A Test Bed for Evaluating the Performance of IoT Networks

Hope Harvey

A Senior Thesis submitted in partial fulfillment
of the requirements for graduation
in the Honors Program
Liberty University
Spring 2020

Acceptance of Senior Honors Thesis

This Senior Honors Thesis is accepted in partial
fulfillment of the requirements for graduation from the
Honors Program of Liberty University.

____
Feng Wang, Ph.D.
Thesis Chair

____
Eduard Babulak, Ph.D.
Committee Member

____
David Schweitzer, Ph.D.
Assistant Honors Director

____
Date

Abstract

The use of smaller, personal IoT networks has increased over the past several years. These devices demand a lot of resources but only have limited access. To establish and sustain a flexible network connection, 6LoWPAN with RPL protocol is commonly used. While RPL provides a low-cost solution for connection, it lacks load balancing mechanisms. Improvements in OF load balancing can be implemented to strengthen network stability. This paper proposes a test bed configuration to show the toll of frequent parent switching on 6LoWPAN. Contiki's RPL 6LoWPAN software runs on STM32 Nucleo microcontrollers with expansion boards for this test bed. The configuration tests frequency of parent changes and packet loss to demonstrate network instability of different RPL OFs. Tests on MRHOF for RPL were executed to confirm the working configuration. Results, with troubleshooting and improvements, show a working bed. The laid-out configuration provides a means for testing network stability in IoT networks.

Keywords: 6LoWPAN, RPL, MRHOF, load balancing, stability

A Test Bed for Evaluating the Performance of IoT Networks

Introduction

Future advancements in cutting-edge technology depend on the Internet of Things (IoT). New implementations of IoT devices enable easy execution of ordinary tasks. Many beneficial uses have been developed; however, many difficulties remain unresolved. IoT devices in Low-Power and Lossy Networks (LLN) work under tight resource constraints, such as short radio communication ranges and limited energy resources. With the growth of practical applications, the demand on IoT networks continually increases. Without a proper load balancing mechanism, nodes may quickly exhaust their energy causing the whole network to disconnect or become unstable. IPv6 over Low-Power Wireless Personal Area Networks (6LoWPAN) and IPv6 Routing Protocol for LLNs (RPL) achieve a reliable and energy efficient routing for LLNs. Improvements to IoT protocols equip networks to remain stable.

Unfortunately, protocols lack stability measures. RPL does not implement a load balancing mechanism. RPL topologies consist of children connecting through parents. One root node grounds the system. In a case where a parent node has many children, the children to parent allocation may result in instability. Uneven load distribution consumes more energy with less efficiency than a balanced network consumes. As protocols look to form strong connections, parent and child relations increase in complexity. RPL lacks mechanisms to establish proper network load balancing, resulting in loss of resources, packets, and stability.

This paper seeks to demonstrate the importance of load balancing stability and provide a test bed configuration for IoT. The protocols for 6LoWPAN and RPL are overviewed followed by parent selection methods. The proposed configuration and results are detailed. Parent selection stability under different loads of traffic can be analyzed on a test bed to see the variation in network stability and packet loss. From this knowledge, mitigation plans can be considered and tested.

Background

6LoWPAN is consistently used to connect LLN devices. This network provides a universal infrastructure of Internet Protocol (IP) that works with the unique characteristics of LoWPAN devices. RPL has established itself as a valuable routing protocol; however, it falls short in some areas. The following sections explain 6LoWPAN and RPL further.

6LoWPAN

6LoWPAN is a powerful, low cost network connecting devices characterized by limitations on one or more of the following: computational power, memory, and energy. Constraints of the network include small packet size, flexibility in location and number of devices needed, and unpredictability due to devices sleeping, losing power or connectivity, etc. Two main types of devices are Reduced Function Devices (RFD) and Full Function Devices (FFD). The device types refer to resources available and power required (Kushalnagar, Montenegro, & Schumacher, 2007). To achieve wireless connection, 6LoWPAN combines IPv6 and IEEE802.15.4 protocols.

Flexible protocol is required for 6LoWPAN devices. All protocols have different layers running in parallel to manage different tasks connecting a device to a network. The layers of 6LoWPAN incorporate an additional, unique Adaptation Layer as can be seen in Figure 1. The Adaptation Layer supports necessary compression, fragmentation, and other forwarding processes bridging the layers (Ma & Luo, 2008). To support this connectivity and easily incorporate it into larger existing networks, different protocols, IPv6 and IEEE802.15.4, were incorporated at different layers.

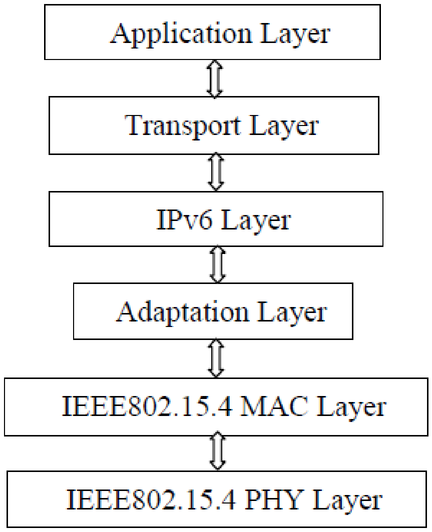

Figure 1. Reference model of 6LoWPAN protocol stack. This figure shows the protocols and their order for 6LoWPAN.
From "The Analysis of 6LowPAN Technology," by X. Ma and W. Lou, 2008, *2008 IEEE Pacific-Asia Workshop on Computational Intelligence and Industrial Application, 1,* p. 964. © 2008 IEEE.

