The Report Committee for Joshua David Tharp Certifies that this is the approved version of the following report:

Replication System for Low Power Internet of Things Devices

APPROVED BY

SUPERVISING COMMITTEE:

Supervisor:

_____

Vijay Garg

_____

Constantine Caramanis

Replication System for Low Power Internet of Things Devices

by Joshua David Tharp, B.S

Report

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

Master of Science in Engineering

The University of Texas at Austin

May 2017

Dedication

This report is dedicated to my Wife and Son who inspire me to keep learning.

Acknowledgments

I would like to thank Dr. Vijay Garg for his classes in Distributed Computing. I drew inspiration from his teachings for this report. I would also like to thank Dr. Christine Julien for her class in Mobile Computing which gave me the idea to use mobile devices in a new way.

Abstract

Replication System for Low Power Internet of Things Devices

Joshua David Tharp, M.S.E

The University of Texas at Austin, 2017

Supervisor: Vijay Garg

As technology evolves to where persistent and ubiquitous computing devices exist the possibilities for shared computing increases. The size and power requirements of Internet of Things(IoT) devices varies from each device. However, each of these devices contains processing power, storage, and network connectivity. Many of these devices passively communicate small pieces of data to a central device. For example, a motion sensor may report times of activity to a central hub or a phone could connect to a media device. Typically, these devices communicate through wireless technology such as Wi-Fi (IEEE 802.11) or Bluetooth (IEEE 802.15). With Bluetooth 4.0 there exists another option which is Bluetooth Low Energy(BLE). With BLE devices can more efficiently query and send data to other devices with minimal power requirements. This paper aims to expand IoT devices to support an advanced protocol for replication of data suitable for

low power devices communicating over BLE. Different methods of forming a network are considered for optimal replication of data using mesh network topologies. The report will also evaluate methods of data transfer using solely BLE or a combination of BLE and high speed high power protocol such as Bluetooth or Wi-Fi.

Table of Contents

List of Figures

INTRODUCTION

Always on networked devices are becoming more ubiquitous through our lives and in our homes. Recent years have seen the advancement of low power computing combined with the computing power available in small form factors. These recent advancements have enabled devices to present a small footprint where they are unobtrusive, quiet, and provide always on services to end users. Devices such as Internet of Things devices in our home automation system or cars, as well as devices such as Arduino and Raspberry Pi (along with many other similar devices). Each of these devices has the capability to communicate with one another but rarely do.

Previously to communicate meant using expensive network protocols such as Wi-Fi or Bluetooth which may not be ideal for devices on battery power, or cumbersome to setup each device to the network. With the introduction of Bluetooth Low Energy(BLE) in Bluetooth 4.0 and the upgrade to the protocol in the Bluetooth 5.0 release there exist a low power protocol that can communicate at distance. Some devices use this protocol as a handshake for other protocols such as Wi-Fi or Bluetooth, where other devices will use it for transferring instrument data to a central device.

There exists the opportunity to utilize this protocol for a more advanced purpose. These devices could participate in a dynamic network that replicates data over the low power BLE protocol. Replication can serve several purposes that utilize these devices such as sharing data on your phone, using the network for secure storage of information or replicating information relevant to the devices location and intended use.

LITERATURE REVIEW

Communicating over wireless networks between nodes is something that happens every day. Nodes utilize any number of wireless technologies, typically Wi-Fi, to join a to a central network and will be able to route messages to nodes in that network. As devices get smaller and are constrained by power or battery requirements then devices can no longer maintain a persistent connection to a central network using existing wireless technologies. To utilize these devices in a smart way to replicate data to each node would be a challenge to do efficiently. There must be a novel construction of a network between the devices and an algorithm of replication that works well in this environment.

There has been previous work in creating mesh networks intelligently. When connecting nodes to a network where each node cannot connect to every other node some type of routing and management of messages and nodes must occur. The authors in [1] implement an algorithm that replicates data using a root spanning tree to help create a mesh network and additional algorithms to then replicate data over the network. Most algorithms do not consider the Bluetooth Low Energy (BLE) protocol or are constrained by the hardware chosen to communicate with. As BLE creates its own set of problems for existing algorithms in wireless networks there has been research in using Bluetooth Classic to create mesh networks in [2]. The authors create a mixed Wi-Fi and Bluetooth network by creating bridge nodes to connect to two network technologies together.

