Department of Computer Science
George Mason University Technical Reports

4400 University Drive MS#4A5
Fairfax, VA 22030-4444 USA
http://cs.gmu.edu/ 703-993-1530

Simple Agile RPL Multicast (SARCAST)

Kevin Andrea
kandrea@masonlive.gmu.edu

Robert Simon
simon@gmu.edu

Technical Report GMU-CS-TR-2016-3

Abstract

Denial of Service (DoS) attacks serve to diminish the ability of the network to perform its intended function over time. As DoS attacks may permeate the veil of the Internet of Things (IoT) devices from the broader internet, it is vital that these devices are able to mitigate these attacks. If such a network of IoT devices were to fail during time of need, lives of patients, war fighters, or the sustainability of manufacturing processes may be placed at risk. This work implements and provide initial assessment on a strategy to mitigate a resource exhaustion DoS attack using a network multicast service, wherein such an attack may target the critical energy reserves of low power devices. The Simple Agile RPL Multicast (SARCAST) concept implemented herein uses an agile addressing scheme to reduce the efficacy of multicast-based DoS attacks on IoT devices.

1 Introduction

Denial of Service (DoS) is a type of attack on networked resources in which the attacker is preventing legitimate resource requests from succeeding. This type of attack may target either the network resources itself, in the form of denial of bandwidth by flooding the network with traffic, or may directly attack the individual system resources by inducing computation or memory exhaustive operations. As the packets forming these attacks may come from either seemingly legitimate or otherwise unknown sources, mitigating such an attack prima facie becomes difficult, which in turn leads to the expenditure of computational and memory resources to ascertain the true nature of the incoming packets.

DoS attacks may also serve to diminish the ability of the network to perform its intended function over time. Instead of causing higher than expected packet loss due to collisions, these attacks may target energy-constrained devices, causing them to use their radio more frequently, thereby reducing the operational life of the network[1]. Such energy-constrained devices are now becoming more widely used under the Internet of Things (IoT) movement [2]. One popular class of IoT devices that is currently being used in both consumer and industrial settings are small, embedded networking devices that provide sensing capabilities to myriad environments ranging from consumer use in homes to industrial automation systems.

Wireless Sensor Networks (WSNs) are small networking devices that fall under this class of Low-Power and Lossy Networks (LLNs). These low-powered radio devices have the capabilities of sensing the external environment and reporting the information collected back to a base station for analysis. As these devices are intended to be left for long-term deployments with little need of maintenance [3], energy-efficiency becomes a key concern.

The nature of these devices, which may be deployed from the back of a moving aircraft [4] to rugged and untraversable terrain for remote environmental monitoring, additionally necessitates some form of ad-hoc routing. The Internet Engineering Task Force (IETF) has specified such a protocol in the IPv6 Routing Protocol for Low-Power and Lossy Networks (RPL) [5], which is designed to create Destination-Oriented Directed Acyclic Graphs (DODAGs) for the purposes of forwarding message traffic to the common root. Since RPL provides the routing core for these devices, it becomes a key aspect of the networking system and forms the basis for the exploration of much of this work.

As DoS attacks may permeate the veil of the IoT from the broader internet, it is vital that this class of devices is able to mitigate the efficacy of such an attack on an RPL-centric network. An attack on one Internet accessible device may now be extendable to small subsets of such devices at the IoT level, bringing the frontier of vulnerability directly into the homes of unsuspecting users by targeting unprotected appliances.

Any device on the periphery of a multicast-enabled WSN, such as a toaster, for instance, may facilitate a specific DoS attack against the entire network. By flooding requests to the multicast address, a single device may rapidly drain the energy from an entire network of embedded systems that require very careful energy maintenance for long-term viability[6]. This report presents initial work on implementing a strategy for mitigating multicast-based DoS attacks based on a scheme of frequently changing addresses.

This report is organized in the following fashion. Section 2 begins by providing a brief introduction of WSNs and RPL as background for the DoS mitigation methodology, in addition to some common DoS attacks against WSNs, some of their adverse affects, and common mitigation strategies. Section 3 then proposes a technique for mitigating one type of DoS attack involving multicast communications. Section 3 describes the implementation on the technique and the results obtained. Section 4 provides a description of the validation in simulation. Section 5 continues by summarizing the analysis and providing an assessment of the technique. Section 6 then explores work that may be derived from the concept presented herein. This report then concludes with references.

2 Background

This report presents initial work on implementing proposed strategy for mitigating multicast-based DoS attacks, under a WSN running the RPL routing protocol. This section provides a brief overview of WSNs, the chosen routing protocol, and then introduces a series of DoS attack strategies on WSN devices.

2.1 Wireless Sensor Networks

WSNs are multi-hop networks that consist of autonomous devices that carry both wireless communication capabilities and sensors [7]. These devices form a subset of LLNs, a category of networking that is typified by communications involving high packet loss rates. The key aspect leading to these loss rates is the low-power nature of the devices; the energy-efficiency requirements of the system necessitates infrequent use of the radio in particular.

RFC 6550 [5] specifically describes LLN devices as being ”interconnected by lossy links, typically supporting only low data rates, that are usually unstable with relatively low packet delivery rates,” which provides a key insight into the efficacy of DoS attacks in general against this platform of system.

2.1.1 WSN Hardware

WSN devices suffer from ”limited resources in terms of energy, memory and processing power,” [8] in this manner. One of the currently employed WSN devices, the Zolertia Z1 [9], shown in Figure 1, offers many improvements over prior versions like the Tmote Sky, however, is still highly susceptible to any resource-based attacks.

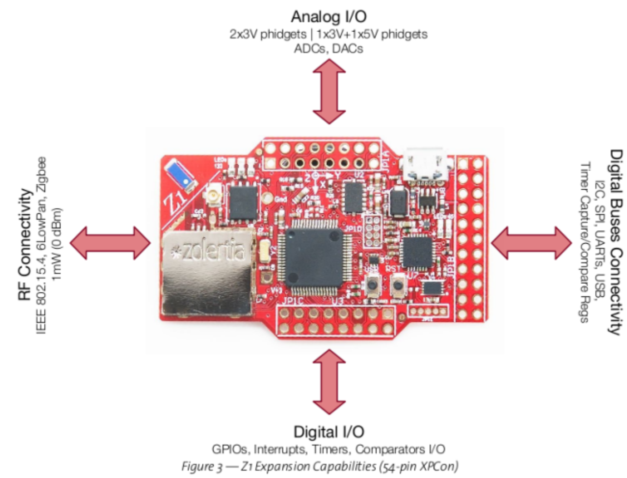

Figure 1: Zolertia Z1 WSN Mote[9]

With only 96KB of flash memory and an additional 10KB of RAM, any resource-sapping operations will unduly affect this device greater than a commensurate attack against an 802.11 wireless device, or a full-powered network computer. Among the chief resources of these devices is power. With two 1500mAh AA-batteries on the current development platform, a Z1 can last for years when using proper radio duty cycling, however, if the device is continually using the radio, this is reduced greatly to an order of 100 hours [9]. This represents the most critical resource of a device that may be left unattended for years given a good energy-management system; as such, a DoS attack targeting the power system can dramatically reduce this lifetime.