IPv6. In the networking layer, 6LoWPAN communicates between networks through IPv6. IPv6 is an industry standard that offers many development and trouble-shooting tools for ease of use. Applying IPv6 is an effective way to connect network devices in the immediate and larger network systems. Various topologies, including star and mesh, can be constructed in 6LoWPAN. In addition, IPv6 allows large address space and ability to compress the header reducing overhead costs (Kushalnagar et al., 2007, p. 6). Choosing to incorporate IPv6 into LoWPAN devices allows flexibility and unites many common goals in small packet overhead, resource consumption, and energy conservation while maintaining processing power.

IEEE802.15.4. The MAC and PHY layers of 6LoWPAN are based on IEEE802.15.4. The small bandwidth protocol consumes low amounts of energy. It works with both RFD and FFD devices over short distances. Like IPv6 it accounts for star and mesh topologies. The one limitation, compensated for by an additional Adaptation Layer as seen in Figure 1, is a difference between header size of IPv6 and IEEE802.15.4. The max MAC payload in IPv6 is 1280 bytes while IEEE802.15.4 max MAC payload is 127 bytes with a 25 byte header (Yang, Guo, Orlik, Parsons, & Ishibashi, 2014). The Adaptation Layer between the two protocols fragments, compresses, and reassembles packets allowing the layers to smoothly work together despite header size difference. IEEE802.15.4 provides the physical foundation for 6LoWPAN to communicate wirelessly with different networks.

RPL

RPL provides a routing mechanism specifically for 6LoWPAN. The network is constructed from a Destination-Oriented Directed Acyclic Graph (DODAG), terminating at a single root or border router. If needed for constraints and performance, RPL can run several instances concurrently. RPL is bi-directional in all links with triggers or monitors to verify connections. Construction of the DODAG always references the root. Moving up reflects moving towards the root. Moving down implies moving away from the root and towards the leaves. Topologies that can be formed include point-to-point, point-to-multipoint, and multipoint-to-point. Separated from other processes, routing optimization objectives are the foundation of forming the DODAG. Routing optimization is achieved through Objective Functions (OF). The OF’s goal ranks the nodes within or near the network building up the DODAG. RPL allows LLN devices to form a network for other networks to interface.

Control messages in RPL pass and acknowledge information about neighbors and the DODAG state. Information from the messages dictates how the DODAG forms. DODAG Information Object (DIO), Destination Advertisement Object (DAO), DODAG Information

Solicitation (DIS), and Destination Advertisement Acknowledgement (DAO-ACK) are four of the main control message types. Most of the information needed for RPL is found in DIO including RPLInstanceID and DODAGID (Winter et al., 2012). DIO contains everything needed for the RPL connection and configuration, choice of node parent within DODAG, and maintenance of DODAG. The upward routes of DODAG are formed from DIO messages. DAO sends information up to the root and receives acknowledgement through DAO-ACK. This builds the downward routes. Two modes are used for DAO: storing and non-storing. In non-storing, the DAO message is sent directly to the root, the only node that stores the data. However, the storing mode has each node the message passes through record the sender’s address in a routing table. DIS messages request DIO from RPL nodes to learn about the surrounding area of DODAGs. This message is valuable for understanding what neighbors a node has, potential parents, and even information about a different DODAG. If a more suitable DODAG appears, a node may switch. When the node is a parent, it may bring or leave its children in the transition (Winter et al., 2012). RPL messages allow different networks to form and change as nodes require.

Objective Functions (OF)

Objective Functions (OF) in RPL protocol navigate formation of topologies through evaluating the Rank while taking other constraints, or metrics, into account (Winter et al., 2012). With OF, RPL remains flexible for different networks (Thubert, 2012). Parent selection in the DODAG, which makes up the formation, is selected by OF through assigning and evaluating the Rank of a node. Rank is the distance from any node to the DODAG root (Winter et al., 2012). Default OF for RPL are Objective Function Zero (OF0) and Minimum Rank with Hysteresis

Objective Function (MRHOF). The OF chosen is highly important as it sets and keeps the network accountable to the goals of the DODAG.

OF0. One of the mostly widely used OFs is OF0. This OF does not use protocol extensions (Thubert, 2012). Working with DIO messages and DODAG Configuration, OF0 adds to the node Rank through rank_increase, step_of_rank, and rank_factor through a hop count method. The rank_increase pulls the parent up in rank closer to the root. The amount of Rank increase is determined by step_of_rank. Application of a rank_factor alters this. Based on the nodes available and their comparative Rank, OF0 will shape a DODAG where the lowest ranking nodes are closer to the root. The goal is to appropriately assign parents that add connection to a *grounded*, or strong, root (Thubert, 2012). Optimization of the path, including load balancing, is not as concerning. The guidelines for preferred parent and feasible successor are fully outlined by Thubert (2012). As OF0 works, it looks for feasible successors for parents farther up the DODAG. OF0 actions are general enough to allow interoperation with a variety of network types. It also allows for implementation dependent functionality offering flexibility for any OF0 use making it a popular choice of OF for many networks.

