

# Adaptive Communication for Battery-Free Devices in Smart Homes

Gaia Maselli<sup>ID</sup>, Member, IEEE, Mauro Piva<sup>ID</sup>, Graduate Student Member, IEEE,  
and John A. Stankovic, Life Fellow, IEEE

**Abstract**—With the ever-growing usage of batteries in the IoT era, the need for more eco-friendly technologies is clear. RF-powered computing enables the redesign of personal computing devices in a battery-less manner. While there has been substantial work on the underlying methods for RF-powered computing, practical applications of this technology has largely been limited to scenarios that involve simple tasks. This paper demonstrates how RFID technology, typically used to implement object identification and counting, can be exploited to realize a battery-free smart home. In particular, we consider the coexistence of several battery-free devices, with different transmission requirements—periodic, event-based, and real-time—and propose a new adaptive and quick-to-learn MAC protocol, called APT-MAC, which dynamically collects information from devices without requiring any *a priori* knowledge of the environment. Extensive simulations clearly show the benefits of using APT-MAC, which is able to successfully deliver 97.7% of new data samples in complex scenarios, including several high traffic demanding devices, such as joysticks and cameras.

**Index Terms**—Backscattering, MAC, reinforcement learning, RFID, sensor augmented RFID tags.

## I. INTRODUCTION

BATTERIES in the IoT era have become ubiquitous, providing energy for a wide range of wireless devices—home, office, and personal devices—and enabling easy user movement. As the number of devices increases in smart environments, more and more batteries need to be replaced, posing significant maintenance issues in terms of cost and replacement time. Widespread battery usage also has a tremendous environmental price to pay. Batteries guarantee only a limited power supply and after some time of daily recharging they lose their charge in a significantly faster time frame than when they were new, eventually becoming unusable. Despite remarkable advancements in prolonging the lifetime of battery-assisted devices through energy-efficient networking protocols [1] and energy harvesting techniques [2], recent developments cannot guarantee power delivery in a continuous

Manuscript received September 5, 2018; revised November 19, 2018 and March 29, 2019; accepted April 14, 2019. Date of publication April 25, 2019; date of current version July 31, 2019. This work was supported in part by the Ministero dell'Istruzione, dell'Università e della Ricerca (MIUR) through the Dipartimenti di eccellenza 2018–2022 of the Department of Computer Science, Sapienza University. (Corresponding author: Gaia Maselli.)

G. Maselli and M. Piva are with the Department of Computer Science, Sapienza University of Rome, 00198 Rome, Italy (e-mail: maselli@di.uniroma1.it).

J. A. Stankovic is with the Computer Science Department, University of Virginia, Charlottesville, VA 22904 USA.

Digital Object Identifier 10.1109/JIOT.2019.2913231

(night and day) and ubiquitous (indoor and outdoor) manner. Moreover, spent batteries contain harmful chemicals and generate huge amounts of toxic waste that has to be disposed. The need for more eco-friendly wireless devices is evident.

The question is: Can we redesign smart objects in a battery-less manner so that they can work without the need for battery recharging or replacement? In particular, can we realize a smart home environment in which everything, from the TV remote to wireless light switches operate without batteries?

The answer is backscatter communication. Many devices that operate at a relatively low power budget can be remotely powered through backscattering of RF signals: modulating the reflection of existing RF signals. Backscattering offers a considerably efficient and increasingly practical alternative to active radio circuits in existing sensor systems [3].

Backscattered signals can be of two types: 1) ambient [4] or 2) RFID [5] signals. Ambient backscattering harvests power from signals available in the environment, such as TV [6], cellular [7], and Wi-Fi [8] transmissions. The main benefit of ambient backscattering is the exploitation of existing RF signals without requiring the deployment of a dedicated device to transmit a high-power signal to nearby devices. The main limitation is that ambient RF energy is not always available, and this can lead to reliability issues. In addition, techniques that have been demonstrated for ambient backscattering have low data rate (1 kb/s in the best signal conditions [6], [8]). The low achievable data rate motivates applications involving occasional (or spot) data transmission, such as money transfer between smart cards or revealing misplaced objects in a grocery store, but are not suitable for applications requiring continuous and real-time communication. When the signal is weak the data rate decreases significantly. For example, when a WiFi station is not transmitting, the achievable data rate using only periodic beacons reduces to 10–40 bps, which is definitively not sufficient for smart home applications. Signal availability is a related limitation. Although TV towers broadcast uninterrupted and continuous signals at all hours of the day and night, the ubiquity of the signal cannot be guaranteed, compromising the effectiveness of continuous, and real-time data transmission. If the signal is weak, sensors cannot operate; they have to accumulate enough energy to perform the required action. Even in metropolitan areas where TV signals are supposed to be ubiquitous, they weaken significantly in indoor environments positioned at more than 8–10 km from the TV tower.

RFID backscattering harvests power from signals emitted by a dedicated RFID reader [5]. In traditional RFID technology, the tags—battery free devices—absorb and reflect the high-power constant signal generated by the reader—a powered device—to send it their unique ID. With the advent of IoT, new applications of RFID technology have emerged: RFID tags can exploit the energy harvested from the reader to run some low power sensors and transmit sensed data [9]–[12]. However, the challenges in building battery free smart objects by exploiting RFID backscattering are multiple.

- 1) How to make battery-free devices to sense and transmit given that they cannot operate spontaneously on their own energy.
- 2) How to guarantee enough energy for multiple battery free devices at the same time.
- 3) How to simultaneously support heterogeneous sensor types, sensor requirements, and their different uses.
- 4) How to effectively cover an entire home.

In this paper, we explore the design of a battery-free smart home from a communication point of view and make the following contributions.

- 1) Invented a new MAC protocol to collect information from smart devices that improves response time and data delivery. Our protocol, called APT-MAC, quickly learns transmission rate requirements of active devices, without having any *a priori* knowledge on the type of devices, and adapt information collection to such requirements. This is critically important since smart homes are outfitted with many heterogeneous devices and people will not want to reconfigure their smart home every time they add or delete devices or applications that use these devices.
- 2) Showed through simulations that our MAC protocol is able to self adapt to devices requirements, reporting very low packet delay—33 ms in a scenario including up to 40 devices—and minimum data loss—below 5% for each type of device.
- 3) Studied what is necessary to cover an entire smart home.
- 4) Addressed energy consumption and health issues.

## II. BATTERY-FREE SMART HOME: ARCHITECTURE

Smart homes are outfitted with a myriad of sensors and smart devices—cameras, presence sensors, smoke sensors, light sensors, thermostats, smart meters, etc.—that are used to reduce resource consumption and improve the quality of life. In a home there are also a variety of everyday devices, such as TV remotes, cooling system remotes, light switches, and video game controllers, that are not “smart,” but are easily used by a vast majority of people to control many home devices/appliances. In this paper, we consider both classes of devices and make them operate without electric cords or batteries in order to realize a battery free Smart Home (see Fig. 1).

SÍ OK HOO CAPITO

To achieve this goal we exploit RFID technology, which is considered a key technology for identification of smart objects and hence it is deployed in any smart environment [13]. In RFID, battery free devices—the tags—send their unique ID to

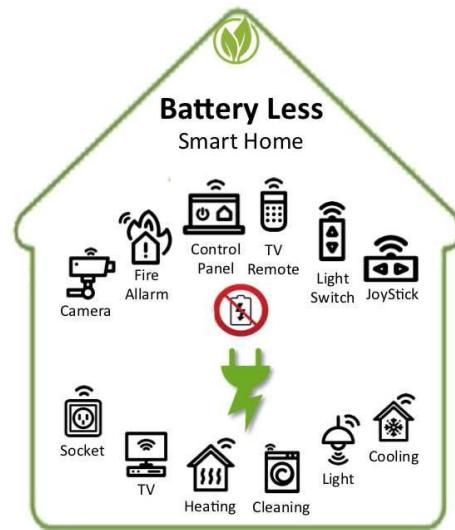


Fig. 1. Battery free smart home.

