



A Novel Discovery Protocol for IoT Based Home Automation

Tyler Steane^{1,*} and PJ Radcliffe¹

¹*School of Engineering, RMIT University, Melbourne, Australia*

(Received 6 December 2018; Accepted 3 January 2019; Published on line 1 September 2019)

*Corresponding author: s3330480@student.rmit.edu.au

DOI: 10.5875/ausmt.v9i3.2076

Abstract: It is proposed that Home Automation systems would be vastly improved by the novel approach of eliminating the Permanent Central Controller as it is the source of significant problems. This leaves the responsibilities of the Permanent Central Controller to be assumed by the remaining devices in the system, including IoT devices and interface devices such as smartphones and computers. While the capacity for joining devices to a network without the Permanent Central Controller has been considered, device discovery has not. This paper examines existing protocols but finds that while many have worthwhile aspects none are suitable for the task. A novel discovery protocol is proposed, using smartphones or computers as intermittent control devices. This new protocol is developed with the aid of a new Robust Network Development Methodology which is able to anticipate problematic use cases prior to implementation. This methodology was very successful in identifying and eliminating significant problems with the new protocol. Implementation and measurement of the novel discovery protocol demonstrates the viability and robustness of discovery without a Permanent Central Controller on low cost ESP8266 family devices.

Keywords: Discovery; Home Automation; Internet of Things; Connectivity Management; Device Management; Home Area Networks; Network Architecture.

Introduction

Home automation has long been an active area of interest for academia and industry since the 1970's when some of the earliest protocols were being developed [1], [2]. Despite decades of work, home automation has yet to make any significant penetration into the average home. This is not for lack of options, many home automation systems are on the market today but these systems are expensive, complex, inflexible and proprietary. The proprietary nature of these systems means that users must buy all parts from the one manufacturer to ensure compatibility with their system, even the simplest components, such as mains switches and light fittings. This limits the accessibility and diversity of features available to users as they are locked in to a single product line. These systems are generally inflexible and require

expert technicians to make any changes, thus preventing the user from re-configuring the system to suit changes in their lifestyle. Even where some flexibility is afforded the systems are often complex and beyond the everyday users' capabilities.

Finally, these systems are expensive not just because they are still niche products, but due to their design. Most current systems revolve around a Permanent Central Controller (PCC) which is a permanent fixed system coordinator, they most often have a wired connection to the network and are an essential indispensable part of the home automation system. Thus, the PCC offers a single point of failure for the whole system. It is frequently expensive on its own but must also be powered on at all times contributing to running costs. Furthermore, each of the mentioned limiting factors increases costs: proprietary systems (which can be priced at the manufacturers whim), expert technicians (which

are expensive), and complexity (which promotes higher build costs).

There are now some devices available which work independent of a PCC, such as smart light or thermostats, however these do not comprise home automation systems. These are individual devices with their own app meaning any system made up of such devices require many apps for the one system. Such devices do not integrate with other devices, unless strictly part of the same product line meaning manufactures are again free to play the lock-in game. Such devices are not a desirable solution but are perhaps the most accessible products for home automation.

The Internet of Things (IoT) has been greatly favored in recent times for innovation in Home Automation but many implementations have maintained the same limitations of the commercially available systems. However, IoT devices could be implemented in a new way to overcome these limitations. Our aim is to develop a new paradigm for home automation using simple IoT devices. In order to be successful this novel paradigm will need to be of low cost, simple to install, simple to maintain and operate, flexible, and manufacturer independent. To achieve this, we believe that the PCC must be eliminated and replaced with protocols allowing M2M (Machine to Machine) communications and machine to human interface in order to redistribute the tasks that were formally the responsibility of the PCC, see Figure 1. To reduce costs and increase usability the user interface runs from one or more smartphones, tablets or computers which acts as an Intermittent Control Device (ICD) which is only sometimes present in the system as a user interface. The ICD is not a replacement central controller it has only assumed some of the responsibilities of the controller to be eliminated as shown in Figure 1. The ICD does not co-ordinate or integrate the system it simply provides users an interface into the decentralized home automation system. The system is fully operational even when the ICD has left the network.

To be successful this new paradigm must be an open standard that will allow any compliant device from any manufacturer to operate on a home automation network with other compliant devices.

Tyler Steane completed a BSc in Applied Physics and a BEng (Honours) in Electronic and Communications Engineering in 2015 at RMIT University in Melbourne. In 2016, he began work on his PhD in electrical and electronic Engineering. His research is focused on home automation and how ubiquitous smart homes might be realised.

PJ Radcliffe received his B.Eng. from Melbourne University and his PhD from RMIT University, both in Australia. He is a senior lecturer in the School of Engineering at RMIT University. His current interests include data driven education research, the Internet of Things, real time embedded systems, and data networks.

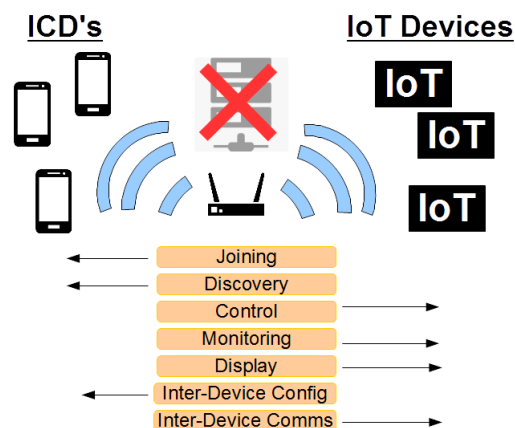


Figure 1. Redistribution of the responsibilities of a Permanent Central Controller.

The vision of the new paradigm is that the everyday user should be able to purchase a new IoT device from the local department or hardware store, install and configure it themselves and connect it to their existing system. This should be achieved with a single Open Source App.

The first step was to devise a secure network joining method that could accommodate simple devices lacking display or input device, without inflating their costs. This was envisaged by Nasrin and Radcliffe [3, 4] and has, more recently, been implemented and refined to provide a practically viable and user friendly protocol [5-7]. The next challenge is to discover the available devices on the network and to allow for remote interface from a smartphone or similar device.