MRHOF. Link metrics consider the quality of connection when assigning Rank values to different nodes. OF0 does not consider metric containers. MRHOF uses the Metric Container in DIO messages to form routes that are low cost and low churn (Gnawali & Levis, 2012). Paths are established through smallest metric values and hysteresis, looking at the quality of a connection. Once the path of least cost is found through minimum Rank, it will not change that path unless the new option is shorter. The comparison mechanism is referred to as *hysteresis* (Gnawali & Levis, 2012). This form of parent stability prevents frequent parent changes by comparing current paths to potential ones. Metrics that are not additive cannot be used by MRHOF (Gnawali & Levis, 2012). Working with MRHOF, Expected Transmission Count (ETX) is used to find stable minimum-ETX node to root routes (Gnawali & Levis, 2012). When metric values are missing, ETX is used by RPL to find the best paths. Using metric values and hysteresis, MRHOF maintains stronger Rank stability. MRHOF provides increased stability and small churn resulting in its common use in RPL.

Parent Selection Stability

A large part of RPL is parent selection. The routes formed are based on parent and child relationships. The cost and functionality of a DODAG depends on the metrics used to select parents and the methods to maintain or change node parents with changing demand. Energy is wasted and packets lost through parent instability that results in inefficient routes or constant changing of parents. This parent oscillation is more common in certain standards such as congestion-aware routing metrics, where load balancing is attempted, or with heterogeneous networks. Sometimes load balancing techniques will unintentionally cause more parent changes. Conversely, if proper methods are not used, a child may remain with an overloaded parent. Both cases of too many or too few parent changes present problems that may lead to network instability and packet loss.

As previously stated, OF is responsible for how parents are selected in RPL; however, there are basic rules all OF hold to. To maintain a proper DODAG in upward routing, a node may never hold Rank equal or higher to its parent set (Winter et al., 2012). An exception exists for INFINITE_RANK. The advertisement of INFINITE_RANK through a DIO message shows a parent that left the set and detached from the DODAG. The parent’s children should move to other parents within the DODAG. They may follow the detached parent when no other options appear (Winter et al., 2012). Not advertising this change, whether the node remains or leaves, results in a poisoned route. When assigning nodes, the Mode of Operation (MOP) field is also considered. MOP of 0 indicates a leaf from which no DAO messages are transmitted (Winter et al., 2012). Maintaining proper Rank and message transmission keeps the DODAG loop free and stable. Different OFs build off this foundation to establish a strong parent selection that focuses on specific routing goals. Each OF provides further detail to the process while staying true to the RPL intent of creating a DODAG.

One of the common issues in parent selection is the *herding effect*. This occurs when nodes recognize congestion on the current parent node and change to another. This becomes a problem when all nodes leave the current parent and select the same parent to go to as seen in Figure 2. The simultaneous changes can loop and continue forever in a vicious cycle. A balanced load will never be fully accomplished if the cycle persists. However, it is not determined to be a persistent problem. The children may switch and then stay with the new parent without inciting a continuous cycle between parent nodes. The herding effect is one to be aware of and avoid in creating metrics but depends on the node's behavior.

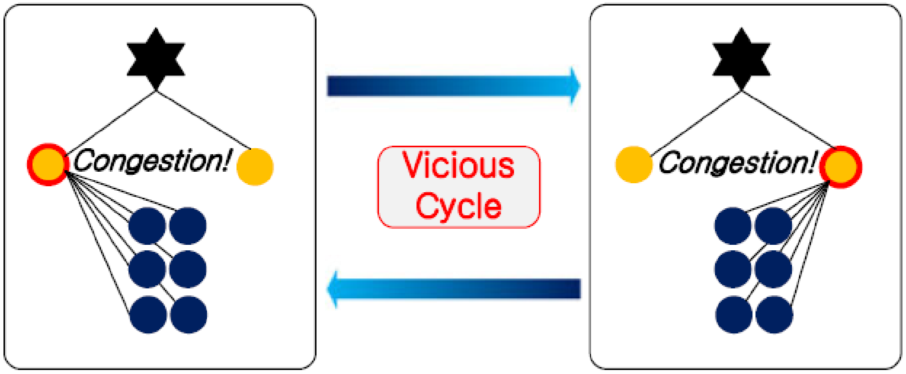

Figure 2. Illustration of herding effect. This figure shows the congestion that can take place when herding effect is occurring. From "Load Balancing Under Heavy Traffic in RPL Routing Protocol for Low Power and Lossy Networks," by H.-S. Kim, H. Kim, J. Paek, and S. Bahk, 2017, *IEEE Transactions on Mobile Computing, 16*(4), p. 970. © 2016 IEEE.

Related Work

Strides have been made in proposing a strong load balancing and stable network protocol for RPL. Different metrics have been proposed to mitigate some of the issues in RPL and LLN device networks. Some of these metrics include MinHop, Delay estimation, Expected Transmission count (ETX), Link Quality Level (LQL), physical layer metrics, and cross layer metrics (Karkazis et al., 2012; Oana Iova et al., 2013). Node congestion and parent stability are other routing concerns. Some routing metrics stand alone, but many are combinations of metrics with separate goals (Karkazis et al., 2012). While each attempt to address an issue, many times the standard will fall short in another area continuing the search for a good OF.