2.2 Contiki Network Stack

While the hardware limitations provide a good framework for the assessment of device vulnerabilities to a DoS attack, it is the software that will provide the means for facilitation. This work uses the Contiki Operating System, an open-source operating system to provide the full network stack necessary for inter-device communications.

The Contiki network stack, Figure 2, provides all of the necessary networking services to participate in larger networked environments. The following sections will describe the relevant aspects of each of the layers as it pertains specifically to the unicast and multicast interdevice communications for this work.

2.2.1 IEEE 802.15.4 (Layers 1 and 2)

IEEE 802.15.4 [10] is the standard that specifies the Physical and Data Link Control Layers for Low Power and

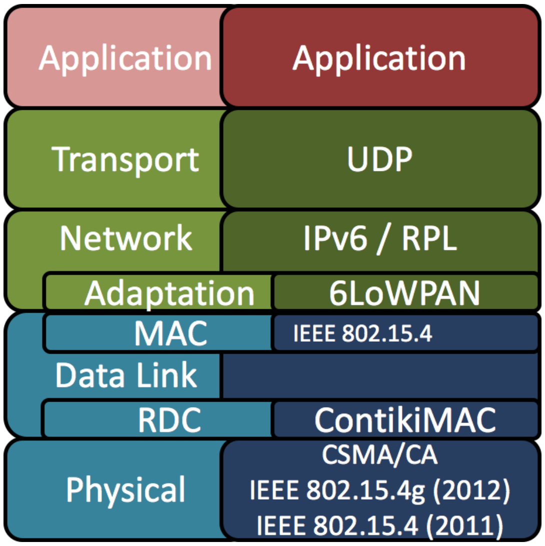

Figure 2: Contiki Network Layers

Lossy Networks. This standard provides the necessary implementation for the Multi-Access Control (MAC) sublayer, through which devices are able to perform direct communication with their neighbors. Each device has a unique MAC address, which is generally formed from a code representing their hardware manufacturer and a production number for the device itself. This also specifies a special MAC address for broadcasting to all devices in range (`FF:FF:FF:FF:FF:FF`). While these layers do not control forwarding of such message to additional devices, this becomes one of the key concepts that underlies the multicast system.

IEEE 802.15.4 also provides a standard mechanism for communicating in a contention-filled network. Carrier Sense Multiple Access (CSMA) with Collision Avoidance (CA) is a means by which the data link control layer is able to send packets in a manner as to reduce collisions [10]. CSMA-CA facilitates this by first performing an Energy Detect (ED) operation by activating the radio in receive mode and assessing the medium for a signal. If a signal is detected, the device will then use use a random back-off timer to delay the next attempt at sending, giving the other signal time to propagate without collisions. This is the technique that enables multiple devices to transmit according to their own schedules, without need for a centralized scheduler.

This also provides the facility for additional vectors into DoS attacks; by flooding the channel, a device performing ED and detecting a message, even if the message would not otherwise interfere with its own communications (Figure 3), would force a delay in sending of its own legitimate traffic, thereby affecting network performance.

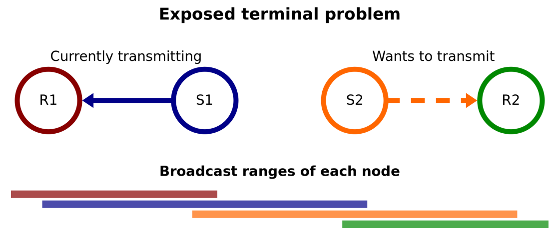

Figure 3: Exposed Terminal Problem *(Public Domain)*

2.2.2 ContikiMAC (Layer 2 - Radio Duty Cycling)

Energy consumption is the key challenge to overcome in this class of energy-constrained devices. As the Z1 data sheet [9] shows, the radio consumes approximately an equivalent amount of energy in receive mode as it does in transmission mode. For this reason, leaving the radio active to continually receive traffic on the network is not viable. To overcome this problem with Contiki, Adam Dunkels developed the ContikiMAC Radio Duty Cycling (RDC) protocol [6].

This protocol works by periodically conducting a pair of Clear Channel Assessment (CCA) probes. Contiki facilitates the efficiency of this RDC by transmitting a message continually until a layer 2 acknowledgement is returned. In this manner, the RDC is able to time the period between each of the pair of probes to be longer than the delay between retransmission events. This enables the receiver to detect an incoming message even if one of the two probes in a pair misses the signal.

By scheduling the CCA probes carefully, the ContikiMAC RDC claims to maintain the radio in the powered off state for approximately 99% of the time, while still supporting normal network traffic [6]. This type of an RDC is not only the key to WSN device power management, but provides possibly the largest resource attack vector for a successful DoS attack. As the RDC will power the radio on with any signal detected during the CCA probe, by flooding the network with traffic, ContikiMAC will be rendered useless for its primary function as it will continually keep the radio activated in receive mode during any down time the device has from its time in transmit mode. As full-time radio use will reduce the operational lifetime from years to days, this becomes a very powerful DoS attack target.

2.2.3 Routing Protocol (Layer 3 - RPL)

The network layer is provided by RPL [5], the IETF standard for IPv6 routing on LLNs. RPL focuses on the creation of a DODAG that is centered on a single node, which serves as the destination for all sensor traffic, the root. This root forms the root of the DODAG and generates the construction of the underlying DAG through the use of a DODAG Information Object (DIO) message. The DIO is generated and periodically transmitted by the root to create and maintain the network.

Upon receipt of a DIO, a device on the network, referred to hereafter as a node, will first assess its current membership in the network being advertised. If this is the first DIO it has received, it will configure itself as a member of the network and then schedule periodic broadcasts of its own copy of the DIO message, so as to propagate it to devices farther from the radio range of the root. As RPL is a distance-vector routing protocol, each node receiving a DIO will use an objective function to calculate its own rank within the network. The rank field is a logical representation of the distance between the node and the root. As a DIO is received, a node will calculate an increase to the received rank value on that message and apply that as its own rank value.

In this manner, routing a packet to a device with a lower rank will make progress in forwarding that packet to the root. This provides a simple, memory-conscious mechanism for creating and maintaining a routing table. Once the network has formed through recursive DIO propagation, nodes will be able to assess each of its neighbors for the best route to the root. This assessment may use different metrics for determining the best route, however, once one is chosen, that neighbor becomes the parent of the current node; all messages destined for the root are forwarded to this parent. A simple DODAG is shown in Figure 4.

