OAK RIDGE
NATIONAL LABORATORY
MANAGED BY UT-BATTELLE
FOR THE DEPARTMENT OF ENERGY

ORNL/TM-2013/180

Communication Requirements and Concept of Operation for Sensor Networks

September 2013

Prepared by

Dwight Clayton
Richard Willems

UT-BATTELLE
ORNL-27 (4-00)

ORNL/TM-2013/180

Measurement Science and Systems Engineering Division

COMMUNICATION REQUIREMENTS AND CONCEPT OF OPERATION FOR SENSOR NETWORKS

Dwight Clayton
Richard Willems

Date Published: September 2013

Prepared by
OAK RIDGE NATIONAL LABORATORY
Oak Ridge, Tennessee 37831-6283
managed by
UT-BATTELLE, LLC
for the
U.S. DEPARTMENT OF ENERGY
under contract DE-AC05-00OR22725

CONTENTS

LIST OF FIGURES

Figure **Page**

LIST OF TABLES

Table **Page**

ACKNOWLEDGMENTS

This research was sponsored by the U.S. Department of Energy Office of Nuclear Energy, for the Nuclear Energy Enabling Technologies effort. The authors would also like to express our appreciation to our summer intern, Kyle Ray, for his hard work and dedication during his internship.

EXECUTIVE SUMMARY

Today's nuclear power plant (NPP) instrumentation uses current loops and voltage-based communications sometimes compromised by radio frequency interference (RFI), also called electromagnetic interference (EMI). Copper-based communications technology also relies on insulation that may degrade after decades of exposure to NPP environments and can be flammable. Wireless technologies offer the potential for greater expansion in instrumentation in a plant that could augment human performance, provide additional data on plant equipment and component status, and facilitate online assessment of the material condition of plants.

Wireless communications capabilities may substantially reduce the cabling cost, but a number of technology and security issues must be resolved. One of the greatest capital cost contribution from I&C systems arises from cable installation. The trend toward more effective, efficient operation and maintenance will require many additional sensors beyond the number of nuclear and process sensors at a conventional NPP. These additional sensors are needed to enable monitoring of real-time process variables, structural components, movable equipment, portable devices, and warehouse/inventory areas, but will exacerbate the cabling complexity and cost. Wireless technologies offer the potential for greater expansion in instrumentation in a plant that could augment human performance, provide additional data on plant equipment and component status, and facilitate online assessment of the material condition of plants.

Robust digital instrumentation communication techniques and architectures are essential to address the potential for greater expansion in industrial environment instrumentation that could augment human performance, provide additional data on plant equipment and component status, and facilitate online assessment of conditions. To develop wireless alternatives to costly hardwired-cabling for real-time, online monitoring, demonstration of a highly reliable, secure wireless communications system is necessary for continuous data transmission. Wireless Sensor Networks (WSNs) hold the promise of realizing the Internet of Things. Various network protocols have been used, e.g., Zigbee, Bluetooth and WirelessHART for WSNs, but not until recently has IPv6 been considered a viable option. Given its huge address space, IPv6 provides a convenient mechanism to communicate with individual nodes in a WSN.

Wireless communications will benefit all new reactor designs and fuel cycle facilities (e.g., enrichment facilities, mixed oxide fuel fabrication facilities, and used fuel deposition facilities) by reducing maintenance and operating costs associated with installing wiring for replacement or temporary diagnostic sensors.

This report examines the communication requirements and an operation concept for wireless sensor networks in a NPP environment. Specifically, this report addresses the necessary power required for each sensor network node, the transmission frequency, network architecture, capabilities required in each sensor network node, and a general concept of operation including "normal" and "off-normal" conditions.

1. INTRODUCTION

Instruments in nuclear power plants (NPPs) and many industrial environments typically are served by at least two sets of wires; one set carries data and control signals and one set provides instrument power. Wireless communication devices are beginning to appear in a limited number of non-safety-related NPP applications [1], [2]. Wireless sensor networks have proven to be less expensive, more flexible, and more reliable in industrial settings than their wired counterparts [3].

Robust digital instrumentation communication techniques and architectures are essential to address the potential for greater expansion in instrumentation in a plant that could augment human performance, provide additional data on plant equipment and component status, and facilitate online assessment of material condition of NPPs. To develop wireless alternatives to costly hardwired-cabling for real-time, online monitoring, demonstration of a high-reliability, secure wireless communications system for continuous data transmission is necessary.

In many industries, wireless mesh networks are beginning to replace conventional point-to-point wiring. Unfortunately, these existing wireless mesh networks do not have the extreme reliability required for NPP safety and control data, which makes additional research and development (R&D) necessary. To develop wireless alternatives to costly hardwired cabling for real-time, online monitoring, demonstration of a high-reliability, secure wireless communication system for continuous data transmission is necessary. This is especially true if wireless communications are used for transmission of measurement and control data as part of plant control systems.