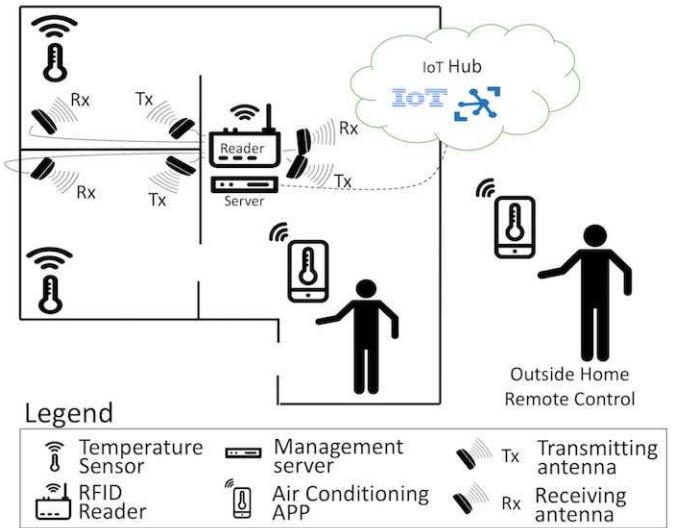


Fig. 2. Smart Home architecture: the system includes an RFID reader, equipped with a transmitting ( $T_x$ ) and a receiving ( $R_x$ ) antenna for each room. Temperature sensors are deployed in different rooms and send data to the air-conditioning app—running on a smart phone—through the RFID reader that is connected to a server. An IoT hub allows for interaction with home devices from both inside and outside the home.

a powered device—the reader—by reflecting the high-power constant signal generated by it. In our system, the reader interacts with sensor-augmented RFID tags, such as the UMich Moo computational RFID tag [14]. These tags are equipped with on-board sensors and/or actuators to provide not only static information such as their ID but also dynamic and real-time information about the state of the tagged object or the environment where these objects reside. The energy necessary for sensing and transmission is harvested from the reader’s transmitted signal.

At the hardware level the system architecture includes several battery-free smart devices and an RFID reader, equipped

with multiple antennas. Battery-free devices that can be realized through sensor augmented RFID tags are: videogame controllers [11], cameras [10], presence detectors, light switches, remotes for appliances, and temperature sensors. They are scattered in different rooms of the home. The RFID reader is equipped with multiple antennas, a transmitting and a receiving antenna for each room. Each antenna has a transmission range that is able to reach all devices in the same room. When the reader issues a query, all the transmitting antennas broadcast the query at the same time, reaching all the devices in the home. When tags receive a query, only the queried tag (whose ID is indicated inside the query message) backscatters the received signal to send its sensed data to the reader; all the other tags use the received signal to power on-board sensors and store the sensed data in a local buffer. At each query only one device in the home is interrogated. Multiple receiving antennas may receive its response (e.g., antennas in nearby rooms). However, the energy emitted by tags naturally decreases with distance; for example the rate of decrease is typically proportional to the inverse fourth power of the distance between the tag and antenna. Therefore, the closest antenna will receive the best signal, while the others will receive a weak signal. The reader will decode only the best signal and discard all the others. In case the reader receives multiple decodable signals, it can decode all of them, getting redundant information. Fig. 2 depicts an example of our system architecture, showing for clarity only two temperature sensors and a smart-phone running the air-conditioning app. The reader is connected to a local server that receives sensed data from the reader and dispatches the data to recipient applications. The server can interact with Smart Home applications running on several devices. For example, in a videogame application the server receives the joystick data from the reader and produces game commands that are sent to the videogame console. In an air-conditioning application the server can send commands to the air-conditioning unit depending on the received temperature data, or can show sensed data on a local screen. The server can also interact with client apps running on smart phones, allowing for interaction from both inside or outside the home—through an Internet IoT hub.

### III. ADVANTAGES AND LIMITATIONS OF RFID-BASED APPROACH

#### A. Advantages

There are important advantages to make smart homes battery free.

1) *E-Waste Reduction:* The majority of batteries currently used in houses and offices are eliminated, significantly reducing electronic waste. The study in [15] reports that China produced 570 000 tons of batteries only in 2013; and this value is growing year over year. The reduction of such an amount of waste is imperative, as the toxic metal required to make rechargeable batteries are harmful to the environment.

2) *Energy Consumption Reduction:* Decreasing the need for batteries implies diminishing also energy consumption for their recharging. It is true that our system requires a continuously operating RFID reader—which consumes energy (see

Section VIII for consumption estimation)—but any smart environment involves the deployment of an RFID reader, as RFID is considered a key enabling technology for many IoT applications [13]. The article in [16] remarks this aspect by stating that RFID technology should be deployed as part of a building's physical infrastructure, just like running water, lights, and heat. Exploiting such readers, our proposed devices can work with the emitted energy, without requiring batteries, or other sources of energy.

3) *Time Saving:* There is no need for batteries charging and/or replacing. Devices are always ready to work.

4) *Increased Device Portability:* Most of the devices become highly portable as electric cords or wires are removed. Consider the case of a light switch, realized through a tag equipped with a pressure button. It can be moved wall to wall without the need for cables and without batteries, with an almost unlimited lifetime. It can send messages to smart connected bulbs, in order to switch ON and OFF. The smart bulb needs only a power socket, but it does not need to be directly connected with the switch as in current homes; power implants are much simplified, as we just have to bring power without paying attention to circuits.

5) *No Need for Direct Visibility:* Most of battery free devices can trigger actions without being in the same room as the actuators. For example the light switch can turn on a light that is located in another room. Similarly, any remotes for appliances can power on devices that are located anywhere inside the home (under the reader coverage).

6) *Increased Environmental Sustainability:* Devices lifetime becomes unlimited: there is no more need for battery recharging or replacement.

#### B. Limitations

The main disadvantages are related to RFID technology and its operational limits.

1) *Limited Range:* Antennas coverage ranges between 1 and 3 m depending on transmission power (around 0.5–1 W for current prototypes), significantly limiting the operational range of devices. However, in Section VII, we discuss how to improve this weakness.

2) *No Outdoor Operation:* Battery-free devices cannot operate outdoors (around the house), or inside where the reader's antennas are not deployed. Although this represents a significant disadvantage, there are positive implications in terms of privacy as none of these devices can be accessed (queried) from outside.

### IV. ZERO CONFIGURATION MAC PROTOCOL

#### A. Protocol Description

A common vision for the IoT foresees a large number of devices in smart homes in the near future [17]. The major issues in dense sensing and actuation environments are system configuration, reconfiguration, and management, as sensors and applications are added and deleted. However, most of people want to use systems without configuring them: manual configuration is tedious and error-prone. Thus, the need for automatic and configuration/management free protocols. We

propose an MAC protocol, called APT-MAC, which quickly learns transmission rate requirements of active devices, without having any a-priori knowledge on the type of devices. We use reinforcement learning to build and continuously update devices transmission behavior. Then APT-MAC adapts information collection (and hence channel access) to such requirements. This avoids configuration overhead.

We now describe the sequence of stages involved in the APT-MAC protocol. At setup, the system sends inventory queries to discover how many devices—sensor-augmented RFID tags—are in the house, and assigns each a unique identifier. At this point the reader knows all devices in the environment and can query them to collect data sensed by their on-board sensors.

The key problem is: In which order should the reader query tags? A TDMA approach with fixed slot assignment queries tags sequentially without giving priority to most demanding devices and does not scale. Our APT-MAC protocol respects device transmission needs and properly rules channel access. We employ reinforcement learning to learn sensor behavior and requirements: it discovers environment patterns, changes in those patterns, and required rates for reading. As an example the protocol is able to understand when a video-game controller is started, and to allow it to send data more frequently than periodic-based devices (e.g., environmental sensors or information displays). The main goal of the protocol is to query tags so as to minimize the time between the generation of new sensor data and its delivery to the reader. To achieve this goal the reader implements the multiarm bandit algorithm [18], a reinforcement learning algorithm, based on the action-reaction paradigm. This algorithm involves five components: 1) the agent that performs the actions on the environment, i.e., the reader; 2) the set  $A = a_i, i = 1, \dots, n$ , ( $n$  is the number of sensors) of actions the agent can do, e.g., query tag  $i$ , query tag  $j$ , etc.; 3) the set  $S$  of states in which the agent can be (in our case the only state is “ready to perform a new query”); 4) the formula  $Q(a_i)$  to evaluate the expected reward of action  $a_i$ ; and 5) The vector  $Q$  to store the expected reward of each action.

