand manner, i.e., it stays in deep sleep under normal conditions and wakes up only when it is required to transmit or receive data. Using a dedicated WuRx, which is attached to the main radio (MR) of an IoT device, no DC operation is needed. As such, the two downsides of DC MAC (i.e., idle listening and overhearing by the MR of the sensor devices) are avoided. It has been shown that the average energy consumption of a WuRx is 1000 times lower than that of

the MRs [7] [14]. Furthermore, the developed WuR in [15] achieves approximately 70 times longer lifetime than what is obtained in DC protocols under light traffic load.

In WuR enabled IoT networks, each end device is equipped with a WuRx that is always awake and consumes ultra-low energy. Whenever the WuRx correctly receives a WuC that matches its address, it interrupts its micro-controller unit (MCU) to switch on the MR for packet exchange. As soon as the MRs of both transmitter and receiver are active, a packet communication procedure proceeds. The procedure finishes with an acknowledgment (ACK) from the receiver. Then, the MRs of both sides go back to deep sleep, whereas their WuRxs still remain active listening to channel continuously.

WuR enabled data transmissions can be performed either in the transmitter-initiated (TI) or in the receiver-initiated (RI) mode. While the TI mode focuses on event-triggered *data reporting*, the RI mode is better suited for aperiodic *data collection*. In the TI mode, when an event is observed by more than one device, they may initiate data transmissions by emitting their WuCs at (almost) the same time, leading to WuC collision(s). For data collection, a data collector needs to initiate data transmissions by sending a WuC to the intended device. Typically, WuCs are sent as a unicast message in order to avoid collisions. However, this kind of operation induces long delay when collecting data from a set of devices since multiple individual based WuCs and packet transmissions are needed for both downlink and uplink in order to collect data consecutively from these devices. Alternatively, multiple devices may be waked up simultaneously by one common WuC. However, a collision among data transmissions occurs if another device or other devices transmit their packets before the ongoing packet transmission is complete.

B. State-of-the-art and Related Work on WuR

While the research and development efforts on WuR have been focusing on circuit implementation, system prototyping, and protocol design during the past decade [9]–[11], the most recent progress is the trend of coupling WuR towards real-life systems [23]–[26]. This trend is represented by the recent standardization activities to enable WuR in both 5G and wireless fidelity (WiFi) networks [23] [24]. In what follows, we summarize a few aspects of the state-of-the-art in WuR research which are most relevant to our work.

1) WuR prototype: Range, sensitivity, and WuC duration: Targeting at providing decade-long lifetime for IoT devices, various WuR receivers have been implemented, achieving a power consumption level of a few microwatts or even nanowatts [7] [8], [14]–[16]. In our earlier work [8], a WuR prototype achieving a current consumption level of 390 nA and offering cellular IoT connectivity via a smartphone has been implemented. In [16], the authors presented a WuR implementation which achieved an ultra-low power consumption level of 13 nW. Operated often at the carrier frequency of 868 MHz, the receiver sensitivity of these μW or nW WuR prototypes is typically around -55 dBm or higher [15] [16]. Consequently, the transmission ranges of WuCs are comparatively short [7] [16].

On the other hand, improving WuRx sensitivity could significantly increase the transmission range of WuC. For example, the WuRx design reported in [27] achieved a sensitivity level of -83 dBm, able to cover an estimated distance of 1200 meters. Another implementation achieved a sensitivity level of -80 dBm at a cost of much higher current consumption of 1 mA [28]. With this sensitivity level, it is estimated that a distance of up to 8.7 km may be achieved. Furthermore, WuC signals are modulated by simple modulation schemes and transmitted at a very low data rate, e.g., 1 kbps [16].

2) WuR data transmissions: Synchronous vs. asynchronous: While traditional MAC protocols in wireless sensor networks (WSNs) focus on how to control power consumption for data exchange among devices efficiently, WuR MAC protocols need to coordinate the operations between each WuRx and its MR (for WuR enabled devices) as well as among neighboring devices (the same as in WSNs). A data exchange in a WuR enabled IoT network can be performed either in an asynchronous or a synchronous manner, triggered by a WuC initiated from an MR.

For light and/or sporadic IoT traffic, an asynchronous MAC protocol is the favorite solution. In [29], an opportunistic WuR MAC protocol has been proposed through which the best intermediate/relay node among an IoT device's neighbors is selected based on a predefined metric. In [30], a WuR based MAC protocol for autonomous WSNs was proposed, enabling asynchronous data communications between a sink and member IoT devices. To further improve energy efficiency and reduce end-to-end delay, we proposed two techniques for data transmissions in WuR enabled IoT networks, known as *early sleeping* based on bit-by-bit WuC address decoding and *early data transmission*, respectively [19].

On the other hand, synchronous MAC protocols may be advantageous when multiple devices intend to transmit at the same time (triggered by an event or/and a common data collector) or for periodic reporting. Considering that a device may have multiple packets to transmit, an RI consecutive packet transmission WuR MAC protocol was proposed to eliminate multiple competitions [13]. To avoid collision due to simultaneous WuC or/and data transmissions, various BO procedures can be employed [12] [19]. However, performing clear channel assessment (CCA) will consume extra energy. For RI-based transmissions, introducing a BO procedure could eliminate collision but would introduce extra delay.

3) Modeling WuR behavior: While earlier work on WuR performance evaluation focused on simulations and testbeds [7] [15], analytical models are emerging [25] [35]. However, the mathematical models developed for carrier sense multiple access (CSMA) with collision avoidance (CSMA/CA) based wireless networks, e.g., [31] and [32], are not directly applicable to WuR enabled IoT networks due to WuR's on-demand communication nature.

When it comes to modeling the protocol performance of WuR, most work targeted at either TI-WuR or RI-WuR protocols. For example, a discrete-time Markov chain (DTMC) was developed to model BoWuR for TI-WuR based WSNs in [18]. An absorbing Markov chain framework was developed for assessing the performance of TI-WuR in [33] by assuming that

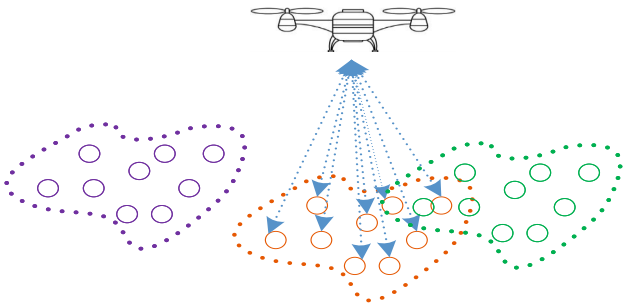

Fig. 1: A WuC network where a UAV collects data from a cluster of IoT devices by multicasting a common WuC.

the number of packet failures follows a geometric distribution. In [12], we presented an M/G/1/2 queuing model to evaluate the performance of TI-WuR protocols for asynchronous operations. Moreover, TI-WuR was evaluated analytically in [34] but collisions were neglected in their calculations. To assess the performance of RI-WuR protocols, we developed two associated DTMCs for modeling the behavior of consecutive packet transmissions in [19]. More recently, a semi-Markov process based model was developed to analyze the behavior of WuR compared with discontinuous reception in cellular networks [25].

However, the aforementioned models were developed from an individual device point of view without considering the dependency and synchronous operations among multiple IoT devices when competing with each other for data transmission in a network cluster. Furthermore, these existing models considered only the steady state regime and none of these models applies to both TI and RI modes. In a nutshell, the following salient features make the developed model in this study distinct from the existing ones. That is, 1) the behavior of a WuR cluster with multiple devices is modeled in our framework by considering the dependency and synchronous transmissions among the member devices in the cluster; 2) our model applies to synchronous WuR data transmissions in both TI and RI modes with transient behavior. As an example, we only report a network operated in the RI mode in Sec. VI of this paper. When the proposed protocol is operated in the TI mode, we consider that a common event has been observed simultaneously by multiple devices. In that case, no common WuC from the data collector is needed. Instead, the BO procedure is triggered when a common event is observed.

III. Network Scenario and Assumptions

Consider a large-scale IoT network that is composed of multiple network clusters, each with a set of WuR enabled IoT devices. The coverage of neighboring clusters may or may not overlap, but each cluster has a unique cluster netmask address. To collect data, a UAV acts as a data collector. For each data collection mission, the UAV flies from a base and hovers above the region of interest for data collection. During one mission, multiple rounds of data collection will be performed, each responsible for collecting data from only one cluster.

In Fig. 1, we illustrate an envisaged scenario where a UAV is collecting data from a region with three clusters. These end devices are deployed and grouped into multiple clusters to monitor various environmental, asset, or infrastructure parameters. As a motivating example, consider that these devices are deployed along a bridge for critical infrastructure surveillance. A mission of data collection performed by a UAV requires therefore high probability of successful transmissions and low delay. Hereby it is worth highlighting that this type of scenarios and the proposed protocol fit better event-driven *data collection*, whereas DC based transmission protocols by LoRaWAN, among others, suit better *data reporting* under normal circumstances.

Each IoT device in such a network is equipped with a WuRx associated with its MR. The main task of the WuRx is to detect a WuC and trigger the MR of the device for data communication if the WuC address matches. After receiving a WuC from the UAV, the devices with the same netmask address, e.g., the member devices in the same cluster, wake up simultaneously. They will then compete with each other for data uploading towards the UAV over a single hop. In this way, the IoT devices in the same cluster perform data transmissions in a synchronous manner. Following the protocol to be presented in Sec. IV, each round of data collection is triggered by a multicast WuC message emitted from the UAV and operated in the RI mode.

In this study, we assume that the downlink WuC transmission from the UAV can reach all member IoT devices in a cluster but it does not cover all devices in the same region. The same assumption applies to uplink traffic for data packet transmissions. During the data collection procedure of each round for any cluster, the UAV remains stable in the air. However, how to partition devices into clusters and how to design an optimal trajectory for flying the UAV across multiple clusters in the region of interest are beyond the scope of this paper. Similarly, we do not consider the power consumption for UAV aviation. Furthermore, we assume that the channel between the UAV and any member device is error-free for both uplink and downlink, and the propagation delay is negligible.

As a WuR enabled device, each end device is operated in three different operation modes: *deep sleep, light sleep*, or *active*. In the deep sleep mode, the MR sleeps, whereas its WuRx is always on. In the light sleep mode, the function of the MR is partially on, being able to decode and validate WuCs. In the active mode, the MR is entirely operating and it performs data transmissions. As an example, the current consumption for these three modes is 0.39 μA for deep sleep, 1.9 μA for light sleep, and 5.3/5.4 mA for data transmission and ACK reception, respectively [8].

IV. MURIST: Principle and Operations

In this section, we propose a MAC protocol for synchronous data transmissions of a cluster of WuR enabled IoT devices focusing on collision avoidance, shortening delay, and reducing energy consumption.