Wiring associated with delivery of electrical power can be minimized through the use of local energy harvesting. Fortunately, NPP facilities are replete with environmental energy sources that have potential to power wireless sensor nodes. The use of wireless communications eliminates the second set of wires.

By combining wireless communications technologies with power harvesting techniques, development of truly wireless sensor nodes becomes a possibility. When wireless communications technologies and power harvesting techniques are ready for the NPP environment, the benefits will extend far beyond a reduction in cable installation and maintenance cost. Self-powered WSNs will provide a cost-effective way to add new or redundant measurements to existing plant instrumentation systems. Because nodes scavenging certain types of energy could continue to operate during extended station blackouts (SBOs) and during periods when operation of the plant's internal power distribution system has been disrupted, measurements identified as critical to accident management should be among the first targeted. The availability of this data would be invaluable not only to operators trying to manage an accident situation but also to the teams responsible for post-incident analyses as well. Self-powered WSNs and the networks that tie them together will provide an opportunity for substantial improvements in the reliability and safety of modern NPPs.

The demand for smaller packages and longer battery life in consumer electronics has driven the development of ultra-low power circuitry for the last decade; self-powered WSN technology will benefit from these advances. The architecture of a self-powered wireless sensor node (Fig. 1) will be largely independent of the harvesting technology employed and the wireless communications method used – assuming low power consumption is kept as a key feature. The power management block would vary slightly according to the type of harvester used, but circuitry implementing the remaining functions would not be radically modified.

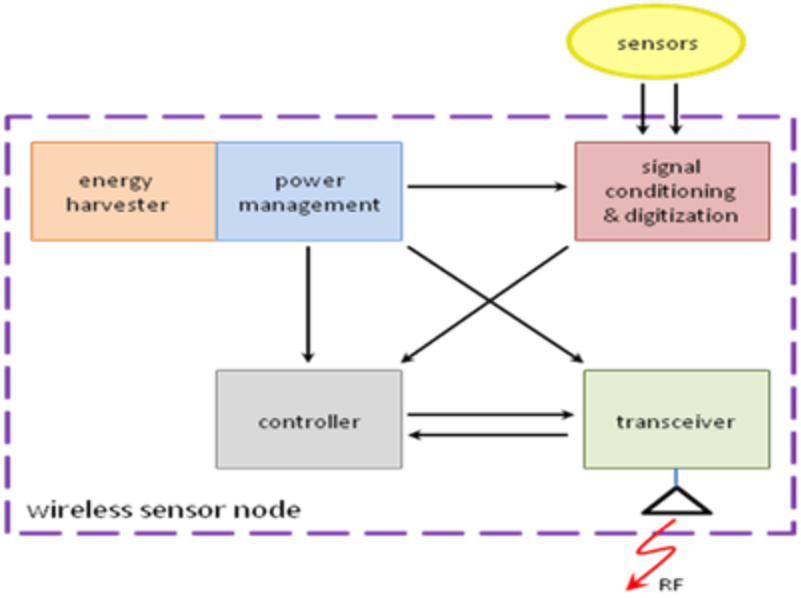

Fig. 1. Elements of a self-powered wireless sensor node.

2. COMMUNICATIONS REQUIREMENTS

It is important to note that ad-hoc mesh networks generally do not operate in a stand-alone mode. They usually have one or more other mesh networks (mesh clouds) to interact with, along with wired infrastructure. Typical security concerns for mesh networks include both passive and active attacks. In a passive attack, the attacker does not insert any information into the network, but listens, and attempts to retrieve vulnerable information. In active attacks, messages are inserted and the operation is disrupted or nodes are harmed – impersonation and spoofing are examples of active attacks. An attacker may also attempt to disrupt the operation of the network by causing a large amount of control packets overloading wireless links and rendering the network unavailable.

2.1 INTRODUCTION TO NETWORK SECURITY

Wireless packets are transmitted through the air where theoretically anyone can eavesdrop. Indeed, there are so-called packet sniffers that can listen to all 16 channels in the 802.15.4 spectrum (the 2.4 GHz ISM band) at the same time.This makes channel hopping alone insufficient to protect data from outside listeners [4]. Security protocols must be designed to prevent a listener hearing from the raw bits of every packet and decrypting any of the information.

A secure wireless network must have the following properties:

- Message Integrity – Data received at the destination should not be accepted if it has been modified in transit. This is an end-to-end property that must be maintained even in the presence of a malicious router and even when the packet goes through many hops from source to destination. This is also called Authentication, as it is intended to confirm the identity of the sender to prevent Man-in-the-Middle attacks where each side in a conversation is unknowingly talking to a third party.