Time is slotted and each slot, also called epoch, involves a reader’s action, (to query a tag). The pseudocode for slot assignment is given in Algorithm 1. The tag with the highest expected reward is selected to be queried next (chooseNext method). Let us suppose the reader is starting epoch  $n + 1$  by querying tag  $\text{next} = i$  and thus taking action  $a_i$ . When the reader receives response from tag  $i$ , it updates the expected reward  $Q(a_i)$  for the taken action by summing the expected reward  $Q(a_i)(n)$  calculated at the previous query, and the difference between the current outcome and the expected reward at the previous query, as defined in

$$Q(a_i)(n + 1) = Q(a_i)(n) + \alpha(\text{Reward} - Q(a_i)(n)) \quad (1)$$

where  $\alpha$  is the learning rate and is fixed to 0.1 (we empirically found this as the best value). Reward is the outcome of the current query: it is a positive value if the queried tag sent fresh data ( $R[\text{next}] = \text{updateReward}(\text{next}, \text{bonus})$ ), and negative if the sensor mounted on the queried tag did not produce new data since the last query ( $R[\text{next}] =$

---

**Algorithm 1** APT-MAC: Pseudocode for Slot Assignment
 

---

```

1: Master M;                                ▷ The reader
2: Set D;                                    ▷ All devices
3: Map R:(d∈D)→double;                      ▷ Rewards Map
4: /* Initialization of the Rewards map */
5: for d ∈ D do
6:   R[d] = 1.0;
7: end for
8: R = softmax(R);
9: /* Each cycle corresponds to a time slot */
10: while true do
11:   Device next = chooseNext(R);
12:   Bool goodQuery = M.query(next);
13:   if goodQuery then
14:     R[next]=updateReward(next,bonus);
15:   else
16:     R[next]=updateReward(next,malus);
17:   end if
18:   R = softmax(R);
19: end while
  
```

---

updateReward(next, malus)). Specifically, the reward is calculated as  $\text{Reward} = \text{bonus} - \text{malus}$ , where  $\text{bonus} = 0.4$  and  $\text{malus} = 0.01$ . If the tag sent new data then  $\text{Reward} = 0.39$ . In case the reader queried a tag that did not have new data since the last query, the reward is negative, i.e.,  $\text{Reward} = -0.01$ . After each reward update, we perform a Softmax [18] on the  $Q$  vector ( $R = \text{softmax}(R)$ ). With Softmax we compress the  $Q$  vector’s values into the range of  $[0, 1]$ . All the values of  $Q$  after Softmax add to 1.

i.e., nel dubbio  
ti metto il malus,  
poi in caso di  
aggiungo il bonus  
(?)

A positive reward allows a tag to gain and keep channel access: a videogame controller (i.e., joystick) just started will send new data every time it is queried and thus will have a very high reward, accessing the channel very often. However, no sensor is so fast to generate new data every slot (experimental results [11] show that slots last 6 ms). Furthermore, tags need a query to read data from on-board sensors and record new values. As a consequence, a tag cannot be queried at each slot, and thus we fixed a minimum query delay (MinQD), that is the minimum interval of time at which tags can be queried (MinQD is fixed at 50 ms based on empirical study). To avoid any starvation problem, in which a tag is never queried because the others have always higher reward values, we also set a maximum query delay (MaxQD), that is the interval of time at which tags have to be queried regardless the expected reward, meaning that each tag will not wait longer than this time between two consecutive queries.

MaxQD time guarantees fairness. While MinQD can be fixed to a common value for all devices, MaxQD depends on devices sampling rate requirements—a joystick or a camera needs to send data more frequently than an environmental sensor. Thus, MaxQD is dynamically set for each device during the start-up phase. When the system starts, MaxQD is fixed at 2000 ms for each device. While the system is operating, the reader updates the MaxQD based on the data loss it observes for each device. Data loss is calculated over intervals

i.e.  $\frac{R}{P} < 1$   
 ↓  
 i.e. se  
 $\frac{R}{P} < 0.85$   
 ↓  
 i.e.  
 E-greedy

of 50 queries (per device) as the number of new data samples received by the reader over the number of new data samples produced by the device (this information is provided by the device that is able to keep a counter of changes, i.e., the number of new samples between two consecutive queries). If the number of sample changes is greater than the number of new data samples received by the reader, then the reader should query the tag more frequently. Specifically, we set a 15% the tolerable data loss. When the reader loses more than 15% of new samples it reduces the MaxQD of 150 ms. This process continues until the data loss is below 15% or the system reaches the MinQD (i.e., 50 ms).

To allow new tags to join the system and at the same time less demanding tags to be queried more often than once every the MaxQD, action selection is probabilistic: the reader queries the device with the highest expected reward with high probability,  $1 - \epsilon$ , and with a small probability,  $\epsilon$ , it queries another randomly chosen device. The epsilon value is set to 0.1 based on an empirical tuning.

The APT-MAC protocol can be deployed on any RFID reader running EPC Global standard [19], by exploiting EPC primitives. The inventory command can be adopted to discover devices in the environment, the read command can be used to query devices, and the write command to send commands to actuators devices. The reader continuously sends commands to devices. In case of sensors readings, commands are scheduled through the mechanism based on reinforcement learning. In the case of an actuator writing instead, the command is scheduled based on requirements coming from the application: the reader receives a request from the application and interleaves the write command between scheduled readings.

### B. Computation Time

In terms of time, APT-MAC protocol costs  $O(n)$ , where  $n$  is the number of devices. At each slot, the reader chooses the next device to query by executing the *chooseNext* method that searches for the maximum value in  $R$ , analyzing in the worst case all  $n$  devices ( $|R| = n$ ). Even the softmax operation can be executed in  $O(n)$ , as it goes by every element in  $R$  and performs a constant time operation for each (i.e., division), while the *updateReward* function can be executed in  $O(1)$ . Thus at each time slot, the time complexity of APT-MAC is  $O(n)$ . With respect to TDMA, the choice of the next device to query is slower in APT-MAC, as TDMA has a predefined slot assignment that has constant cost  $O(1)$ . However, APT-MAC's cost is negligible if we consider that in smart homes the number of devices is in the order of tens.

VABBÈ UN  
ODG NON SO  
SE DEFINIRLO  
"NEGLIGIBILE"  
HA OK

## V. PERFORMANCE EVALUATION

We perform simulations to evaluate the performance of our APT-MAC protocol and compare it with a TDMA protocol that sequentially queries all tags, and the optimal query strategy, called *Optimum*, that always knows the best action to perform (i.e., which device to query next to avoid any data loss). At each epoch, the Optimum knows which sensor has

TABLE I  
WORKLOAD SCENARIOS DESCRIPTION (ENV. SENSORS INCLUDE TEMPERATURE AND PRESENCE SENSORS)

# of Sensors	Scenario	Joystick	Remote	Env. Sensors
20	Case 1	1	2	17
	Case 2	2	3	15
	Case 3	3	3	14
	Case 4	4	4	12
30	Case 1	1	2	27
	Case 2	2	3	25
	Case 3	3	3	24
	Case 4	4	4	22
40	Case 1	1	2	37
	Case 2	2	3	35
	Case 3	3	3	34
	Case 4	4	4	32

produced new data, and queries exactly that device, performing optimal data collection.

### A. Scenarios

We considered a wide range of use devices that should cover most homes: the number of battery-free devices ranges from 20 to 40 ( $n = 20, 30, 40$ ), including different types of devices to realize different workloads. Specifically, for each number  $n$  of devices we simulated four different cases. Table I details the numbers of different types of devices included in each case. Environmental sensors include temperature and presence sensors, as their transmission rate requirement is similar and very low. Complexity increases from cases 1 to 4 and from 20 to 40 sensors, as the number of real-time devices increases.

### B. Metrics

In our performance evaluation we focus on the following metrics: packet delay and data loss rate.

- 1) *Packet Delay*: The time between the generation of new sensor data and its delivery to the reader. In case a tag reads new data multiple times before being queried, the packet delay is measured since the first, undelivered, data collection.
- 2) *Data Loss*: The amount of new data samples delivered to the reader over the amount of new data samples generated by the sensor. Thus, data loss reflects how much new data is lost because tags are not queried on time. As tags have a small memory when they read new data from the on-board sensors they have to overwrite the previous data to store the new one. However, they can maintain a counter of changes that keeps the number of data updates performed since the last data transmission.

### C. Device Model

Smart homes include mainly three types of devices: 1) real-time (e.g., joysticks and cameras); 2) periodic (e.g., temperature sensors); and 3) event-based (presence detectors and remotes for appliances). To build a realistic simulation environment we modeled devices behavior through Markov chains. We watched real operation for a temperature sensor, a TV remote, and a joystick, and built devices models based on the observed behavior. The goal of our

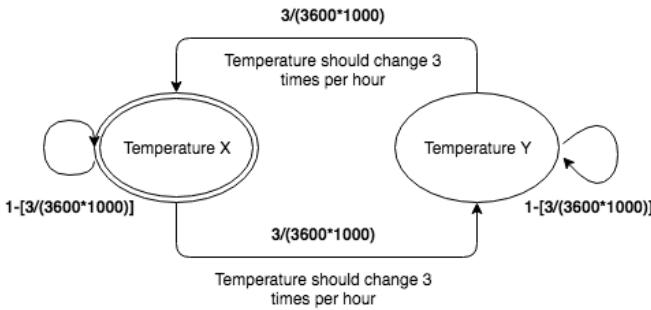


Fig. 3. Temperature sensor model.

model is to focus on state changes rather than on specific device states—values sampled by the on-board sensors. In the case of a temperature sensor, the model represents if the current value is different (it may be higher or lower) with respect to the previous reading, but does not keep track of the sampled value. Thus, a temperature sensor model includes only two states: 1) Temperature X and 2) Temperature Y with a transition from a state to the other whenever a new temperature value is observed. To introduce randomness in device behavior, state transition is probabilistic and is calculated as the number of expected changes over time.