This paper has found existing discovery protocols to be insufficient when there is no Permanent Central Controller and so a novel and robust protocol has been developed. However, it is a significant challenge to develop a new protocol which is robust, since it is easy to make mistakes or embed errors in the design. To overcome this problem a methodology is required to reduce the number and magnitude of flaws. In development of this novel protocol it was found that no formal development methodology was suited to assist in minimizing design flaws. While there are many project development methodologies none suits the niche area of protocol development and so a new methodology is required. Therefore, this paper also defines and implements the Robust Network Development Methodology. This, novel method will allow for greater confidence in the design, implementation, testing and use of the novel discovery protocol.

This paper is organized as follows; section 2 will make a review of existing literature considering device discovery, while section 3 will outline the requirement for a new approach to discovery. This will lead to section 4's consideration of the need for a network protocol

development methodology taking lessons from existing development methods. Section 5 will propose the Robust Network Development Methodology learning from the insights of section 4. Section 6 will apply this methodology and propose a discovery protocol, which is implemented and assessed in section 7. Finally, future work is considered in section 8.

Related Work

Discovery over a network has been an active area of research for some time and a variety of methods exist each catering for different environments. The environment of particular interest to us is the home which will normally have one or more wireless LANs to service the entire house. The following review is grouped by the underlying protocols that existing work has built upon. The Universal Plug and Play (UPnP) protocol has long been a favored protocol for home automation systems particularly for the Simple Service Discovery Protocol (SSDP) included in its stack [8-10]. Zhang et al. [11] combined UPnP with OSGi to develop a control system architecture that kept a registry of discovered devices that could be search for a desired device. This stage of registration is a valuable one, but the system focuses on a larger network with several controllers and multiple home gateways. This does not make it easy for home owners to develop their own system.

Hsu et al. [12] have taken advantage of the discovery capabilities of UPnP to develop an interface framework for device discovery and management within the home. The focus on a user-friendly interface is commendable and sorely needed. However, the concept is still reliant on a Permanent Central Controller.

Zhang et al. have used UPnP simply to discover devices for their control system [13]. Much of the protocol goes unused and is integrated only to service the discovery event; this overhead is mitigated by their use of an external server. While the server is used to create “mash-up”, where a light bulb and light sensor can be virtually combined into one smart bulb, the demands on the user's skills are still high and a Permanent Central Controller is still essential.

Thus, while UPnP offers much in the way of a general framework and ideology for device discovery its architecture leaves it reliant on a Permanent Central Controller and leaves it unsuitable for achieving the aims of a viable home automation system.

Lee et al. have addressed the issue of “auto-configuration” using a software defined network [14]. While the system still relies on a Permanent Central Controller it is able to discover new devices and register them to a database using the MAC address for

identification. While the system is no doubt powerful it would not easily integrate in to the average household and still requires a central controller.

Kim et al. looked to DPWS for a device discovery protocol when developing an IoT home gateway [15]. However, they found it too demanding for more constrained devices and so developed their own ad-hoc protocol for these devices using UDP packets. This protocol starts with a UDP advertisement message from the constrained device alerting the gateway to register the device. This protocol is not very robust and assumes all transmissions are successful, furthermore it requires predefined device information which can be associated with a registered device.

In their work developing an application framework Kamilaris et al developed an ad-hoc discovery protocol which requires new devices to advertise themselves until acknowledged [16, 17]. This approach depends on a Permanent Central Controller for the replies and assumes the reliable delivery of all transmissions.

Datta et al [18], have suggested a smartphone application that implements an IoT framework for home automation, and have even demonstrated how this can be implemented for personalized healthcare [19]. This approach however treats discovery as a selection process to find the appropriate resource for a given task, thus the system is largely static and all devices are already known. Discovery in the sense of new device additions is a manual process.

A more recent addition to the plethora of IoT protocols has been CoAP, an RFC draft published in 2014 [20]. It has many virtues including its focus on constrained devices making it low power and not computationally demanding [21-23]. CoAP offers a method by which UDP can be made sufficiently robust for most applications. Discovery has two components under CoAP, firstly a server must be discovered and then its resources. For our purposes the discovery of servers is of interest. This is achieved via a multicast packet, to which discoverable devices reply. However, these messages are considered “Non-confirmable”, meaning an acknowledge is not required. If there is any packet loss then CoAP discovery messages may be lost or not reach all devices.

From this review of the present literature there is no approach to device discovery that suits a system without a central controller where devices can arbitrarily be brought in and out of service. All are heavily entrenched in the topology of the Permanent Central Controller and few offer a user-friendly solution fit for the home environment. In order to meet the stated goals a light weight, novel solution is needed, which can eliminate the Permanent Central Controller and offer an “easy to install and use” approach that requires little of the end

user.

A Novel Protocol

A protocol is required to distribute the roles and responsibilities of the controller in order to eliminate the Permanent Central Controller. The key roles for distribution are device joining, discovery and operation. We intend to eliminate the need for a Permanent Central Controller with a new protocol called the Decentralised Home Automation Protocol (DHAP) that will handle the three key roles. Joining has already been described and implemented [6, 7] this paper focuses on the development of the discovery part of the protocol, with operation to follow in future work.

DHAP allows an Intermittent Control Devices (ICD), such as a smartphone or a laptop to join, discover and control IoT devices around the home. Without a Permanent Central Controller an ICD needs to be able to determine what devices exist on the network, and how to display and control them. Such a list needs to be built up progressively over time as new devices are added but must also be up-to-date as devices are replaced or go offline. This maintenance activity goes beyond regular discovery and is better described as a census of the network devices.

DHAP is intended for use in the home environment and is expected to operate over a wireless network. However, it is not intended to be limited to Wi-Fi unnecessarily. Wireless allows for more flexibility and is also the more challenging typology and will thus be the focus of this paper. This paper will focus on development of the ICD protocol to implement the census function.

Existing Development Methodologies

The process by which a concept is developed into a network protocol is a critical one as mistakes here may not be discovered until practical use and require an expensive rework process. This section considers existing development methodologies, highlighting the key strengths and weaknesses that can be used to develop a method tailored to network protocol development.

It has been known for many years that mistakes made in the development process may not be seen until the protocol is implemented, or even later when it is used in practice [24]. Such mistakes are expensive both in terms of time and money. Minimization of errors in all areas of design is crucial and many authors have proposed formal methodologies for minimizing errors in the development phase. We can take advantage of decades of experience in the area of project life cycle research and learn the lessons from various methodologies, and then adapt these

lessons to the niche area of network protocol development. The literature in the area of life cycles is vast, but there is little devoted to the specifics of network protocol development, with the notable exception of [25] which offers a methodology based largely on the waterfall model.