- Access Control – Nodes should only accept data from authorized nodes. This is an end-to-end property, though it also has a link-layer corollary. Data from unauthorized nodes should not be permitted to result in a denial of service (DoS) attack.

- Confidentiality – An eavesdropper that intercepts any encrypted data should not be able to determine anything about the plaintext data except the plaintext length (semantic security).

- Replay Protection – If an adversary captures legitimate encrypted traffic and re-injects it into the network (possibly at a different location), that traffic must not be accepted at the destination without detection. This is an end-to-end and a link layer property.

- DoS resistance – It should be difficult to inject packets into the network and congesting it to a point that prevents the network from operating normally.

2.2 INTERNET PROTOCOL (IP)

The Internet Protocol is unquestionably the standard for interconnecting intelligent computing and communications devices. The Internet Protocol exhibits:

- Extensive interoperability
- Established security
- Established naming, addressing, and discovery
- Established application level data model
- Established network management tools

Based on a layered architecture (Fig. 2), this protocol has been used and adapted for a large percentage of world electronic communications.

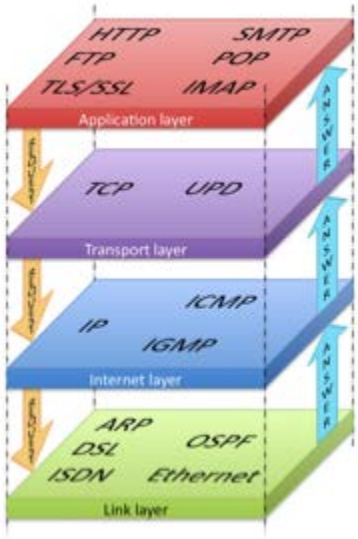

Fig. 2. The Internet Protocol is a layered architecture.

In 1998, the Internet Engineering Task Force (IETF) introduced IPv6. It was primarily designed to replace IPv4 as the network protocol of the Internet. With an increasing number of networked devices, one of the key driving forces for developing IPv6 was the realization that the current Internet protocol IPv4 was rapidly running out of unique IP addresses. To avert the threat of address space exhaustion, IPv6 expands the address space of IPv4 from 32-bits to 128-bits giving a total of 2^{128} or $3.4x10^{38}$ unique network addresses [5]. In addition to a large address space, IPv6 supports network-layer encryption and authentication. Through the use of header extensions, IPv6 implements L2 encryption and authentication with IPSec to ensure both data confidentiality and authenticity [6].

A typical WSN, as shown in Fig. 3, consists of a self-forming, multi-hop mesh of nodes, also known as motes, which collect and relay data. An edge router monitors and manages network performance, provides security, and exchanges data with a host application. Packets traveling through the wireless network are based on a compatible sensor network protocol.

The Edge Router typically has two interfaces – a hardwired IPv4 or IPv6 Ethernet connection and a wireless sensor network connection. The Edge Router converts wireless packets originating from the sensor network protocol to IPv4 or IPv6 (the two most common Ethernet protocols) datagrams, and inserts them into the Internet. Likewise, it transforms IPv4 or IPv6 datagrams from the Internet into wireless sensor packets. In a NPP environment, it is envisioned that the Edge Router would be located at the containment boundary.

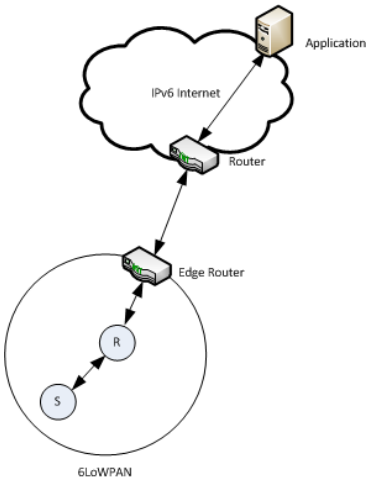

Fig. 3. Typical WSN Configuration

While the transition from IPv4 to IPv6 is far from complete, this report will focus on IPv6 since implementation of a WSN in a NPP environment is deemed to be a long term proposition, deploying these systems for years if not decades. As the "Internet of Things" becomes a reality, the transition to IPv6 is expected to accelerate. For the purposes of this report, the technical differences between these two protocols are not critical. IPv6 will be used to show that WSNs can function in networks of the future.

An IPv6 data packet is comprised of two main parts: the header and the payload. The IPv6 header format is streamlined to keep packet header overhead to a minimum by moving both non-essential fields and optional fields to extension headers that are placed after the IPv6 header. The first 40 bytes/octets of an IPv6 packet comprise the header (Fig. 4) and contains the following fields [5]:

Version	Traffic Class	Flow Label	
Payload Length		Next Header	Hop Limit
Source Address			
Destination address			

Fig. 4. IPv6 Header Packet Composition

- The first four bits of the header packet represent the Internet Protocol version number and is set to 0110b or 6.