The following are some proposals for load balancing issues. Using Packet Transmission Rate (PTR) to find the parent with the lowest rate, the Traffic Aware Objective Function for RPL-based networks (TAOF) seeks to balance the load and extend the lifetime of the node (Ji et al., 2018). Congestion causing packet loss and parent selection in routing is considered in queue utilization based RPL (QU-RPL) (Kim, Kim, Paek, & Bahk, 2017). The QU-RPL congestion indicator, sent in the DIO message, is used to propagate node overload information with a probabilistic parent change mechanism to add stability (Kim et al., 2017). A Stability Index (SI) encouraged stability with new metrics for new nodes to measure node and DODAG stability before joining in stability metric based RPL sRPL (Yang et al., 2014). SRPL adds to the address length which will take more energy to propagate through the network. Often the goal to achieve one aspect of RPL is achieved at the cost of another characteristic.

Various combinations of two different matrices are considered in other works (Karkazis et al., 2012). OF-FL, an OF that uses fuzzy logic, is another protocol addition to RPL that incorporates four different metrics (Gaddour, Koubâa, Baccour, & Abid, 2014). The combination of end-to-end delay, hop count, link quality (ETX), and node energy are considered for OF-FL attempting to cover many of the LLN limitations (Gaddour et al., 2014). An amount of user control over routing is found in a new internet routing architecture (NIRA) (Yang, Clark, & Berger, 2007). NIRA give a user control over source and destination addresses, providing full power to a user to control their own routing and load balancing instead of sensing it (Yang et al., 2007). Such potential would need to be monitored carefully and does not help the autonomous nature of most IoT networks. The previous works all propose ways to balance the load of a network using different OF metrics within RPL.

Many other proposals have been made with similar goals of seeking to mitigate existing load balancing problems. Even so, load balancing is still not perfected. The need for load stability remains especially in the area of parent selection. In addition, many have not been fully tested on actual test beds. Many of the concepts presented are confirmed though simulations. This paper will propose a test bed configuration and propose another OF to stabilize frequent parent switching.

Test Bed

The test bed is purposed to facilitate load balancing testing for different protocols in IoT. The goal is to demonstrate the effect of packet frequency changing on the network stability. This is determined by looking at the number of parent changes and packets lost on each child node. Balancing demanding children on an IoT network is the focus of the layout.

Testbed Set-up

The configuration consists of a basic tree with the receiver as the root, two senders as second layer nodes, and the last three senders as leaves as shown in Figure 3. A serial sniffer can be added to the configuration to capture packets. The root and parents need to maintain a decent distance between themselves and the child nodes. In addition, the children should remain outside the range of the root node, never selecting it as the preferred parent. The distance between children is not important. The sniffer should be positioned near the nodes where packet capture is desired. Placing the sniffer among the children will allow analysis of packet loss. All parents should keep the root as the preferred parent and the children should choose one of their two closest parents. Each node's preferred parent should be checked before running cases. After initial set up, only the leaves, the third layer, are monitored for preferred parent change and packet loss.

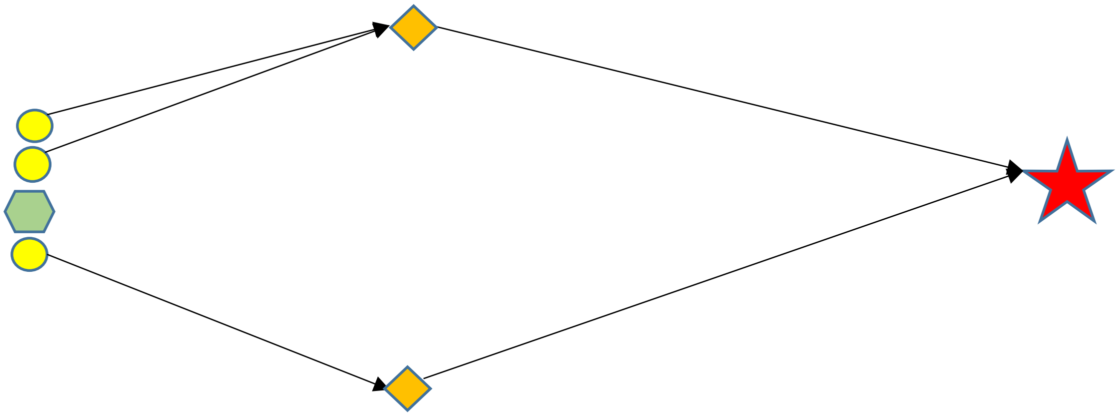

Figure 3. Test Bed Layout. This figure shows the node type and placement. The red star is the root, orange diamonds the parents, yellow circles children, and the green hexagon the optional sniffer.

Testbed Configuration

The test bed should be a simple set up looking to observe a controlled network of devices communicating over 6LoWPAN with RPL. The hardware and software need to be compatible and flexible to change different parameters.

Few components are needed to make up the test bed. Microcontrollers, their power supply, and a PC with serial port communication software form the test bed. The microcontrollers, each a STM32 Nucleo with expansion board as seen in Figures 4 and 5, work with the Contiki RPL 6LoWPAN software, osxContiki6LP.