Each state is characterized by a *stand-by time*, intended as the minimum amount of time in which the state cannot change due to physical aspects and users speed in performing actions. For example, we measured the user speed to press buttons on a device, and we found that the interval of time between two consecutive pressures on the same button does not last less than 70 ms. This time increases to about 98 ms in the case of two different buttons. Environmental devices, not sensing user actions, present also a stand-by time, due to physical sensor aspects (e.g., sensing resolution, refresh rate, ADC conversion, etc.). Considering these factors we fixed the stand-by time at 50 ms: if at time  $x$  the sensor has sensed value  $y$  then the sensor will report the value  $y$  for the following 50 ms (up to time  $x + 50$  ms). This time motivates also the choice of fixing the MinQD at 50 ms (see Section IV).

1) *Temperature Sensor*: The model for this sensor (see Fig. 3) has only two states—a) Temperature X and b) Temperature Y—and the transition probability between them is  $3/(3600 \times 1000)$ , where three represents the number of times we expect the temperature to change, and  $3600 \times 1000$  represents the interval of time—1 h—in milliseconds in which we expect these three changes to happen (3600 is the number of seconds in an hour and 1000 is the number of milliseconds in a second). This model is confirmed by results we obtained measuring temperature with our sensor: on average temperature may change of 1 degree three times per hour, not only due to changes in daily climate, but also to temporarily opening of windows/doors.

2) *Presence Sensor*: The model for this sensor (see Fig. 4) has three states: a) No Presence; b) Short Presence; and c) Long Presence. From the No Presence state, we may transit in the Short Presence state ten times a day ( $10/(24 \times 3600 \times 1000)$ ), and in the Long Presence state four times a day ( $4/(24 \times 3600 \times 1000)$ ). From the Short Presence state, we may transit back to the No Presence state ten times a day ( $1/(10 \times 1000)$ ), and to the Long Presence state once a day ( $1/(60 \times 10 \times 1000)$ ). From the Long Presence state, we may transit back to the No Presence state once a day ( $1/(10 \times 1000)$ ), and to the Short Presence state nine times a day ( $9/(10 \times 1000)$ ).

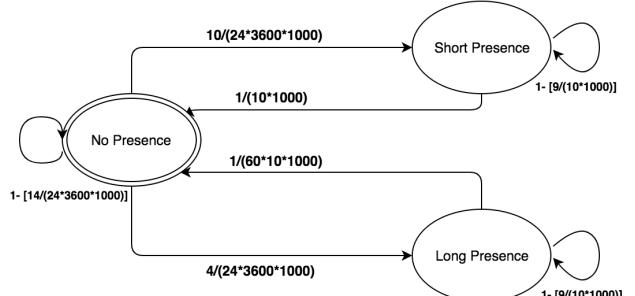


Fig. 4. Presence sensor model.

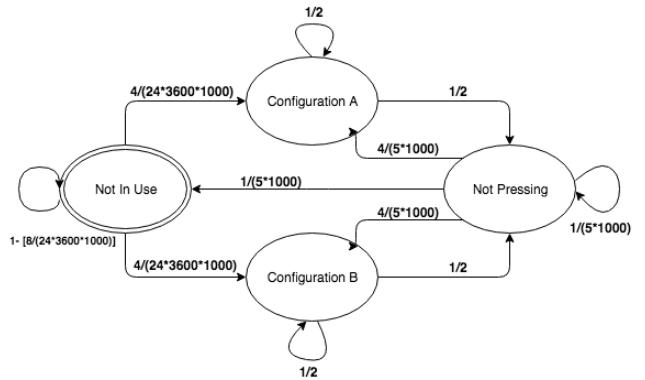


Fig. 5. TV remote model.

1000)), or in the Long Presence four times a day ( $4/(24 \times 3600 \times 1000)$ ). As a presence sensor detects people movements, there is a permanence for each state, that we modeled as **Short Presence** state for a number of seconds less than or near to ten ( $1/(10 \times 3600)$ ) s, and as **Long Presence** for times near to 10 min ( $1/(60 \times 3600 \times 1000)$ ).

3) *TV Remote*: For the simulation of a TV remote we modeled four states: a) Not in Use; b) Configuration A; c) Configuration B; and d) Not Pressing (see Fig. 5). From the Not in Use state we expect to start using the TV remote eight times per day. Remote buttons can be pressed with the same probability, and a button pressure makes the system to transit in a different state. After pressing a button, it is possible to move only to the Not Pressing state, which is different from the Not in Use state, as it represents the situation in which the user is going to press another button. We expect the user to change state about every second, and to perform around eight pressures before stop using the TV remote.

4) *Joystick*: The joystick represents the most complex device to model (see Fig. 6). It includes five states: a) Not Playing; b) Playing; c) Configuration A; d) Configuration B; and e) Not Pressing. Although having only five states may lead to the consideration of an oversimplified model, they are sufficient to realistically represent the behavior of such a device, as we are interested in changes and not in values. The state Configuration A represents a state in which multiple buttons may be pressed and the analog joystick reports some coordinates. When there is a change in the sensed values (even just

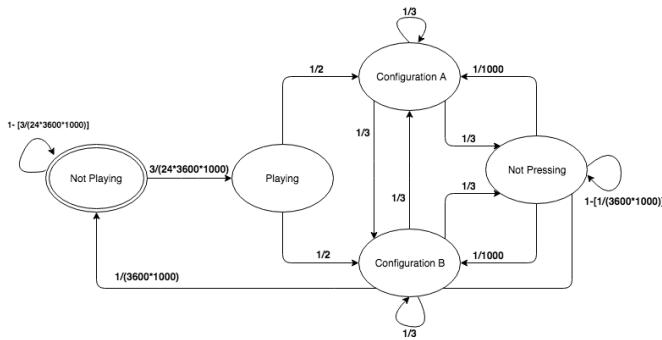


Fig. 6. Joystick model.

one pressure buttons reports a different value) the system transits in the Configuration B state—something has changed, we do not care what. Then, if there is another change (for example the joystick sensed a new value while buttons reports the same last values) the system transits again in Configuration A. This means that each state, Configuration A and Configuration B, correspond to different instances of values. If the system transits from Configuration A to Configuration B and then again to Configuration A, the current sensors values may be different from the values sensed when transiting the previous time to Configuration A. This model has been created to simulate a user who uses the controller three times a day and each time plays for one hour or less. These values have been validated by observing real players.

5) **Camera:** The camera is the most demanding device in terms of required data rate. Shot images need to be fragmented before being transferred. We modeled a camera that takes a shot and sends the image fragmented in multiple packets at the highest rate allowed by the network. Only after sending the last fragment it can take a new shot. If the image size is 25 kB [10], and time slots last 6 ms, the camera can send 40 bits at each query. Thus it can take a new shot every 30 s (if the number of slots required to send the entire image is  $(25 \times 10^3 \times 8)b/40b = 5000$ , then it takes  $5000 \times 6 \text{ ms} = 30000 \text{ ms} = 30 \text{ s}$  to send an entire image). As other devices also need to access the channel and send their data, the camera takes a few seconds more to send an image, depending on how many devices are active at the same time (see Section V-D).

#### D. Results

The evaluation looks at the speed and reliability of our APT-MAC protocol. In terms of speed we first evaluate the “start-up” transient time—where devices have larger losses and then they settle into great performance—and then the packet delay at steady state.

The start-up transient time is the time the protocol takes to learn transmission device requirements and minimize data losses (tuning the MaxQD value). Fig. 7 shows a snapshot of the number of lost data samples per device, when devices are started for the first time. Each device keeps a counter of sample changes since the last query and sends this value together with the last sample when queried by the reader.

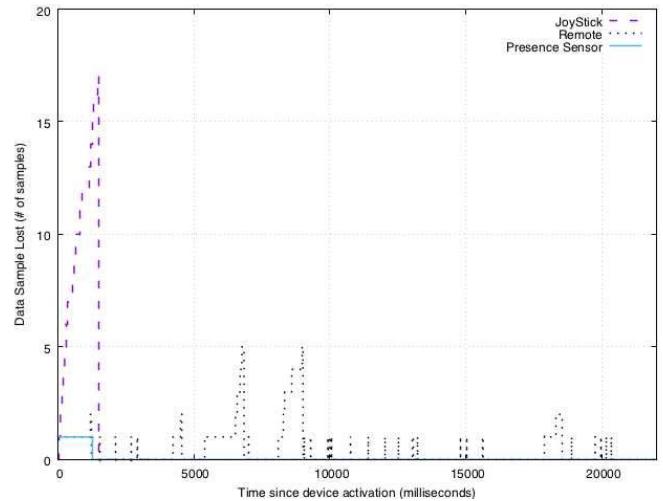


Fig. 7. Devices transient time.