A methodology which can learn from the experience of life cycle processes and create a tailored solution for network protocol design will be far better than an ad-hoc approach which is likely to create more errors, development delays, and cost over runs.

A network protocol development methodology needs to start with a clear goal for the basic functionality of the protocol. The final output should be a robust protocol design that can be handed to those responsible for its implementation. There is little tolerance for error in network protocols so the tests and use case assessments must be extensive. This suggests that high quality documentation is essential, not only for a successful implementation but also to instill confidence in those testing and commissioning systems and integrating the protocol around the world. Network protocol development is somewhat unique in that there is no traditional customer interface in the development phase, and thus the requirements are driven by the initial concept and end goal.

With the key characteristics of a network protocol life-cycle identified it is now possible to look at existing methodologies to see what lessons can be learned. The Waterfall life cycle [26] prioritizes planning early on and promotes rigorous documentation. These are essential elements in general and specifically for network protocol development. The life cycle promotes formalized reviews with the potential for guidelines and check-lists to manage errors but also facilitate the maturation of less experienced developers and to reduce errors made by less experienced staff. A key problem is that specifications are not tested until customer use, at which time changes are very expensive.

The prototype life cycle [27] suggest a helpful emphasis on prioritization, starting with key functionality first and building up the design from there. Furthermore, its iterative approach allows for improvement of an idea as well as repeated scrutiny which fosters a robust design. The spiral model [28] takes advantage of an experienced engineer in-order to make rigorous risk assessments, when a team consists of members with varied experience this offers the potential to develop experienced engineers as the less experienced learn from the more experienced. Additionally, like the prototype process the spiral model benefits from the refinement of iteration, and adds an extra opportunity to detect errors, especially those due to revisions in various aspects of the design.

The V-model [29], while very similar to the waterfall model offers the valuable contribution of highlighting the relationship between where errors are made and where they will be discovered and the corresponding degree of cost incurred.

The extreme (or XP) model [30] develops test cases prior to programming, and this offers a unique and effective strategy for error minimization and cost efficiency.

König's work [25] is to be commended for its rare consideration of life cycles in Network protocol development. The proposed methodology is a well thought out life cycle which owes much to the waterfall model. However, the late consideration of constraints in König's life cycle will inevitably require a reworking of the design, which increases costs and turnaround times.

Many design life cycles offer great assistance to network protocol development. However, none offers a robust way to move from the initial idea\concept to the verification phase. This is crucial for a robust protocol and thus we have devised the following methodology which is loosely based on the extreme model and the prototype life cycle.

Robustness Network Development Methodology

This methodology takes the key aspects of network protocol development and the lessons from existing methodologies to develop a method specifically for network protocol design, termed the Robust Network Development Methodology (RNDM).

Step 1 - The naive protocol: The initial network protocol concept should be summarized in a naive protocol. A naive protocol should represent the minimal functional outcome desired for a new protocol, it should be inherently optimistic of the environment and all other processes. It is not expected to be a viable solution and should avoid anticipating any problems in its operation. This prevents the analysis being driven in a particular direction that might narrow the solutions considered in later stages. The naive protocol is not expected to necessarily work and will be refined in later steps.

Step 2 - Model: choose a modeling method and tool. There are a variety of ways to represent a network protocol. Common methods include bounce diagrams (also called sequence diagrams) [31], Finite State Machine (FSM) models such as SDL [32], and Petri nets [33]. FSM and Petri nets have the advantage that they can be used to verify the protocol works as required. Bounce diagram are adequate when the protocol is simple, such as with a master-slave relationship, and where timing variations are not significant.

Step 3 - Assess: The naive protocol is rigorously assessed in a method akin to Gedanken experiments [34]. This is the scenario generation phase in which all problematic use cases are identified. One way to achieve this would be to consider a check-list of known general problematic scenarios or categories; this has a great advantage for maintaining uniform and comprehensive assessment with less experienced staff.

The drawback, however, is that this assessment will only be as good as the check-list, and the unique aspects of the protocol and its environment may be overlooked. In order to combat this one may also apply a more creative brainstorming approach answering a simple question of 'what could possibly go wrong anywhere in this protocol?' This method favors the experienced developer, while the former is of great assistance to the novice.

A combination of check-list and brainstorming can achieve the best of both worlds, but the brainstorm should be applied first as the check-list can suppress creative thought and appear to be comprehensive. Once the brainstorm and then the check-list have been applied an assessment of the two should allow for additions to the check-list for future application. Thus, the check-list is strengthened with time, and experience within the team is grown.

The check-list should be applied rigorously; this is best done by applying each item to every event or process in the protocol. This is best demonstrated by assessing a bounce diagram, run the check list for each vector, its beginning (Tx) and end (Rx) and anything in-between. And then what could go wrong between each vector and the next.

Step 4 - Enhance: Each problematic case or scenario should be addressed in the protocol design. This is greatly aided if the cases are numbered and categorized. Once these issues have been addressed the more robust protocol is again modeled and assessed (as in Steps 2 & 3). This is repeated until all issues have been adequately addressed, it should be expected that several iterations will be required as some changes will create problematic scenarios of their own, however experience will reduce the frequency of such problems. Again, the traceability of these problematic cases is essential and a detailed cataloguing of them should ensure that all known concerns are traced through development, implementation and testing.

The advantage of this approach, summarized in Figure 2, of use case generation is that it not only provides test cases for those implementing the protocol in code, but it also provides a comprehensive description of how the protocol should behave and for what conditions it is known to operate thus informing those contemplating the use of the protocol in their systems.

RNDM can and should be followed up by standard

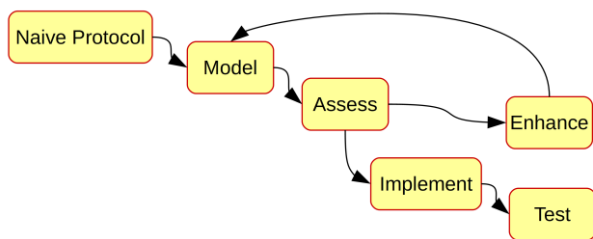


Figure 2. The Robust Network Development Methodology.

penetration and load testing to verify the assumptions made and to quantify the performance of the protocol.

Architecture of a Discovery Protocol

Applying RNDM to develop a novel discovery protocol begins with a naive protocol shown in Figure 3. This protocol is suitable for a single LAN home environment protected by WPA2. Any smartphone, even multiple smartphones, can act as Intermittent Control Devices (ICD) and will need to create or update their list of IoT devices which have already been joined to the LAN.