A. The Principle of MURIST

For each round of data collection, a WuC with the common netmask address of a cluster is emitted by the data collector. In

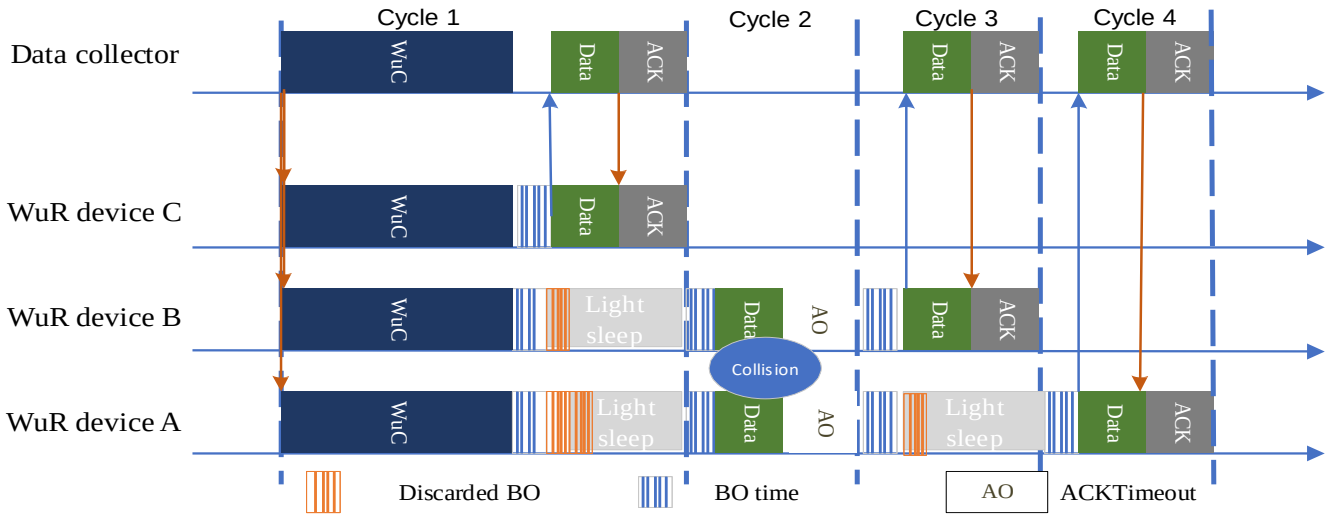

Fig. 2: Illustration of packet transmissions in MURIST MAC: An example with three devices competing with each other. There are four cycles of competition for this small-scale cluster. A collision occurred during the second cycle.

MURIST, we adopt a multicast address to wake up a cluster of devices synchronously instead of waking them up individually. There are two reasons that explain the rationale behind our design: 1) the data rate for transmitting WuCs is typically low, especially when a longer range is considered [7] [16]; and 2) a netmask (common for a cluster of devices) is shorter than a unicast address (unique for each individual device). For instance, a netmask for a cluster of devices could be adopted as the multicast address [17]. Following the bit-by-bit WuC address decoding principle implemented in our earlier work [19], those devices which lie in the overlapping area but do not belong to the corresponding cluster will go back to deep sleep before the netmask address is fully decoded.

Clearly, adopting a multicast address instead of a unicast address would lead to shorter delays. While a *mulicast* WuC is targeted at waking up a cluster of devices in a one-to-many manner, a *unicast* WuC is intended to wake up a specific device which is uniquely identified. Consider a network cluster with N devices and a data collector. By unicast transmission, it is meant that such a data transmission occurs in a one-to-one manner *for both uplink and downlink*. Therefore, the duration of a data collection round would require N WuCs plus N data frame transmissions when unicast transmission is adopted. Using our protocol, the duration contains only one WuC transmission plus N data frame transmissions. However, collisions need to be resolved, as explained in the next subsection.

Algorithm 1: MURIST Operation for Each Device

Start channel access competition: A WuC from a data collector is correctly decoded
Input : $Max\_attempt$; CW
Output: $S_T = 1$ (successful transmission) or $S_T = 0$ (unsuccessful transmission)
Initialization: $Num\_of\_attempt = 0$; $S_T = 0$
while $S_T = 0$ & $Num\_of\_attempt < Max\_attempt$ **do**
  $Channel\_status = idle$;
  $BO\_slot\_counter = random[0, CW - 1]$; //: Select a random BO interval from $[0, CW - 1]$
  $Num\_of\_attempt = Num\_of\_attempt + 1$;
  while $Channel\_status = idle$ & $BO\_slot\_counter > 0$ **do**
    $BO\_slot\_counter = BO\_slot\_counter - 1$;
    $Channel\_status = EnergyDetection()$; //:Channel status assessment. Returns *idle* or *busy*
  end
  if $Channel\_status = busy$ **then**
    Clear the $BO\_slot\_counter$; //: Discard the remaining BO interval;
    Go to light sleep for the ongoing packet transmission duration; //: Another device is sending
  else
    Transmit a packet; //: Send a packet to the data collector since the channel is idle
    Set ACKTimeout timer; //: Wait for an ACK from the data collector
    if *ACK is received within ACKTimeout* **then**
      $S_T = 1$; //: Transmission is successful
    end
  end
end
Go to deep sleep.

B. The Operation of MURIST

In an IoT network operated based on MURIST, each device needs to follow the principle of the proposed protocol which is illustrated in Algorithm 1. When the *preamble* of a WuC is detected by the WuRx of a device, it triggers the MR to switch on partially to the light sleep mode for WuC address validation. If the decoded address matches the multicast address and the device has data to report, the MR of the device will fully wake up to the active mode. On the other hand, those devices that do not have a packet to transmit will return to deep sleep right after light sleep and therefore will not participate in channel access competition.

To collect data from a cluster, multiple rounds of transmission competitions (each referred to as a BO cycle or simply a cycle) may be needed. During any BO cycle, each active device selects a random BO interval uniformly from the range of $[0, CW - 1]$, where CW represents the contention window

size and the BO interval is equal to the selected integer number multiplied by the BO slot duration. Then it starts to count down its BO interval slot-by-slot. During each BO slot, the device performs *energy detection*, which is one of the five CCA modes defined in the IEEE 802.15.4 standard [20], to check channel status. Since energy detection is performed by simply integrating the square of the received signal or signal envelope without any a priori knowledge on the type of underlying modulation scheme, it consumes less energy as no decoding by MCU is needed [21]. Therefore, we adopt energy detection in MURIST.

When multiple devices compete for channel access during any BO cycle, a device may regard itself as the winner of the current channel access contention when its BO counter reaches 0 and the channel is assessed as being idle. Immediately, it transmits its packet. However, since there is no coordination among competing devices, two or more devices may have the same assessment, resulting in a collision. In other words, a device wins the competition and transmits its data frame successfully if and only if it is the sole device that selects the minimum BO interval. A frame transmission finishes when an ACK message is received by the device. Immediately afterwards, the winning device goes back to deep sleep.

On the other hand, the remaining devices which did not win the channel access competition would discard their remaining BO slot countdowns and compete for access in the next round by selecting another BO interval, still from [0, CW-1]. During the period when the winning device is transmitting, the MRs of the other devices would remain in the light sleep mode. This implies that each device knows the duration of an ongoing frame transmission, which is assumed to be the same for all frame transmissions. This procedure will continue until all devices have finished their data packet exchanges, meaning that they have either successfully transmitted their packets or the maximum number of transmission attempts has been reached. Note that no ACK from an end device upon the reception of a WuC is needed prior to data transmission since the UAV is always active, ready to receive data during the whole period of each data collection mission.

As an example, Fig. 2 illustrates the operation of the MURIST protocol for a single packet transmission in a small-scale network with three IoT devices A, B, C, and one data collector. Upon receiving a common WuC from the data collector, these three devices compete for data transmission. At the first round of competition, i.e., during Cycle 1, device C is the only one that selects the lowest BO slot number, hence winning the competition. Then, it transmits a data packet to the data collector and waits for an ACK. As soon as an ACK from the UAV is received, device C goes back to the deep sleep mode. In the next competition round, i.e., during Cycle 2, devices A and B select the same BO slot interval. Thus a collision occurs. Thereafter, devices A and B follow the same procedure until they have successfully transmitted their packets (as an example shown in Fig. 2) or have reached the maximum transmission limit (as a general rule).

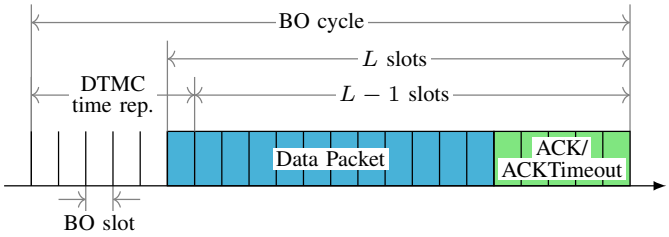

Fig. 3: Illustration of a BO cycle which consists of multiple BO slots and a successful or an unsuccessful packet transmission. The DTMC models represents the sum of the BO slots and the first slot of the packet transmission.

V. An Absorbing Markov Chain Based Model

To evaluate the performance of the proposed protocol, we develop an absorbing Markov chain based model as presented below. The model describes the state of a network cluster from the perspective of a randomly selected device in it, which we refer to as a *tagged device*. The remaining devices in the network cluster are referred to as *other devices*. The model captures the behavior of the network over a finite time period that goes from the occurrence of a triggering event, which initiates the transmission of a packet from each of the devices in the cluster, to the moment when the tagged device finishes. During this period, the network is regarded as being operated in a *transient* state. The tagged device finishes when it either acquires access to the channel and transmits the packet successfully or discards the packet because the maximum number of channel access attempts, denoted by M, has been reached. From the analysis of the model, we derive several performance metrics for the tagged device. Note, however, that these performance metrics are representative for the whole cluster of devices since the tagged device is randomly selected among the N devices in the cluster. A list of notations that are used in our analytical framework is summarized in Table I.

A. Modeling the Network Cluster Behavior via an Absorbing Markov Chain

Our model is based on a DTMC, $\{\boldsymbol{X}_t\}_{t\geq 0}$, with $M+1$ absorbing states, denoted by $a_1, a_2, \ldots, a_{M+1}$, respectively. Absorption into a_i, $i = 1, \ldots, M$, corresponds to the case in which the tagged device managed to transmit at the i-th attempt, whereas absorption into a_{M+1} corresponds to the case in which the packet was discarded after M unsuccessful attempts.

Throughout this section, we use a discrete-time framework with a time unit being the duration of one BO slot. The evolution of the DTMC only includes the time spent in BO and the first slot used for transmission in each BO cycle, i.e., for each round of transmission competition, as illustrated in Fig. 3. In other words, if L denotes the duration of the transmission of a packet measured in terms of number of BO slots, there are $L-1$ slots that are used for transmission in each BO cycle but they are not counted in the DTMC. Note that these $L-1$ slots will nevertheless be counted for delay calculation. Bearing this in mind, we can easily map the DTMC time to the actual time (as seen by the tagged device) as follows.