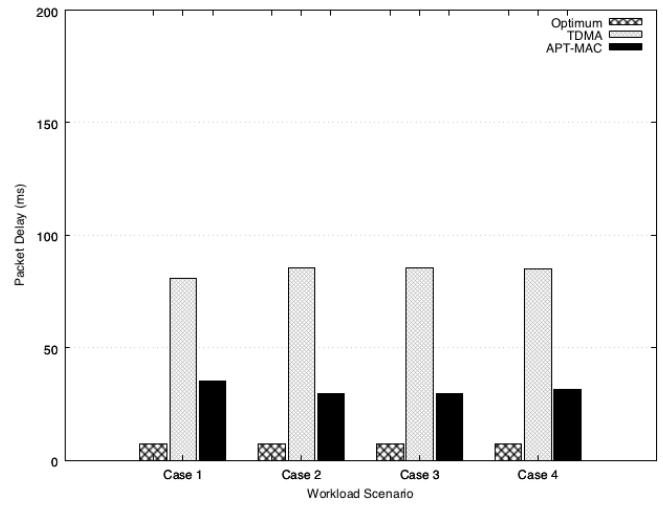


Fig. 8. Packet delay with 20 devices.

The joystick and the presence sensor take very short time—respectively, 1.5 and 1.29 s—to settle into great performance, eliminating sample losses. The remote instead needs longer time to reduce sample losses—after 20 s it loses on average 1 sample every 200 ms. As working sessions for remotes last on average 20 s, at the end of the first session the remote has reduced data loss, but has not reached the minimum MaxQD, so it will need another session to reach tolerable data loss.

The packet delay is a measure of the system speed to deliver sensed data to the reader in the steady state (once the reader has tuned the MaxQD per device). We estimate the transient state as the first 12 h of operation, as the user may start a device, such as TV remotes, joysticks, and light switches, even few hours after the system is launched.

Fig. 8 shows the packet delay for 20 devices by varying device complexity (see Table I). The Optimum shows the minimum achievable delay, i.e., 7.5 ms regardless the case complexity. APT-MAC takes on average between 29 and 35 ms. Compared with TDMA, APT-MAC is up to 2.8 times faster (in case 3, TDMA takes on average 85.56 ms to deliver

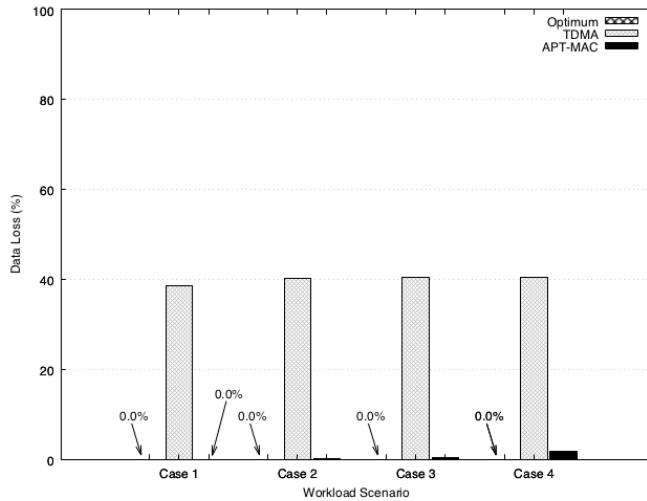


Fig. 9. Data loss with 20 devices.

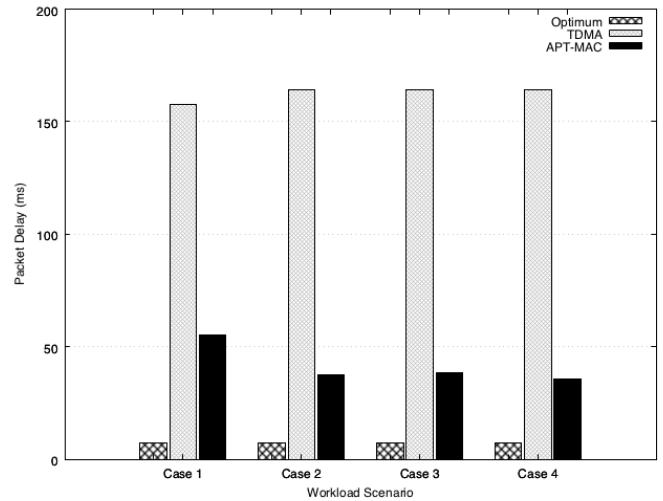


Fig. 11. Packet delay with 40 devices.

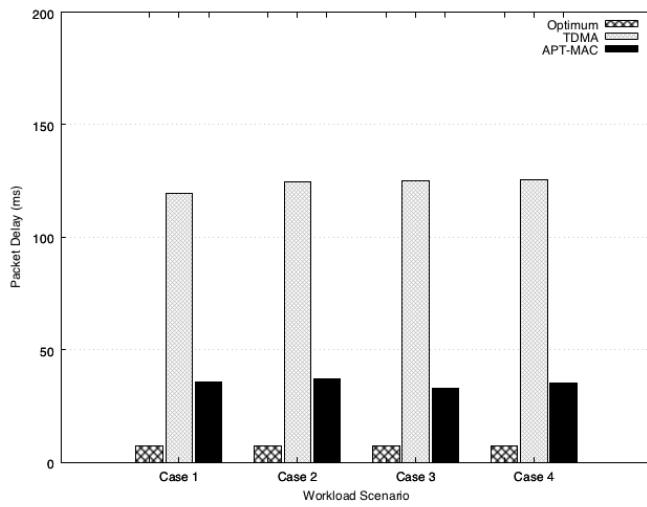


Fig. 10. Packet delay with 30 devices.

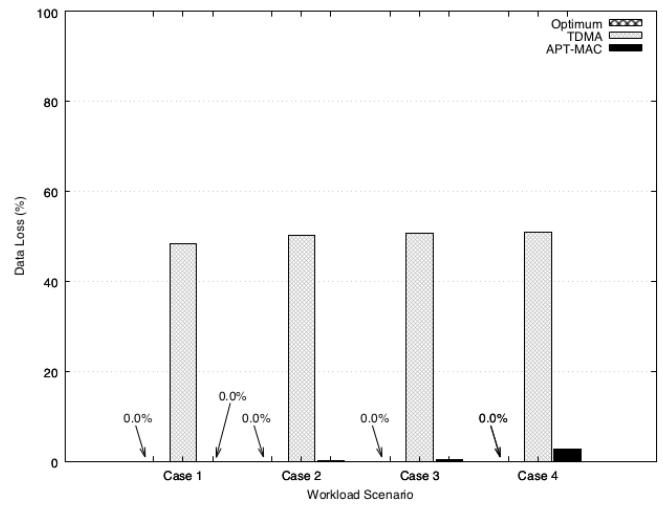


Fig. 12. Data loss with 30 devices.

data to the reader while APT-MAC is able to send data on average in only 29.9 ms). Although both solutions seem reasonable from an application requirements point of view—at the user level the difference in time is not noticeable—there is significant improvement by using APT-MAC in terms of data loss. As shown in Fig. 9, TDMA loses between 38.61% (case 1) and 40.46% (case 4) of new data samples, while APT-MAC does not lose more than 1.8% of new data in the worst case scenario (case 4), being very close to the Optimum—whose data loss is always 0—in the other three cases.