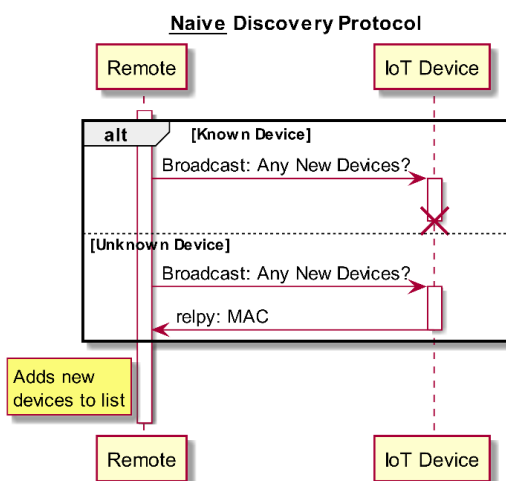


Figure 3. Naive Discovery Sequence.

A. Step 1 & 2: Naive Protocol

1. An ICD broadcasts a UDP request for any new devices to announce themselves and then waits for a response.
2. Each device receives this broadcast and if they have not previously announced themselves, they reply with a UDP packet containing their ID, which will be their MAC address.
3. The ICD will add any new devices that respond to its existing list and end the task.

B. Step 3: Problematic Use-Cases

Each use case states a problem and its possible causes.

1. Generic transmission problems for any transmission:

- 1.1. Failure to properly initialize the wireless system.
- 1.2. No network connectivity.

2. Generic reception problems for any reception:

- 2.1. Packet loss due to poor signal strength or occasional packet collision.
- 2.2. Packet loss due to deliberate jamming.
- 2.3. No network connectivity.

3. ICD Failed Broadcast:

- 3.1. ICD wireless transmission does not succeed due to initialisation issues.
- 3.2. No network connection.

4. Device Failed to Receive:

- 4.1. Packet loss due to interference or poor signal strength.

5. Failed Reply:

- 5.1. There are no new devices.
- 5.2. UDP means there is no guarantee of delivery.
- 5.3. These networks may easily have 150+ devices; this may increase traffic and packet collisions.

6. Forgotten Devices:

- 6.1. If a device replies but is not heard, it assumes it has been found and will not answer again. Thus it is undiscoverable.
- 6.2. If a device is removed at the remote how can it be re-discovered without re-setting the device
- 6.3. What if a device has to be reset for some reason, e.g. firmware update, reconfigure.

7. Lost devices:

- 7.1. What happens when a device goes off-line: crashes, is removed etc. This would invalidate the list.

8. Multiple Remotes:

- 8.1. Home networks will need to be able to support multiple remotes, but currently a device will only announce itself once.

9. Multiple Networks:

- 9.1. What will happen if a user wishes to use the same remote at different locations on different networks?

10. Network Re-configuration:

- 10.1. Users may change their SSID or Password
- 10.2. Users could change their Gateway IP.

11. Security & visibility (I.e. Locks):

- 11.1. Should any remote be allowed to know about any device? This maybe better handled at the next layer.

12. Removing a Device:

- 12.1. How can a device be removed from the network?

13. Ignoring a Device:

- 13.1. Should a remote ignore some devices? How should this be achieved?

14. Ignoring a Request:

14.1. Should the device be intentionally undiscovered?

C. Step 4: Resilient Protocol

The Robust protocol is shown in Figure 4 and the general process, shown in Figure 5, is as follows:

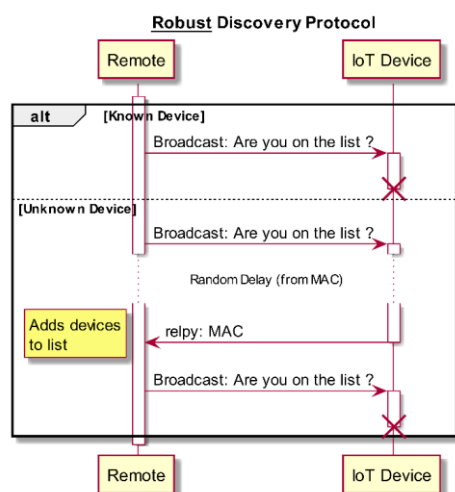


Figure 4. Robust Discovery Sequence.

1. An ICD broadcasts a UDP packet to all devices. It will include the census list of all know devices.
2. Devices will check the census list and those not listed or whose details are out of date will reply to the ICD after a delay of a random duration.
3. The ICD will wait for 1 second for all devices to reply and then either add or update all replying devices to the list.
4. Steps 1-3 will then be repeated until no replies are received within 1 second.
5. If three consecutive attempts then receive no replies the process is ended. Otherwise the Process begins again.

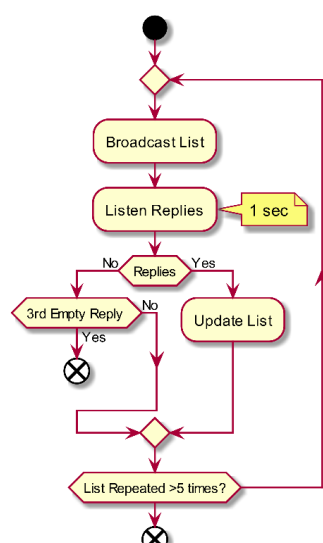


Figure 5. Logic flow of discovery broadcast attempts.

6. The protocol may also be ended if 5 identical broadcasts are made or additionally as a catch all to prevent run away a time-out should be implemented.

The Census List will contain each device's:

- MAC Address
- Last known IP Address
- Status bit (Online\off-line)
- Visibility bit (permission level)
- Last contact date

These parameters in this protocol have been chosen to attain balance where possible between competing concerns. For example, more attempts and longer wait times or delays will often allow for greater transmission success; however, such parameters would also extend cycle times and slow down the discovery process and burden the network longer and irritate users. These values may be refined further through detailed assessment but at present these values strike a helpful balance.

D. Problematic Use-Cases addressed:

Connectivity and initialization issues can easily be handled by software, to diagnose the status of connections and advise users or other programmatic functions which can respond appropriately. Thus, for example manufactures would be responsible for leaving there device in a safe mode when connectivity is lost. Therefore, the issues in 1.1 to 3.2 can be easily and somewhat trivially handled.