If the DTMC is absorbed into a_i at time t', the time elapsed, in terms of number of BO slots, until the tagged device

TABLE I: List of notations used in the analytical framework

Notation	*Description*
N	Number of nodes in the network cluster
M	Maximum number of transmission attempts
CW	Contention window size
L	Duration of a packet measured in terms of number of BO slots
$a_i,\ i=1,\ldots,M$	Absorbing state, corresponding to the case in which the tagged device managed to transmit at the i-th attempt
a_{M+1}	Absorbing state, corresponding to the case in which the packet was discarded after M unsuccessful attempts
$X_t=1,\ldots,M$	BO stage
$Y_t=1,\ldots,\bar{y}(X_t)$	Number of devices that have already managed to transmit
$\bar{y}(X)=\min(X-1,N-1)$	Maximum number of devices that may have transmitted before BO stage X
$Z_t=1,\ldots,W_{X_t}$	Slot counter of the current BO cycle
W_m	Length of the contention window at BO stage m
$\mathcal{B}_{m,n}$	Set of states of a BO cycle, at BO stage m, with $N-n$ devices that have not been able to transmit yet
$V_m^{(n)}$	Sojourn time in $\mathcal{B}_{m,n}$
$\boldsymbol{t}_i$	Column vector with the transition probabilities from each of the transient states to a_i
$\boldsymbol{T}$	Square matrix that contains the transition probabilities between transient states
$\boldsymbol{\alpha}$	Initial probability vector of the transient states
γ_i	Probability of the tagged device transmitting at the i-th attempt
P_s	Probability of the tagged device transmitting successfully a packet
P_d	Probability of the tagged device discarding a packet
$c_r(m,n)$	Probability that from the m-th BO cycle onward, with n other devices having transmitted successfully over the $m-1$ previous BO cycles, the tagged device will suffer r collisions and eventually make a successful transmission
$\mathbb{E}[C]$	Expected number of collisions
$\mathbb{E}[X_{\text{slots}}^{\text{BO}}]$	Expected number of BO slots for a successful device
$\mathbb{E}[X_{\text{cycles}}^{\text{BO}}]$	Expected number of attempts for a successful device
D	A random variable representing the delay until the tagged device has either successfully transmitted the packet or discarded it
D_{st}	Expected delay experienced by a packet transmitted successfully
E_{tx}	Energy consumption for a successful packet transmission
E_c	Energy consumption for a collided packet transmission
E_{slot}	Energy consumption for an idle BO slot
E_{id}	Energy consumption for an idle transmission slot
E_{st}	Average energy consumed by the tagged device for a successful transmission

successfully finishes its transmission at the i-th attempt is obtained by

$$t = t' + i(L-1), \qquad i = 1,\ldots,M. \tag{1}$$

Similarly, if the DTMC is absorbed into a_{M+1} at time t', the time elapsed until the tagged device discards the packet is

$$t = t' + (M-1)(L-1) + 1. \tag{2}$$

The transient states (i.e., those that are not absorbing) are represented as $\boldsymbol{X}_t = (X_t, Y_t, Z_t)$, where $X_t = 1,\ldots,M$ is the BO stage, $Y_t = 1,\ldots,\bar{y}(X_t)$ is the number of devices that have already managed to transmit, $\bar{y}(X) = \min(X-1, N-1)$ is the maximum number of devices that may have transmitted before BO stage X, $Z_t = 1,\ldots,W_{X_t}$ is the slot counter of the current BO cycle, and W_m is the length of the contention window at BO stage m.

As mentioned above, the duration of a BO cycle in our model comprises the proper BO, which is equal to the shortest BO period selected among all the active devices in the network plus the first slot of the packet transmission phase for the winning device. At each cycle, the tagged device will make a transmission attempt and its BO stage is increased by one.

The states in $\mathcal{B}_{m,n} = \{(m,n,k)\,|\,k=1,\ldots,W_m\}$ are the possible phases of a BO cycle, at BO stage m, with $N-n$ devices that have not been able to transmit yet. A BO will start at state $(m,n,1)$ and will progress through the states $(m,n,2)$, …, (m,n,k_{tx}), where k_{tx} is the first slot in the cycle with a transmission. The chain then exists $\mathcal{B}_{m,n}$ from (m,n,k_{tx}) and moves to a_m (if the tagged device gained access to the channel); to $\mathcal{B}_{m+1,n+1}$ (if access to the channel was obtained by another device); to $\mathcal{B}_{m+1,n}$ (if a collision occurred); or to a_{M+1}, an absorption state corresponding to a collision or other device gaining access to the channel when $m=M$ (i.e., it is the last attempt). Note that there are (at most) M BO cycles and that the absorbing state a_{M+1} bears a different meaning from the other absorbing states, in which the tagged device could not transmit in any of the M BO cycles.

The duration of such a cycle, k_{tx}, (or equivalently, the sojourn time in $\mathcal{B}_{m,n}$) is described by the following random variable

$$V_m^{(n)} = \min_{j=1,\ldots,n} U_j, \tag{3}$$

where $U_1,\ldots,U_n$ and U_0 (to be used later) are independent and identically distributed discrete uniform random variables with support $\{1,\ldots,W_m\}$. Here we are assuming that the devices draw their BO times from a uniform distribution. However, the model could be easily adapted to other BO-time distributions (e.g., geometric, as in ALOHA-like protocols).

The transition matrix of the DTMC has the following form

$$\boldsymbol{P} = \begin{bmatrix} \boldsymbol{T} & \boldsymbol{t}_1 & \cdots & \boldsymbol{t}_{M+1} \\ \boldsymbol{0} & 1 & & \\ \vdots & & \ddots & \\ \boldsymbol{0} & & & 1 \end{bmatrix}, \tag{4}$$

where $\boldsymbol{t}_i$ is a column vector with the transition probabilities from each of the transient states to a_i, and $\boldsymbol{T}$ is a square matrix that contains the transition probabilities among transient states.

The detailed block-structure of $\boldsymbol{T}$ is given by (5) at the top of the next page.

From state (m,n,k), there are four possible transitions corresponding to the four different events that can occur at the k-th slot of the BO procedure:

1) no device starts transmitting, $(m,n,k) \to (m,n,k+1)$; this transition is not possible if $k = W_m$;
2) the tagged device starts transmitting and wins access to the channel (since the other devices have selected larger BO values), $(m,n,k) \to a_m$;
3) any of the $N-n-1$ other devices starts transmitting and wins access to the channel, $(m,n,k) \to (m+1,n+1,1)$, if $m < M$; and $(m,n,k) \to a_{M+1}$, if $m = M$;
4) two or more devices start transmitting and a collision occurs, $(m,n,k) \to (m+1,n,1)$, if $m < M$; and $(m,n,k) \to a_{M+1}$, if $m = M$.

Denote the probabilities of these transitions by $p_0(m,n,k)$, $p_t(m,n,k)$, $p_o(m,n,k)$ and $p_c(m,n,k)$, respectively. They

$$\boldsymbol{T} = \begin{bmatrix}
\boldsymbol{T}_{1,0} & \boldsymbol{C}_{1,0} & \boldsymbol{O}_{1,0} & & & & & & & & & \\
& \boldsymbol{T}_{2,0} & & \boldsymbol{C}_{2,0} & \boldsymbol{O}_{2,0} & & & & & & & \\
& & \boldsymbol{T}_{2,1} & & \boldsymbol{C}_{2,1} & \boldsymbol{O}_{2,1} & & & & & & \\
& & & \boldsymbol{T}_{3,0} & & & & & & & & \\
& & & & \boldsymbol{T}_{3,1} & & & & & & & \\
& & & & & \boldsymbol{T}_{3,2} & & & & & & \\
& & & & & & \ddots & & & & & \\
& & & & & & & \boldsymbol{T}_{M-1,0} & & \boldsymbol{C}_{M-1,0} & \boldsymbol{C}_{M-1,0} & \\
& & & & & & & & \ddots & & \ddots & \ddots \\
& & & & & & & & \boldsymbol{T}_{M-1,\bar{y}(M-1)} & & & \\
& & & & & & & & & \boldsymbol{T}_{M,0} & & \\
& & & & & & & & & & \ddots & \\
& & & & & & & & & & & \boldsymbol{T}_{M,\bar{y}(M)}
\end{bmatrix}, \tag{5}$$

are given as follows:

$$p_0(m,n,k) = \frac{\mathrm{P}\big(V_m^{(N-n)} \geq k+1\big)}{\mathrm{P}\big(V_m^{(N-n)} \geq k\big)} = \frac{(1-k/W_m)^n}{(1-(k-1)/W_m)^n} = \frac{(W_m-k)^n}{(W_m-k+1)^n}, \tag{6}$$

$$p_t(m,n,k) = \frac{\mathrm{P}(U_0 = k, V_m^{(N-n-1)} > k)}{\mathrm{P}(V_m^{(N-n)} \geq k)} = \frac{1/W_m\,(1-k/W_m)^{n-1}}{(1-(k-1)/W_m)^n} = \frac{(W_m-k)^{n-1}}{(W_m-k+1)^n}, \tag{7}$$

$$p_o(m,n,k) = (N-n-1)p_t(m,n,k) = (N-n-1)\frac{(W_m-k)^{n-1}}{(W_m-k+1)^n}, \tag{8}$$

$$p_c(m,n,k) = 1 - p_0(m,n,k) - p_t(m,n,k) - p_o(m,n,k) = 1 - \frac{(W_m-k)^n + (N-n)(W_m-k)^{n-1}}{(W_m-k+1)^n}. \tag{9}$$

Furthermore, we introduce the probabilities $p_{ct}(m,n,k)$ and $p_{co}(m,n,k)$, which will be used later in Subsection V-C2). They are the probabilities of the two options when there is a collision: either the tagged device is involved in the collision,

$$p_{ct}(m,n,k) = \frac{\mathrm{P}(U_0 = k, V_m^{(N-n-1)} = k)}{\mathrm{P}(V_m^{(N-n)} \geq k)} = \frac{(W_m-k+1)^{n-1} - (W_m-k)^{n-1}}{(W_m-k+1)^n}, \tag{10}$$

or only other devices were involved, $p_{co}(m,n,k) = p_c(m,n,k) - p_{ct}(m,n,k)$.

Let us group these probabilities into block vectors as

$$\boldsymbol{t}_{m,n}^x = \begin{bmatrix} p_x(m,n,1) \\ \vdots \\ p_x(m,n,W_m) \end{bmatrix}, \tag{11}$$

where x is a placeholder for 't' (tagged), 'o' (other), 'c' (collision), 'ct' (collision tagged), or 'co' (collision others).