This paper aims to use commercial products that can interop with each other, however there is a novel framework called FruityMesh [3]. FruityMesh uses Bluetooth Low Energy to create a mesh topology that can forward messages through the network. It creates small groups of nodes to try to create efficient groupings of nodes as to avoid to many connections and more efficient broadcasting of messages. It does all this with a Bluetooth Low Energy developer kit from Nordic. The kit supports advance features that allow devices to concurrently operate as a peripheral, central, advertiser and observer. FruityMesh flashes their own firmware onto the devices which allows them access to the bare metal hardware to implement their mesh network. The framework that FruityMesh provides is mature and open source. I hope to see their work or something similar eventually adopted into the Bluetooth specification.

APPROACH

BLUETOOTH TECHNOLOGY

The method of communication must use power efficient hardware to be useful for small devices that my run off batteries. There are a few technologies such as ZigBee and Z-Wave that are utilized in low power devices. However, these technologies are not as widespread as Bluetooth. Bluetooth can be found in the majority of personal computing devices such as laptops, phones, and computers. ABI Research estimates that by 2021 5 billion Bluetooth devices with Bluetooth Low Energy will have shipped [4]. Bluetooth is also used as a general-purpose network hardware for small devices such as Raspberry PI and Arduino. Bluetooth can be split into two technologies: Bluetooth and Bluetooth Low Energy. This research will focus on Bluetooth Low Energy for replicating data through a network. With Bluetooth advancing to support more concurrent connections, faster speeds, and longer range it becomes a technology that many see powering the IoT devices into the future.

Bluetooth has a long history of wireless communication among small devices. Many may be familiar with wireless devices such as headphones, speakers, and wireless devices in our cars and phones. These modes of communication rely on a constant connection to a device and work at ranges of up to 100 meters [5]. Under this type of usage Bluetooth saw wide adoption to where it now can be found in the majority of smart phones and small devices. Rarely has Bluetooth been exploited to create small mesh or pico networks that communicate over to perform a common goal. Part of the reason has

been the limitation of Bluetooth for simultaneous connections, data rate, and power requirements.

BLUETOOH LOW ENERGY

Bluetooth Low Energy (BLE) is a newer specification of the Bluetooth standard that provides a low power solution for devices to communicate. The standard was developed for smaller devices that still needed wireless range to communicate with other devices. Today communication typically happens between a peripheral device and central device. The peripheral device will periodically announce that it has information to share to those listening for it. It does this by writing data to a characteristic. This characteristic is a unsigned 16 bit integer UUID that identifies a unique topic that peripheral supports. This could be a temperature sensor writing data to the temperature UUID that central devices could then scan and read. Each characteristic advertises using a BLE packet. This packet identifies the characteristic and provides space for data to be transmitted. The amount of data that can be transmitted varies between chipsets as some manufactures provide additional support for larger packets. However, the specification in 4.0 and 4.0 defines the data size to be 31 bytes and in 4.2 this is extended to 257 bytes.

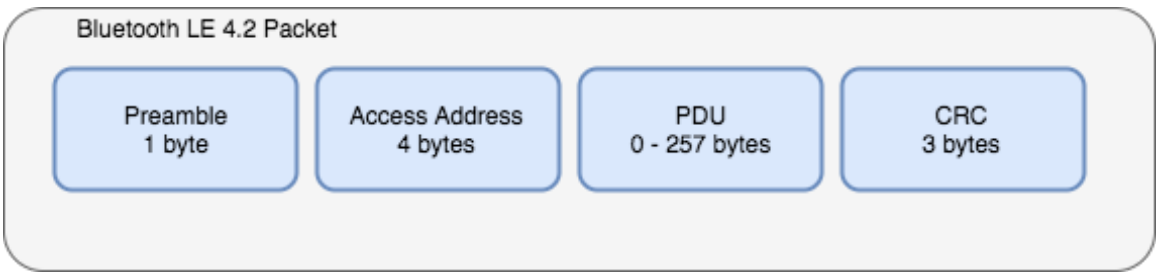

Figure 1: Bluetooth Low Energy packet structure.