Failed deliveries as in to 4.1 will not be addressed directly in this solution. The protocol will only be as good as the network on which it is operating. Successful transmissions that have struggled to make contact should be observed on ICD's and may need to trigger a notification to the User that their network is struggling, under a de-auth attack, or may have performance issues.

Packet loss due to collisions, however, is of particular concern. A fully developed home automation system may have 150+ devices all replying at once. In order to keep devices simple UDP is the preferred transport protocol however this does not provide any guarantee of transmission. The repeated broadcasts, with the list updated in between, will function as a kind of acknowledgement and ensure that any devices within range can be found. This will address the concerns of 5.2.

Further to the issue of packet collisions the IEEE 802.11 standard utilizes the Distributed Coordination Function (DCF) which implements collision avoidance (CSMA/CA); while this does not prevent collisions it should reduce the likelihood of such events [35]–[37]. This procedure, however, does not address the potential for simultaneous replies to collide. With the potential for 150+ devices to be replying at once this protocol will also

implement a staggered response for all devices, to reduce the number of devices trying to transmit at the same time. The response delays will be randomized based on the devices MAC address and will address the concerns of 5.3.

Case 5.2, highlights the potential for packet loss due to the unreliability of UDP. While the 802.11 DCF mechanism requires an acknowledge packet from the receiving station, if no successful transmission is achieved by the transmitting station and the maximum attempt limit is breached then no attempt to recover the transmission will be initiated. In this case devices will remain unaware of whether or not they have been registered by an ICD, instead ICDs will broadcast a list of all known devices and only those devices not listed will reply.

As has been discussed failed replies from a device can be negated using this protocol. Devices are now ignorant of their status with any given ICD until they receive a broadcast with the ICD's latest list. So, if a device replies to the initial broadcast but it fails it will be prompted to reply again when the ICD repeats the broadcast as the new list will still not include the device. If a device does not recognize that it has been included in the list, Figure 5 shows that the same census list will not be sent out more than 5 times. Case 6.1-6.3 & 8.1 are therefore no longer of concern as devices will always respond if not listed.

Case 7.1, still leaves open the possibility that the list may include a device that is no-longer connected. The list of known devices will include several parameters beyond those necessary to identify the device and these will include a status bit (online\offline) and users should be alerted and given the option to drop the device from the list. If any other parameters are changed the device should reply and will be treated as an update that is it will reply like normal and the ICD will recognize that the device already exists and then make an update.

ICD's should construct separate lists for different network SSID's in order to address case 9.1. While, for cases 10.1 & 10.2, users should be permitted to edit the network details of a list but the list will need to be re-validated by a scan for all devices. This will not require the re-acquisition of higher level details (from the detailed info layer) as this can be carried over from the original list.

Simple device permissions should be established at the joining protocol, as well as sending the local SSID and password, 2 passwords should be sent to allow for 3 permission levels an Administrator, a regular user and a Guest user (No password other than local Wi-Fi). No devices should ignore a discovery requests. Instead they may respond with a visibility declaration to set the permission level required for them to be visible, so an ICD may know they are present but may not be shown to all

users. This resolves cases 11.1 & 14.1.

In response to case 13.1, ICD's should not ignore any device, however additional software may hide devices from a user's list for aesthetic reasons or convenience. Devices may be ignored in the sense of not being counted during the repeat broadcasts if they have been added to the list but continue to reply as though they have not found themselves on the list, in this case the user should be notified that the device is not behaving as appropriate.

Experimentation

One of the great values of RNDM is that it automatically provides a comprehensive set of test case scenarios which can be used to test the protocol design at a conceptual level, in simulation, or as a real physical experiment. Four problematic use cases, 2.1, 4.1 & 5.2-5.3, are chosen for further analysis here as they lend themselves to physical implementation and measurement.

The test bed developed simulates the ICD with a script running on a virtual machine with a wired connection to a home network Wi-Fi router. The wireless IoT devices were represented by 3 ESP-01 and 2 ESP-12 modules.

These ESP Modules are members of the ESP8266 chip family which present a great opportunity for home automation as they are of very low cost and Wi-Fi enabled. They have previously been identified as ideal candidates from implementation of the Steane's Joining protocol [7].

The assessment consisted of 3 different experiments, each broken into multiple tests with different conditions. These tests were then repeated 200 or 1000 times for statistical rigor. Ideally the results from these experiments would be compared to existing technologies, however due to the ubiquity of the Permanent Central Controller topology there is no protocol supporting a decentralised approach with which to make a meaningful comparison.

Exp. 1: Demonstrating Interference

The first experiment assessed the concerns raised by use cases 2.1, 4.1 & 5.3, namely that packets may be lost and disrupt the protocol. In this experiment the ICD's make a single (unrepeated) attempt to discover any devices, 5 different tests were run: the first had only one ESP device available, the second had 2 devices and a new device was added to each test until the fifth with 5 devices. Each test was performed 200 times and the percentage of transmitted reply packets successfully receive by the ICD was recorded.

Exp. 1: Results and Discussion

The results of this Experiment are shown in Figure 6, and as anticipated the modules appear to be interfering with one another even with just 2 devices where nearly 15% of packets are lost. Significantly more losses are seen with more devices as the interference is compounded to the point that for 5 devices less than 60% of packets are transmitted successfully. This validates the concerns of

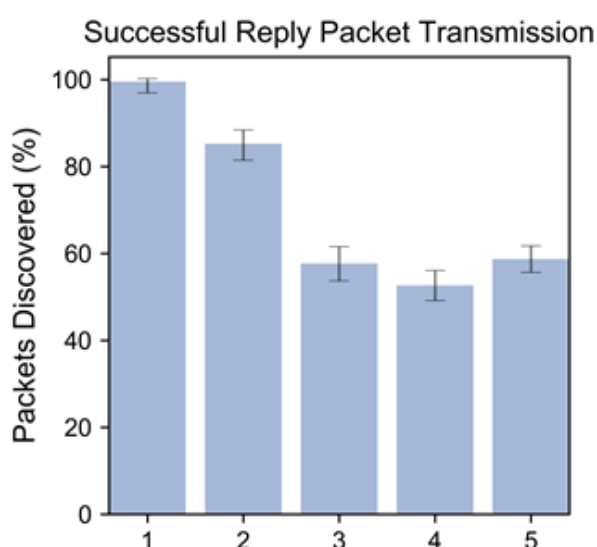


Figure 6. Percentage of device replies successfully received by the ICD as the number of replying devices is increase from 1 to 5 devices.

use cases 2.1, 4.1 & 5.3 and confirms the need for the protocol to address the issue of simultaneous transmissions and packet collisions.