Now, we can give the expressions of the blocks of $\boldsymbol{T}$, and of the vectors $\boldsymbol{t}_1, \ldots, \boldsymbol{t}_{M+1}$ introduced in (4) as follows

$$\boldsymbol{T}_{m,n} = \begin{bmatrix} 0 & p_0(m,n,1) & & \\ & \ddots & \ddots & \\ & & 0 & p_0(m,n,W_m-1) \\ & & & 0 \end{bmatrix}, \tag{12}$$

$$\boldsymbol{C}_{m,n} = \begin{bmatrix} \boldsymbol{t}_{m,n}^c & \boldsymbol{0} & \cdots & \boldsymbol{0} \end{bmatrix}, \tag{13}$$

$$\boldsymbol{O}_{m,n} = \begin{bmatrix} \boldsymbol{t}_{m,n}^o & \boldsymbol{0} & \cdots & \boldsymbol{0} \end{bmatrix}, \tag{14}$$

$$\boldsymbol{t}_i = \begin{bmatrix} \boldsymbol{0} \\ \vdots \\ \boldsymbol{0} \\ \boldsymbol{t}_{i,0}^t \\ \vdots \\ \boldsymbol{t}_{i,\bar{y}(i)}^t \\ \boldsymbol{0} \\ \vdots \\ \boldsymbol{0} \end{bmatrix}, \quad \boldsymbol{t}_{M+1} = \begin{bmatrix} \boldsymbol{0} \\ \vdots \\ \vdots \\ \vdots \\ \boldsymbol{0} \\ \boldsymbol{t}_{M,0}^o + \boldsymbol{t}_{M,0}^c \\ \vdots \\ \boldsymbol{t}_{M,\bar{y}(M)+\boldsymbol{t}_{M,\bar{y}(M)}^c}^o \end{bmatrix}, \tag{15}$$

where $i = 1, \ldots, M$, and $\boldsymbol{0}$ denotes a column vector of zeros of an appropriate size.

B. An Example with Three End Devices

Before we continue with performance analysis, let us present a simple example of the model by considering a small-scale cluster as shown in Fig. 2 with $N = 3$ devices (the tagged device plus two other devices). The maximum number of transmission attempts is configured as $M = 2$, and the contention window sizes are set as $W_1 = 2$ and $W_2 = 4$ for the first and second transmission attempt, respectively.

Accordingly, there are three absorbing states (a_1, a_2, a_3). The absorption of the DTMC into each of them represents the three possible outcomes of a transmission round: a_1, the tagged device manages to transmit successfully at the first attempt (i.e., it is the unique device that selected the minimum

BO interval in the first BO cycle); a_2, the tagged device manages to transmit successfully in its second attempt; a_3, the tagged device was not able to transmit successfully neither in the first nor in the second attempt.

Then, there are $W_1 + 2W_2 = 10$ transient states that are organized into three blocks. That is, $\mathcal{B}_{1,0} = \{(1,0,1),\ (1,0,2)\}$, $\mathcal{B}_{2,0} = \{(2,0,1),\ (2,0,2),\ (2,0,3),\ (2,0,4)\}$, and $\mathcal{B}_{2,1} = \{(2,1,1),\ (2,1,2),\ (2,1,3),\ (2,1,4)\}$. The states in $\mathcal{B}_{1,0}$ represent the two possible phases of the first BO cycle. Similarly, the states in $\mathcal{B}_{2,0}$ or $\mathcal{B}_{2,1}$ represent the four possible phases of the second BO cycle. The matrices $\boldsymbol{T}_{1,0}$, $\boldsymbol{T}_{2,0}$ and $\boldsymbol{T}_{2,1}$ contain the transition probabilities between the states inside each block.

The DTMC starts at the first state in $\mathcal{B}_{1,0}$ and will leave that block of states because one of the following three events happens: a transition into the $\mathcal{B}_{2,0}$ block (a collision occurs); a transition into the $\mathcal{B}_{2,1}$ block (one of the two non-tagged devices manages to transmit successfully); or absorption into state a_1 (the tagged device manages to transmit successfully). The transition probabilities for each of these three outcomes are the entries of $\boldsymbol{C}_{1,0}$, $\boldsymbol{O}_{1,0}$, and $\boldsymbol{t}^t_{1,0}$, respectively.

Now, consider that the DTMC is in block $\mathcal{B}_{2,1}$, which means that it is at the second BO cycle and one of the non-tagged devices transmitted successfully in the first BO cycle. Since the maximum number of attempts (i.e., BO cycles) is $M = 2$, there are only two possible outcomes for the DTMC to abandon $\mathcal{B}_{2,1}$: absorption into state a_2 (the tagged device manages to transmit successfully) or absorption into state a_3 (the tagged also fails in the second, which is also its last, attempt, and another device wins the contention or a collision occurs). The transition probabilities for each of these two possible outcomes are the entries of $\boldsymbol{t}^t_{2,1}$ and $\boldsymbol{t}^o_{2,1} + \boldsymbol{t}^c_{2,1}$, respectively.

Correspondingly, the transition matrix of the DTMC is given as

$$\boldsymbol{P} = \left[\begin{array}{ccc:ccc} \boldsymbol{T}_{1,0} & \boldsymbol{C}_{1,0} & \boldsymbol{O}_{1,0} & \boldsymbol{t}^t_{1,0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{T}_{2,0} & \mathbf{0} & \mathbf{0} & \boldsymbol{t}^t_{2,0} & \boldsymbol{t}^o_{2,0} + \boldsymbol{t}^c_{2,0} \\ \mathbf{0} & \mathbf{0} & \boldsymbol{T}_{2,1} & \mathbf{0} & \boldsymbol{t}^t_{2,1} & \boldsymbol{t}^o_{2,1} + \boldsymbol{t}^c_{2,1} \\ \hdashline \mathbf{0} & \mathbf{0} & \mathbf{0} & 1 & 0 & 0 \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & 0 & 1 & 0 \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & 0 & 0 & 1 \end{array}\right], \quad (16)$$

where the first three block rows, and block columns, are associated with the state blocks $\mathcal{B}_{1,0}$, $\mathcal{B}_{2,0}$ and $\mathcal{B}_{2,1}$ respectively; and the last three rows (and columns) are associated with the absorbing states a_1, a_2, and a_3.

The entries of the block matrices corresponding to the transitions between transient states (the upper-left part of $\boldsymbol{P}$) are obtained as follows:

$$\boldsymbol{T}_{1,0} = \begin{bmatrix} 0 & p_0(1,0,1) \\ 0 & 0 \end{bmatrix}, \ \boldsymbol{C}_{1,0} = \begin{bmatrix} p_c(1,0,1) & 0 & 0 & 0 \\ p_c(1,0,2) & 0 & 0 & 0 \end{bmatrix},$$

$$\boldsymbol{O}_{1,0} = \begin{bmatrix} p_o(1,0,1) & 0 & 0 & 0 \\ p_o(1,0,2) & 0 & 0 & 0 \end{bmatrix}. \quad (17)$$

$$\boldsymbol{T}_{2,0} = \begin{bmatrix} 0 & p_0(2,0,1) & 0 & 0 \\ 0 & 0 & p_0(2,0,1) & 0 \\ 0 & 0 & 0 & p_0(2,0,1) \\ 0 & 0 & 0 & 0 \end{bmatrix},$$

$$\boldsymbol{T}_{2,1} = \begin{bmatrix} 0 & p_0(2,1,1) & 0 & 0 \\ 0 & 0 & p_0(2,1,1) & 0 \\ 0 & 0 & 0 & p_0(2,1,1) \\ 0 & 0 & 0 & 0 \end{bmatrix}. \quad (18)$$

Finally, $\boldsymbol{t}^t_{1,0}, \boldsymbol{t}^t_{2,0}, \boldsymbol{t}^o_{2,0}, \boldsymbol{t}^c_{2,0}, \boldsymbol{t}^t_{2,1}, \boldsymbol{t}^o_{2,1}, \boldsymbol{t}^c_{2,1}$ are column vectors whose general form is given in (11).

C. Performance Analysis

The metrics that we consider in the performance analysis are the following: 1) distribution of successful and discarded transmissions, as well as number of BO cycles prior to a successful transmission; 2) distribution of the number of collisions; 3) access delay distribution; and 4) energy consumption. These metrics are defined and obtained as follows.

1) Distribution of successful and discarded transmissions, and number of BO cycles for a successful transmission: Let $\boldsymbol{\alpha} = [1\ 0\ \cdots\ 0]$ be the initial probability vector of the transient states. Then, the probability of absorbing into a_i at time t is given by

$$\gamma_i(t) = \boldsymbol{\alpha}\boldsymbol{T}^{t-1}\boldsymbol{t}_i, \qquad i = 1, \ldots, M+1; \quad t \geq 1. \quad (19)$$

Furthermore, it is evident that

$$\gamma_i(t) = 0, \qquad i = 1, \ldots, M+1; \quad t \leq 0. \quad (20)$$

From here, we obtain the probability that the tagged device manages to transmit at the i-th attempt as

$$\gamma_i = \sum_{t \geq 1} \gamma_i(t) = \boldsymbol{\alpha}(\boldsymbol{I} - \boldsymbol{T})^{-1}\boldsymbol{t}_i, \qquad i = 1, \ldots, M; \quad (21)$$

the probability of the tagged device discarding the packet as

$$P_d = \boldsymbol{\alpha}(\boldsymbol{I} - \boldsymbol{T})^{-1}\boldsymbol{t}_{M+1}, \quad (22)$$

and the probability of successful transmission as

$$P_s = \sum_{i=1}^{M} \gamma_i = \boldsymbol{\alpha}(\boldsymbol{I} - \boldsymbol{T})^{-1} \sum_{i=1}^{M} \boldsymbol{t}_i = 1 - P_d, \quad (23)$$

where $\boldsymbol{I}$ denotes the identity matrix of an appropriate size.

Using (21), we can write the mean number of attempts (or BO cycles) for a successful device as

$$\mathbb{E}[X^{\text{BO}}_{\text{cycles}}] = \frac{1}{P_s} \sum_{i=1}^{M} i\gamma_i = \frac{\boldsymbol{\alpha}(\boldsymbol{I} - \boldsymbol{T})^{-1} \sum_{i=1}^{M} i\boldsymbol{t}_i}{\boldsymbol{\alpha}(\boldsymbol{I} - \boldsymbol{T})^{-1} \sum_{i=1}^{M} \boldsymbol{t}_i}. \quad (24)$$

The time to absorption measured in number of slots is equal to the number of BO slots plus one transmission slot per BO cycle. Thus, the mean number of BO slots to absorption can be obtained as

$$\begin{aligned} \mathbb{E}[X^{\text{BO}}_{\text{slots}}] &= \frac{1}{P_s}\boldsymbol{\alpha}(\boldsymbol{I} - \boldsymbol{T})^{-2} \sum_{i=1}^{M} \boldsymbol{t}_i - \mathbb{E}[X^{\text{BO}}_{\text{cycles}}] \\ &= \frac{\boldsymbol{\alpha}(\boldsymbol{I} - \boldsymbol{T})^{-1} \left((\boldsymbol{I} - \boldsymbol{T})^{-1} \sum_{i=1}^{M} \boldsymbol{t}_i - \sum_{i=1}^{M} i\boldsymbol{t}_i \right)}{\boldsymbol{\alpha}(\boldsymbol{I} - \boldsymbol{T})^{-1} \sum_{i=1}^{M} \boldsymbol{t}_i}. \end{aligned} \quad (25)$$

2) Distribution of the number of collisions: Herein, we derive the distribution of the total number of collisions for a device that manages to transmit successfully in the end.

