CORA Cork Open Research Archive
Cartlann Taighde Oscailte Chorcaí

Title	Home automation system coordinator replacement with one-touch network recommissioning
Authors	Farooq, Muhammad Omer;Wheelock, Ian;Pesch, Dirk
Publication date	2020-06-15
Original Citation	Farooq, M. O., Wheelock, I. and Pesch, D. (2020) 'Home automation system coordinator replacement with one-touch network recommissioning', IEEE Consumer Electronics Magazine, 9(6), pp. 57-65. https://doi.org/10.1109/MCE.2020.3002483
Type of publication	Article (peer-reviewed)
Link to publisher's version	10.1109/MCE.2020.3002483
Rights	© 2020, IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.
Download date	2026-10-07 17:12:11
Item downloaded from	https://hdl.handle.net/10468/10487

UCC
University College Cork, Ireland
Coláiste na hOllscoile Corcaigh

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/MCE.2020.3002483, IEEE Consumer Electronics Magazine

Home Automation System Coordinator Replacement With One-Touch Network Recommissioning

Muhammad Omer Farooq
Carleton University, Canada

Dirk Pesch
University College Cork, Ireland

Ian Wheelock
Commscope, Ireland

***Abstract*—The home automation systems (HASs) and products market has seen significant growth over the past decade. The current HAS architecture is based on a centralized network coordinator to form, manage, and supervise the network system. Each smart home device is typically manually on-boarded onto the network through the coordinator. If the network coordinator fails or a user wishes to change the technology provider, the HAS needs to be recommissioned. Recommissioning can be a tedious task, as it involves manual on-boarding of possibly a multitude of existing smart home devices onto the new network coordinator. This tight coupling between the devices and network coordinator is seen as a significant road-block for further expansion of the HAS market. Here, we present a HAS proxy that incorporates mechanisms to eliminate tight coupling between smart home devices and a centralized network coordinator. Our HAS proxy approach enables one-touch on-boarding of existing smart home devices onto any new network coordinator with devices being oblivious of the process rendering network recommissioning unnecessary. We evaluated the concept in a simulator for HAS along with our HAS proxy.**

INTRODUCTION

A networked system formed by smart devices to automate a range of tasks in a home environment is typically refereed to as a home automation system (HAS). The market for HASs has been growing significantly over the past decade. Hence, the market development has been leading to a multitude of networking technologies to support smart home applications. ZigBee [1], Z-Wave [2], and Thread [3] are among the most noticeable networking technologies in this domain. These technologies use a central network coordinator that forms, manages, and supervises a HAS. Each networking technology comes with a procedure to on-board a device onto a HAS network. To initiate the on-boarding procedure a manual interaction with the smart device and coordinator is required. Hence, commissioning such a network can become a tedious task.

To highlight the issue caused by tight coupling between smart devices and a network coordinator, let us consider a HAS that consists of even a moderate number of devices. Initially, to commission a network, manual interaction with each device is required. If after operating for some time, the coordinator malfunctions, the HAS will stop functioning. To restore the HAS functionality a new network coordinator supporting the same technology will need to be installed, and a user has to again manually on-board devices onto the new coordinator. This situation can also happen if a user wants to switch to a new HAS provider as often a change in telecommunications service provider results in changing the HAS network coordinator. The scenario is depicted in Fig. 1. This tightly coupled design can become an obstacle for further HAS market development.

Here we present a HAS Proxy that incorporates mechanisms to avoid network recommissioning in case an existing network coordinator needs to be replaced. Our HAS Proxy can on-board devices onto the new coordinator using a one-touch on-boarding procedure where the individual smart devices are oblivious of the process. The HAS Proxy overcomes the tight coupling between smart devices/end devices (EDs) and the centralized network coordinator, hence it has a potential to make the HAS consumer market more open.