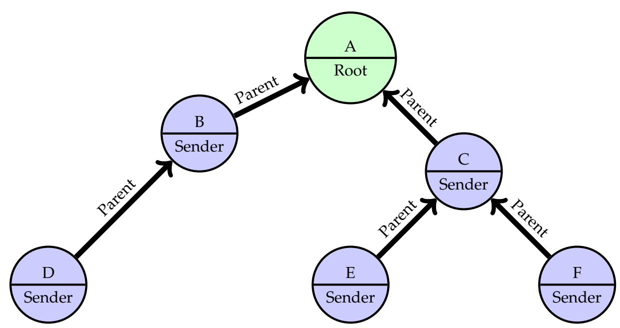

Figure 4: Sample RPL DODAG

2.3 Multicast

Multicast in IPv6 makes use of the `FFfs::` addressing scheme. In this scheme, `FF` is the first octet, representing a multicast address in general. The following code (`f`) is a 4-bit flag that is used to indicate whether the address is permanent or transient in nature. The final code (`s`) is a 4-bit value indicating the scope of the address, where valid scope values indicate whether it is link-local, site-local, or global, among others. The following 14 octets of the multicast address represent the Group ID.

Multicast, in its simplest form over wireless links, works by sending a packet using the MAC broadcast address for delivery to all neighboring devices within range. Once the message is received, it is sent up to Layer 3, where the message is processed under two different objectives.

Forwarding Decision First, the message is assessed for forwarding to neighboring devices. This may be automatic, however, it will involve some check to ensure it will not ultimately be rebroadcast to the sending device without some means of controlling retransmission loops or rebroadcasting duplicate copies of the message.

Message Delivery Decision Second, the message is assessed for the group membership of the current node. If the node has a multicast Group ID that matches the Group ID field in the message, then the payload is delivered to the application layer.

2.4 Denial of Service

As WSNs are being fielded in both consumer and commercial settings, the stakes for their proper performance can be quite high. A WSN deployed in a hospital may be responsible for the timely notification of the duty physician when any of the patients in an emergency ward need assistance. If such a network were to fail during time of need, lives may be placed in jeopardy. Industrial applications may likewise see the damage of equipment in the event the monitoring and actuator control network is similarly compromised. Such compromises may be brought on by specific vulnerabilities, making the networks susceptible to a simple DoS style of attack.

An article on the Denial of Service in Sensor Networks [1] describes several different DoS attacks at each layer of the networking stack. At the physical layer, jamming is a powerful attack that can be used to both induce the RDC to keep the radio on and prevent legitimate traffic from being sent, resulting in a loss of network capabilities. One of the chief defenses against this form of attack is spread-spectrum frequency hopping.

At the data link control layer, flooding the network can result in collisions and exhaustion. CSMA/CA will respond to such attacks by continuing to attempt retransmissions of the packet, increasing backoff times until a maximum number of attempts has been exhausted, at which point the transmission will fail. By flooding the network with a DoS attack, these layer two mechanisms will continually attempt retransmissions, both exhausting the energy resource of the device and amplifying the flooding of the network during any down-time between the attacking packets.

Routing attacks are common at the network layer, involving DoS packets that understand the routing mechanism to abuse how messages are forwarded. By sending messages with false rank values, for example, a DoS attacker may change the topology of the DODAG temporarily, effecting a denial of service of the legitimate packets of all of the new descendants of the attacker. This can be mitigated through message authentication or encryption.

The transport layer also provides vulnerabilities to DoS in the form of flooding resource-intensive requests. At this layer, TCP is commonly abused through the SYN flood DoS attack. By flooding a server with SYN messages, a server will create state for each incoming SYN request and then hold that state until a timeout occurs. This type of an attack is a resource-exhaustion attack and may be mitigated through the use of puzzles that the server can send back to the client to solve before it invests any resources in the connection.

2.5 Denial of Service Objectives

The attack vector of interest in this work is a resource exhaustion attack that propagates to affect the entire network. In this case, this work explores a DoS attack using multicast capabilities of a WSN to flood a large number of devices simultaneously. In this attack, two vulnerabilities are being exploited.

The first involves the RDC at the data link control layer. By receiving a large number of multicast packets, each node in the network will be forced to turn its radio on, listed using the CCA technique, receive and process the packet, then transmit the multicast packet to all other neighboring devices. By carefully timing this attack and through multiple attack fronts (DDoS technique), each node in the network can be kept in a perpetual state of radio activation, draining the energy of the entire network at once.

The second target of this attack is also at the data link control layer. CSMA/CA will hold its traffic as long as any other signals are perceptible within range. By using a DDoS attack, amplified by each legitimate node also forwarding the attacking multicast messages, traffic will have to engage in transmission delays, ultimately aiming for traffic to be dropped entirely.

2.6 DoS General Attack Strategies

A node engaging in a DoS attack is able to exploit the vulnerability in the multicast system by simply following the protocol as intended. By sending a message using the broadcast MAC address, all neighboring devices in radio range will receive the packet. Using a valid multicast address will furthermore cause the message to be forwarded. Even if the Group ID does not match the node, it will still have expended energy to receive the packet and then again to transmit it further, as the group may be valid for other nodes in the network. Every node on the network network will then receive and similarly expend energy processing each packet sent by the attacking device.

3 Simple Agile RPL multiCAST (SARCAST)

SARCAST is an IPv6 Multicast system that mitigates the efficacy of DoS attacks by using address agility to prevent malicious packets from entering a network. The agile address system consists of a field added within the destination address of the packet. On a periodic interval, each system on the network changes the valid agile address; any messages with an expired or otherwise invalid agile address will be rejected.

3.1 SARCAST Algorithm

The SARCAST algorithm, depicted with the below flowchart in Figure 5 performs layer 3 processing on incoming multicast packets.

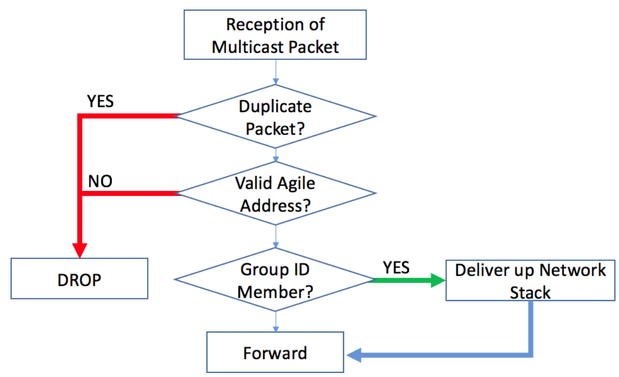

Figure 5: SARCAST Algorithm

SARCAST performs this algorithm upon the arrival of any multicast packet to layer 3 for processing. Not depicted in the flowchart is the check for a valid Time-To-Live (TTL) field or the IPv6 Address processing to ascertain disposition of the packet. The base check on this, and all multicast packets, is valid membership in the multicast group. This is performed by a comparison of the addressed Group ID with the set of Group IDs that the node is a member of. Regardless of membership, valid multicast packets are forwarded to neighboring devices wirelessly.

3.2 SARCAST Addressing Scheme

To facilitate the agility and packet duplication operations, SARCAST modified the general format for multicast addressing, as shown in Figure 6 below.