Denote by $c_r(m,n)$ the probability that from the m-th BO cycle onward, with n other devices having transmitted successfully over the $m-1$ previous BO cycles, the tagged device will suffer r collisions and eventually make a successful transmission. The distribution of the number of collisions for a successful node, $\{c_0, c_1, \ldots, c_{M-1}\}$, is simply given by

$$\mathrm{P}(C=r) = \frac{c_r(1,0)}{P_s}, \qquad r = 0, \ldots, M-1. \tag{26}$$

Before we provide a recursive method to compute $c_r(1,0)$, we need first to introduce further notation and results.

It is not difficult to show that in the m-th BO cycle, with $N-n$ devices trying to transmit, the probability of outcome $x \in \{$ 't', 'o', 'c', 'ct', 'co' $\}$ is

$$p_x(m,n) = [1\ 0\ \cdots\ 0](\boldsymbol{I} - \boldsymbol{T}_{m,n})^{-1}\boldsymbol{t}^x_{m,n}. \tag{27}$$

Now, it is easy to check that the following recursion applies for $m = 1, \ldots, M-1$ and $n = 0, \ldots, \min(M-1, N-2)$:

$$\begin{aligned} c_0(m,n) = p_t(m,n) + p_o(m,n)c_0(m+1,n+1) \\ + p_{co}(m,n)c_0(m+1,n), \end{aligned} \tag{28}$$

$$\begin{aligned} c_r(m,n) = p_o(m,n)c_r(m+1,n+1) \\ + p_{co}(m,n)c_r(m+1,n) + p_{ct}(m,n)c_{r-1}(m+1,n), \\ r = 1, \ldots, M-m; \end{aligned} \tag{29}$$

and the initial conditions are:

$$c_0(m, N-1) = 1, \qquad m \geq N; \tag{30}$$

$$c_0(M,n) = p_t(M,n), \qquad n \leq \min(M-1, N-2). \tag{31}$$

3) Delay distributions: Let D be a random variable that represents the delay until the tagged device has either successfully transmitted the packet or discarded it. Then,

$$\mathrm{P}(D=t) = \sum_{i=1}^{M} \gamma_i\big(t - i(L-1)\big) + \gamma_{M+1}\big(t - (M-1)(L-1) + 1\big). \tag{32}$$

The delay distributions conditioned on the tagged device transmitting successfully and discarding the packet are given respectively as

$$\mathrm{P}_s(D=t) = \frac{1}{P_s}\sum_{i=1}^{M} \gamma_i\big(t - i(L-1)\big) \tag{33}$$

and

$$\mathrm{P}_d(D=t) = \frac{1}{P_d}\gamma_{M+1}\big(t - (M-1)(L-1) + 1\big). \tag{34}$$

From the deduced delay distribution, we can obtain the average access delay, which is meaningful for successfully transmitted packets. The expected access delay, denoted by D_{st}, is defined as the average duration from the instant when the first bit of a WuC message sent by the data collector arrives at the WuRx until the moment when the data packet is successfully transmitted from the MR and an ACK is received. Accordingly, D_{st} can be calculated as

$$D_{st} = T_{wuc} + \mathbb{E}[X^{\mathrm{BO}}_{\mathrm{cycles}}]T_t + \mathbb{E}[X^{\mathrm{BO}}_{\mathrm{slots}}]T_{slot}, \tag{35}$$

TABLE II: Network Parameter Configurations [7] [19] [22]

Radio type	*Parameter*	*Value*	*Unit*
Common	Supply voltage	3	V
WuR device (Main radio)	Data rate	250	kbps
	CCA current	20.28	mA
	CCA duration	128	μs
	Transmission current	17.4	mA
	BO current	5.16	mA
	Reception current	18.8	mA
	Idle current	20	μA
	SIFS duration	192	μs
	Payload size	35	bytes
	Slot duration	320	μs
	Contention window size	16, 32	slots
	Maximum transmission attempts (M)	7, 4, 29	times
WuR device (WuRx)	WuC reception current (light sleep)	8	μA
	Deep sleep current	3.5	μA
	MCU switching current	2.7	μA
	Time to switch on MCU	1.79	ms
Data Collector (e.g., UAV)	WuC duration	12.2	ms
	WuC address length	16	bits
	WuC transmission current	152	mA
	ACK frame size	11	bytes
	Reception current	18.8	mA
	SIFS duration	192	μs

where T_{wuc}, $\mathbb{E}[X^{\mathrm{BO}}_{\mathrm{cycles}}]$, and $\mathbb{E}[X^{\mathrm{BO}}_{\mathrm{slots}}]$ are WuC duration, average number of BO cycles, and average number of BO slots; T_t and T_{slot} are the expected delays from the time a device wins a competition until the packet is successfully received by the data collector (including the reception of ACK) and an idle slot-time, respectively. Furthermore, $\mathbb{E}[X^{\mathrm{BO}}_{\mathrm{cycles}}]$ and $\mathbb{E}[X^{\mathrm{BO}}_{\mathrm{slots}}]$ can be obtained by (24) and (25), respectively. T_t is given by

$$T_t = T_{MST} + T_{data} + T_{SIFS} + T_{ack}, \tag{36}$$

where T_{MST}, T_{data}, T_{SIFS}, and T_{ack} are the time needed for fully activation of MCU, data packet transmission, short inter-frame space (SIFS), and ACK frame transmission, respectively. Note that, in the above calculations, D is a random variable and it does not include T_{wuc} whereas D_{st} represents a mean value including all delays. The other delays in (36) will be included in the calculation of D_{st} in (35) considering that $L = T_t/T_{slot}$.

4) Energy consumption: Denoted by E_{st}, it represents the average energy consumption required by a device to successfully transmit a packet. E_{st} is calculated as the average energy consumed by the tagged device from the moment its MR wakes up from deep sleep upon arrival of a WuC until a data packet is successfully delivered and an ACK is received. E_{st} can be calculated as

$$\begin{aligned} E_{st} = \mathbb{E}[X^{\mathrm{BO}}_{\mathrm{slots}}]E_{slot} + E_{tx} + \mathbb{E}[C]E_c + \\ (\mathbb{E}[X^{\mathrm{BO}}_{\mathrm{cycles}}] - \mathbb{E}[C] - 1)E_{id}, \end{aligned} \tag{37}$$

where E_{tx}, E_c, E_{slot}, and E_{id} are the energy consumption for a successful packet transmission, collided packet transmission, for an idle BO slot, and for an idle transmission slot, respectively. The mean number of collisions, $\mathbb{E}[C]$, can be obtained from the distribution of the number of collisions presented in (26). E_{tx} and E_c can be respectively obtained by

$$E_{tx} = E_{MST} + E_{data} + E_{SIFS} + E_{ack},$$
$$E_c = E_{MST} + E_{data} + E_{SIFS} + E_{out},$$

TABLE III: Analytical vs. Simulation Results

CW	N	P_s		Avg. BO slots		Avg. # attempts	
		Anal.	Sim.	Anal.	Sim.	Anal.	Sim.
16	08	0.730	0.734	7.455	7.512	4.110	4.120
16	10	0.543	0.549	5.199	5.292	4.105	4.152
16	12	0.420	0.421	3.883	3.892	4.100	4.098
16	14	0.334	0.337	3.018	2.992	4.095	4.071
16	16	0.270	0.269	2.407	2.366	4.09	4.086
16	18	0.222	0.221	1.955	1.972	4.085	4.086
16	20	0.184	0.185	1.610	1.644	4.08	4.113
32	08	0.804	0.805	17.320	17.277	4.059	4.039
32	10	0.622	0.622	12.558	12.543	4.058	4.045
32	12	0.501	0.504	9.770	9.853	4.056	4.081
32	14	0.415	0.415	7.917	7.824	4.055	4.034
32	16	0.350	0.352	6.591	6.510	4.054	4.011
32	18	0.301	0.302	5.595	5.526	4.052	4.031
32	20	0.261	0.261	4.819	4.806	4.051	4.031

where E_{data}, E_{MST}, E_{SIFS}, E_{ack}, and E_{out} are the energy consumption for data packet transmission, fully activation of MCU, SIFS, ACK reception, and ACKTimeout, respectively.

VI. Simulations and Numerical Results

In this section, we first validate the mathematical model by comparing the results obtained from the analytical framework with that obtained from simulations. Then, the performance of the MURIST protocol is evaluated by considering a variable number of IoT devices in the considered network cluster. We also compare the performance of MURIST with that of CSMA wake-up radio (CSMA-WuR) [12] and backoff enabled sub-carrier modulation wake-up radio (BoWuR) [18], as well as with *unicast* based data collection. For the sake of simulation simplicity, we keep the contention window size identical for both initial and retransmission attempts. Note nevertheless that the DTMC model is generic and it applies to both identical and variable window sizes.

A. Simulation Setup and Model Validation

Consider a WuR enabled IoT network cluster operated in the RI mode. A data collection round is initiated by a data collector (e.g., a UAV) by sending a multicast WuC to all devices in the cluster. Upon decoding and validating the multicast WuC, all devices simultaneously start the BO and data transmission procedure following the MURIST protocol. In the meantime, each IoT device maintains a timestamp from the instant it receives the WuC in order to track the status of the current packet transmission. The timestamp expires either with a successful transmission or an unsuccessful transmission when the maximum number of transmission attempts (i.e., M) has been reached.

To verify the accuracy of the model, we developed a custom-built discrete-event simulator in MATLAB which mimics the behavior of the studied protocols. Simulations were carried out for a network cluster with the number of devices varying in the range of $N \in \{4, 8, 10, \ldots, 20\}$. Unless otherwise stated, the network and device parameters are configured according to Table II. It is worth highlighting that the simulation results are independent of the analytical framework presented in the previous section. On the other hand, to investigate the performance of MURIST when multiple clusters are involved is beyond the scope of this paper as it requires designing a device clustering algorithm [17] and UAV flying trajectory planning.