The BLE standard allows for devices to read characteristic values ad-hoc or to subscribe to changes. That way when data is announced by a peripheral it is seamless to notify application code of a changed value.

Previously in BLE 4.0 it was impossible to create concurrent connections that both advertise data and connect to other devices. This limited the usefulness of BLE in mesh networks. The BLE standard in 4.1, 4.2, and 5.0 (2017 & 2018) have further addressed these concerns by increasing throughput, concurrent connections, range, and power efficiency.

HARDWARE

Ideal hardware would be capable of running in an environment with multiple devices and support the transfer of files to storage on the device. To test and evaluate the replication protocol a mix of Raspberry Pi boards and a laptop were selected. All the hardware used supports Bluetooth 4.1 or above. By using 4.1 and above each device has access to more advertising channels and larger data throughput while using Bluetooth Low Energy.

- 2 x Raspberry Pi 3
    - 1.2GHz 64-bit quad-core ARMv8 CPU
    - 1GB RAM
    - 802.11n Wireless LAN
    - Bluetooth 4.1 (Broadcom)
- 1 x MacBook Pro 2017
    - Intel Core I7 2.7Ghz
    - 16GB Ram
    - 802.11ac Wireless LAN
    - Bluetooth 4.2 (Broadcom)

REPLICATION

NETWORK ARCHITECTURE

Currently each individual Bluetooth Low Energy device is limited by the number of advertisements it can create and the intervals at which it advertises and scans for advertisements. This means that a device may not see another device's advertisement purely due to missing the window in which the Bluetooth protocol decided to advertise in. Different devices support different connection parameters. Most devices will choose the parameters that the peripheral is set to. Meaning the central device does not choose the parameters when communicating with a peripheral (but can suggest parameters in some cases). The relevant parameters that affect throughput between two Bluetooth Low Energy devices are:

- Connection Interval
- Number of packets per interval
- Length of data sent

When the two devices are connected, they do not directly send data to each other. Instead they have a window to communicate which is determined by the connection interval. One device will send a packet to the other. If it is received the other device sends a packet back.