Removed for Copyright

Figure 4. *STM32 Nucleo-L152RE. This figure shows the microcontroller used. From "UM1724: User manual," by STMicroelectronics, 2019. Retrieved from https://www.st.com/resource/en/user_manual/dm00105823-stm32-nucleo64-boards-mb1136-stmicroelectronics.pdf. © 2019 STMicroelectronics.*

Removed for Copyright

Figure 5. X-NUCLEO-IDS1A SPIRIT1 expansion board. This figure shows the expansion board used with the microcontroller. From "Getting Started with the Contiki OS/6LoWPAN on STM32 Nucleo with SPIRIT1 and Sensors Expansion Boards," by STMicroelectronics, 2016. Retrieved from https://www.st.com/content/ccc/resource/technical/document/user_manual/group0/13/c8/99/6c/b7/38/4d/90/DM00255309/files/DM00255309.pdf/jcr:content/translations/en.DM00255309.pdf. © 2016 STMicroelectronics.*

Hardware configuration. STM32 Nucleo-L152RE, seen in Figure 4, with expansion board X-NUCLEO-IDS01A*, seen in Figure 5, are the microcontrollers recommended to be used as nodes. The STM32 Nucleo-L152RE is a flexible, low-power microcontroller with an ST-LINK/V2-1 debugger/programmer attached. There are different expansion boards and sensors that work with it. The expansion board, X-NUCLEO-EDS01A*, is a low data rate and low power sub-1GHz option (STMicroelectronics, 2016a). It works with a SPIRIT1 transceiver at 868 MHz when X-NUCLEO-IDS01A4 is selected (STMicroelectronics, 2016a). Seven STM32 Nucleos with expansion boards can be used to form a network.

Software configuration. The software configuration provides the means to send messages among the nodes and monitor how this connection was established through printed messages. A PC will alter code, flash the nodes, and monitor the leaf send rate and preferred parent changes. Depending on the distance of the children, a second PC with a serial port terminal can be used. The STM32 system workbench on Eclipse is the recommended IDE.

Contiki's osxContiki6LP is the protocol to use on the boards from the en.x-cube-subg1_firmware found on STMicroelectronics website. The example code for the UDP-receiver and UDP-sender needs to be compiled and flashed to the respective boards. Figure 6 shows the set-up for the sender and receiver connections between two boards both powered through a PC. When directly connected to a PC, a serial monitor can display and collect the debug messages. All the children should be directly connected to a PC for message detection. The UDP-sender can be altered to enable debug messages for viewing preferred parent information and neighbor lists, to change the packet rate, and to customize the MRHOF for the test bed. If using a sniffer, software can be found in the serial-sniffer folder in the same directory as the receiver and sender software. The flashed sniffer board needs to stay connected to a PC with Wireshark to collect information on the captured packets.

Removed for Copyright

Figure 6. UDP Sender and Receiver Configuration. This figure shows the configuration of an UDP sender receiver set-up. From “Quick Start Guide: Contiki OS and 6LoWPAN sub-1GHz RF communication software expansion for STM32 Cube (osxContiki6LP),” by STMicroelectronics, 2016. Retrieved from https://www.st.com/content/ccc/resource/sales_and_marketing/presentation/product_presentation/group0/01/f9/75/c0/32/9d/48/bb/osxContiki6LP%20Quick%20start%20guide/files/osxContiki6LP_quick_start_guide.pdf/jcr:content/translations/en.osxContiki6LP_quick_start_guide.pdf. © 2016 STMicroelectronics.

Variable and message configuration. Several sender firmware files have adjustable debug messages and variables for obtaining the desired information. The sender folder, TX_SPIRIT1_STM32L152RE_Nucleo, is used here to show different changes available. The file with sender components is unicast-sender.c. Several variables for basic configuration changes are here including UDP_PORT, MESSAGE_SIZE, and APP_DUTY_CYCLE_SLOT. Equation 1 gives the equation used by Contiki to assign APP_DUTY_CYCLE_SLOT for a desired packet per minute send rate. This is the variable to adjust for adjusting sending rates. The periodic_sender() functions facilitates sending messages and printing the communication results.

The mode RPL runs in can also be set in rpl.c through the static enum rpl_mode mode. Mesh, feather, and leaf modes are all available to configure what nodes can forward information and if they are reachable.

$$\text{APP\_DUTY\_CYCLE\_SLOT} = \frac{\text{1 packet}}{\text{Number of desired packets per minute}} * 60 \text{ seconds per minute } (1)$$

Parameters for the RPL protocol are included in the Middlewares > Contiki > core > net path. MAX_LINK_METRIC, PARENT_SWITCH_THRESHOLD, and MAX_PATH_COST are integral variables for the MRHOF metrics system. Default values are in the RFC 6719 MRHOF guidelines. Contiki may have altered one or two of these values. The OF0 variables can be found in rpl-of0.c. RANK_FACTOR, RANK_STETCH, MIN_STEP_OF_RANK, AND MAX_STEP_OF_RANK can all be manipulated. Depending on the needs of the individual test bed, these parameters can be adjusted.

If more information is needed for troubleshooting or logging information, debugging messages can be enabled. Macros for defining debug properties are defined in most of the rpl files. DEBUG_PRINT or ‘1’ will enable all messages for the file. A list of neighbor information and the preferred parent changes are obtained through rpl-dag.c debug messages. This file’s messages should be printed and logged. To see estimated ETX, enable debugging in link-stats.c. Further information on ETX and the DAG formation could be output through rpl-ext-header.c. Individual messages can be commented out or adjusted to return only necessary information. Too many messages can convolute the logging and analysis. Conversely, files may always be filtered later to seen only certain messages.

Expected Measurement Results

Each test case looks at the leaf node behavior. Several different cases should be planned and then executed for best use of the bed. The length of the test should be at least 20-30 minutes long. A baseline for packet rate should be selected for all senders to run on. Different cases of increased frequency should be planned to run on one of the children nodes. These cases will be compared against the base case for correlation of packet frequency increase and parent change occurrence.