- The traffic class field is an 8-bit field and is used to implement Quality of Service (QoS) markings based on data loss, latency and/or bandwidth.

- The 20-bit flow label field allows the marking of packets so they belong to a particular traffic flow for which the sender requires special handling, i.e., real-time.

- The payload length is a 16-bit unsigned integer and represents the number of bytes/octets following the packet header. As noted earlier, any header extensions are treated as part of the payload.

- The next header field represents an 8-bit selector that identifies the next header type immediately following the IPv6 packet header.

- The hop limit field is an 8-bit field that is decremented by one each time the packet is forwarded. When the hop limit reaches zero the packet is discarded.

- The source address is the 128-bit address of the originator of the packet.

- The destination address is the 128-bit address of the intended recipient of the packet.

2.3 IEEE 802.15.4

The Institute of Electrical and Electronics Engineers (IEEE) released the 802.15.4 low power wireless personal area network (WPAN) standard in 2003 [7]. The standard attempts to achieve several goals simultaneously: extremely low cost, short-range wireless communication with reasonable power consumption. Security under 802.15.4 can be broken down into four kinds of service: access control, message integrity, message confidentiality and replay protection. Access control is accomplished through access control lists, i.e., data from unauthorized sources is not permitted. Message integrity ensures that the data received at the destination is unaltered. Data encryption provides confidentiality of the message and prevents eavesdropping on the payload. Replay protection prevents an adversary from capturing encrypted traffic and re-injecting it into the network.

There are three fields in the IEEE 802.15.4 Message Authentication Control (MAC) frame that are related to security: the Frame Control, the Auxiliary Security Header and the Data Payload (Fig. 5).

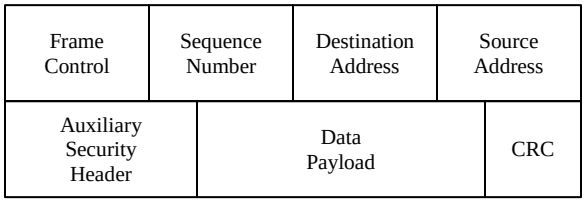

Fig. 5. IEEE 802.15.4 MAC Frame Composition

To enable the Auxiliary Security Header and thereby enable link-layer security, the Security Enabled bit of the Frame Control field must be turned on. The Auxiliary Security Header shown in Fig. 6 has three fields: Security Control, Frame Counter and Key Identifier.

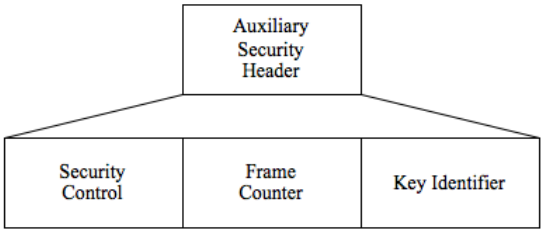

Fig. 6. IEEE 802.15.4 MAC Frame Security Composition

The Security Control is a 1-byte field that specifies the global Security Policy for the frame and is comprised of two bit fields: Security Level and Key Identifier Mode (Fig. 7).

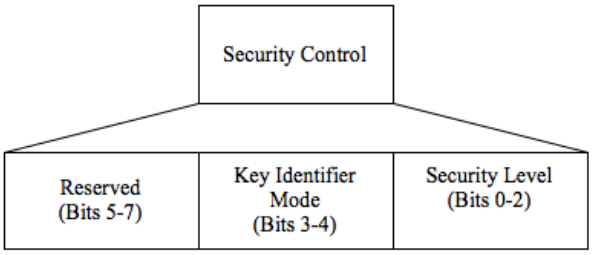

Fig. 7. Security Control Field Composition

Within the Security Control field, the Security Level bits specify the encryption level and the key length. The Security Level values along with their corresponding security properties are shown in Table 1 [8].

Table 1. IEEE 802.15.4 Security Properties

Security Level	Security Property	Description
0x00	No security	Data unencrypted Data not authenticated
0x01	AES-CBC-MAC-32	Data unencrypted Data authenticated
0x02	AES-CBC-MAC-64	Data unencrypted Data authenticated
0x03	AES-CBC-MAC-128	Data unencrypted Data authenticated
0x04	AES-CTR	Data encrypted Data not authenticated
0x05	AES-CCM-32	Data encrypted Data authenticated
0x06	AES-CCM-64	Data encrypted Data authenticated
0x07	AES-CCM-128	Data encrypted Data authenticated

The value of 0x00 specifies no data encryption and no data authentication. Values 0x01-0x03 specify the data are authenticated using the encrypted MAC but the payload content is transmitted in plaintext. The MAC can be 32, 64 or 128-bits. The 0x04 value specifies the packet is encrypted but not authenticated. Values in the range of 0x05-0x07 specify that the data are encrypted and authenticated.