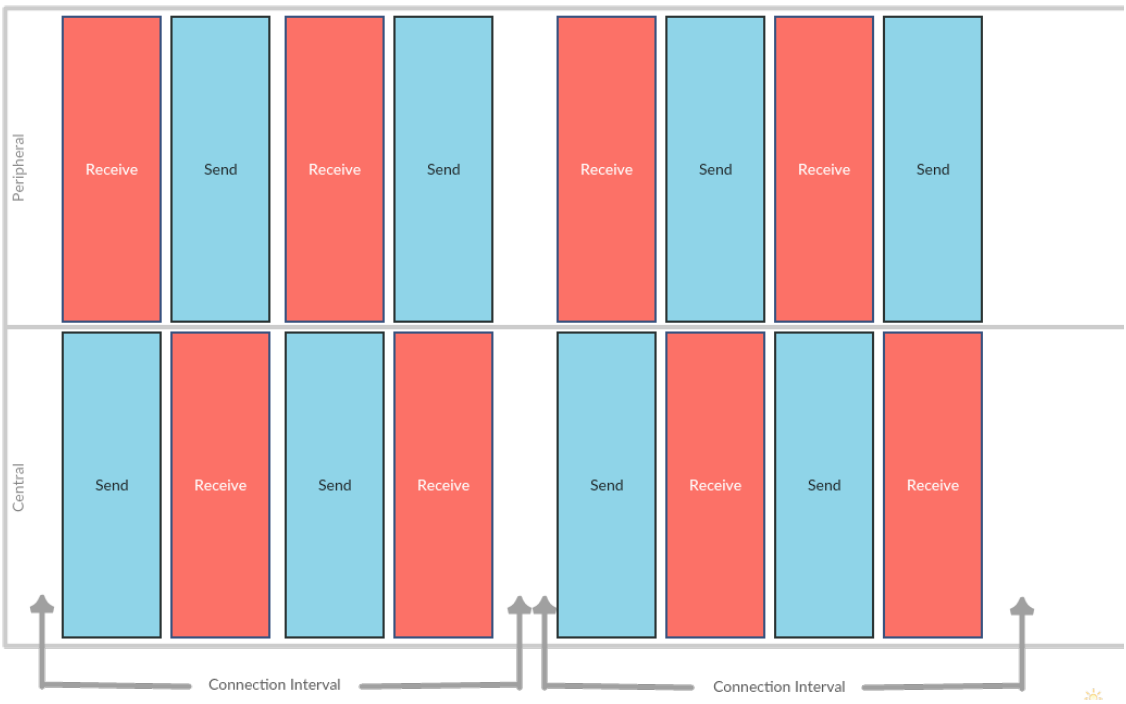

Figure 2: Exchange of packets in connection interval

The number of these packets sent in one connection interval is referred to as the packets per interval. Finally, for each packet there is a data length and the number of packets delivered per interval times the data length gives us the throughput of the connection based on these parameters. The formula can be seen to be:

$$Throughput = \frac{Packets\ Per\ Interval * Data\ Length}{Connection\ Interval}$$

Using this formula, we can derive the maximum throughput that the devices can communicate at. The Bluetooth spec states the minimum connection interval guideline to be 20ms but typically devices will have a larger connection interval. The hardware used

has a 30ms connection interval. In iOS and Mac OS, 4 packets can be sent per connection interval. Using the PDU size for BLE 4.0 at 31 bytes the throughput becomes:

$$4133\ bytes/s = \frac{4 * 31\ bytes}{30\ ms} * 1000\ ms$$

With BLE 4.1 data extensions allows for larger payloads of 257 bytes:

$$34{,}266\ bytes/s = \frac{4 * 257\ bytes}{30\ ms} * 1000\ ms$$

With the asynchronous connection logic that Bluetooth Low Energy utilizes combined with mobile devices such as phones, which move around frequently, the network topology becomes less predictable then standard protocols such as Wi-Fi. As such a broadcast network was chosen that can quickly initiate replication between nodes even if it might not be considered the most optimal topology at that time. This allows for devices that can sync up in their connection window quickly to unitize the opportunity to transfer data rather than seek out other nodes and wait for their advertisements. A device may not advertise continuously to save power so nodes that can be found by the device that is scanning are given priority based on their signal strength defined by RSSI.

MESSAGE TYPES

Devices will communicate by sending messages between each other. Each device will have a characteristic that it will advertise for each message in the system. The Bluetooth standard states that a characteristic may be a read, write, or read and write a value. Characteristics may be used to present data in their PDU field which is limited to 31 bytes in Bluetooth 4.0 and 257 bytes in Bluetooth 4.2. Messages in the system are

defined using Protocol Buffers [6]. Protocol Buffers allow efficient binary storage of a structured message that is suitable for data transfer between machines. It is an improvement on text formats such as JSON and XML. Since the PDU size in a BLE packet is quite small any wasted space will adversely affect the throughput between nodes. The messages defined in the system are for the coordination of file status and file transfer.

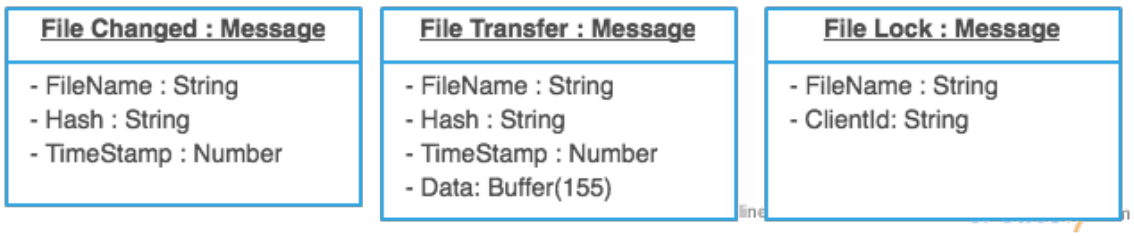

Figure 3: Messages used in replication protocol

File Changed Message – When a file is updated on a node a vector clock is updated to record the change so that nodes do not try to update if they have a new version of the file or there exists a newer version of the file. The filename and a md5 hash of the file is also transmitted to allow nodes requesting the file to validate it after the file transfer completes.