Runs should be completed before running each set of tests. These runs confirm the test bed set up. Nodes need to be checked to be at correct distances, mostly to ensure the children do not pick the root as a parent.

The test bed results should give several insights on the impact of frequency in parent stability. The baseline will not show many preferred parent changes. As the frequency of all children are similar, the parent nodes quickly adjust and handle packets. The results are expected to show that increasing the frequency of just one child node alters the stability of the network. As the frequency increases the parent of the child needs to accommodate the increased load with the normal load. The debug messages should indicate an even distribution of parent changes throughout the run. If both children select the same parent, the parent will be overloaded.

Data collection methods. Primary means of gathering data is accomplished through serial monitoring. Tera Term is a good serial port terminal with logging functions that can be used to collect serial messages. The port should run at 115200 pbs with 8 bits, no parity, and 1 stop bit. The generated logs should be used to evaluate the preferred parent changes. The log files can be filtered allowing the collection of more debug messages and then later parsed for the desired message. In Linux or Cygwin, the grep command can be used.

Sniffer based data collection method. Packet loss can also be monitored through the Serial-Sniffer application. Cygwin64, a terminal window, calls the serial-sniffer application in administrative mode to capture UDP packets with Wireshark. The serial-sniffer relies on two executable files in the “/Utilities/serial-sniffer” folder within the firmware package. This path and the path to the Wireshark executable are both needed. Once in the Utilities folder, the following command is run for the serial-sniffer: `./serialdump-windows.exe -b115200 /dev/ttyS<#> | ./convert-to-binary | <path to Wireshark executable>/wireshark.exe -k -i -` filling in the necessary serial port number and path to Wireshark. The sniffer captures all packets from the children, or other close nodes, and displays the information through Wireshark. This allows for packet loss analysis.

Metrics for performance measurements. The method of analysis on the gathered data provides a variety of information. This test bed is flexible, able to run with several protocols so analysis of results will vary. Frequency of shifting, other DODAG component changes, packet loss, RTT latency, and message overhead power can be observed.

The main considerations for a functioning IoT should be in the resources consumed and network formed. Looking at the protocol’s main objective and seeing if it is met is vital. In OF0, the Rank calculation should be analyzed. For MRHOF, this would include seeing the EXT and rank changes as parents are selected and loads change. If the sniffer is used, the loss and timing of packets will confirm the configuration.

This configuration attempts to focus on the load balancing between children and their parents looking for specific metrics in these areas. The preferred parent and DODAG neighbor list configurations in correlation to the calculated rank should be considered.

Measurement Results

A basic tree was established to test this configuration with the receiver as the root, two senders as second layer nodes, and the last three senders as leaves as seen in Figure 7. The serial sniffer was not included in these runs. The distance between the root and parents is approximately 5 meters and the distance between the parents and children is 4 meters. For general ease of testing, the IP addresses of each board used were recorded. Knowing the address and location of each board is important as the debug messages use the board IP addresses to give information about the DODAG. The signal strength of each board should also be considered and tested against the root. Strength variation was compensated for in the distance between the layers. For example, weaker boards can be used as children to decrease their connection to the

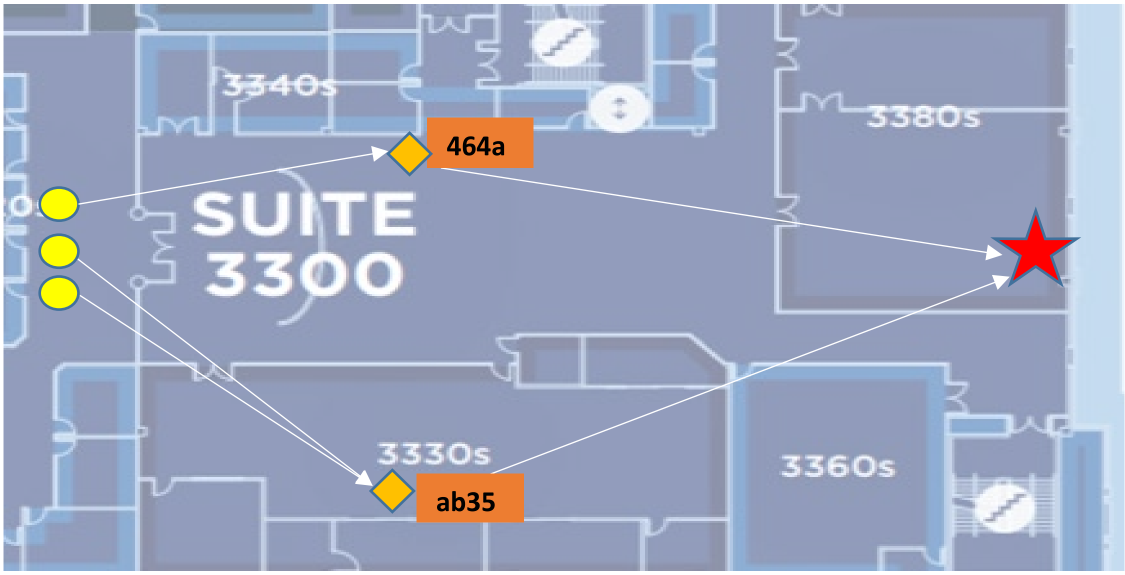

Figure 7. Test Bed Set Up. This figure shows the test bed set up at Liberty University, Demoss Hall 3rd Floor, Engineering Department Atrium Area. The red star is the root, orange diamonds the parents, and yellow circles children.

root node. This configuration is proposed to show parent selection and the potential of switching when using RPL in conjunction with MRHOF.