The 14-octet block for Group ID in the canonical IPv6 Multicast packet has been replaced by a 10-octet Agile Address and a 1-octet Sequence Number. The Group ID has been relegated to the final 3-octets of the address, still leaving a possible 16,777,216 valid groups on the network. The Agile Address is used for verifying packet

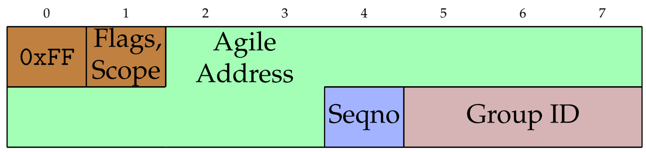

Figure 6: AMASS Multicast Addressing Scheme

validity while the Sequence Number is used for duplicate packet rejection. Both of these are described in the following subsections.

3.3 Duplicate Packet Detection

Duplicate packet detection has been difficult for multicast systems as there are no layer 3 fields that facilitate this function. Performing deep packet inspection at this point is also futile as systems in this networking class generally run over the more resource efficient UDP transport protocol, which similarly is devoid of unique message identifiers in its header. As such, other currently used RPL multicast systems for embedded devices, such as SMRF[11], are stateless in this regard. SMRF relies on the RPL DAG hierarchy to send messages from the parent device to its children; any message received from a child node is immediately dropped. In practice this solves the packet duplication problem, while further imposing a restriction that multicast traffic can only be forwarded to the subtree for which it is the root device. This presents a problem for general communication over a multicast system.

SARCAST uses a different approach by incorporating a sequence number for each generated multicast message. This sequence number is embedded within the addressing scheme as depicted in Figure 6 above. This 1-octet field is paired with the IPv6 Source Address to form a unique message identifier that is used to reject duplicate messages upon reception. SARCAST implements this check using a simple LRU table; which stores the two-field identifier in the last recently updated position of the table, replacing the oldest entry. This provides immediate duplicate message protection while minimizing the space allocated for state. The size of this table in entries is a SARCAST configurable option.

3.4 Agile Addressing

The core of SARCAST is the concept of agile addressing of multicast packets. The agility here is in the form of a pseudo-random 10-octet string that serves as a validation check for incoming messages. On a customizable interval, the root device for the RPL DAG updates a 32-bit counter value that serves as the basis for generating the agile address. This counter value and the current value of the corresponding 16-bit real time clock that drives it are both sent across the network as a part of the routine DODAG Information Object (DIO) message, shown in Figure 7. Each node in the network, after receiving the initial DIO configuration message, will periodically generate their own copy of the message for RPL route maintenance. This periodic mechanism is used to achieve network-level synchronization of the agility timing.

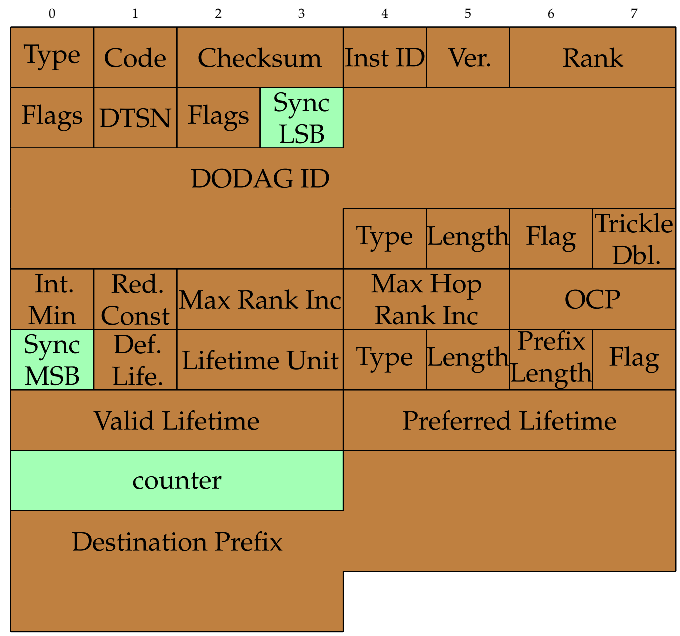

Figure 7: DIO, SARCAST Synchronization Framing

In this message, the 32-bit counter and the 16-bit synchronization clock are both encoded within reserved blocks of the DIO. As there were no available 16-bit reserved fields, the clock was split into two 8-bit blocks to be stored in the two remaining 8-bit reserved fields of the DIO.

$$
\begin{aligned}
counter &= ROR(counter, 3) \\
SyncMSB &= ROR(clock, 3) \wedge 0xFF \\
SyncLSB &= SHR(ROR(clock, 3), 8) \wedge 0xFF
\end{aligned}
$$

The ROR function achieves a bitwise right rotation operation of the specified number of bits, and SHR is a logical right shift operation. The purpose of these rotations is to decouple the synchronization updates from the values transmitted in the messages, so as to further increase the difficulty of an analytic attack on sniffed packets. These messages are also infrequent, also increasing the difficulty of eliciting the location, size, and purpose of each octet of the data encoded in the DIO.

Synchronization Issues As an RPL network grows with respect to hop-count, synchronization becomes more difficult. This implementation of SARCAST uses the periodic DIO message that is self-generated by RPL for network maintenance for this synchronization. The problem with this approach is the inherent infrequency of the DIO message sends. Initially, each node in the network will transmit their own copy of this message using the minimum timing interval. If no network changes are detected after the sends, then the Trickle Algorithm[12] doubles the period between these transmissions. On a stable network, this doubling will continue until a maximum number of doublings is reached. The purpose of this approach is to reduce the overhead traffic for RPL network maintenance, however, this property leads to fewer SARCAST synchronization messages, leading to a propensity for counter drift on larger networks.

Synchronization Compensation SARCAST compensates for differences in synchronization by considering a number of both past and future valid addresses when processing a message. Each of these two limits are individually configurable in SARCAST as best fits the size of the network and the settings governing DIO send times. While this greatly increases synchronization, it also prolongs any DoS attack against the system from an intelligent attacker. The settings for these limits should be selected to provide the minimum acceptable service to nodes in the network, which will prevent the number of agile address changes the DoS attack can succeed when using a captured valid address.

3.5 Agile Address Generation

The address itself is generated from the network synchronized 32-bit counter. The counter is updated using the `SARCAST_INCREMENT_TIME` value, which for this validation work was set at 250ms. Every time a DIO is received, the timer is reset based on the 16-bit value of the clock in the message. The local clock is used to compute the offset at which the counter increment should occur, so as to synchronize all counter updates with the root device. From this point on, the real time clock will fire and increment the counter using the set increment time.