File Transfer Message – Wraps the data in a message similar to the File Changed message. If the size of the file does not fit in the buffer then the file will be split into multiple messages. On each read of the message the peripheral service the file will update the message with a new one containing the next slice of bytes.

File Lock Message – Locking is required as the File Transfer characteristic is shared among nodes. Any device wishing to retrieve a file must obtain a lock first. This is

done by checking if there is empty data in the File Lock characteristic. If empty the device will write its mac address into the File Lock characteristic. Before reading the file, it will check once more that it owns the lock. If the node does not own the lock then it will use a back off period to wait for the lock to become free.

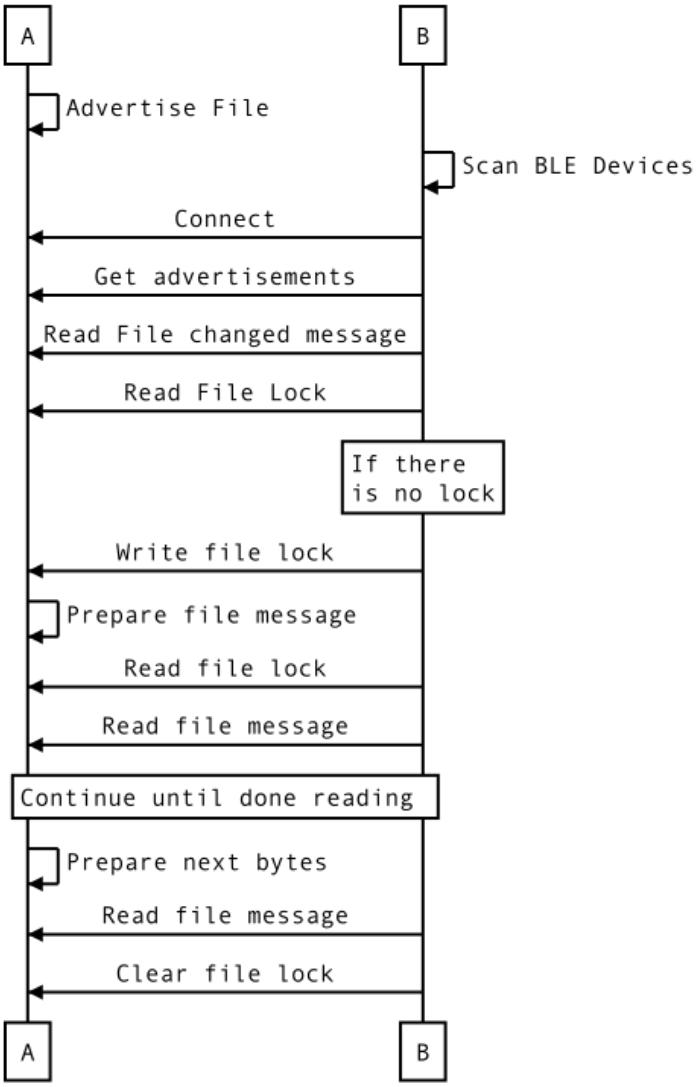

Figure 4: Replication protocol message flow

RESULTS

To test the protocol out an environment of three devices were used. Each device could update a file and advertise its changes at different intervals. Devices would then exchange messages by scanning for devices that contain Bluetooth Low Energy characteristics that correspond to file updates. The experiments were run with devices at different distances from each other as well as different file lengths. Depending on how often devices wake up to scan for updates the replication would occur soon after discovering a node with a file update newer than the file on the node.

The time it took for a node to start to scan before it found a node with a characteristic that corresponded to nodes in the experiment was on average 1.6 seconds. This may be due to the fact that there were other Bluetooth Low Energy devices in the area that it had to scan through first.

The time it took a node to start transfer of a file to when it finished varied with distance and environmental factors such as walls and furniture. The results were below the theoretical calculated throughput for both BLE 4.0 and 4.2 but the files did transfer as expected.