The test run looked at the leaf node behavior. The parent nodes were kept at 5 packets per minute (PPM). Each test was run for a half hour. MRHOF, which does not consider load balancing, was used as the OF. The duty cycle was changed for all or just one node to increase the packet speed showing how the increase in traffic affects the number of parent changes.

Several test runs were completed before running each of the cases mentioned. Nodes were checked to be at correct distances, mostly where the children would not pick the root as a parent. The root was moved from the original position outside a classroom to inside for the purpose of decreasing the signal strength. Preferred parent data logging and packet capture were collected at the same time so the data collected would be consistent in relation to the testing conditions.

Preferred Parent Changes

Two different cases with a baseline were considered. All nodes were set to send packets at 5 PPM for a baseline. This was the control case that the others would be compared against. Case 1 has one child increase its speed to 20 PPM. This should not greatly affect the number of parent switches or magnitude of packet loss as the load is increasing uniformly. Case 2 increases the node to 100 PPM. A change should be seen when the packet load sent is not consistent among the children. More instances of parent switching are expected to occur. This will show the largest difference in parent switching and packet loss of all the cases.

The test bed results showed the test bed configuration can be used to look at parent stability but has some room for improvements. Table 1 holds accounts for the cases, runs, and results of the tested configuration. The baseline does not show many preferred parent changes. When the demand is consistent, nodes tend to select and stay with their parent. As the frequency of all children are similar, the parent nodes quickly adjust and handle the packets. In Case 1, where one child's PPM is increased, the preferred parent change is not dramatic. Case 2 has only one child increase to 100 PPM from the original baseline of 5 PPM. The results collected show a low, consistent amount of change from child to child. The expectation would have been for more fluctuation as one child's demand is very different from the others. This could have been an outlying case that an average would show differently.

Table 1

Preferred Parent Change Results

Case	Run	Time (hour)	COM8		COM9		COM11	
			PPM	Preferred Parent Changes	PPM	Preferred Parent Changes	PPM	Preferred Parent Changes
B	1	0.5	5	2	5	1	5	1
	2	0.5	5	1	5	1	5	1
	3	0.5	5	1	5	2	5	1
1	1	0.5	5	4	20	4	5	4
	2	0.5	5	1	20	1	5	1
2	1	0.33	5	1	100	1	5	1

The results loosely show that increasing the frequency of just one child node alters the stability of the network despite the results not yielding as many changes as expected. Looking at the debug messages, the rank did not always change as was expected for the load on the system. It is a change from the baseline; however, more tests should be completed to have a well-rounded study.

Future Improvements

There was some difficulty in selecting an appropriate area to fit the desired topology while maintaining proper connectivity. The children had to be far enough away as to not pick the root node as the preferred parent. The area needed to be in a place out of the way for all the tests to be run without interference but have a power source. Having easy access to the nodes to reset or test the connection was an important consideration too.

Further cases at more consistent, smaller intervals on the same topology would more accurately show parent stability and packet loss. This set up of the test bed configuration would take 30-60 minutes to properly connect. Even then, some of the runs, not included in the given results, would use other children as preferred parents. The use of rpl_mode in the children to prevent connection would be an improvement.

Running similar tests with other OF on the same configuration would demonstrate how different metrics work in balancing increased loads. Different OFs could be compared to see the effect on objective goals on parent stability. The test bed could also be used to test other OF as a consistent baseline exists to compare the simulations against.

Conclusions

The rising popularity of IoT devices has not reached its peak. IoT networks provide a significant portion of device connectivity that requires consistent and dependable connectivity. 6LoWPAN has provided the means to wirelessly connect nodes and networks. RPL offers a foundational protocol flexible enough to adapt to the network desired. OF metrics can be adjusted to fit a specific system. Looking into the limiters of the system, it is beneficial to improve the protocol’s methods of parent selection, an area of weakness for an OF. The network stability, efficiency and reliability will be improved.

The test bed configuration for load balancing in IoT seeks to test the different protocols proposed. Cases can be evaluated on the proposed STM32 test bed to show the impact packet frequency has on a network. The functionality of the bed is shown through a test run. Improvements on the testing could be made to increase functionality and repeatability.

The test bed is proposed to provide a configuration where an increase in packet frequency results in fluctuation of parent selection as load becomes unbalanced and more demanding for an IoT network. The loss of resources and packets does not benefit any network. When protocols are improved, the overall stress on the network decreases and further advances in IoT network are opened.

References

Baer, M. B. (2008). Prefix codes for power laws. *2008 IEEE International Symposium on Information Theory*, 2464–2468. https://doi.org/10.1109/ISIT.2008.4595434

Fraenkel, A. S., & Kleinb, S. T. (1996). Robust universal complete codes for transmission and compression. *Discrete Applied Mathematics*, *64*(1), 31–55. https://doi.org/10.1016/0166-218X(93)00116-H

Gaddour, O., Koubâa, A., Baccour, N., & Abid, M. (2014). OF-FL: QoS-aware fuzzy logic objective function for the RPL routing protocol. *2014 12th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt)*, 365–372. https://doi.org/10.1109/WIOPT.2014.6850321

Gnawali, O., & Levis, P. (2012). *The Minimum Rank with Hysteresis Objective Function* (Report No. RFC6719; p. RFC6719). https://doi.org/10.17487/rfc6719