Figs. 10 and 12, and Figs. 11 and 13 report, respectively, packet delay and data loss for 30 and 40 devices. The results do not deviate from those that we have seen for 20 devices—irrespective of the number of devices and their complexity, we generally see that APT-MAC is always superior to the TDMA. In the case of 40 devices, TDMA doubles the packet delay (164.38 ms in case 4) with respect to the scenario with 20 devices, while APT-MAC takes no more than 35.85 ms in the same case to deliver new data to the reader (4.59 times

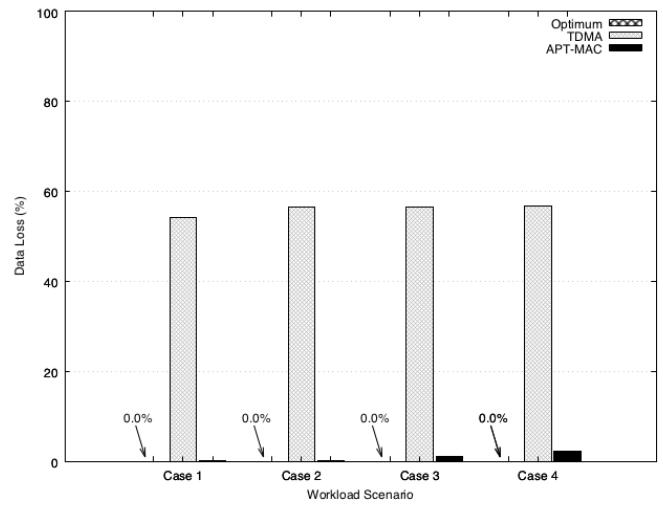


Fig. 13. Data loss with 40 devices.

faster than TDMA). Also data loss increases significantly with TDMA, up to 56.74% (case 4 with 40 devices), while it increases slightly with APT-MAC, which loses 2.3% of the

TABLE II  
GLOBAL AND PER-DEVICE PERCENTAGE OF  
DATA LOSS IN PRESENCE OF A CAMERA

	Global	Env	Remote	Joystick
Camera OFF	2.61	1.18	2.86	2.61
Camera ON (60sec)	2.88	1.30	4.04	3.68
Camera ON (45sec)	3.19	1.38	4.07	3.07
Camera ON (30sec)	4.91	1.23	4.77	5.10

new data samples. Thus, our first set of results clearly shows the benefits of using a reinforcements learning-based approach.

The second set of simulations investigates system performance in case of stressed conditions. We fix the number of devices at 40 and add a camera for home surveillance, which always has new data to send (continuous burst of data). Specifically, we consider three different cases. First, the camera sends data at the maximum allowed rate, almost saturating the channel. In this case, the reader gets a new image about every 33 s. This value decreases to 30 s in case the camera is the only active device in the system, meaning that the system cannot support cameras with higher shooting frequency. In the other two cases the camera reduces its shooting frequency to 45 and 60 s, giving more chance to the other devices to access the channel. Table II reports the average percentages of packet losses per type of devices in absence and presence of camera. The global data loss does not include data generated by the camera because it never experiences loss: it takes a new shot only after having sent all data related to the previous shot. When the camera is off, the system loses 2.61% of data samples, with comparable loss for the different types of devices (i.e., 1.18% for environmental devices, 2.86% for remote, and 2.61% for joystick). When the camera is on the global data loss increases as the shooting frequency increases (from 2.88% of data loss with shots every 60 s to 4.91% for shots every 30 s). In the last case, joysticks experience the highest data loss (i.e., 5.10%), that however remains tolerable to the user. The system degrades its performance if the number of cameras increases. In the case of two cameras with a shooting frequency of about 30 s each, the system data loss grows to 10.5%, becoming more difficult to tolerate from a user perspective.

In conclusion, a reinforcement learning-based approach, employing the bandit algorithm, is able to efficiently manage systems of 40 battery free smart devices, including up to four joysticks and four remotes, and a camera with a shooting frequency of one image every 30 s, without requiring any reconfiguration when devices enter and leave. Our approach is less suitable to more complex systems, involving multiple cameras—the algorithm becomes less efficient in assigning transmission slots to devices—or video cameras—due to limitations of the involved RFID technology, which does not allow sending images more frequently than every 30 s [10].

#### E. Evaluation in Noisy Environment

Our discussion so far assumes that the communication channel is free of errors. However, as battery-free smart devices rely on backscattering, the signal they reflect back to the reader is weak and subject to interference from any nearby device operating on the same frequency, such as IEEE 802.11ah,

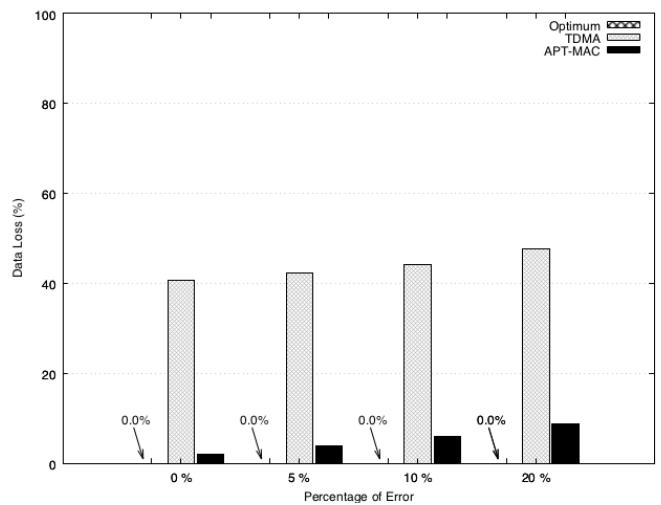


Fig. 14. Data loss in noisy environment.

amateur radio, ISMs, walkie talkies, and old cordless phones, depending on the regional allocation of the radio spectrum).

The obvious question is how APT-MAC behaves in presence of a noisy channel. RFID transmissions—both reader-to-tag and tag-to-reader—include a cyclic redundancy check (CRC) code that allows to detect accidental changes to raw data [19]. If the CRC verification fails the received packet is discarded, without sending any feedback to the sender.

To evaluate the impact of a noisy channel, we evaluate protocol performance in the case of 40 devices (case 1 as defined in Table I) by including a packet error rate,  $\text{PER} = \{5\%, 10\%, 20\%\}$  on the communication channel. Fig. 14 presents data loss by varying PER. The Optimum shows no impact regardless the PER value. APT-MAC experiences an increasing (+2.9%, +3.9%, +6.9%) but still tolerable data loss, with respect to the case of no error, while TDMA worsens its performance achieving almost 50% of lost packets.

The main ability to tolerate data loss of APT-MAC is motivated by data redundancy: the reader queries sensors at a higher frequency than that one at which sensors generate new data samples. For this reason, a lost packet does not imply a lost data sample. In case instead a new data sample is lost, the APT-MAC protocol automatically reacts by querying the sensor more frequently, diminishing the amount of lost data. In addition, in many cases, applications that use such data are implemented to be robust to some data loss.

Results on packet delay (see Fig. 15) show how the time between the generation of new sensor data and its delivery to the reader keeps below 40 ms, which is reasonable for most of home applications. A commercial joystick, one of the most demanding devices inside a smart home, can work with a delay of about 100 ms, as found in [20].

#### F. Fairness

We now show that APT-MAC protocol guarantees a fair access to the channel, fairly distributing data loss among all devices. We define the fairness index—based on Jain's

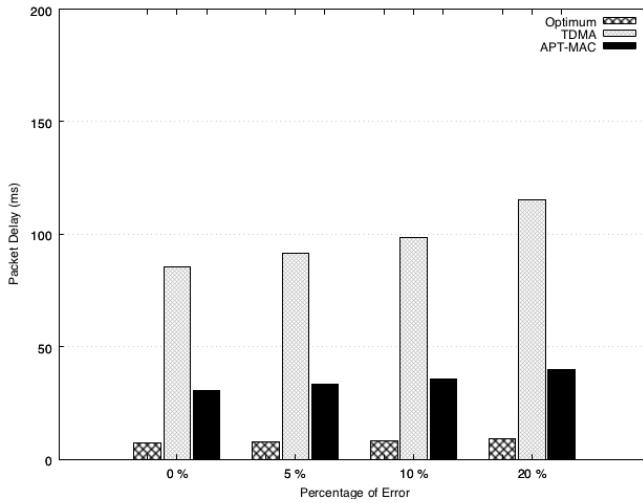


Fig. 15. Delay in noisy environment.

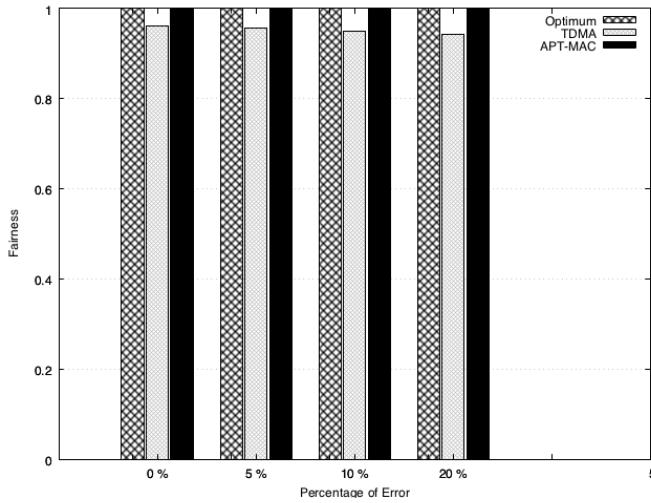


Fig. 16. Fairness on data loss.

fairness [21]—as described in

$$\psi(x_1, x_2, \dots, x_n) = \frac{\left(\sum_{i=1}^n x_i\right)^2}{n \cdot \sum_{i=1}^n x_i^2} \quad (2)$$

where  $x_i$  is data loss for device  $i$ .