The Key Identifier Mode bits specify the kind of key to be used (implicit or explicit) by the sender and receiver. Table 2 lists the possible values.

Table 2. Key Identifier Modes

Key Identifier Mode	Description
0	The sender and receiver know the Key ID implicitly. Key ID is not sent in the message.
1	The Key ID is determined explicitly by the Key Index subfield of Key Identifier.
2	The Key ID is determined explicitly by the Key Index and 4-bytes of the Key Source.
3	The Key ID is determined explicitly by the Key Index and 8-bytes of the Key Source.

The Frame Counter is a 4-byte counter given by the source of the current frame and is used to guard against message replay.

The Key Identifier field is used if the Key Identifier Mode value is non-zero. The Key Identifier is a 10-byte field that is further divided into the Key Source subfield (9-bytes) and the Key Index subfield (1-byte) shown in Fig. 8.

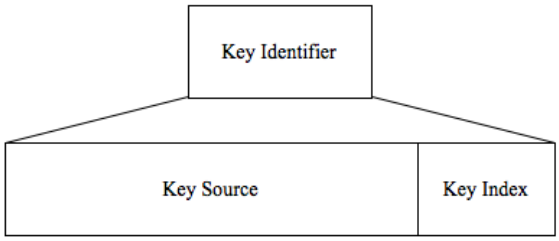

Fig. 8. Key Identifier Field Composition

For non-zero values, the Key Source specifies the group key originator and the Key Index specifies different keys from a particular Key Source. Although IEEE 802.15.4 supports encryption keys, the standard does not specify how the keys are managed or how authentication policies should be applied. It is assumed that the high layer protocols handle the key management.

The encryption algorithm used in IEEE 802.15.4 is the Advanced Encryption Standard (AES) with a 128-bit key length. Not only is AES used to encrypt the payload, but also to authenticate it. For authentication, a 128-bit key is used but the resulting MAC is appended to the payload as 32, 64 or 128-bits. Fig. 9 shows the formatting of the data payload for the three main security suites [9].

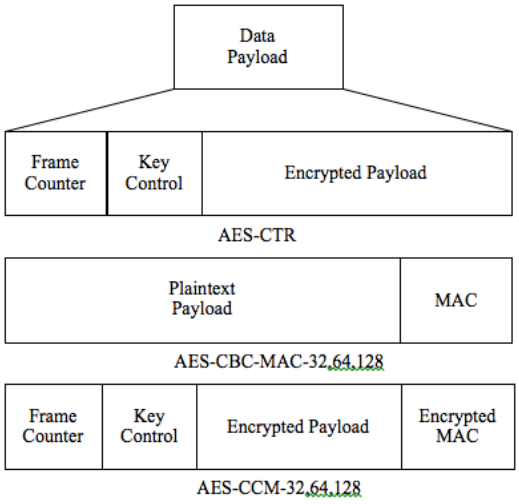

Fig. 9. IEEE 802.15.4 MAC Frame Data Payload for three main security suites

2.4 6LOWPAN

As discussed above, an IEEE 802.15.4 MAC frame is 127 octets; however, adding the MAC frame header overhead and including the AES-CCM-128 security feature, only 81 octets remain for the upper network layers. Given the minimum transmission unit (MTU) size of IPv6 is 1280 octets (bytes), a translation layer is required.

In 2007, the Internet Engineering Task Force (IETF) developed the 6LoWPAN standard for mapping IPv6 over low-power IEEE 802.15.4 wireless networks. The standard deals primarily with the frame format as well as the link-local addresses and stateless auto configured addresses of IPv6 packets over IEEE 802.15.4 networks.

Although the underlying IEEE 802.15.4 standard uses 4 different wireless frame types – beacon, MAC control, data, and acknowledgement, the 6LoWPAN specification only concerns itself with data frames that embed the IPv6 packets.

In order to accommodate IPv6 packets over IEEE 802.15.4 wireless networks, the ITEF developed an adaptation layer to translate the larger IPv6 datagrams to the smaller 802.15.4 data frames. The adaptation layer sits above the link layer and below the transport layer in the network stack as shown in Fig. 10.