Exp. 2: Assessing Randomised Delay

The second experiment consisted of 2 tests aimed at assessing use case 5.3, where high traffic rates may result in high packet losses and is addressed with random response delays in the proposed protocol. In both tests the ICD broadcasts an empty device list (indicating that all devices should reply), in the first test devices responded without delay while in the second test the IoT devices would seed a random number using the last octet of their mac address and then wait a randomised time between 0-200ms (thus implementing the proposed protocol). Each experiment was run 1000 times.

Exp. 2: Results and Discussion

The results, shown in Figure 7, indicate high packet loss without the randomised delay meaning only 2-4 of the 5 devices can be discovered with any regularity. Here as in the following figures, error bars have been omitted for very small samples which have large confidence intervals. Given the built-in capacity of the 802.11 standard to handle packet collision, including CSMA/CA

and MAC level acknowledge packets; it is surprising that so few replies are transmitted successfully. It is therefore worthy of further investigation to question if the standard is fully and properly implemented. The addition of DHAP's randomized delay vastly improved the results to the point that all five devices replied and where discovered successfully. This experiment proves the utility of the Robust Network Development Methodology (RNDM) as it

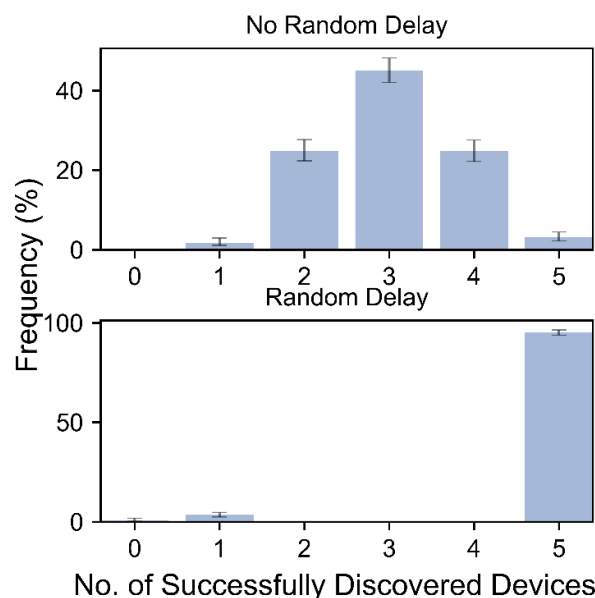


Figure 7. Number of devices discovered after one discovery broadcast without (Top) and with (bottom) a random delay. Shows a vast improvement in the average no. of devices discovered when using a random delay.

was able to anticipate and effectively address this issue.

Exp. 3: Assessment of Robustness measures

The third experiment assessed the proposed protocols capacity to cope with uses cases 2.1, 4.1 & 5.2, which relate to packet loss and the less than robust nature of UDP. Thus this demonstrates the effectiveness of the multiple attempts made within the protocol to discover all devices. This experiment comprised 4 tests, in all the ICD began by broadcasting an empty device list, then the following broadcast would update the list to include all devices whose replies where received. As listed devices are not required to respond it was expected that as the list grew fewer devices would be attempting a reply, make it easier for devices to be discovered.

Each of the four tests where run 1000 times, the first two tests were run with and without the random delay seen in experiment 2. The final two test where a repeat of the first but with the addition of randomised packet loss achieved by having the IoT devices randomly dropped their reply packets 50% of the time. This simulated the effect of a noisy Wi-Fi environment in which

packets may be lost for other reasons than the contention between replying IoT devices.

Exp.3: Results and Discussion

Figure 8 shows the results of the four tests in experiment 3. The two plots, A & C, are similar to those obtained in experiment 2 and show that with the random delay; usually just one attempt is required to capture all 5 devices while 2-3 are more common without the delay. Finally the right 2 plots show the impact of randomly dropping packets, as expected the average number of attempts is increased but the random delay still gives a better result with an average of 3 attempts needed compared to 4 attempts without.

These experiments demonstrate that the proposed robust solution successfully addresses and handles the problematic uses cases considered. The protocol is shown to be suitable for improving the successful response rate for discovery replies from an unreliable 2-4 device to a reliable 5 out of 5. As well as recovering from missed packets due to packet collisions as a result of high traffic or simultaneous transmissions, with typically less than 5 attempts needed to discover all devices even with 50% packet drop rates. Furthermore it demonstrated the ability of this protocol to discover devices without a central controller, without compromising robustness or functionality.

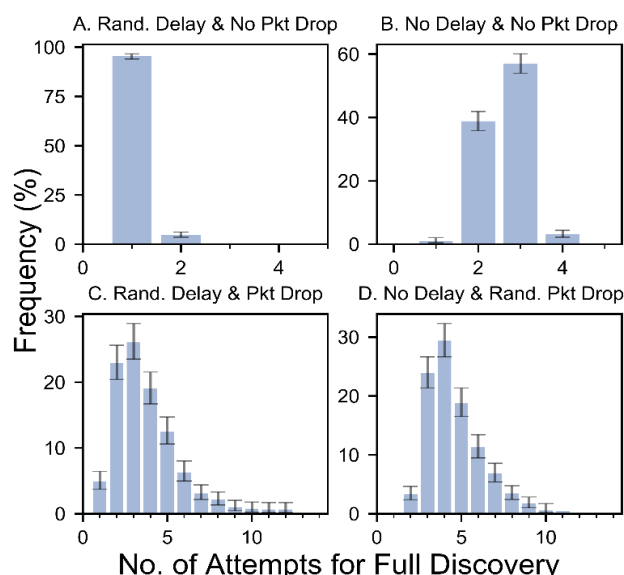


Figure 8. Shows Number of attempts needed to discover all 5 devices with and without the randomized response delay (A & B). This is repeated with reply packets randomly dropped (C & D).

Future work

The novel device discovery protocol is currently being integrated with the previous secure joining protocol and this will be tested in a variety of situations. One

element is still missing for making a complete, central controller free, IoT home system: the display and control of the IoT device. The concepts for this have been laid out and the required novel protocols are being developed. If all goes to plan the entire protocol suite will be release as an open standard with a reference implementation. Using this standard one smartphone app will be able to join, discover, display, and operate any compatible IoT device from any manufacturer.