2162-2248 (c) 2020 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.
Authorized licensed use limited to: UNIVERSITY COLLEGE CORK. Downloaded on July 09,2020 at 11:45:38 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/MCE.2020.3002483, IEEE Consumer Electronics Magazine

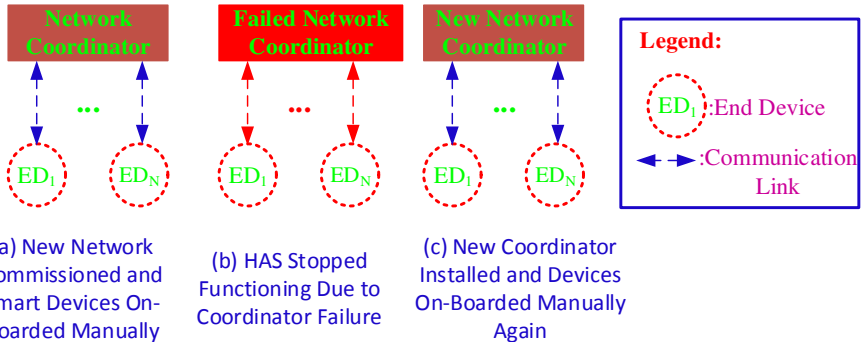


Fig. 1. Tight Coupling between Smart Devices and HAS Network Coordinator

Home Automation System Coordinator Replacement Methods

Smart home has a number of use cases, such as intelligent home energy management [4], intelligent home care [5], home security [6] to name, but a few. A large number of home automation system networking technologies exist [7]. Among the existing technologies, ZigBee, Bluetooth, Z-Wave, and Thread are most prominent technologies. These technologies come with their own networking protocol, and they form a network using a centralized coordinator. Hence, they are tight closely to a particular HAS network coordinator. If the coordinator fails, it will need to be replaced. HAS coordinator replacement methods can be categorised as follows: (i) single backup coordinator, and (ii) multiple backup coordinators.

A. Single Backup Coordinator Methods

Replacements methods presented in [8], [9], and [10] are based on single backup coordinator. This category of coordinator replacement method is based on the following principles: (i) primary coordinator's MAC address, security keys, device bindings, routing and neighbor tables are backed up on a secondary coordinator, (ii) a back up coordinator periodically sends data packets to the primary coordinator, and it infers the coordinator failure if ACKs for the transmitted packets are not received, and (iii) in case of the primary coordinator failure, the backup coordinator takes on the role of primary coordinator and restore network functionality using the backed up network information.

Multiple Backup Coordinator Methods

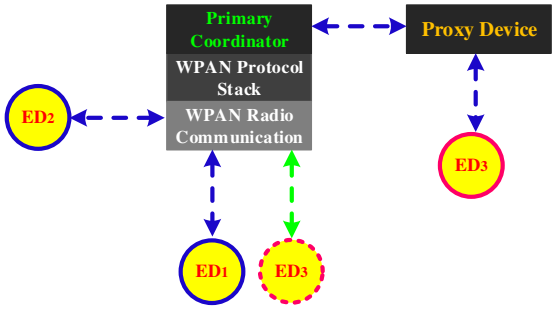
Replacement methods presented in [11], and [12] are based on multiple backup coordinators. These methods work on the following principles: (i) each backup coordinator maintains a live data connection with the primary coordinator, (ii) the primary coordinator's failure is inferred if ACKs for the transmitted packets are not received, and (iii) in case of the primary coordinator failure one of the backup coordinator is selected as the primary coordinator (usually backup coordinator's priority is used to select a new primary coordinator).

Where Do Current Methods Fail

The state-of-the-art solutions to replace a network coordinator fail in a scenario where a user wishes to change the HAS service provider. Suppose that a customer is using a HAS from company A, and the company uses one of the backup coordinator schemes to replace the primary network coordinator in case that one fails. However, due to some reason a user needs to switch to another HAS provided by company B. In order to use Company B's HAS, the user installs a new network coordinator from company B. As company A's coordinator shares its network information with its own company's backup coordinator, therefore company B's coordinator cannot obtain data from company A's coordinator. This results in vendor lock-in. Here, the network needs to be recommissioned, which tends to be road-block for most users to change to a new or more innovative technology. Along with with interoperability with HAS networking protocols [7], this is one of the main roadblocks for innovation in this space.