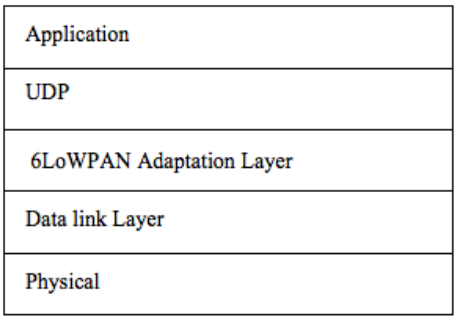

Fig. 10. Network Stack Employing 6LoWPAN Adaptation Layer

The IEEE 802.15.4 standard includes a built-in 128-bit AES encryption feature that secures each link along the way. Link-layer encryption only protects the payload for a single hop and is vulnerable at intermediate hops in a typical multi-hop WSN. A compromised wireless router could easily eavesdrop on the network traffic. Therefore, link-layer encryption by itself is not sufficient for securing application-level information. Moreover, the standard recommends that acknowledgments are requested so that data frames lost on the wireless link can be recovered at the link layer. On the surface this is a good idea, however, a potential security hole in the currently defined IEEE 802.15.4 acknowledgement frames cannot make use of the link layer security. An attacker can easily simulate successful reception of data frames that were lost, thereby having the overall effect of jamming the wireless signal.

Due to the dissimilar domains, the adaptation layer is required to allow interoperability between the two domains. However, this also has potential to lead to security risks. As Kim has shown, the adaptation layer is vulnerable to potential threats through packet fragmentation attacks on the IPv6 side [10].

On the wireless side, embedded 6LoWPAN devices do not have the capabilities for complicated firewalls and are autonomous; therefore, unrestricted data coming into the WSN from the Internet can easily overwhelm the wireless nodes, causing a DoS.

Clearly the potential of various security threats exists with 6LoWPAN. Ideally, IPSec will be used between the application and the sensor node. Although IPv6 offers a complete end-to-end secure solution through IPSec, the current IEEE 802.15.4 frame size prevents its use. Raza, et al. have developed a way of using a compressed form of IPSec to secure end-to-end communication. However, it is limited to IPSec AH mode and data confidentially cannot be achieved [11]. To overcome these shortcomings, IPSec can be used between the application and the Edge Router to ensure data integrity and confidentiality, along with firewall technology on the Edge Router to limit traffic flow in and out of the WSN. On the wireless side, strict security policies managed by the Edge Router could be used to prevent unauthorized nodes from joining the network. Once authenticated to the network, a session key dynamically created by the Edge Router could be issued to allow encrypted traffic between the node and the Edge Router.

3. CONCEPT OF OPERATION

As illustrated in Fig. 3, a typical use case of a WSN is a sensor node, S, that performs a measurement on some physical parameter (e.g., temperature or humidity), and passes the data frame to an intermediate node acting as a router, R, and transmits the packet to the Edge Router that routes the datagram through the Internet to the application. While using environmental energy harvesting techniques, local energy storage in a supercapacitor will allow periodic, short-duration periods of elevated power consumption. If conditions limit the amount of power that is available for prolonged periods, the frequency of data transmissions can be dynamically adjusted to reduce consumption.

3.1 SELF-FORMING AND SELF-HEALING MESH NETWORKS

Since the power source for each sensor is somewhat independent, it is essential that a wireless sensor network be capable of self-forming (ad hoc network structure), and self-healing (tolerant of nodes entering and leaving the sensor network). In such an arrangement, each node is a router that can transmit and receive sensor data. A new node can join anywhere on the mesh. This feature (Fig. 11) allows the simple expansion of measurement locations whether temporary or permanent.

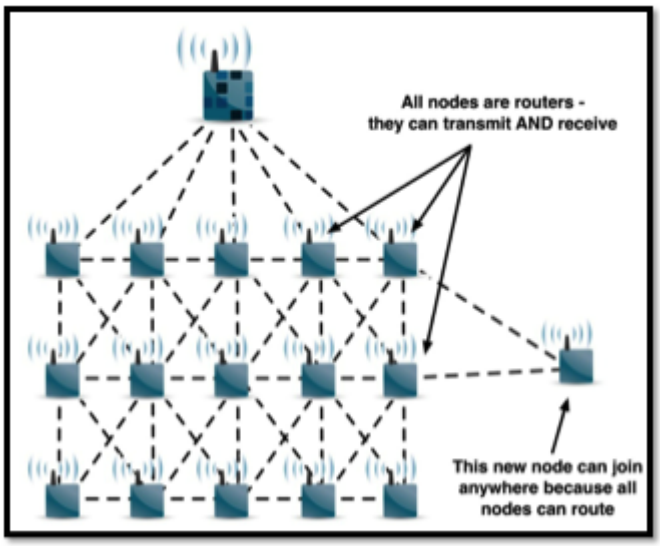

Fig. 11. Self-forming, self-healing mesh networks allow for reliable and robust WSNs.