Results (see Fig. 16) confirm that APT-MAC protocol is fair independently of channel noise, presenting a fairness index equal to 1 when PER varies from 0% to 10% and equal to 99% when PER = 20%. This is because the MaxQD time avoids any starvation problem, guaranteeing that even less operating devices are queried at regular intervals. In presence of channel errors this mechanism remains effective. TDMA instead decreases its fairness from 96% in absence of noise to 94.23% when PER = 20%.

## VI. RELATED WORK

We discuss related work that we have not touched upon in the previous sections.

**Information Collection From Sensor-Augmented RFID Tags:** A few solutions have been proposed to collect information from sensor-augmented RFID tags. A couple of recent solutions are based on Hash functions [22], [23]. The idea behind these works is to make the reader send only one query. Many of the devices in the environment will answer subsequently, without the need of a query for each of them. Exploiting a hash function as an index, the reader is able to understand in which frame it will receive no answer, one answer or a collision. With this information available the reader can send a bit vector in which a 1 means that the tag with that index should transmit, otherwise it should not. In this way the reader is able to avoid collisions and optimize transmission. Differently from our system, these protocols can only reduce the number of empty slots and optimize transmissions, but they are not able to adapt to current tags needs (e.g., give priority to more demanding devices). The vector index is defined by the hash function, and cannot be defined depending on the device need to transmit. Furthermore, as the reader can send only one query which contains information regarding many successive slots, these protocols are not able to dynamically adapt to burst data. The backscatter concept appears also in [24], but this paper presents mainly a modeling contribution, based on Poisson assumptions, stochastic geometry, and evaluated only via simulation.

**MAC Protocols for Smart Homes:** Current smart home devices are mainly based on technologies, such as ZigBee and Wi-Fi, or mainstream systems like Amazon Echo. At the MAC layer, these technologies operate according to a CSMA approach, that cannot be adopted in the context of this paper, because RFID tags cannot communicate spontaneously: they need a centralized entity (e.g., the reader) that energizes and queries them.

## VII. TECHNOLOGY POTENTIAL OVER TIME

Fig. 17 shows the range of devices that can be handled today with RFID technology, by varying the sensing frequency and the data rate. A reader that has a transmission power of  $P_t = 0.5$  W achieves less than 1 m of distance between the antennas and the tags. With this technology we can realize several smart devices (e.g., videogame controller, light switch), but we cannot realize a videocamera. A more powerful reader (e.g., with  $P_t = 1$  W) allows for longer distance (up to 3 m) between the reader’s antennas and the tags, but cannot satisfy a real-time frequency. WiFi technology achieves the best results, allowing for data transmission also from a videocamera, but is it not battery free.

Many constraints of our RFID-based solution are related to technology, and introduced by our prototypical implementation. There are a number of improvements that may be employed in a real environment in order to achieve better results in term of bit rate, distance, and energy available.

### A. Transmission Power

In our scenario we exploit a USRP RF daughterboard modified in order to transmit 500 mW of power, only half of a commercial reader. Exploiting the Friis equation and the

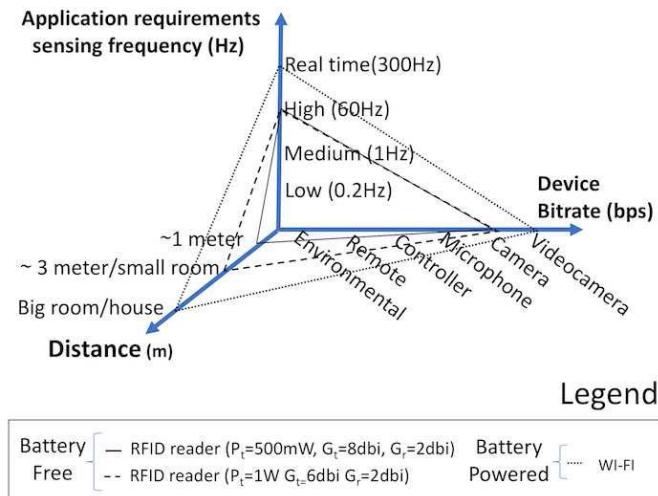


Fig. 17. Devices that can be handled today with RFID (compared with Wi-Fi) technology depending on the distance between devices, datarate and data frequency required by applications.

work [25] we can predict an operative range of 3.3 m, just changing the utilized reader (e.g., with transmission power equal to 1 W) and without hardware modification to RFID devices.

#### B. Exploiting Hybrid Harvesting (RF + Light)

In [26], we can find an example of an RFID device powered from both RF harvesting and a small solar panel ( $3 \text{ cm} \times 3 \text{ cm}$ ). In this case, it was shown to reach a distance of 21 feet and a maximum bitrate of 21.7 kb/s. In our specific case, the smart home, we should consider the energy harvested inside a room. The minimum amount of Lux required in this case in order to operate is 150 Lux, which is less than standard natural indoor light.

### VIII. ENERGY CONSUMPTION AND HEALTH ISSUES

We now evaluate the energy consumed by a continuously transmitting reader and its impact on users' health.

With the help of an amperometer, we quantified the energy consumption of our prototypal RFID reader in 1.32 A, with 6 V, for a total of about 8 W. In one year, our reader consumes less than 70 KW/h. In USA, the maximum price for one KW/h is always less than 20 cent, involving a cost of less than \$20 per year. We should even note that consumption of a commercial reader is expected to be less than that of our prototype.

From a health standpoint, the work in [27] studied the absorption rates in the human head and shoulder for Passive UHF RFID Systems at 915 MHz and found that in an ideal absorption environment, an RFID reader located within 10 cm (3.9 In) from the human head presents a specific absorption rate above the maximum value allowed by the United States' Federal Communications Commission (1.6 W/kg for both the spatial-peak 1 g and 10 g cube of tissue). Consequently, really close proximity to UHF RFID readers has potential health issues, particularly when close to the eyes. To avoid any potential harm to humans, UHF RFID transmitting antennas should

be set back at least 0.5 m (1.6 feet) from anyone who might receive constant exposure. If the antenna is within legal power output limits, and is kept at least 1 m (3.3 feet) from the human body, the incident radiation—even on the eyes—is at a level well below maximum allowable levels. In a battery free smart home there is one transmitting antenna per room, typically located on walls. A user inside a smart home rarely walks or resides next to the walls. In addition, RFID antennas can be placed far away from sofas or tables with chairs.

### IX. OPEN ISSUES AND FUTURE WORK

Besides the issues related to the RFID technology (discussed in Section III), open issues related to the APT-MAC protocol are mainly related to specific use cases. The protocol has been designed for environments with heterogeneous devices: in smart homes we expect to find a set of real time devices, a set of event-based devices and a set of periodic devices. In such a case, APT-MAC outperforms TDMA, as it is able to differentiate channel access based on estimated devices behavior. But when the environment is atypical—featuring only real time or periodic devices—APT-MAC loses effectiveness, converging to a TDMA-like protocol with slots equally allocated. We believe that although very rare, these situations should be properly addressed by the protocol, to guarantee high data collection performance. We leave this issue as future work.

### X. CONCLUSION

In this paper, we show how RFID technology can be employed to realize a variety of battery-free smart devices, performing real-time, periodic, and event-based sensing. Considering a wide set of battery-free devices, we defined a system to realize a battery-free smart home, specifying the system architecture—at both the hardware and software level—and the MAC protocol—called APT-MAC, that dynamically allocates transmission slots to devices without requiring any *a priori* knowledge of the environment. The key aspect in APT-MAC is the use of reinforcement learning that allows to dynamically adapt to user and environmental behavior. Results clearly show the benefits of our approach. The system efficiently handles scenarios with 40 different devices including a camera shooting a new image every 30 s.

We believe that our battery-free smart home offers a path forward for practical use of RFID technology in the development of battery free devices, motivating further work aimed at investigating techniques to support more demanding devices (videocameras) and more powerful technology (getting longer transmission ranges).