The addresses changes on fixed multiples of the counter updates, using the `SARCAST_DIVISIONS` configured value. This work uses a value of 5, leading to an agile address change every 1.25 seconds. The address generation uses the following formula:

$$hash\left(ROR\left(\left(\frac{sarcastCounter + offset}{SARCAST\_DIVISONS}\right) \oplus SALT, 6\right)\right)$$

Where `ROR(X,S)` performs a circular right rotation on `X` by `S` bits, and the hash function is SHA-1. The purpose of these operations is to decouple the input of the hash function from the synchronization values as transported in the DIO message. This is meant as an augment to message authentication as a means to increase the difficulty of analytic attacks on message patterns. In this implementation the `SALT` is a fixed 32-bit hexadecimal string that is pre-loaded on each valid node.

The `offset` used in the address generation is a counter offset that corresponds to either positive or negative shifts in the address generated. This is the method used by the validation function to compare the received agile address against the set number of past or future addresses to compensate for synchronization drift or network size.

4 Validation

This initial validation of SARCAST involved four principle phases of assessment. The first phase is the control, with SARCAST set to use a static multicast address, without an attacker present. The second phase uses the same static multicast address with an attacker. The third phase employs agile multicast address changing against a DoS adversary using a static multicast address. The fourth phase finally employs agile multicast address changing with an adversary that is able to sniff and change its own multicast addressing scheme in response to a received message, transmitting a new attack for each received message.

The validation was performed using the COOJA network simulator, using the msp430sim MSP430 hardware emulator.

4.1 General Testing Procedures

The testing procedure involves the scenario depicted in Figure 8. Node 1, depicted in green, is the RPL root device. Node 7, the device on the bottom of the figure, depicted in purple, is the multicast DoS adversary. The remaining 10 nodes are senders.

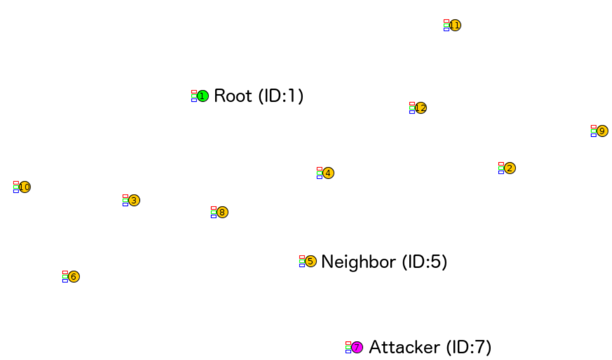

Figure 8: Testing Topology

All of the tests emulate a WSN that is currently operational within the normal bounds of expected use. Each sender device is acting as a sensor in the network, outputting a packet every 4 seconds to the root, via the RPL DODAG. The message is a standard unicast transmitted UDP packet containing the payload, "`SEQNO ### Good teaching is one-fourth preparation and` `three-fourths theater.`", a quote from the novelist Gail Godwin. The ### in this message represents a simple tracking value for testing. In addition to this routine WSN traffic, the root also sends a multicast control packet to all nodes every 15 seconds. This packet represents a control message whose payload consists of a simple logging message, "`[#] SARCAST Message`",where the # value represents the sending node ID.

The attacker is configured to send a multicast message every 3 seconds when enabled. The configuration of the attacker will change for each of the three active phases, as described in the following subsections.

4.2 Phase One - Static Addressing without Attacker

For this phase, all devices operate under their normal mode of operations as described in Section 4.1. For this phase, the multicast address is fixed with the value `FF1E::89:ABCD`. Figure 9 shows the normal operation in this mode, with only the root (ID:1) transmitting using the multicast system.

```
00:06.269   ID:1    [fe80::c30c:0:0:1] MSEND [ff1e::89:abcd] SEQNO 000
00:06.279   ID:8    [fe80::c30c:0:0:1] MRECEIVE [ff1e::89:abcd] SEQNO 000
00:06.279   ID:3    [fe80::c30c:0:0:1] MRECEIVE [ff1e::89:abcd] SEQNO 000
00:07.476   ID:10   [fe80::c30c:0:0:1] MRECEIVE [ff1e::89:abcd] SEQNO 000
00:07.476   ID:6    [fe80::c30c:0:0:1] MRECEIVE [ff1e::89:abcd] SEQNO 000
00:07.977   ID:5    [fe80::c30c:0:0:1] MRECEIVE [ff1e::89:abcd] SEQNO 000
00:07.977   ID:4    [fe80::c30c:0:0:1] MRECEIVE [ff1e::89:abcd] SEQNO 000
00:08.142   ID:7    [fe80::c30c:0:0:1] MRECEIVE [ff1e::89:abcd] SEQNO 000
00:09.808   ID:12   [fe80::c30c:0:0:1] MRECEIVE [ff1e::89:abcd] SEQNO 000
00:10.233   ID:11   [fe80::c30c:0:0:1] MRECEIVE [ff1e::89:abcd] SEQNO 000
00:10.233   ID:2    [fe80::c30c:0:0:1] MRECEIVE [ff1e::89:abcd] SEQNO 000
00:10.945   ID:9    [fe80::c30c:0:0:1] MRECEIVE [ff1e::89:abcd] SEQNO 000
00:21.269   ID:1    [fe80::c30c:0:0:1] MSEND [ff1e::89:abcd] SEQNO 001
00:21.279   ID:8    [fe80::c30c:0:0:1] MRECEIVE [ff1e::189:abcd] SEQNO 001
00:21.279   ID:3    [fe80::c30c:0:0:1] MRECEIVE [ff1e::189:abcd] SEQNO 001
00:21.430   ID:5    [fe80::c30c:0:0:1] MRECEIVE [ff1e::189:abcd] SEQNO 001
00:21.430   ID:4    [fe80::c30c:0:0:1] MRECEIVE [ff1e::189:abcd] SEQNO 001
00:21.604   ID:12   [fe80::c30c:0:0:1] MRECEIVE [ff1e::189:abcd] SEQNO 001
00:22.194   ID:11   [fe80::c30c:0:0:1] MRECEIVE [ff1e::189:abcd] SEQNO 001
00:22.194   ID:2    [fe80::c30c:0:0:1] MRECEIVE [ff1e::189:abcd] SEQNO 001
00:22.226   ID:9    [fe80::c30c:0:0:1] MRECEIVE [ff1e::189:abcd] SEQNO 001
00:22.401   ID:7    [fe80::c30c:0:0:1] MRECEIVE [ff1e::189:abcd] SEQNO 001
00:22.647   ID:10   [fe80::c30c:0:0:1] MRECEIVE [ff1e::189:abcd] SEQNO 001
00:22.648   ID:6    [fe80::c30c:0:0:1] MRECEIVE [ff1e::189:abcd] SEQNO 001
00:36.269   ID:1    [fe80::c30c:0:0:1] MSEND [ff1e::89:abcd] SEQNO 002
00:36.279   ID:8    [fe80::c30c:0:0:1] MRECEIVE [ff1e::289:abcd] SEQNO 002
00:36.279   ID:3    [fe80::c30c:0:0:1] MRECEIVE [ff1e::289:abcd] SEQNO 002
```

Figure 9: Phase One Simulation Sample

Figure 9 shows the a sample execution of the system with no address agility and no attacker. In this model, the node with ID 7 is a receiver of multicast messages from the root. In this sample, the first event at time `00:06.269` shows the root (ID:1) sending a multicast message to the address `FF1E::89:ABCD`. All 11 of the nodes in the network report receiving this message.