2162-2248 (c) 2020 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.
Authorized licensed use limited to: UNIVERSITY COLLEGE CORK. Downloaded on July 09,2020 at 11:45:38 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/MCE.2020.3002483, IEEE Consumer Electronics Magazine



The same device ED_3 on-boarded onto the primary coordinator's network

Fig. 2. HAS Architecture Based on Our HAS Proxy

Existing network coordinators for HAS come from a variety of companies, such as Amazon, Google, Philips, Nest, Samsung. These companies' systems do not have a backup mechanism to replace a faulty network coordinator without requiring a network recommissioning. For example, an existing HAS uses a Philips coordinator. if the user wants to replace the Philips coordinator with a coordinator that comes from Nest, the user needs to recommission the network using the Nest coordinator, even if Philips's HAS uses a coordinator backup method (although this feature is currently not available in any commercial product). In the following, we present a proxy HAS that can eliminate the stated problem.

HOME AUTOMATION SYSTEM PROXY

In this section, we present our concept for a HAS proxy that can eliminate the stated issues with network recommissioning. In general, the following are distinguishing features of our solution:

- Ability to form a wireless personal area network (WPAN), and on-boards smart home devices onto a network coordinator.
- One-touch network recommissioning.
- Incorporates method to replace faulty HAS proxy.
- It works with any existing networking protocol stacks for smart homes, such as ZigBee, Z-Wave, Thread, and Bluetooth.

Fig. 2 shows a HAS network architecture based on our HAS proxy. Our HAS proxy is a software component that can easily run over a relatively low-end computing device, such as, Raspberry Pi. To run our HAS proxy a computing device should support relevant WPAN radio and networking protocol stack. For example, for a ZigBee-based HAS it must include IEEE 802.15.4 radio support and implementation of ZigBee networking protocol stack. Moreover, to facilitate communication over the Internet the computing device should have Ethernet and/or WiFi connection along with TCP/IP implementation. The relevant functionality of our HAS proxy can be implemented using mentioned available hardware and software components. As the HAS proxy uses any existing protocol stack, hence its capacity is limited by an underlying protocol's design. In this article, we use ZigBee protocol stack to elaborate the HAS proxy functionality. In the short addressing mode, ZigBee uses 2-byte for assigning network layer addresses to nodes. Hence, in this case the maximum number of nodes that the HAS proxy can handle is 2^{12}. Our HAS proxy is a software proxy implementation, therefore it does not eliminate the need for a primary network coordinator.

Forming a Network and On-Boarding Devices onto the Primary Coordinator

As shown in Fig. 3(a) our HAS proxy acts as a man-in-the-middle between a smart home device and the primary network coordinator. Our HAS proxy implements some of a network coordinator's functionalities, however by its nature it is not a network coordinator. It has the capabilities to form, manage, and supervise a network based on a certain networking protocol stack, for example ZigBee. The HAS proxy periodically broadcasts a beacon frame with its personal area network (PAN) ID. Any new smart device is initially manually on-boarded onto the HAS proxy. During the on-boarding process the sequence of messages exchanged between the smart device and the HAS proxy is shown in Fig. 3(b). The sequence of messages shown in Fig. 3 are standard ZigBee messages. At the end of the on-boarding process with the HAS proxy, the smart device has acquired the following information: PAN ID, smart device network address, and link security key. Similarly, the HAS proxy has stored the smart device's MAC address, network address, and link key in its internal data structure.

2162-2248 (c) 2020 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.
Authorized licensed use limited to: UNIVERSITY COLLEGE CORK. Downloaded on July 09,2020 at 11:45:38 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/MCE.2020.3002483, IEEE Consumer Electronics Magazine