Conclusion

The new Decentralised Home Automation Protocol (DAHAP) provides a complete home based IoT system without the need for a Permanent Central Controller PCC. The duties of the PCC are distributed between IoT devices and one or more Intermittent Control Devices (ICDs) such as a smartphone. This reduces costs and allows a home owner to "plug and play" any DHAP compatible device. The secure joining protocol has already been described; this paper describes the novel device discovery protocol. The need for a novel discovery protocol was established having found the current methods for device discovery to be dependent on the problematic Permanent Central Controller topology. Consideration of a solution highlighted the need for a development methodology specifically tailored to the needs of protocol design. Thus, the novel Robust Network Development Methodology (RNDM) was developed and used to develop the new discovery protocol.

Further highlighting the value of RNDM, the discovery protocol was easily assessed using the problematic uses-cases identified in the development process. This assessment demonstrated the need for the protocols features including multiple discovery broadcasts as well as randomized response delays.

The experimental results obtained in this work demonstrate not only the success of this protocol but the viability of device discovery without a Permanent Central Controller (PCC) or any additional infrastructure or services unlike alternative discovery approaches discussed in the related work section. This work has shown that the dependence of Home Automation systems on PCC's is not at all necessary for a robust discovery protocol, this builds on work showing that the PCC is not needed for joining either and anticipates future work aiming to eliminate the PCC for devices display and Control.

The development of a complete, central controller free, home automation system is a step nearer with only one further protocol to develop, the display and control protocol. Once this is complete a single smartphone app will be able to join, discover, display and control any IoT

device that conforms to the DHAP protocol suite.


Acknowledgment

This research was supported by an Australian Government Research Training Program Scholarship.

REFERENCES

- [1] D. C. Campbell and D. R. Thompson, "Appliance control," US4200862 A, 29-Apr-1980.
- [2] Dave Rye, "Dave Rye @ X10: My life at x10," 01-Oct-1999. [Online]. Available: <https://www.hometoys.com/content.php?url=/htinews/oct99/articles/rye/rye.htm>. [Accessed: 04-Apr-2018].
- [3] S. Nasrin and P. J. Radcliffe, "Novel protocol enables DIY home automation," in proceeding of *Telecommunication Networks and Applications Conference (ATNAC), 2014 Australasian*, Australia, Nov. 26-28, 2014, pp. 212–216. doi: [10.1109/ATNAC.2014.7020900](https://doi.org/10.1109/ATNAC.2014.7020900)
- [4] S. Nasrin and P. J. Radcliffe, "A Novel Three Stage Network Joining Protocol for Internet of Things based Home Automation Systems," *Computer Communication & Collaboration*, vol. 4, no. 3, pp. 1–11, 2016. doi: [2292-1036-2016-03-001-68](https://doi.org/10.2292-1036-2016-03-001-68)
- [5] T. N. E. Steane and P. J. Radcliffe, "An enhanced implementation of a novel IoT joining protocol," in proceeding of *26th International Telecommunication Networks and Applications Conference (ITNAC)*, New Zealand, Dec. 7-9, 2016, pp. 22–25. doi: [10.1109/ATNAC.2016.7878776](https://doi.org/10.1109/ATNAC.2016.7878776)
- [6] T. N. E. Steane and P. J. Radcliffe, "An Evaluation and Enhancement of a Novel IoT Joining Protocol," *Australian Journal of Telecommunications and the Digital Economy*, vol. 5, no. 2, 2017. doi: [10.18080/ajtde.v5n2.92](https://doi.org/10.18080/ajtde.v5n2.92)
- [7] T. N. E. Steane and P. J. Radcliffe, "A universal iot joining protocol for DIY applications," in proceeding of *27th International Telecommunication Networks and Applications Conference (ITNAC)*, Australia, Nov. 22-24, 2017, pp. 1–3. doi: [10.1109/ATNAC.2017.8215360](https://doi.org/10.1109/ATNAC.2017.8215360)
- [8] UPnP Forum, "UPnP Device Architecture 2.0," 2015. [Online]. Available: <http://upnp.org/specs/arch/UPnP-arch-DeviceArchitecture-v2.0.pdf>.
- [9] Verma, P.K., Verma, R., Prakash, A., "Machine-to-Machine (M2M) communications: A survey," *Journal of Network and Computer Applications*, vol. 66, pp. 83–105, 2016. doi: [10.1016/j.jnca.2016.02.016](https://doi.org/10.1016/j.jnca.2016.02.016)
- [10] M. Can Filibeli, O. Ozkasap, and M. Reha Civanlar, "Embedded web server-based home appliance networks," *Journal of Network and Computer Applications*, vol. 30, no. 2, pp. 499–514, 2007. doi: [10.1016/j.jnca.2006.04.001](https://doi.org/10.1016/j.jnca.2006.04.001)
- [11] H. Zhang, F.-Y. Wang, and Y. Ai, "An OSGi and agent based control system architecture for smart home," in proceeding of *2005 IEEE Networking, Sensing and Control*, USA, March 19-22, 2005, pp. 13–18. doi: [10.1109/ICNSC.2005.1461152](https://doi.org/10.1109/ICNSC.2005.1461152)
- [12] C. W. Hsu, S. T. Cheng, and C. F. Chen, "Widget-based framework for web service discovery on multiple home social network," in proceeding of *2011 IEEE International Conference on Granular Computing*, Taiwan, Nov. 8-11, 2011, pp. 250–255. doi: [10.1109/GRC.2011.6122603](https://doi.org/10.1109/GRC.2011.6122603)
- [13] J. Zhang, Z. Wang, Z. Yang, and Q. Zhang, "Proximity based IoT device authentication," in proceeding of *IEEE INFOCOM 2017 IEEE Conference on Computer Communications*, USA, May 1-4, 2017, pp. 1–9. doi: [10.1109/INFOCOM.2017.8057145](https://doi.org/10.1109/INFOCOM.2017.8057145)
- [14] M. Lee, Y. Kim, and Y. Lee, "A home cloud-based home network auto-configuration using SDN," in proceeding of *12th International Conference on Networking, Sensing and Control*, Taiwan, April 9-11, 2015, pp. 444–449. doi: [10.1109/ICNSC.2015.7116078](https://doi.org/10.1109/ICNSC.2015.7116078)
- [15] S. M. Kim, H. S. Choi, and W. S. Rhee, "IoT home gateway for auto-configuration and management of MQTT devices," in proceeding of *2015 IEEE Conference on Wireless Sensors (ICWiSe)*, Malaysia, Aug. 24-26, 2015, pp. 12–17. doi: [10.1109/ICWiSe.2015.7380346](https://doi.org/10.1109/ICWiSe.2015.7380346)
- [16] A. Kamilaris, A. Pitsillides, and V. Trifa, "The Smart Home Meets the Web of Things," *Int. J. Ad Hoc Ubiquitous Comput.*, vol. 7, no. 3, pp. 145–154, 2011. doi: [10.1504/IJAHUC.2011.040115](https://doi.org/10.1504/IJAHUC.2011.040115)
- [17] A. Kamilaris, V. Trifa, and A. Pitsillides, "HomeWeb: An application framework for Web-based smart homes," in proceeding of *18th International Conference on Telecommunications*, Cyprus, May 8-11, 2011, pp. 134–139. doi: [10.1109/CTS.2011.5898905](https://doi.org/10.1109/CTS.2011.5898905)
- [18] S. K. Datta and C. Bonnet, "Connect and Control Things: Integrating Lightweight IoT Framework into a Mobile Application," in proceeding of *9th International Conference on Next Generation Mobile Applications, Services and Technologies*, UK, Sept. 9-11, 2015, pp. 66–71. doi: [10.1109/NGMAST.2015.23](https://doi.org/10.1109/NGMAST.2015.23)
- [19] S. K. Datta, C. Bonnet, A. Gyrard, R. P. F. da Costa,