For a 100 second run, this phase resulted in the root sending 7 multicast messages. The 11 nodes reported a total of 76 proper receives out of 77. There were an additional 238 unicast messages sent from the nodes to the root, of which 211 were received.

4.3 Phase Two - Static Addressing with Attacker

For this phase, the multicast address system remains static, however a DoS attacker is introduced. Figure 10 shows the normal operation in this mode, with both the root (ID:1) and the attacker (ID:7) transmitting using the multicast system.

```
00:04.289   ID:7    [fe80::c30c:0:0:7] MSEND [ff1e::89:abcd] SEQNO 000
00:04.299   ID:5    [fe80::c30c:0:0:7] MRECEIVE [ff1e::89:abcd] SEQNO 000
00:05.111   ID:4    [fe80::c30c:0:0:7] MRECEIVE [ff1e::89:abcd] SEQNO 000
00:05.111   ID:7    [fe80::c30c:0:0:7] MRECEIVE [ff1e::89:abcd] SEQNO 000
00:05.111   ID:8    [fe80::c30c:0:0:7] MRECEIVE [ff1e::89:abcd] SEQNO 000
00:05.995   ID:12   [fe80::c30c:0:0:7] MRECEIVE [ff1e::89:abcd] SEQNO 000
00:06.269   ID:1    [fe80::c30c:0:0:1] MSEND [ff1e::89:abcd] SEQNO 000
00:06.279   ID:3    [fe80::c30c:0:0:1] MRECEIVE [ff1e::89:abcd] SEQNO 000
00:06.279   ID:8    [fe80::c30c:0:0:1] MRECEIVE [ff1e::89:abcd] SEQNO 000
00:07.289   ID:7    [fe80::c30c:0:0:7] MSEND [ff1e::89:abcd] SEQNO 001
00:07.298   ID:5    [fe80::c30c:0:0:7] MRECEIVE [ff1e::189:abcd] SEQNO 001
00:07.476   ID:10   [fe80::c30c:0:0:1] MRECEIVE [ff1e::89:abcd] SEQNO 000
00:07.476   ID:6    [fe80::c30c:0:0:1] MRECEIVE [ff1e::89:abcd] SEQNO 000
00:07.516   ID:4    [fe80::c30c:0:0:1] MRECEIVE [ff1e::89:abcd] SEQNO 000
00:07.516   ID:5    [fe80::c30c:0:0:1] MRECEIVE [ff1e::89:abcd] SEQNO 000
00:07.632   ID:2    [fe80::c30c:0:0:7] MRECEIVE [ff1e::89:abcd] SEQNO 000
00:07.632   ID:11   [fe80::c30c:0:0:7] MRECEIVE [ff1e::89:abcd] SEQNO 000
00:07.679   ID:9    [fe80::c30c:0:0:7] MRECEIVE [ff1e::89:abcd] SEQNO 000
00:08.745   ID:12   [fe80::c30c:0:0:1] MRECEIVE [ff1e::89:abcd] SEQNO 000
00:08.920   ID:7    [fe80::c30c:0:0:1] MRECEIVE [ff1e::89:abcd] SEQNO 000
00:10.030   ID:11   [fe80::c30c:0:0:1] MRECEIVE [ff1e::89:abcd] SEQNO 000
00:10.030   ID:2    [fe80::c30c:0:0:1] MRECEIVE [ff1e::89:abcd] SEQNO 000
00:10.289   ID:7    [fe80::c30c:0:0:7] MSEND [ff1e::89:abcd] SEQNO 002
```

Figure 10: Phase Two Simulation Sample

Figure 10 shows the a sample execution of the system with no address agility and a simple attacker. In this model, the node with ID 7 sends a multicast message every 3 seconds. In this figure, the top entry is the attacker (ID:7) sending a multicast message, which is received by most of the nodes in the network. The root (ID:1) then sends its first message in concert with the second multicast message from the attacker.

For a 100 second run, this phase resulted in the root sending 7 multicast messages. The 11 nodes reported a total of 56 proper receives out of 77. There were an additional 235 unicast messages sent from the nodes to the root, of which 209 were received. The attacker sent 32 multicast messages, of which nodes reported receiving a total 273 out of 320 possible messages. This attack demonstrates a congested network with the loss of message traffic. The goal of this DoS attacker is to contribute to the energy drain of network and system resources, which is accomplished.

4.4 Phase Three - Agile Addressing, Static Adversary

The SARCAST agile addresses enabled in this phase are generated as described earlier in this document and are synchronized across the network to all non-adversary nodes. Figure 11 shows the normal operation in this mode, with both the root (ID:1) and the attacker (ID:7) using the multicast system, however, the only immediate neighbor of the attacker (ID:5) is rejecting all of these messages as the agile address is invalid, sparing the rest of the network from the resource attacks.

```
00:04.289  ID:7   [fe80::c30c:0:0:7] MSEND [ff1e::89:abcd] SEQNO 000
00:04.313  ID:5   [fe80::c30c:0:0:7] MADDR_REJECT [ff1e::89:abcd] SEQNO 000
00:06.274  ID:1   [fe80::c30c:0:0:1] MSEND [ff1e:4fe2:b82b:2fd1:7b21:6366:89:abcd] SEQNO 000
00:06.299  ID:8   [fe80::c30c:0:0:1] MRECEIVE [ff1e:4fe2:b82b:2fd1:7b21:6366:89:abcd] SEQNO 000
00:06.299  ID:3   [fe80::c30c:0:0:1] MRECEIVE [ff1e:4fe2:b82b:2fd1:7b21:6366:89:abcd] SEQNO 000
00:07.289  ID:7   [fe80::c30c:0:0:7] MSEND [ff1e::89:abcd] SEQNO 001
00:07.312  ID:5   [fe80::c30c:0:0:7] MADDR_REJECT [ff1e::189:abcd] SEQNO 001
00:07.509  ID:10  [fe80::c30c:0:0:1] MRECEIVE [ff1e:4fe2:b82b:2fd1:7b21:6366:89:abcd] SEQNO 000
00:07.509  ID:6   [fe80::c30c:0:0:1] MRECEIVE [ff1e:4fe2:b82b:2fd1:7b21:6366:89:abcd] SEQNO 000
00:08.018  ID:4   [fe80::c30c:0:0:1] MRECEIVE [ff1e:4fe2:b82b:2fd1:7b21:6366:89:abcd] SEQNO 000
00:08.018  ID:5   [fe80::c30c:0:0:1] MRECEIVE [ff1e:4fe2:b82b:2fd1:7b21:6366:89:abcd] SEQNO 000
00:08.199  ID:7   [fe80::c30c:0:0:1] MRECEIVE [ff1e:4fe2:b82b:2fd1:7b21:6366:89:abcd] SEQNO 000
00:09.856  ID:12  [fe80::c30c:0:0:1] MRECEIVE [ff1e:4fe2:b82b:2fd1:7b21:6366:89:abcd] SEQNO 000
00:10.289  ID:7   [fe80::c30c:0:0:7] MSEND [ff1e::89:abcd] SEQNO 002
00:10.297  ID:11  [fe80::c30c:0:0:1] MRECEIVE [ff1e:4fe2:b82b:2fd1:7b21:6366:89:abcd] SEQNO 000
00:10.297  ID:2   [fe80::c30c:0:0:1] MRECEIVE [ff1e:4fe2:b82b:2fd1:7b21:6366:89:abcd] SEQNO 000
00:10.313  ID:5   [fe80::c30c:0:0:7] MADDR_REJECT [ff1e::289:abcd] SEQNO 002
00:11.024  ID:9   [fe80::c30c:0:0:1] MRECEIVE [ff1e:4fe2:b82b:2fd1:7b21:6366:89:abcd] SEQNO 000
00:13.289  ID:7   [fe80::c30c:0:0:7] MSEND [ff1e::89:abcd] SEQNO 003
00:13.313  ID:5   [fe80::c30c:0:0:7] MADDR_REJECT [ff1e::389:abcd] SEQNO 003
00:16.289  ID:7   [fe80::c30c:0:0:7] MSEND [ff1e::89:abcd] SEQNO 004
00:16.313  ID:5   [fe80::c30c:0:0:7] MADDR_REJECT [ff1e::489:abcd] SEQNO 004
00:19.289  ID:7   [fe80::c30c:0:0:7] MSEND [ff1e::89:abcd] SEQNO 005
```