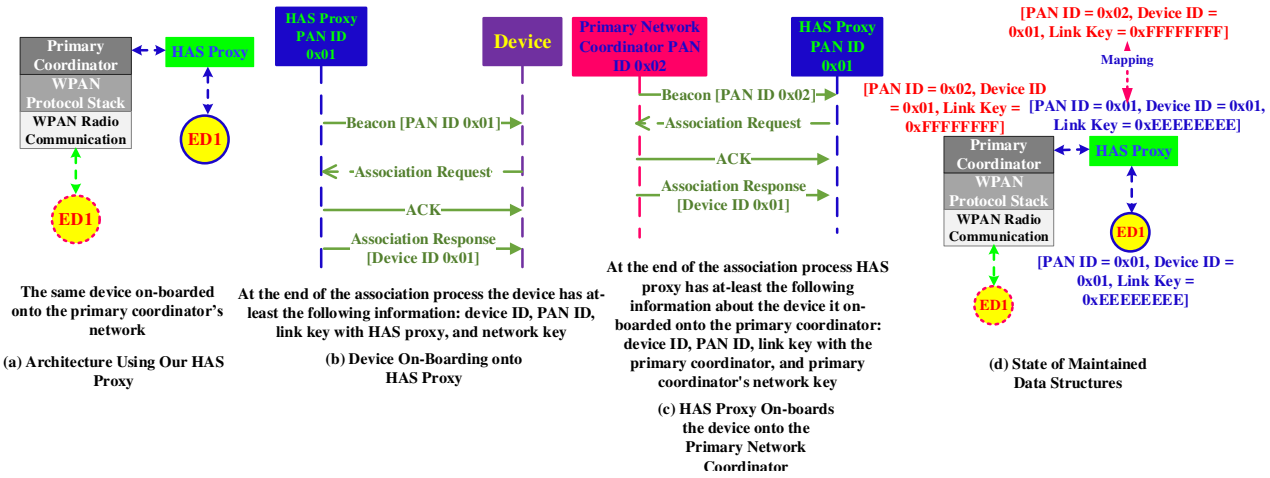


Fig. 3. Network Architecture and Device On-Boarding Using HAS Proxy

As the HAS proxy acts as a man-in-the-middle, the next step in the process is that the proxy on-boards the smart device onto the primary coordinator, and this step is oblivious to the smart device. The sequence of messages exchanged between the proxy and the primary coordinator to on-board the smart device onto the primary coordinator is shown in Fig. 3(c). As this process does not require any human interaction, hence this process is orders of magnitude faster compared to manually on-boarding of a device onto the proxy. Thus, the additional time required by the proxy to on-board devices onto the primary coordinator is negligible. At the end of this step, the HAS proxy would have obtained the following information: PAN ID, smart device network address on the primary coordinator's network, and link security key for the smart device. Now, the proxy keeps the mapping between the smart device's network credentials issued by the primary coordinator and the credentials issued by the proxy to the smart device in its internal data structure. The proxy can also maintain a copy of this data structure in the cloud. As the proxy has to on-board multiple smart devices onto the primary coordinator, it executes this on-boarding process in a promiscuous mode using the actual MAC address of the device already on-boarded onto the proxy. Fig. 3(d) shows the state of data structures stored at the proxy, smart device, and the primary coordinator. Apart from the shown data structures, the proxy and the primary coordinator also maintain device binding information.

Communication Among Devices Using HAS Proxy

To illustrate the way communication takes place in the presence of a HAS proxy, let us consider a scenario in which a HAS consists of a smart light and a switch. The proxy on-boards the smart devices onto a network identified with PAN ID 0x01. Afterwards, it automatically on-boards the devices on the primary coordinator's network identified with PAN ID 0x02. The on-boarding process is the same as described in the preceding section. At the end of the on-boarding process, the HAS proxy maintains mappings of the credentials assigned to the smart devices by the proxy and the primary coordinator as shown in Fig. 4.

Fig. 4 shows the exchange of message between two devices using our HAS proxy. A switch sends a TURN ON command to the light, the proxy receives message, and sends an acknowledgment back to the switch. Afterwards, using its internal device credential mapping data structure, it replaces PAN ID, source ID, and destination ID in the message header, and forwards the message to the primary coordinator. After consulting its device association table, the primary coordinator forwards the message to the light. As the light's PAN ID is 0x01 and the PAN ID in the message header is 0x02, therefore the light ignores the message. However, as the proxy is working in the promiscuous mode, it captures the message, and forwards the message to the light after replacing PAN ID, source ID, and destination ID in the message header.