Iova, O., Picco, P., Istomin, T., & Kiraly, C. (2016). RPL: The routing standard for the internet of things...or is it? *IEEE Communications Magazine*, *54*(12), 16–22. https://doi.org/10.1109/MCOM.2016.1600397CM

Iova, Oana, Theoleyre, F., & Noel, T. (2013). Stability and efficiency of RPL under realistic conditions in wireless sensor networks. *2013 IEEE 24th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, 2098–2102. https://doi.org/10.1109/PIMRC.2013.6666490

Iova, Oana, Theoleyre, F., & Noel, T. (2015). Using multiparent routing in RPL to increase the stability and the lifetime of the network. *Ad Hoc Networks*, *29*, 45–62. https://doi.org/10.1016/j.adhoc.2015.01.020

Ji, C., Koutsiamanis, R.-A., Montavont, N., Chatzimisios, P., Dujovne, D., & Papadopoulos, G. Z. (2018). TAOF: Traffic aware objective function for RPL-based networks. *2018 Global Information Infrastructure and Networking Symposium (GIIS)*, 1–5. https://doi.org/10.1109/GIIS.2018.8635699

Karkazis, P., Leligou, H. C., Sarakis, L., Zahariadis, T., Trakadas, P., Velivassaki, T. H., & Capsalis, C. (2012). Design of primary and composite routing metrics for RPL-compliant wireless sensor networks. *2012 International Conference on Telecommunications and Multimedia (TEMU)*, 13–18. https://doi.org/10.1109/TEMU.2012.6294705

Kim, H.-S., Kim, H., Paek, J., & Bahk, S. (2017). Load balancing under heavy traffic in RPL routing protocol for low power and lossy networks. *IEEE Transactions on Mobile Computing*, *16*(4), 964–979. https://doi.org/10.1109/TMC.2016.2585107

Kushalnagar, N., Montenegro, G., & Schumacher, C. (2007). *IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs): Overview, Assumptions, Problem Statement, and Goals* (Report No. RFC4919). https://doi.org/10.17487/rfc4919

Ma, X., & Luo, W. (2008). The analysis of 6LowPAN technology. *2008 IEEE Pacific-Asia Workshop on Computational Intelligence and Industrial Application*, *1*, 963–966. https://doi.org/10.1109/PACIIA.2008.72

Mulligan, G. (2007). The 6LoWPAN architecture. *Proceedings of the 4th Workshop on Embedded Networked Sensors - EmNets '07*, 78. https://doi.org/10.1145/1278972.1278992

Peres, B., P. Santos, B., de O. Souza, O. A., Goussevskaia, O., M. Vieira, M. A., M. Vieira, L. F., & F. Loureiro, A. A. (2018). Matrix: Multihop address allocation and dynamic any-to-any routing for 6LoWPAN. *Computer Networks*, *140*, 28–40. https://doi.org/10.1016/j.comnet.2018.04.017

Shang, W., Yu, Y., & Droms, R. (2016, February 10). Challenges in IoT networking via TCP/IP architecture. *NDN Technical Report, NDN-0038*. https://named-data.net/wp-content/uploads/2016/02/ndn-0038-1-challenges-iot.pdf

STMicroelectronics. (2016, January). Getting started with the Contiki OS/6LoWPAN on STM32 Nucleo with SPIRIT1 and sensors expansion boards. Retrieved from https://www.st.com/content/ccc/resource/technical/document/user_manual/group0/13/c8/99/6c/b7/38/4d/90/DM00255309/files/DM00255309.pdf/jcr:content/translations/en.DM00255309.pdf

STMicroelectronics. (2016, April 21). Quick Start Guide: Contiki OS and 6LoWPAN sub-1GHz RF communication software expansion for STM32 Cube (osxContiki6LP). Retrieved from https://www.st.com/content/ccc/resource/sales_and_marketing/presentation/product_presentation/group0/01/f9/75/c0/32/9d/48/bb/osxContiki6LP%20Quick%20start%20guide/files/osxContiki6LP_quick_start_guide.pdf/jcr:content/translations/en.osxContiki6LP_quick_start_guide.pdf

STMicroelectronics. (2019, April). UM1724: User manual. Retrieved from https://www.st.com/resource/en/user_manual/dm00105823-stm32-nucleo64-boards-mb1136-stmicroelectronics.pdf

Thubert, P. (2012). *Objective Function Zero for the Routing Protocol for Low-Power and Lossy Networks (RPL)* (Report No. RFC6552; p. RFC6552). https://doi.org/10.17487/rfc6552

Winter, T., Thubert, P., Brandt, A., Hui, J., Kelsey, R., Levis, P., … Alexander, R. (2012). *RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks* (Report No. RFC6550). https://doi.org/10.17487/rfc6550

Yang, X., Clark, D., & Berger, A. W. (2007). NIRA: a new inter-domain routing architecture. *IEEE/ACM Transactions on Networking*, *15*(4), 775–788. https://doi.org/10.1109/TNET.2007.893888

Yang, X., Guo, J., Orlik, P., Parsons, K., & Ishibashi, K. (2014). Stability metric based routing protocol for low-power and lossy networks. *2014 IEEE International Conference on Communications (ICC)*, 3688–3693. https://doi.org/10.1109/ICC.2014.6883895

Appendix

In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of Liberty University's products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to http://www.ieee.org/publications_standards/publications/rights/rights_link.html to learn how to obtain a License from RightsLink.