### REFERENCES

- [1] I. Demirkol, C. Ersoy, and F. Alagoz, "MAC protocols for wireless sensor networks: A survey," *IEEE Commun. Mag.*, vol. 44, no. 4, pp. 115–121, Apr. 2006.
- [2] Y. He, X. Cheng, W. Peng, and G. L. Stuber, "A survey of energy harvesting communications: Models and offline optimal policies," *IEEE Commun. Mag.*, vol. 53, no. 6, pp. 79–85, Jun. 2015.
- [3] P. Zhang, J. Gummesson, and D. Ganesan, "Blink: A high throughput link layer for backscatter communication," in *Proc. 10th Int. Conf. Mobile Syst. Appl. Services*, 2012, pp. 99–112.

- [4] S. Gollakota, M. S. Reynolds, J. R. Smith, and D. J. Wetherall, "The emergence of RF-powered computing," *IEEE Comput.*, vol. 47, no. 1, pp. 32–39, Jan. 2014.
- [5] R. Want, "Enabling ubiquitous sensing with RFID," *IEEE Comput.*, vol. 37, no. 4, pp. 84–86, Apr. 2004.
- [6] V. Liu *et al.*, "Ambient backscatter: Wireless communication out of thin air," in *Proc. ACM SIGCOMM*, Hong Kong, 2013, pp. 39–50.
- [7] A. N. Parks, A. P. Sample, Y. Zhao, and J. R. Smith, "A wireless sensing platform utilizing ambient RF energy," in *Proc. IEEE Topical Meeting Wireless Sensors Sensor Netw. (WiSNet)*, Austin, TX, USA, Jan. 2013, pp. 1–3.
- [8] B. Kellogg, A. Parks, S. Gollakota, J. R. Smith, and D. Wetherall, "Wi-Fi backscatter: Internet connectivity for RF-powered devices," in *Proc. ACM SIGCOMM*, Chicago, IL, USA, pp. 607–618.
- [9] V. Talla, B. Kellogg, S. Gollakota, and J. R. Smith, "Battery-free cellphone," *Proc. ACM Interact. Mobile Wearable Ubiquitous Technol.*, vol. 1, no. 2, Jun. 2017, Art. no. 25.
- [10] S. Naderiparizi, A. N. Parks, Z. Kapetanovic, B. Ransford, and J. R. Smith, "WISPCam: A battery-free RFID camera," in *Proc. IEEE Int. Conf. RFID*, San Diego, CA, USA, 2015, pp. 166–173.
- [11] G. Maselli, M. Piva, G. Ramponi, and D. Ganesan, "JoyTag: A battery-less videogame controller exploiting RFID backscattering: Demo," in *Proc. ACM MOBICOM*, New York, NY, USA, pp. 515–516.
- [12] J. Gummesson *et al.*, "RFID light bulb: Enabling ubiquitous deployment of interactive RFID systems," *Proc. ACM Interact. Mobile Wearable Ubiquitous Technol.*, vol. 1, no. 2, Jun. 2017, Art. no. 12.
- [13] D. Miorandi, S. Sicari, F. De Pellegrini, and I. Chlamtac, "Internet of Things: Vision, applications and research challenges," *Ad Hoc Netw.*, vol. 10, no. 7, pp. 1497–1516, 2012.
- [14] *A Batteryless Programmable RFID-Scale Sensor Device*. Accessed: Jan. 2018. [Online]. Available: <https://spqr.eecs.umich.edu/moo/>
- [15] X. Song, S. Hu, D. Chen, and B. Zhu, "Estimation of waste battery generation and analysis of the waste battery recycling system in China," *J. Ind. Ecol.*, vol. 21, no. 1, pp. 57–69, 2017. doi: 10.1111/jiec.12407.
- [16] M. Roberti, "RFID needs to be part of the building," *RFID J.*, Oct. 2018. [Online]. Available: <https://www.rfidjournal.com/articles/view?17892>
- [17] J. A. Stankovic, "Research directions for the Internet of Things," *IEEE Internet Things J.*, vol. 1, no. 1, pp. 3–9, Feb. 2014.
- [18] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, vol. 1. Cambridge, U.K.: MIT Press, 1998.
- [19] *EPC™ Radio-Frequency Identity Protocols Class-1 Generation-2 UHF RFID Protocol for Communications at 860MHz–960MHz*. EPCglobal Inc., Lawrenceville, NJ, USA, 2006. [Online]. Available: <http://www.epcglobalinc.org/standards/uhfc1g2>
- [20] G. Maselli and G. Salierno, "Performance evaluation of a battery-free videogame controller," in *Proc. 14th ACM Symp. Perform. Eval. Wireless Ad Hoc Sensor Ubiquitous Netw.*, Miami, FL, USA, 2017, pp. 71–75.
- [21] R. Jain, *The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation, and Modeling*. New York, NY, USA: Wiley, 1990.
- [22] S. Chen, M. Zhang, and B. Xiao, "Efficient information collection protocols for sensor-augmented RFID networks," in *Proc. IEEE INFOCOM*, Shanghai, China, 2011, pp. 3101–3109.
- [23] Y. Qiao, S. Chen, T. Li, and S. Chen, "Tag-ordering polling protocols in RFID systems," *IEEE/ACM Trans. Netw.*, vol. 24, no. 3, pp. 1548–1561, Jun. 2016.
- [24] K. Han and K. Huang, "Wirelessly powered backscatter communication networks: Modeling, coverage, and capacity," *IEEE Trans. Wireless Commun.*, vol. 16, no. 4, pp. 2548–2561, Apr. 2017.
- [25] A. P. Sample, D. J. Yeager, P. S. Powledge, and J. R. Smith, "Design of a passively-powered, programmable sensing platform for UHF RFID systems," in *Proc. IEEE Int. Conf. RFID*, Grapevine, TX, USA, 2007, pp. 149–156.
- [26] P. Zhang and D. Ganesan, "Enabling bit-by-bit backscatter communication in severe energy harvesting environments," in *Proc. 11th USENIX Conf. Netw. Syst. Design Implement. (NSDI)*, Seattle, WA, USA, 2014, pp. 345–357.
- [27] D. D. Arumugam and D. W. Engels, "Specific absorption rates in the human head and shoulder for passive UHF RFID systems at 915 MHz," *Int. J. Radio Freq. Identification Technol. Appl.*, vol. 2, nos. 1–2, pp. 1–26, 2009.



**Gaia Maselli** (M'09) received the Ph.D. degree in computer science from the University of Pisa, Pisa, Italy.

She is an Assistant Professor with the Department of Computer Science, Sapienza University of Rome, Rome, Italy. In the past, she has contributed to research on cross-layer design for ad-hoc networks. Her current research interests include design and implementation aspects of mobile networks and wireless communications systems with a particular focus on backscattering networks for the Internet of

Things and design and performance evaluation of networking protocols for RFID systems.

Dr. Maselli serves as a member of TPC for several international conferences, is an Associate Editor of *Computer Communications* (Elsevier), and serves as a Reviewer for several journals, such as the *IEEE TRANSACTIONS ON MOBILE COMPUTING*, the *IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS*, the *IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS*, and the *IEEE/ACM TRANSACTIONS ON NETWORKING*. She has participated in many European Community research projects such as CHIRON, eDIANA, SENDORA, SENSEI, E-Sense, WiseNts, and MobileMAN.



**Mauro Piva** (GS'19) received the Laurea degree (Highest Hons.) in computer science from the Sapienza University of Rome, Rome, Italy, in 2017, where he is currently pursuing the Ph.D. degree at the Department of Computer Science.

His current research interests include backscattering communication, reinforcement learning for networking, Internet of Things networks, and software defined networking.



**John A. Stankovic** (S'77–M'79–SM'86–F'93–LF'14) received the Ph.D. degree from Brown University, Providence, RI, USA, and the Honorary Doctorate degree in real-time systems from the University of York, York, U.K.

He is the BP America Professor with the Computer Science Department, University of Virginia, Charlottesville, VA, USA, and the Director of the Link Lab. He has over 57000 citations with an H-index of 115. His current research interests include real-time systems, wireless sensor networks, smart and connected health, cyber physical systems, and Internet of Things.

Dr. Stankovic was a recipient of the IEEE Technical Committee on Distributed Processing's Distinguished Achievement Award (Inaugural Winner), seven Best Paper Awards, including one for ACM SenSys 2006, the University of Virginia Distinguished Scientist Award in 2015, the School of Engineering's Distinguished Faculty Award in 2010, the Distinguished Faculty Award from the University of Massachusetts, and the IEEE Real-Time Systems Technical Committee's Award for Outstanding Technical Contributions and Leadership. He has given over 40 keynote talks at conferences and many distinguished lectures at major universities. He also served on the National Academy's Computer Science Telecommunications Board. He was the Editor-in-Chief of the *IEEE TRANSACTIONS ON DISTRIBUTED AND PARALLEL SYSTEMS* and was the founder and the Co-Editor-in-Chief of *Real-Time Systems*. He is a Fellow of the ACM.