2162-2248 (c) 2020 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.
Authorized licensed use limited to: UNIVERSITY COLLEGE CORK. Downloaded on July 09,2020 at 11:45:38 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/MCE.2020.3002483, IEEE Consumer Electronics Magazine

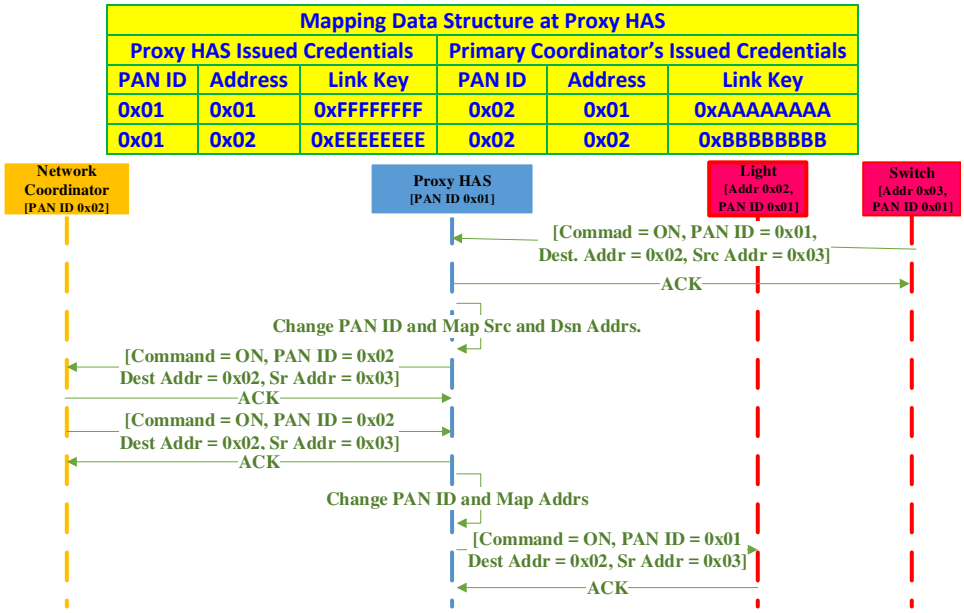


Fig. 4. Communication Among Devices

In general, our HAS proxy relays messages, but before relaying a message it replaces the PAN ID, source node ID, destination node ID in the MAC layer message header with the information maintained in the credential mapping data structure. Moreover, it relays the message using the device specific link key maintained at the proxy. It is important to note here that whenever the proxy makes any changes to the message header, it has to recalculate the message checksum. From the above description, it appears that the proxy has all the information to relay the message directly to the destination node rather than going through the primary coordinator, as it can possibly also maintain associations among devices. However, the primary network coordinator may employ different policies related to billing, network statistics, etc, therefore the communication has to go through the primary coordinator. Using our HAS proxy each message has to be transmitted twice. This increases data activity, however mostly HASs generate low amounts of data, hence additional transmissions will not have any negative impact on the performance and timely execution of commands.

Restoring Network Functionality After Replacing a Primary Network Coordinator

If an existing primary network coordinator malfunctions or a user wishes to change a HAS service provider, the existing primary network coordinator needs to be replaced. These are the scenarios where our proxy HAS becomes really useful as it can avoid network recommissioning and manual onboarding of already existing devices onto the newly installed primary network coordinator. In such scenarios, a user can activate our HAS proxy one-touch network recommissioning mechanism. As soon as the HAS proxy's one-touch network recommissioning mechanism is activated, the proxy sequentially on-boards each device in the proxy's database onto the newly installed primary network coordinator, and it also updates its credentials mapping data structure. The HAS proxy's credentials mapping data structure must be updated as the newly installed primary coordinator may have a different PAN ID, and it may also assign different addresses and link security keys to devices compared to what was assigned to them by the previous primary coordinator. The sequence of messages exchanged between the proxy and the new coordinator to onboard a device onto the new primary coordinator is shown in Fig. 3(c). Existing smart devices are

2162-2248 (c) 2020 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.
Authorized licensed use limited to: UNIVERSITY COLLEGE CORK. Downloaded on July 09,2020 at 11:45:38 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/MCE.2020.3002483, IEEE Consumer Electronics Magazine

oblivious of this whole process as for them our proxy is completed transparent and appears to be the network coordinator. Similarly, our proxy managed to recommission the network as it maintained all necessary information about the smart devices in its internal data structures. The proxy also maintains previously existing devices' associations, therefore if those associations are required by the new primary coordinator, it shares them with the new coordinator. The time required to recommission the network will be substantially lower compared to manual recommissioning of the network.