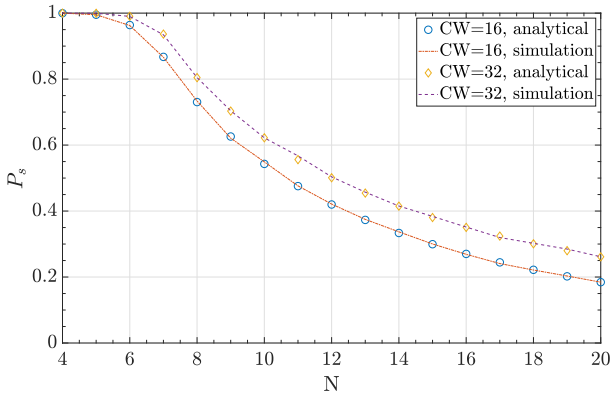

Fig. 4: MURIST: Successful transmission probability when the number of devices, N, varies in the network.

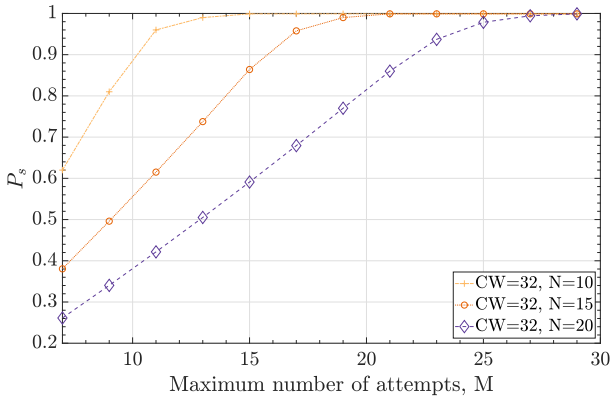

Fig. 5: MURIST: Successful transmission probability as the maximum number of transmission attempts, M, varies.

Let us first compare the analytical and simulation results presented jointly in Table III with three parameters. They are, successful transmission probability P_s, average total number of BO slots before a successful transmission, and average number of transmission attempts which is equal to the number of BO cycles required by a device to achieve a successful transmission. For the results shown in Table III, the maximum number of transmission attempts is configured as $M = 7$. Moreover, the curves in Figs. 4 and 8 include the values obtained from both the analytical framework and simulations. From the values shown in the table and the curves depicted in these two figures, it is evident that the analytical and simulation results match precisely with each other. Therefore, the accuracy of the developed Markov model is validated. For the sake of illustration clarity, we do not plot the curves obtained from both methods in subsequent figures.

B. Performance of MURIST

1) Successful transmission probability: The obtained probability of successful transmissions for MURIST, P_s, as the number of devices varies in the network is illustrated in Fig. 4 with $M = 7$. As can be observed, the network experiences a lower P_s with a larger number of devices, i.e., when N increases. This is because more collisions occur when the device population becomes larger in the network. On the other

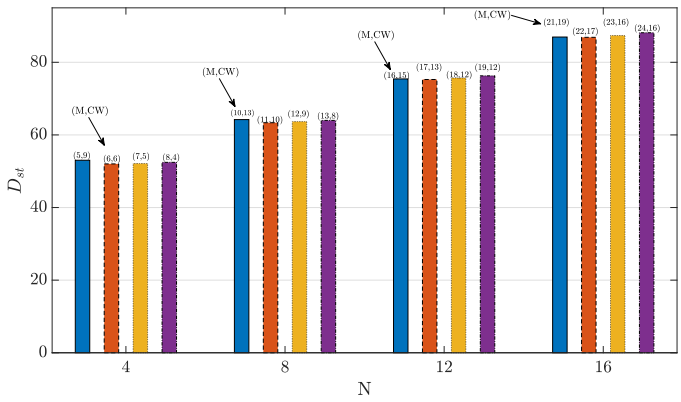

Fig. 6: MURIST: Average access delay $E[D]$ for achieving $P_s \geq 0.95$ with proper CW sizes and M values, for four different cluster sizes.

hand, a higher P_s is achieved with a larger CW size. This is because a larger CW size would give devices a wider range of values for their BO slot selections, leading to a lower collision probability in each BO cycle.

Furthermore, let us explore how reliable transmission could be improved via appropriate parameter configurations while keeping the performance of access delay in mind. To investigate this behavior, we reconfigure M to larger values for another set of numerical evaluation. Fig. 5 illustrates the trend of P_s for different maximum number of attempts for packet transmissions. It is clear from the figure that P_s increases significantly with M, and it can reach almost 100% successful transmission probability when M is sufficiently large. In general, to achieve $P_s \geq 80\%$, M should be configured to a value that is greater than N, as observed in Fig. 5. On the other hand, a larger M leads to a longer access delay since more transmission attempts are allowed. As such, there is a tradeoff between reliable transmissions and delay.

2) Further discussions on proper parameter configuration: Configuring network parameters to proper values is scenario or service dependent. For delay-tolerant services where high-reliable transmissions are vital, larger values for the maximum number of transmission attempts M and contention window size CW with respect to network cluster size should be configured. On the other hand, services with low-latency requirements could be provided by adopting smaller values for M or/and CW. As a rule of thumb, proper configuration of M and CW values should be jointly considered based on the service requirement.

As an example, let us consider a high level of reliable transmission targeted at $P_s = 95\%$. We first determine an optimal value for CW and M from a set of values that yield the target $P_s \geq 0.95$. To do so, we identify the minimum value of CW for a given M that leads to a P_s equal to or greater than 0.95, i.e., $CW(M) = \min\{CW | P_s(CW, M) \geq 0.95\}$, for a given cluster population N. Then, we assess the expected access delay that meets the target reliable transmission requirement for different network cluster sizes. Fig. 6 illustrates the expected access delay in terms of number of slots for $P_s \geq 0.95$, given four cluster sizes as $N = \{4, 8, 12, 16\}$. For $N = 8$, for instance, M and CW should be configured as $(M, CW) = (10, 13), (11, 10), (12, 9)$, or $(13, 8)$, achieving a reliable transmission level of $P_s = 95.288\%, 95.395\%, 96.659\%$, or 97.174%, respectively.

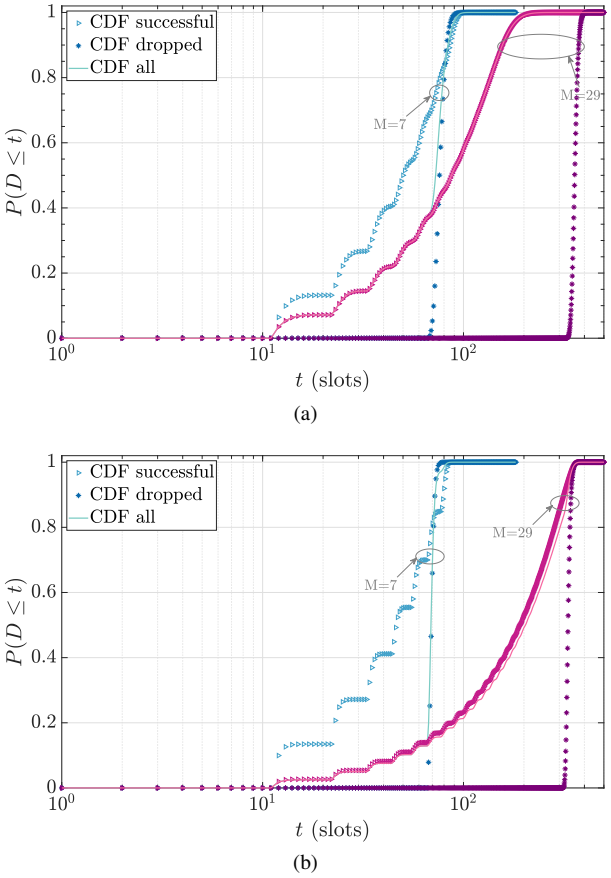

Fig. 7: MURIST: Delay CDF for two sets of device population and different re-transmission limits; $CW = 16$; and $L = 11$ slots. (a) $N = 10$, $M \in \{7, 29\}$. (b) $N = 20$, $M \in \{7, 29\}$.

Accordingly, the achieved average access delay for these four cluster sizes is approximately $52, 63, 75, 87$ slots $\times$ 320 μs/slot $= 16.6, 20.2, 24.0, 51.8$ ms, respectively. In general, access delay will increase monotonically as the number of devices in a cluster increases. Therefore, for a large-scale network consisting of potentially many clusters, how to partition the network into clusters with proper sizes appears as an interesting task. However, it is beyond the scope of this paper.

3) Access delay distributions: In Fig. 7, we illustrate the delay performance represented by the cumulative distribution function (CDF) of the access delay with two different configurations: (a) $N = 10$ and (b) $N = 20$, respectively, both with $M = \{7, 29\}$. The reason we set $M = 29$ is that this value leads to 100% successful transmission probabilities for all configured device populations, as shown in Fig. 5. However, this high reliability is achieved at the cost of longer delays when comparing the delay distribution with that of $M = 7$. This is due to the fact that devices in the network obtain more transmission opportunities when M is larger and that induces longer delay, i.e., a packet remains longer in the buffer before it is successfully transmitted. As observed, this behavior is in accordance with the tradeoff mentioned above.

To further explore the delay performance, let us observe the CDF curves depicted in Figs. 7(a) and 7(b). For a lower

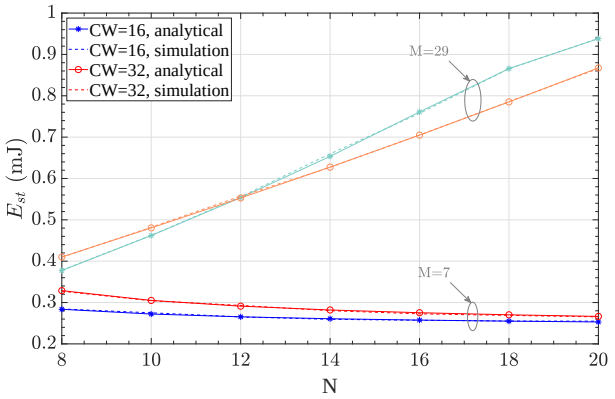

Fig. 8: MURIST: Energy Consumption when the number of devices, N, varies in the network.

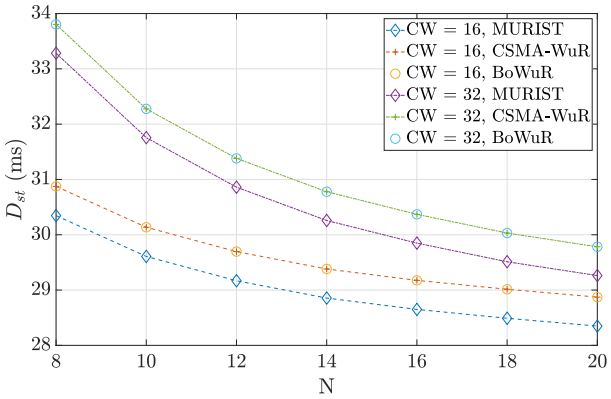

Fig. 9: MURIST vs. BoWuR vs. CSMA-WuR: Access delay comparison when the number of devices, N, varies in the network.