3.2 POWER COMSUMPTION

One of the main concerns with deploying any wireless sensor network is power consumption. It is vital to know how much power the network will consume so that the most efficient and cost effective method can be chosen to power the devices. Nodes in the network follow a general state machine during their lifetime (Fig. 12).

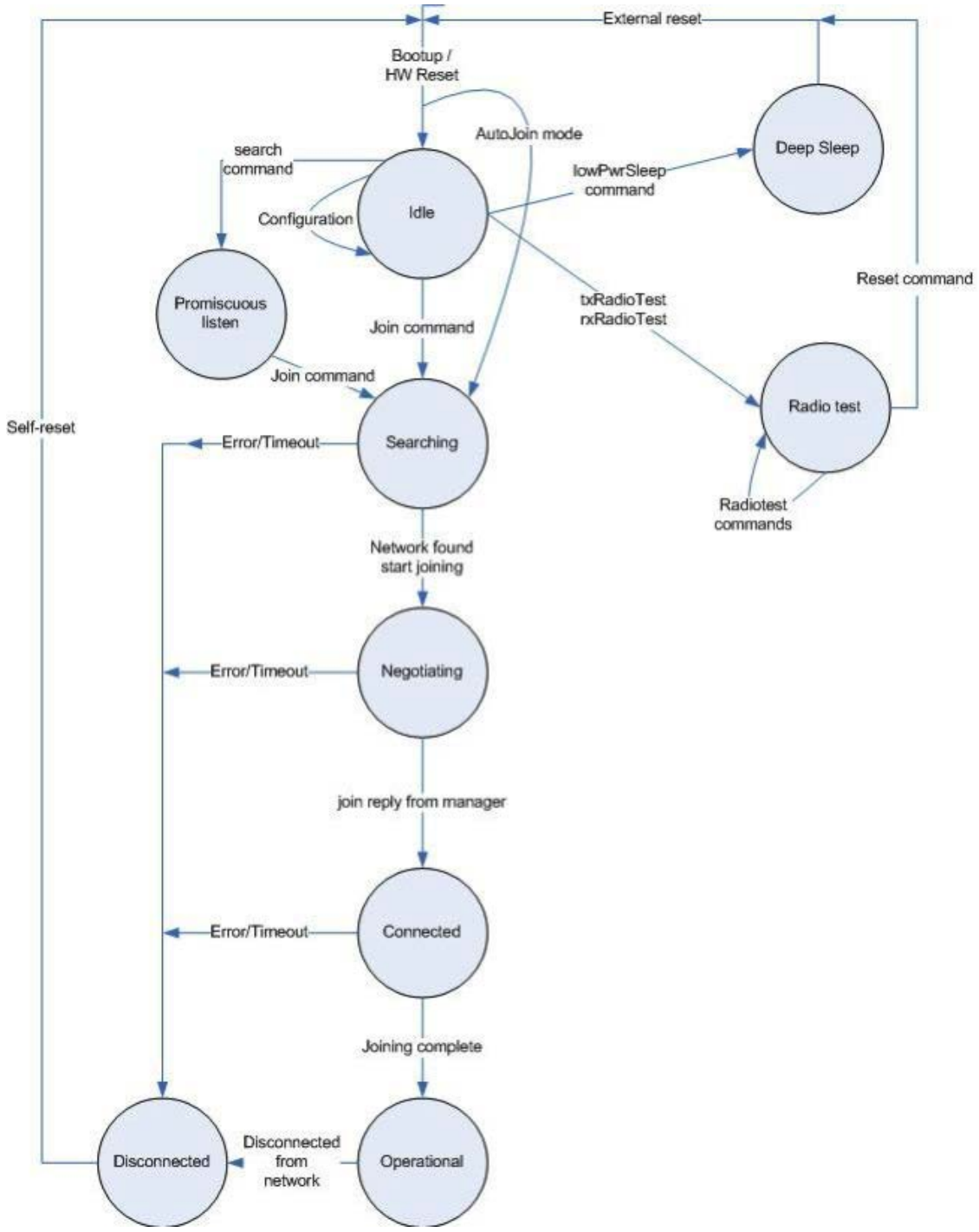

Fig. 12. Node (Mote) State Diagram

To arrive at a baseline power estimate for a hypothetical sensor node (Table 3), it is assumed that the wireless sensor node includes signal conditioning and digitization electronics for four thermocouples, a small microprocessor, and a radio transceiver that consumes twice as much power during the transmit cycle as the most efficient available ZigBee [12] transceivers – note ZigBee is not an IP compatible device so twice the power consumption is assumed.

It is also assumed one transmission of data from each node every 10 seconds as well as several relays of data from other nodes every second. It is also assume that low-power, commercial off-the-shelf components are used and that power to the thermocouple cold-junction compensation (CJC) sub-circuits can be turned off between measurements.