HAS Proxy Functionality in the Presence of an Existing HAS

It is possible that our HAS proxy is brought into an already existing HAS. In this case, the proxy does not on-board devices that are already on-boarded onto the primary coordinator's network. Fig. 2 shows that before the proxy was brought in the network, the primary coordinator already on-boarded devices ED1 and ED2. However, ED3 was brought into the network after the proxy was brought into the network. Therefore, the proxy on-boards ED3, and afterwards it automatically on-boards ED3 onto the primary network coordinator, and maintains the required network credential mapping in its internal data structure. In this case, if the primary coordinator needs to be replaced, the proxy can only on-board ED3 automatically onto any new primary coordinator, hence ED1 and ED2 need manual on-boarding. Any communication among devices that are directly on-boarded onto the primary coordinator takes place as per an underlying communication standard/protocol stack. However, if any of the device involved in communication includes a device that was on-boarded onto the primary coordinator through the proxy, the communication takes place as described in the previous sub-sections.

Replacing a Faulty HAS Proxy

It is possible that the HAS proxy develops a fault, hence it needs to be replaced. In such an event, the network stops functioning. Here, we also present a method to replace a faulty proxy. Our proxy maintains essential information about itself, such as its MAC address, PAN ID, etc along each device information, and each device mapped information in the cloud. Fig. 5 represents a network architecture that uses our proxy and our methods to replace primary network coordinator and a faulty proxy. In case our HAS proxy is replaced by our new HAS proxy, the user interacts with the new proxy to instruct the proxy to fetch the stored PAN's settings and information related to devices from the cloud. Once, the new proxy fetches the required information from the cloud, the network becomes operational again. As the mentioned process is activated through one-touch, hence the process can execute really fast.

Implementation and Verification of the Proposed System

We have implemented a ZigBee HAS simulator, and it also contains a software-based implementation of our HAS proxy. The simulator was implemented in the Java programming language. The simulator contains an implementation of the following aspects of the ZigBee networking protocol stack: IEEE802.15.4 MAC layer framer, application support sub-layer, ZigBee cluster library frame, ZigBee simple descriptor, device association table, and group/scene table. Apart from the primary network coordinator and our HAS proxy the simulator supports the following smart devices: light bulb, dimmable light, door lock, toggle switch, dimmer, and door lock handler. Our simulator also implements ZigBee clusters required to support the functionality of the stated devices.

To use the simulator, a user needs to provide a configuration file to the simulator. A sample configuration file is shown in Fig. 6. As shown in the figure, the configuration file has 5 sections: (i) Devices, (ii) Associations, (iii) Groups, (iv) ChangeCoordinator, and (v) Simtime. The devices section lists the type and number of devices for each listed type of device to be created in a simulated HAS. The associations section lists the association among the devices. The groups section lists the groups in a simulated HAS. The ChangeCoordinator section lists the time at which the primary coordinator needs to be replaced, the time is in seconds, and the Simtime section lists the total duration of simulation in seconds. In the simulation

2162-2248 (c) 2020 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.
Authorized licensed use limited to: UNIVERSITY COLLEGE CORK. Downloaded on July 09,2020 at 11:45:38 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/MCE.2020.3002483, IEEE Consumer Electronics Magazine