Figure 11: Phase Three Simulation Sample

The figure shows the mass sending of multicast messages by the adversary, however, in this case the neighboring legitimate node continues to reject them all, while normal traffic is still being received by the root. As no propagation of these messages is possible, the only compromise to the network would be resource exhaustion on the part of node 5. The top of this image again shows the attacker (ID:7) initiating a multicast message to address `FF1E::89:ABCD`, however, its only immediate neighbor (ID:5) rejects this address, preventing any further propagation of the attack throughout the network. The multicast messages sent by root (ID:1) use the valid agile address for each send and its messages are received by the network.

For a 100 second run, this phase resulted in the root sending 7 multicast messages. The 11 nodes reported a total of 76 proper receives out of 77. There were an additional 237 unicast messages sent from the nodes to the root, of which 214 were received. The attacker sent 32 multicast messages, of which nodes reported receiving a total 0 out of 320 possible messages. This attack demonstrates a congested network with the loss of message traffic. The goal of this DoS attacker is to contribute to the energy drain of network and system resources, which is accomplished.

4.5 Phase Four - Agile Addressing, Dynamic Adversary

Figure 12 shows the normal operation in this mode, with both the root (ID:1) and the attacker (ID:7) using the multicast system. In this scenario the attacker will send a multicast message every 3 seconds, however, upon reception of a multicast message, it immediately sends another multicast message using the agile address sniffed from the received message. This attacker will then be able to successfully send multicast messages until the agile addresses change sufficiently on the network. For this test, the 2 prior addresses, the current address, and the next 2 future addresses are all considered valid.

Figure 12 shows the first two attempts at a multicast attack by the attacker (ID:7) failing with immediate rejections. When the attacker receives a proper multicast message that originated with the root (ID:1), it immediately sends its own next message, using the agile address it just extracted from the received packet. This succeeds, causing the neighbor (ID:5) to accept it and forward it as valid. These successes will persist until the agile address system changes the address, invalidating the stolen address used by the attacker.

```
00:04.289  ID:7   [fe80::c30c:0:0:7] MSEND [ff1e::89:abcd] SEQNO 000
00:04.313  ID:5   [fe80::c30c:0:0:7] MADDR_REJECT [ff1e::89:abcd] SEQNO 000
00:06.274  ID:1   [fe80::c30c:0:0:1] MSEND [ff1e:4fe2:b82b:2fd1:7b21:6366:89:abcd] SEQNO 000
00:06.299  ID:8   [fe80::c30c:0:0:1] MRECEIVE [ff1e:4fe2:b82b:2fd1:7b21:6366:89:abcd] SEQNO 000
00:06.299  ID:3   [fe80::c30c:0:0:1] MRECEIVE [ff1e:4fe2:b82b:2fd1:7b21:6366:89:abcd] SEQNO 000
00:07.289  ID:7   [fe80::c30c:0:0:7] MSEND [ff1e::89:abcd] SEQNO 001
00:07.312  ID:5   [fe80::c30c:0:0:7] MADDR_REJECT [ff1e::189:abcd] SEQNO 001
00:07.509  ID:10  [fe80::c30c:0:0:1] MRECEIVE [ff1e:4fe2:b82b:2fd1:7b21:6366:89:abcd] SEQNO 000
00:07.509  ID:6   [fe80::c30c:0:0:1] MRECEIVE [ff1e:4fe2:b82b:2fd1:7b21:6366:89:abcd] SEQNO 000
00:08.018  ID:4   [fe80::c30c:0:0:1] MRECEIVE [ff1e:4fe2:b82b:2fd1:7b21:6366:89:abcd] SEQNO 000
00:08.018  ID:5   [fe80::c30c:0:0:1] MRECEIVE [ff1e:4fe2:b82b:2fd1:7b21:6366:89:abcd] SEQNO 000
00:08.199  ID:7   [fe80::c30c:0:0:1] MRECEIVE [ff1e:4fe2:b82b:2fd1:7b21:6366:89:abcd] SEQNO 000
00:08.207  ID:7   [fe80::c30c:0:0:7] MSEND [ff1e:4fe2:b82b:2fd1:7b21:6366:89:abcd] SEQNO 002
00:08.232  ID:5   [fe80::c30c:0:0:7] MRECEIVE [ff1e:4fe2:b82b:2fd1:7b21:6366:289:abcd] SEQNO 002
00:09.731  ID:7   [fe80::c30c:0:0:7] MRECEIVE [ff1e:4fe2:b82b:2fd1:7b21:6366:289:abcd] SEQNO 002
00:09.731  ID:8   [fe80::c30c:0:0:7] MRECEIVE [ff1e:4fe2:b82b:2fd1:7b21:6366:289:abcd] SEQNO 002
00:09.731  ID:4   [fe80::c30c:0:0:7] MRECEIVE [ff1e:4fe2:b82b:2fd1:7b21:6366:289:abcd] SEQNO 002
00:09.739  ID:7   [fe80::c30c:0:0:7] MSEND [ff1e:4fe2:b82b:2fd1:7b21:6366:289:abcd] SEQNO 003
00:09.764  ID:5   [fe80::c30c:0:0:7] MRECEIVE [ff1e:4fe2:b82b:2fd1:7b21:6366:389:abcd] SEQNO 003
00:10.091  ID:12  [fe80::c30c:0:0:7] MRECEIVE [ff1e:4fe2:b82b:2fd1:7b21:6366:289:abcd] SEQNO 002
00:10.291  ID:7   [fe80::c30c:0:0:7] MSEND [ff1e:4fe2:b82b:2fd1:7b21:6366:89:abcd] SEQNO 004
00:10.316  ID:5   [fe80::c30c:0:0:7] MRECEIVE [ff1e:4fe2:b82b:2fd1:7b21:6366:489:abcd] SEQNO 004
00:10.532  ID:11  [fe80::c30c:0:0:7] MRECEIVE [ff1e:4fe2:b82b:2fd1:7b21:6366:289:abcd] SEQNO 002
00:10.532  ID:2   [fe80::c30c:0:0:7] MRECEIVE [ff1e:4fe2:b82b:2fd1:7b21:6366:289:abcd] SEQNO 002
```