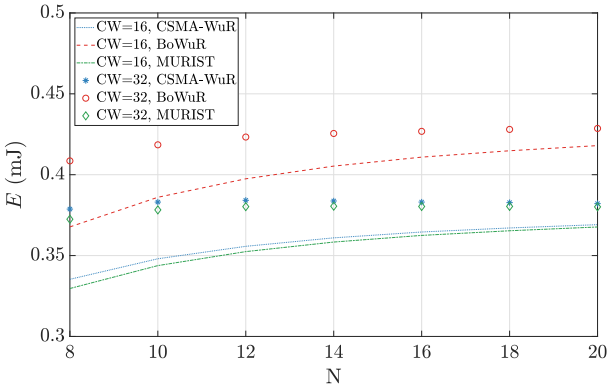

Fig. 10: MURIST vs. BoWuR vs. CSMA-WuR: Energy Consumption comparison when the number of devices, N, varies in the network.

value of M (i.e., $M = 7$), a higher percentage of devices will perform successful transmissions requiring fewer slots when the network is larger (i.e., $N = 20$), compared with a small network when $N = 10$. This is because a larger network with a lower M experiences higher packet loss as devices obtain fewer transmission opportunities. The packets in network configurations where $M < N$ experience shorter delays for successful transmissions (i.e., fewer BO slots are required for higher values of N when $M = 7$ as evident in Table III).

4) Energy consumption: With respect to energy consumption *per device*, we can observe in Fig. 8 that it increases as the number of devices in the cluster grows when $M = 29$. That means, for a higher M, a packet stays longer in the buffer before it gets transmitted. Thus, it consumes more energy. On the other hand, for lower values of M, energy consumption decreases slightly as the number of devices in the cluster grows. This is due to the fact mentioned above, i.e., a larger network with a lower M experiences higher packet loss. That means, the average number of BO slots required per successful transmission decreases as the device population becomes larger (evident from Table III) when M is smaller. This decrement is caused by the fact that the value of M is configured to be relatively low for a given population N. With less retransmission opportunities, higher packet loss is expected. However, for those 'lucky' devices which transmit their packets successfully, this result implies that they have selected a lower value of BO slots.

Furthermore, the larger the value of the CW, the higher the energy consumption. This is because a larger CW implies that, on average, more BO slots are selected for a successful transmission. As a device needs to assess the transmission status of other devices by checking channel occupancy in each BO counting-down step, more energy is consumed with a larger CW.

C. Performance Comparison with CSMA-WuR and BoWuR

CSMA-WuR and BoWuR are two MAC protocols for WuR enabled IoT networks proposed in [12] and [18], respectively. A common feature of these two protocols and MURIST is that a BO procedure is needed in all these three protocols. However, distinctions exist. While BoWuR performs CCA and BO prior to data transmission, CSMA-WuR follows BO first and then CCA before data transmission. On the other hand, MURIST also performs BO before data transmission but checks channel status via energy detection.

Fig. 9 illustrates the average access delay obtained from these three protocols, where the analytical values are obtained from (35). As CSMA-WuR and BoWuR perform the same procedure except CCA at the beginning of a data transmission, the access delay curves from them overlap (with a minor difference of 128 μs which is the CCA duration and it is not visible in Fig. 9). On the other hand, the data transmission delay for MURIST has been reduced by 18.35% on average compared with that of CSMA-WuR and BoWuR when $CW = 16$. A similar trend is observed for $CW = 32$. The reason for this behavior is that the devised MURIST protocol requires a lower number of attempts for each successful packet transmission. More specifically, all devices in MURIST wake up at the same time and a device goes back to the light sleep mode once it observes the transmission of another device during each BO slot. On the other hand, a device in CSMA-WuR has to wait until its BO counter reaches zero before it performs CCA and goes back to sleep. From Fig. 9, it is also clear that configuring a higher value for CW leads to a longer data transmission delay. The observed longer delay is due to the increment of the number of BO slots with a larger CW size. Moreover, for a given CW size, a larger N means more competitors for each transmission attempt. This implies that these 'lucky' devices have selected smaller BO duration, leading to a slightly lower D_{st} on average as shown in Fig. 9.

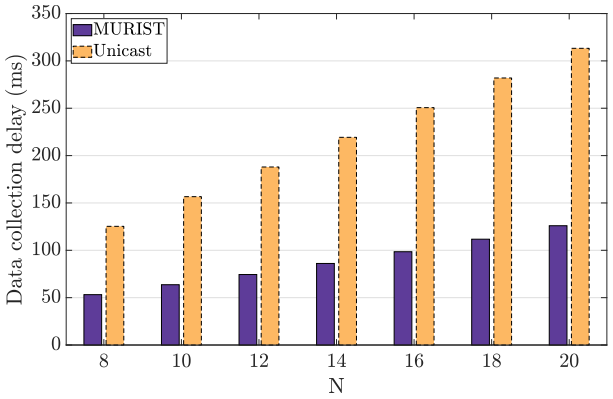

Fig. 11: MURIST vs. unicast: Data collection delay when the number of devices, N, varies.

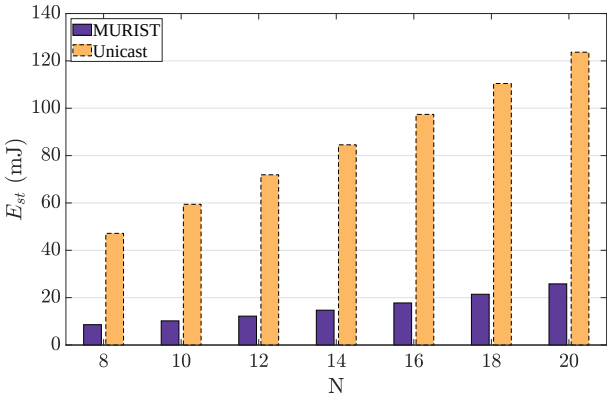

Fig. 12: MURIST vs. unicast: Energy Consumption of the network when the number of devices, N, varies.

Furthermore, we illustrate the average energy consumption for the studied three protocols in Fig. 10. For both $CW = \{16, 32\}$, the energy consumption by MURIST is slightly lower than that of CSMA-WuR, whereas much higher energy is consumed by BoWuR. This significant improvement is achieved thanks to the feature of MURIST that allows devices go back to sleep immediately when observing the transmission of another device. Moreover, it is observed that the average energy consumption increases with the network size for all the studied protocols.

D. Performance Comparison: Multicast vs. Unicast

Finally, we compare the performance of MURIST versus unicast based data collection. For unicast based operation, the data collector sends a unicast WuC to each specific device, identified by its unique WuC address. When a unicast WuC is emitted by the data collector, all devices start to decode and validate the WuC address following the bit-by-bit decoding principle [19]. The unintended devices will then go back to sleep once a mismatch of any bit is detected. When the address of the received WuC matches its own address, the target device wakes up its MR for data transmission. Since only one device is waked up by each WuC, there is no collision for data transmission. Still, an ACK is kept as it indicates the time instant when either the MR can go back to sleep again or a retransmission is needed if no ACK is received after ACKTimeout. For one round of unicast based data collection, the required number of WuC (downlink) and packet (uplink) transmissions follows the unicast data collection principle presented in Subsection IV-A.

However, the merit of collision-free transmission is achieved at a cost of both longer data collection delay and higher energy consumption of the network, as illustrated in Figs. 11 and 12. By *data collection delay*, it is meant the total amount of time used for one round of data collection, i.e., the accumulated access delay for a cluster of devices. The reason that a unicast based data collection round takes a longer time for a cluster of devices is obvious. While each transmission round in unicast consists of the transmission of one WuC, one Data, and one ACK for each packet, a MURIST based data collection round consists of the transmission of multiple Data (including BO and ACK) transmissions plus *only one* common WuC transmission *for the whole cluster of devices. When comparing the trend of access delay (shown in Fig. 9) and of data collection delay (shown in Fig. 11) as cluster size varies, one may notice that D_{st} decreases while data collection delay increases with a larger N. This is because a longer time is needed to collect data from a larger cluster although it takes a slightly shorter time on average to collect data from each individual device.*

With respect to the average energy consumption of a network cluster with various device populations, the advantage of adopting MURIST is convincing, as illustrated in Fig. 12. Again, this benefit is brought by the procedure of eliminating $N - 1$ WuCs in a network. The larger the cluster size, the higher the benefit.

VII. Conclusions and Future Work

In this paper, we proposed a multicast based data collection protocol for WuR enabled IoT networks with a procedure triggered by a common WuC to wake up a cluster of devices to initiate contention-based synchronous packet transmissions in a distributed manner. The devised analytical framework based on an absorbing Markov chain captures the behavior of the cluster of devices operated in the synchronous mode. Through extensive simulations and quantitatively comparison with two other existing WuR protocols as well as with unicast based transmissions, we demonstrate the accuracy of the developed model and the superiority of the proposed protocol over its counterparts in terms of both data transmission latency and energy consumption. To achieve ideal performance for a network with expected service requirements, the tunable parameters of the protocol need to be properly configured. As our future work, we plan to integrate error-prone channel conditions into our analytical framework and to implement the devised protocol in a prototype testbed.

References

[1] 3GPP TR 38.913, "Study on scenarios and requirements for next generation access technologies," R16, v16.0.0, Jun. 2020.
[2] 3GPP TS 22.104, "Service requirements for cyber-physical control applications in vertical domains," R17, v17.4.0, Sep. 2020.
[3] N. H. Motlagh, M. Bagaa, and T. Taleb, "UAV-based IoT platform: A crowd surveillance use case," *IEEE Commun. Mag.*, vol. 55, no. 2, pp. 128–134, Feb. 2017.
[4] T. Long, M. Ozger, O. Cetinkaya, and O. B. Akan, "Energy neutral Internet of drones," *IEEE Commun. Mag.*, vol. 56, no. 1, pp. 22–28, Jan. 2018.

[5] ITU-R M.2410-0, "Minimum requirements related to technical performance for IMT-2020 radio interface(s)," Nov. 2017.

[6] F. Adelantado, X. Vilajosana, P. Tuset-Peiro, B. Martinez, J. Melià-Seguí, and T. Watteyne, "Understanding the limits of LoRaWAN," *IEEE Commun. Mag.*, vol. 55, no. 9, pp. 34–40, Sep. 2017.

[7] J. Oller, I. Demirkol, J. Casademont, J. Paradells, G. U. Gamm, and L. Reindl, "Has time come to switch from duty-cycled MAC protocols to wake-up radio for wireless sensor networks?" *IEEE/ACM Trans. Netw.*, vol. 24, no. 2, pp. 674–687, Apr. 2016.