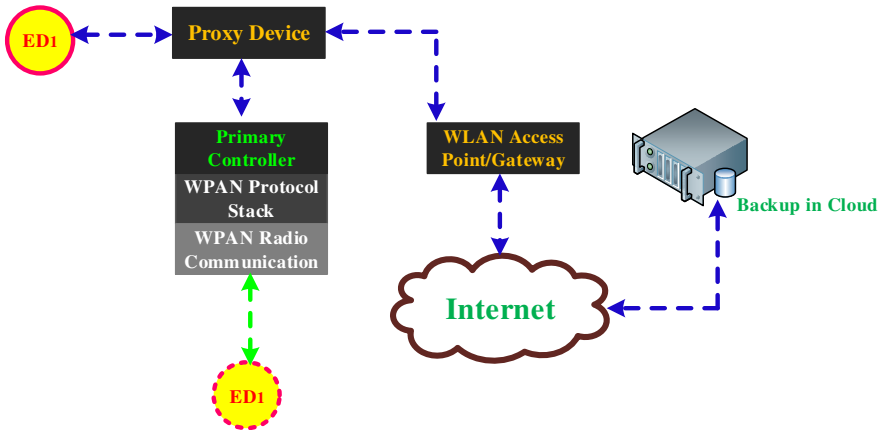


Fig. 5. Complete HAS Architecture with Proxy HAS

configuration file, ChangeCoordinator section is optional. In our current simulator implementation devices, such as switch, dimmer, and door lock handler send commands at a random time intervals.

We ran a simulation using the configuration file shown in Fig. 6. Our simulator creates the following ZigBee devices: 7 lights, 6 switches, 1 door lock, 1 door lock handler, 1 dimmer, and 1 dimmable light. The simulator creates associations between the following devices: (i) switch 1 $\leftrightarrow$ light 1, (ii) switch 2 $\leftrightarrow$ light 2, (iii) doorhandler 1 $\leftrightarrow$ doorlock 1, and (iv) dimmer 1 $\leftrightarrow$ dimmablelight 1. The first group section in the configuration file creates a group among switch 3, light 3, and light 4. In this particular group configuration switch 3 controls both light 3 and light 4. Similarly, there are other groups sections in the file that create groups of different smart home devices. The Changecoordinator section, instructs the simulator to change the primary network coordinator at 900 seconds. Hence, at 900 simulation seconds our HAS proxy will automatically on-board the existing devices onto a new network coordinator. The Simtime section instructs the simulator to end the simulation after 2000 seconds. At the end of the simulation, the simulator outputs the number of control packets and data packets exchanged in the network.

The number of data packets exchanged in the network in the presence of our HAS proxy was exactly two times the number of packets exchanged when there was no HAS proxy in the HAS. This is due to the fact that each device is on-boarded onto our HAS proxy, therefore devices send their packets to our proxy , which relays the messages to the primary network coordinator and vice versa. The introduction of a HAS proxy results in relatively higher data activity in the network, however it is manageable for the following reasons: (i) packets exchanged between HAS proxy and primary network coordinator can take place on a separate communication channel, (ii) 1x extra messages are exchanged between the proxy and primary coordinator, and no extra messages are transmitted by the smart devices, hence their power consumption with and without our proxy device is the same, and (iii) our proxy device and primary network coordinator are mains powered, hence extra power consumption by these devices is not an issue. It is also important to note here that when a primary network coordinator is replaced, the number of control messages exchanged to on-board already existing devices onto the new coordinator is same with and without our HAS proxy, however the proxy eliminates the need to manually on-board all end devices on the new coordinator.

CONCLUSIONS

We presented a Home automation System proxy and accompanying methods to avoid network recommissioning when either an existing network coordinator malfunctions or a user switches to a new network service provider. In such an event, our proxy can seamlessly on-board existing devices onto the new coordinator. Hence, it avoids a tedious manual on-boarding process. Moreover, it also achieves seamless communication between devices

2162-2248 (c) 2020 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.
Authorized licensed use limited to: UNIVERSITY COLLEGE CORK. Downloaded on July 09,2020 at 11:45:38 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/MCE.2020.3002483, IEEE Consumer Electronics Magazine


```
Devices:
Zigbee, Light, 7
Zigbee, Switch, 6
Zigbee, doorlock, 1
Zigbee, doorhandler, 1
Zigbee, dimmer, 1
Zigbee, dimmablelight, 1

Associations:
Zigbee, Switch, 1, Zigbee, Light, 1
Zigbee, Switch, 2, Zigbee, Light, 2
Zigbee, doorhandler, 1, Zigbee, doorlock, 1
Zigbee, dimmer, 1, Zigbee, dimmablelight, 1

Groups:
Zigbee, Switch, 3
Zigbee, Light, 3
Zigbee, Light, 4

Groups:
Zigbee, Switch, 4
Zigbee, Light, 5
Zigbee, Light, 6

Groups:
Zigbee, Switch, 5
Zigbee, Light, 7

ChangeCoordinator:
900

Simtime:
2000
```