Figure 12: Phase Four Simulation Sample

For a 100 second run, this phase resulted in the root sending 7 multicast messages. The 11 nodes reported a total of 65 proper receives out of 77. There were an additional 237 unicast messages sent from the nodes to the root, of which 207 were received. The attacker sent 67 multicast messages, of which nodes reported receiving a total 136 out of 670 possible messages. This attack demonstrates a congested network with the loss of message traffic due to the success of each attack. The strength of SARCAST here is that even though the attacks increased dramatically in frequency, the immediate neighbor of the attacker began rejecting the attacks shortly after they were launched, greatly limiting the effect of the attack on the network as a whole.

5 Conclusion

This work was able to successfully validate the core principles behind the SARCAST implementation. While this is very much representative of a prototype for validation purposes, it fulfilled the needs of testing the proposed strategy and demonstrated that a WSN can synchronize agile multicast addresses in such a manner as to allow multiple multicast group memberships, to serve as valid addresses with respect to the IPv6 multicast addressing scheme, and to deny the propagation of a multicast DoS attack throughout the network.

6 Future Work

There are several aspects of SARCAST that need to be addressed for a proper implementation and rigorous testing within an operational environment.

Synchronization The first aspect is to move the synchronization functionality to encompass more frequently exchanged communications. While the DIO provides necessary fields for this, there are two fundamental and critical flaws. The use of reserved fields is not permitted under RFC6550, rendering this approach invalid for full network implementation. The more practical reason against using the DIO message, however, is that the Trickle algorithm [12] doubles the period between DIO transmissions to reduce the network overhead associated with RPL. Using a message whose nature is to be infrequently encountered as a synchronization primary is not ideal.

Resolving the synchronization update issues will enable a SARCAST implementation to greatly reduce both the delay between agile address updates and the number of past addresses it needs to consider as valid. By addressing this, DoS attacks by attackers will reduce in efficacy.

Scalability The second aspect is to address scalability in the system. The logical modification to increase scalability is to have each node that receives a valid SARCAST message update the agile address before forwarding it. With this modification, once a multicast message is accepted by any node, it will be forwarded across an arbitrarily large network without worry of the agile address being rejected due to turnover. The problem with this approach is seen in Phase Four of the validation testing above. In this phase, an attacker receives a valid multicast address from a message and then immediately sends its own attacking message, which uses the valid address it just acquired. When the neighboring device receives this message, it forwards it to all neighbors, including the attacker. Any updates of the address by the neighbor would be detected by the attacker, allowing it to continually update its own messages with the newest agile addresses. Modifying the nodes for updates to address scalability in this manner leads to complete control of the network resources by the attacker.

Future work will explore alternatives using more accurate synchronization to reach arbitrarily distant nodes without compromising the integrity of the agile addresses to the attacker.

Detection The third aspect is to incorporate a failsafe in the multicast control code of the target device. If a device is able to recognize it is currently the direct recipient of the attacking signal from a multicast DoS adversary due to statistical analysis, it may be able to disable the Radio Duty Cycling (RDC) protocol and power the radio off manually for a set duration. Incorporating a timer similar to that which is used in Trickle may allow the device to weather the DoS attack more gracefully by taking itself off the network until the DoS attack ceases. The purpose of this strategy would be to keep the DoS adversary from draining all of the energy from its neighbors.

References

[1] A. D. Wood and J. A. Stankovic, "Denial of service in sensor networks," *Computer*, vol. 35, pp. 54–62, Oct 2002.

[2] R. Minerva, A. Biru, and D. Rotondi, "Towards a definition of the Internet of Things (IoT)," May 2015. Available at `http://iot.ieee.org/images/files/pdf/IEEE_IoT_Towards_Definition_Internet_of_Things_Revision1_27MAY15.pdf`.

[3] J. Polastre, R. Szewczyk, and D. Culler, "Telos: enabling ultra-low power wireless research," in *Information Processing in Sensor Networks, 2005. IPSN 2005. Fourth International Symposium on*, pp. 364–369, April 2005.

[4] M. Senouci and A. Mellouk, *Deploying Wireless Sensor Networks: Theory and Practice*. Elsevier Science, 2016.

[5] T. Winter, P. Thubert, A. Brandt, J. Hui, R. Kelsey, P. Levis, K. Pister, R. Struik, J. Vasseur, and R. Alexander, "RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks." RFC 6550 (Proposed Standard), Mar. 2012. RFC 6550 - RPL.

[6] A. Dunkels, "The ContikiMAC Radio Duty Cycling Protocol," Tech. Rep. T2011:13, Swedish Institute of Computer Science, Dec. 2011.

[7] E. Hamida and G. Chelius, "Strategies for data dissemination to mobile sinks in wireless sensor networks," *IEEE Wireless Communications*, vol. 15, pp. 31–37, Dec. 2008.

[8] T. Watteyne, A. Molinaro, M. G. Richichi, and M. Dohler, "From manet to ietf roll standardization: A paradigm shift in wsn routing protocols," *IEEE Communications Surveys Tutorials*, vol. 13, pp. 688–707, Fourth 2011.

[9] Advancare, S.L., "Z1 datasheet," 2013. Available at `http://zolertia.sourceforge.net/wiki/images/e/e8/Z1_RevC_Datasheet.pdf`.

[10] J. Adams, "An introduction to IEEE STD 802.15.4," in *2006 IEEE Aerospace Conference*, pp. 8 pp.–, 2006.

[11] G. Oikonomou and I. Phillips, "Stateless multicast forwarding with rpl in 6lowpan sensor networks," in *Pervasive Computing and Communications Workshops (PERCOM Workshops), 2012 IEEE International Conference on*, pp. 272–277, March 2012.

[12] P. Levis, T. Clausen, J. Hui, O. Gnawali, and J. Ko, "The trickle algorithm." RFC 6206 (Standards Track), Mar. 2011. RFC 6206 - Trickle.