- and K. Boudaoud, "Applying Internet of Things for personalized healthcare in smart homes," in proceeding of *24th Wireless and Optical Communication Conference (WOCC)*, Taiwan, Oct. 23-24, 2015, pp. 164–169.
doi: [10.1109/WOCC.2015.7346198](https://doi.org/10.1109/WOCC.2015.7346198)
- [20] Z. Shelby, K. Hartke, and C. Bormann, "The Constrained Application Protocol (CoAP)." [Online]. Available: <https://tools.ietf.org/html/rfc7252>. [Accessed: 28-Jul-2016].
- [21] M. Alaa, A. A. Zaidan, B. B. Zaidan, M. Talal, and M. L. M. Kiah, "A review of smart home applications based on Internet of Things," *Journal of Network and Computer Applications*, vol. 97, pp. 48–65, 2017.
doi: [10.1016/j.jnca.2017.08.017](https://doi.org/10.1016/j.jnca.2017.08.017)
- [22] D. F. S. Santos, H. O. Almeida, and A. Perkusich, "A personal connected health system for the Internet of Things based on the Constrained Application Protocol," *Computers & Electrical Engineering*, vol. 44, pp. 122–136, 2015.
doi: [10.1016/j.compeleceng.2015.02.020](https://doi.org/10.1016/j.compeleceng.2015.02.020)
- [23] M. Castro, A. J. Jara, and A. F. Skarmeta, "Enabling end-to-end CoAP-based communications for the Web of Things," *Journal of Network and Computer Applications*, vol. 59, pp. 230–236, 2016.
doi: [10.1016/j.jnca.2014.09.019](https://doi.org/10.1016/j.jnca.2014.09.019)
- [24] B. Boehm and V. R. Basili, "Top 10 list [software development]," *Computer*, vol. 34, no. 1, pp. 135–137, 2001.
doi: [10.1109/2.962984](https://doi.org/10.1109/2.962984)
- [25] H. König, *Protocol Engineering*. Springer Science & Business Media, 2012.
- [26] W. W. Royce, "Managing the development of large software systems: concepts and techniques," in proceedings of the *9th international conference on Software Engineering*, pp. 328–338, 1970.
- [27] J. Martin, *Rapid Application Development*. Macmillan Publishing Company, 1991.
- [28] B. W. Boehm, "A spiral model of software development and enhancement," *Computer*, vol. 21, no. 5, pp. 61–72, 1988.
doi: [10.1109/2.59](https://doi.org/10.1109/2.59)
- [29] B. W. Boehm, "Verifying and Validating Software Requirements and Design Specifications," *IEEE Software*, vol. 1, no. 1, pp. 75–88, 1984.
doi: [10.1109/MS.1984.233702](https://doi.org/10.1109/MS.1984.233702)
- [30] K. Beck, *Extreme Programming Explained: Embrace Change*. Addison-Wesley, 2000.
- [31] Object Management Group, *OMG Unified Modeling Language™ (OMG UML), Superstructure*. 2012.
- [32] R. Grammes and R. Gotzhein, "SDL Profiles – Formal Semantics and Tool Support," *Fundamental Approaches to Software Engineering*, pp. 200–214, 2007.
doi: [10.1007/978-3-540-71289-3_17](https://doi.org/10.1007/978-3-540-71289-3_17)
- [33] T. Murata, "Petri nets: Properties, analysis and applications," *Proceedings of the IEEE*, vol. 77, no. 4, pp. 541–580, 1989.
doi: [10.1109/5.24143](https://doi.org/10.1109/5.24143)
- [34] N. Kushik, J. López, A. Cavalli, and N. Yevtushenko, "Improving Protocol Passive Testing through 'Gedanken' Experiments with Finite State Machines," in proceeding of *2016 IEEE International Conference on Software Quality, Reliability and Security (QRS)*, Austria, Aug. 1-3, 2016, pp. 315–322.
doi: [10.1109/QRS.2016.43](https://doi.org/10.1109/QRS.2016.43)
- [35] IEEE Computer Society, "IEEE Standard for Information technology–Telecommunications and information exchange between systems Local and metropolitan area networks–Specific requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications," *IEEE Std 802.11-2012 (Revision of IEEE Std 802.11-2007)*, pp. 1–2793, 2012.
doi: [10.1109/IEEESTD.2012.6178212](https://doi.org/10.1109/IEEESTD.2012.6178212)
- [36] Vu, Hai and Sakurai, Taka, "Collision probability in saturated IEEE 802.11 networks," in proceeding of *Australian Telecommunication Networks and Applications Conference*, Australia, Dec., 2006.
- [37] P. Bartolomeu, M. Alam, J. Ferreira, and J. Fonseca, "Survey on low power real-time wireless MAC protocols," *Journal of Network and Computer Applications*, vol. 75, pp. 293–316, 2016.
doi: [10.1016/j.jnca.2016.09.004](https://doi.org/10.1016/j.jnca.2016.09.004)

 ©The Authors. This work is licensed under the Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License.