Fig. 6. A Sample Simulation Configuration File

through the primary coordinator. To accomplish its functionality, the proxy maintains a device mapping data structure and backs-up the data structure in the cloud as well. In case the proxy malfunctions, a new proxy can fetch the data from the cloud and the network becomes operational again straight away. We have evaluated the approach in a Java simulator that can simulate a ZigBee based HAS along with the functionality of our proxy.

Acknowledgments - this research has received support from Science Foundation Ireland under Grant Number 13/RC/2077.

REFERENCES

[1] "ZigBee Cluster Library Specification," [online], http://www.zigbee.org/wp-content/uploads/2014/10/07-5123-06-zigbee-cluster-library-specification.pdf, (Last accessed on 6^{th} June, 2020).

[2] "Z-Wave Protocol Overview," [online], https://wiki.ase.tut.fi/courseWiki/images/9/94/SDS10243_2_Z_Wave_Protocol_Overview.pdf, (Last accessed on 6^{th} June, 2020).

[3] "THREAD," [online], https://www.threadgroup.org/, (Last accessed on 6^{th} June, 2020).

[4] M. Yousefi, A. Hajizadeh, and M. N. Soltani, "A Comparison Study on Stochastic Modeling Methods for Home Energy Management Systems," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 8, pp. 4799–4808, 2019.

[5] T. Tsai, K. Zhang, and Y. Tsai, "Design and Implementation on Intelligent Homecare Appliance System," *IEEE Consumer Electronics Magazine*, vol. 9, no. 1, pp. 16–21, 2020.

[6] Z. Huang, L. Zhang, X. Meng, and K. R. Choo, "Key-Free Authentication Protocol Against Subverted Indoor Smart Devices for Smart Home," *IEEE Internet of Things Journal*, vol. 7, no. 2, pp. 1039–1047, 2020.

[7] M. O. Farooq, I. Wheelock, and D. Pesch, "IoT-Connect: An Interoperability Framework for Smart Home Communication Protocols," *IEEE Consumer Electronics Magazine*, vol. 9, no. 1, pp. 22–29, Jan 2020.

[8] D. Scazzoli, A. Kumar, N. Sharma, M. Magarini, and G. Verticale, "A Novel Technique for ZigBee Coordinator Failure Recovery and its Impact on Timing Synchronization," in *IEEE* 27^{th} *Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, 2016, pp. 1–5.

[9] R. Klln and A. Zimmermann, "Transparent Coordinator Failure Recovery for ZigBee Networks," in *IEEE Conference on Emerging Technologies & Factory Automation*, 2009, pp. 1–8.

[10] S. HU, "ZigBee Network System and the Backup Method Thereof," Patent US 20 170 163 521A1, June 8, 2017.

[11] "Hot Backup Method for Coordinator in ZigBee Networks," Patent CN 102 917 384A, February 6, 2013.

[12] S. Oh and W. Kim, "Coordinator Switching Method and Communication System Using Same," Patent US 20 040 255 001A1, December 6, 2004.

Muhammad Omer Farooq is a Research Professor at Carleton University Canada. Contact him at omer.farooq@sce.carleton.ca.

Ian Wheelock is a Engineering Fellow at CommScope, Ireland. Contact him at ian.wheelock@commscope.com.

Dirk Pesch is a Professor in the School of Computer Science and IT at University College Cork in Ireland. Contact him at d.pesch@cs.ucc.ie.

2162-2248 (c) 2020 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.
Authorized licensed use limited to: UNIVERSITY COLLEGE CORK. Downloaded on July 09,2020 at 11:45:38 UTC from IEEE Xplore. Restrictions apply.