[8] A. Frøytlog, T. Foss, O. Bakker, G. Jevne, M. A. Haglund, F. Y. Li, J. Oller, and G. Y. Li, "Ultra-low power wake-up radio for 5G IoT," *IEEE Commun. Mag.*, vol. 57, no. 3, pp. 111–117, Mar. 2019.

[9] M. Ghribi and A. Meddeb, "Survey and taxonomy of MAC, routing and cross layer protocols using wake-up radio," *J. Netw. & Comput. Appl.*, Elsevier, vol. 149, article 102465, pp. 1–24, Jan. 2020.

[10] F. Z. Djiroun and D. Djenouri, "MAC protocols with wake-up radio for wireless sensor networks: A review," *IEEE Commun. Surveys & Tuts.*, vol. 19, no. 1, pp. 587–618, 1st Quart., 2017.

[11] R. Piyare, A. L. Murphy, C. Kiraly, P. Tosato, and D. Brunelli, "Ultra low power wake-up radios: A hardware and networking survey," *IEEE Commun. Surveys & Tuts.*, vol. 19, no. 4, pp. 2117–2157, 4th Quart., 2017.

[12] D. Ghose, F. Y. Li, and V. Pla, "MAC protocols for wake-up radio: Principles, modeling and performance analysis," *IEEE Trans. Ind. Informat.*, vol. 14, no. 5, pp. 2294–2306, May 2018.

[13] L. Guntupalli, D. Ghose, F. Y. Li, and M. Gidlund, "Energy efficient consecutive packet transmissions in receiver-initiated wake-up radio enabled WSNs," *IEEE Sensors J.*, vol. 18, no. 11, pp. 4733–4745, Jun. 2018.

[14] D. Spenza, M. Magno, S. Basagni, L. Benini, M. Paoli, and C. Petrioli, "Beyond duty cycling: Wake-up radio with selective awakenings for long-lived wireless sensing systems," in *Proc. IEEE INFOCOM, 2015*, pp. 522–530.

[15] M. Magno, V. Jelicic, B. Srbinovski, V. Bilas, E. Popovici, and L. Benini, "Design, implementation, and performance evaluation of a flexible low-latency nanowatt wake-up radio receiver," *IEEE Trans. Ind. Informat.*, vol. 12, no. 2, pp. 633–644, Apr. 2016.

[16] A. Elgani, M. Magno, F. Renzini, L. Perilli, E. Franchi Scarselli, A. Gnudi, R. Canegallo, G. Ricotti, and L. Benini, "Nanowatt wake-up radios: Discrete-components and integrated architectures," in *Proc. IEEE Int. Conf. Electronics, Circuits and Systems (ICECS)*, 2018, pp. 1–4.

[17] C.-A. Hsu, C.-H. Tsai, F. Y. Li, C.-Y. Chen, and Y.-C. Tseng, "Receiver-initiated data collection in wake-up radio enabled mIoT networks: Achieving collision-free transmissions by hashing and partitioning," *IEEE Trans. Green Commun. Netw.*, vol. 5, no. 2, pp. 868-883, Jun. 2021.

[18] D. Ghose and F. Y. Li, "Enabling backoff for SCM wake-up radio: Scheme and modeling," *IEEE Commun. Lett.*, vol. 21, no. 5, pp. 1031–1034, May 2017.

[19] D. Ghose, A. Frøytlog, and F. Y. Li, "Enabling early sleeping and early data transmission in wake-up radio-enabled IoT networks," *Comput. Netw.*, vol. 153, pp. 132–144, Apr. 2019.

[20] IEEE-SA, "Part 15.4: Low-rate wireless personal area networks (LR-WPANs)," *IEEE Standard for Local and Metropolitan Area Networks*, Jun. 2011.

[21] I. Ramachandran and S. Roy, "Clear channel assessment in energy constrained wideband wireless networks," *IEEE Wireless Commun.*, vol. 14, no. 3, pp. 70–78, Jun. 2007.

[22] *JN-AN-1001 JN516x Power Consumption*, Application Note, NXP Semiconductors, Eindhoven, The Netherlands, Jun. 2016.

[23] 3GPP TS 36.213, "LTE; Evolved universal terrestrial radio access (E-UTRA); Physical layer procedures," R16, v16.3.0, Nov. 2020.

[24] D.-J. Deng, M. Gan, Y.-C. Guo, J. Yu, Y.-P. Lin, S.-Y. Lien, and K.-C. Chen, "IEEE 802.11ba: Low-power wake-up radio for green IoT," *IEEE Commun. Mag.*, vol. 57, no. 7, pp. 106–112, Jul. 2019.

[25] S. Rostami, S. Lagen, M. Costa, M. Valkama, and P. Dini, "Wake-up radio based access in 5G under delay constraints: Modeling and optimization," *IEEE Trans. Commun.*, vol. 68, no. 2, pp. 1044–1057, Feb. 2020.

[26] S. Rostami, P. Kela, K. Leppanen, and M. Valkama, "Wake-up radio-based 5G mobile access: Methods, benefits, and challenges," *IEEE Commun. Mag.*, vol. 58, no. 7, pp. 14–20, Jul. 2020.

[27] H. Milosiu, F. Oehler, M. Eppel, D. Früsorger, S. Lensing, G. Popken, and T. Thönes, "A 3-μW 868-MHz wake-up receiver with 83 dBm sensitivity and scalable data rate," in *Proc. European Solid State Circuits Conference (ESSCIRC)*, Sep. 2013, pp. 387–390.

[28] A. Frøytlog, M. A. Haglund, L. R. Cenkeramaddi, T. Jordbru, R. A. Kjellby, and B. Beferull-Lozano, "Design and implementation of a long-range low-power wake-up radio for IoT devices," in *Proc. IEEE WF-IoT*, Apr. 2019, pp. 247–250.

[29] F. Ait Aoudia, M. Gautier, and O. Berder, "OPWUM: Opportunistic MAC protocol leveraging wake-up receivers in WSNs," *J. Sensors*, vol. 2016, article 6263719, pp.1–9, Sep. 2016.

[30] A. Pegatoquet, T. N. Le, and M. Magno, "A wake-up radio based MAC protocol for autonomous wireless sensor networks," *IEEE/ACM Trans. Netw.*, vol. 27, no. 1, pp. 56–70, Feb. 2019.

[31] 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.

[32] D. Malone, K. Duffy, and D. Leith, "Modeling the 802.11 distributed coordination function in non-saturated heterogeneous conditions," *IEEE/ACM Trans. Netw.*, vol. 15, no. 1, pp. 159–172, Feb. 2007.

[33] F. Ait Aoudia, M. Gautier, M. Magno, O. Berder, and L. Benini, "A generic framework for modeling MAC protocols in wireless sensor networks," *IEEE/ACM Trans. Netw.*, vol. 25, no. 3, pp. 1489–1500, Jun. 2017.

[34] N. Seyed Mazloum and O. Edfors, "Influence of duty-cycled wake-up receiver characteristics on energy consumption in single-hop networks," *IEEE Trans. Wireless Commun.*, vol. 16, no. 6, pp. 3870–3884, Jun. 2017.

[35] N. S. Mazloum and O. Edfors, "Performance analysis and energy optimization of wake-up receiver schemes for wireless low-power applications," *IEEE Trans. Wireless Commun.*, vol. 13, no. 12, pp. 7050–7061, Dec. 2014.

Debasish Ghose received the Ph.D. degree in Information and Communication Technology from the University of Agder, Norway in 2019. From 2020 to 2021, he was a System Developer with Confirmit AS, Grimstad, Norway. Currently, he is working as a Post-Doctoral Research Fellow at the University of Agder, Grimstad, Norway. His research interests include protocol design, modeling, and performance evaluation of the Internet of things, system development, edge and fog computing, data analytics, and machine learning.

Luis Tello-Oquendo (Member, IEEE) received the degree (Hons.) in Electronic and Computer Engineering from the Escuela Superior Politècnica de Chimborazo (ESPOCH), Ecuador, in 2010, the M.Sc. degree in Telecommunication Technologies, Systems, and Networks, and the Ph.D. degree (cum laude) in Telecommunications from the Universitat Politècnica de València (UPV), Spain, in 2013 and 2018, respectively. In 2011, he was a Lecturer with the Facultad de Ingeniería Electrónica, ESPOCH. From 2013 to 2018, he was a Graduate Research Assistant with the Broadband Internetworking Research Group, UPV. From 2016 to 2017, he was a Research Scholar with the Broadband Wireless Networking Laboratory, Georgia Institute of Technology, Atlanta, GA, USA. He is currently an Associate Professor at the College of Engineering, National University of Chimborazo, Ecuador. His research interests include wireless communication, software-defined networks, 5G and beyond cellular systems, Internet of things, and machine learning. He is a member of the ACM. He received the Best Academic Record Award from the Escuela Técnica Superior de Ingenieros de Telecomunicaciín, UPV, in 2013, and the IEEE ComSoc Award for attending the IEEE ComSoc Summer School at the University of New Mexico, Albuquerque, NM, USA, in 2017.

Vicent Pla received the Telecommunication Engineering (B.E. & M.E.) and Ph.D. degrees from Universitat Politècnica de València (UPV), València, Spain, in 1997 and 2005, respectively, and the B.Sc. in Mathematics from the Universidad Nacional de Educación a Distancia (UNED), Spain, in 2015. In 1999, he joined the Department of Communications at UPV, where he is currently a Professor. His research interests lie primarily in the area of modeling and performance analysis of communication networks. During the past few years, most of his research activities have focused on traffic and resource management in wireless networks. In these areas he has published numerous papers in refereed journals and conference proceedings, and has been an active participant in several research projects.

Frank Y. Li received the Ph.D. degree from the Department of Telematics (now Department of Information Security and Communication Technology), Norwegian University of Science and Technology (NTNU), Trondheim, Norway, in 2003. He was a Senior Researcher with the UniK-University Graduate Center (now Department of Technology Systems), University of Oslo, Norway, before joining the Department of Information and Communication Technology, University of Agder (UiA), Norway, in Aug. 2007, as an Associate Professor and then a Full Professor. From Aug. 2017 to Jul. 2018, he was a Visiting Professor with the Department of Electrical and Computer Engineering, Rice University, Houston, TX, USA. During the past few years, he has been an active participant in multiple Norwegian and EU research projects. His research interests include MAC mechanisms and routing protocols in 5G and beyond mobile systems and wireless networks, Internet of things, mesh and ad hoc networks, wireless sensor networks, D2D communications, cooperative communications, cognitive radio networks, green wireless communications, dependability and reliability in wireless networks, QoS, resource management, and traffic engineering in wired and wireless IP-based networks, and the analysis, simulation, and performance evaluation of communication protocols and networks. He was listed as a Lead Scientist by the European Commission DG RTD Unit A.03- Evaluation and Monitoring of Programmes in Nov. 2007.