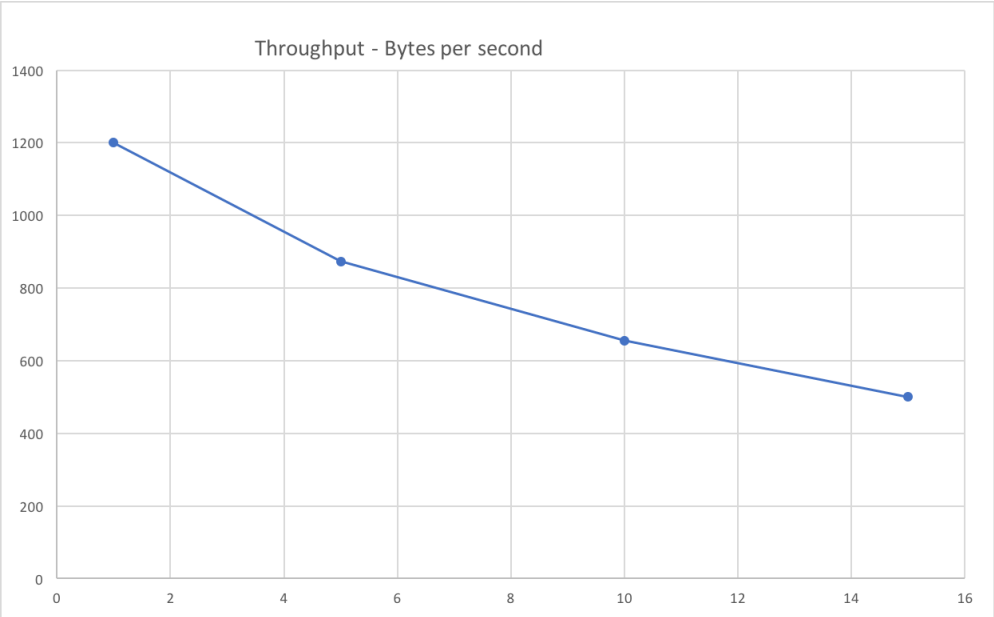

Figure 5: Throughput of protocol by distance

As can be seen in Figure 5, the max speed seen was 1200 Bytes per second. As the distance from the device grew the speed went down. During the experiment the speed sharply dropped as soon as the Bluetooth radio had to communicate through walls.

CONCLUSION

In this paper, I showed how Bluetooth Low Energy works as well as how Bluetooth Low Energy can send large files to other nodes. By utilizing Bluetooth Low Energy devices can transparently transfer files using a very low power. The protocol can handle multiple nodes in the system and replicate files to each other.

Bluetooth Low Energy is theoretically capable of much higher throughout than what was achieved in this paper. The algorithm has the overhead of having to split out files on demand when a node requests. It also must encode and then decode the data transferred between nodes. Further work could see the reduction of this overhead by avoiding encoding the actual file in a message and rely on sending the raw bytes. The application would know to stop reading when the buffer is less than full on a subsequent read. This may also have the effect to triggering BLE optimization in connection intervals when it detects the buffer is full when sending data. It is likely we are missing out on optimizations in the BLE protocol by having the overhead of encoding messages on the fly. Future work could additionally use Bluetooth 5.0 and make use of the increase in connections available to create more advertisements per device connecting with the hope that concurrent data transfers could take place.

Bibliography

[1] Verma, D. C., S. Sahu, S. Calo, A. Shaikh, I. Chang, and A. Acharya. "SRIRAM: A scalable resilient autonomic mesh." IBM Systems Journal 42, no. 1 (2003): 19-28.

[2] Al-Tekreeti, Safa, Christopher Adams, and Naseer Al-Jawad. "Creating Wi-Fi Bluetooth mesh network for crisis management applications." Mobile Multimedia/Image Processing, Security, and Applications 2010, 2010.

[3] "FruityMesh." FruityMesh - GitHub. Accessed May 04, 2017. https://github.com/mwaylabs/fruitymesh/wiki.

[4] "Bluetooth Smart Evolution Helps the Technology Break into Key IoT Market Verticals." ABI Research. Accessed May 04, 2017. https://www.abiresearch.com/press/bluetooth-smart-evolution-helps-technology-break-k/.

[5] "Adopted Specifications." Bluetooth Technology Website. Accessed May 04, 2017. https://www.bluetooth.com/specifications/adopted-specifications.

[6] "Protocol Buffers." Google Developers. Accessed May 04, 2017. https://developers.google.com/protocol-buffers/.