Table 3. Power budget for a self-powered wireless sensor node

Transceiver, including encryption	9 mW average
Microcontroller	200 μW
Four channels of CJC and amplification	1 mW average
Quad 12-bit analog-to-digital converter	18 μW
Miscellaneous circuitry	<3 mW
Power loss in 85% efficient power conversion/management circuit	2 mW
Total	**15 mW average**

An estimate of 20 mW of continuous power consumption should serve as a conservative power target. If this estimate is too low, the frequency of transmission can be decreased (as mentioned above) or the size of the environmental power harvesting apparatus will need to be increased.

3.3 HANDS-ON EXPERIENCE

Two factors led to the desire of gaining some hands-on experience with current state-of-the-art sensor networks. The first motivation was to verify the estimated power consumption is adequate. The second is to gain experience with sensor nodes entering and leaving the mesh network. It should be noted that the commercially available nodes do not have the quality of sensors required for a NPP environment.

The SmartMesh IP wireless sensor network developed by Dust Networks is an embedded wireless sensor network that is based on 6LowPAN and IEEE 802.15.4e standards. The SmartMesh IP includes the breakthrough Eterna SoC technology that enables up to 8x lower power consumption than competitive solutions even in harsh, dynamically changing RF environments. The network is a self-forming multi-hop mesh of nodes, which are known as motes that collect and relay data. The motes pass the data on to a network manager which monitors and manages network performance, security, and exchanges data with a host application [13]. While these commercially available motes are normally battery powered, this technology was selected for hands-on examination since these are the most energy efficient motes currently available. As shown in Fig. 13, the test configuration consisted of one manager board, five motes (nodes) with a resistive temperature detector (RTD) on each mote, and two Universal Serial Bus (USB) interface cards.

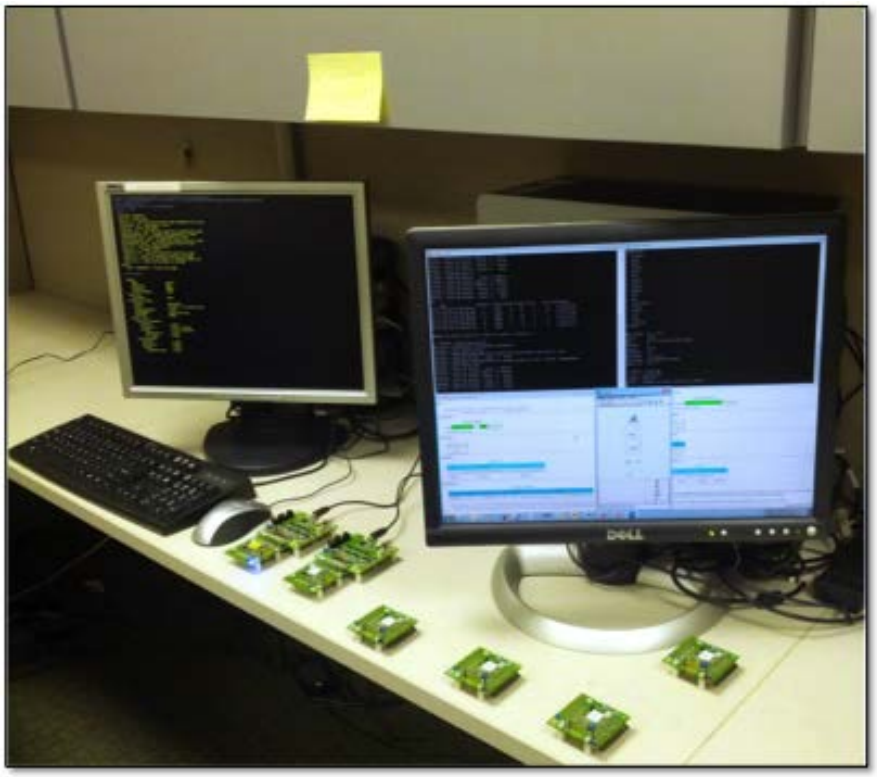

Fig. 13. SmartMesh Test Configuration.

Additional information is included in the Appendix to allow others to duplicate the experience with this wireless sensor network.

3.3.1 Power Consumption Measurements

As shown in Fig. 12, there are three very important states: Idle, Searching, and Operational. These are the three main states that the nodes (motes) will operate. To measure the power consumed in each of these states, a simple experiment was constructed to measure this data. The test setup consisted of a power supply to power the device and a digital multi-meter to measure the current. The node was forced into the various states by using its Command Line Interface (CLI). The current drawn by the device was measured in each of the three states and recorded over a two minute time period. Simple radio tests were conducted to measure how much current the devices would draw when transmitting and receiving packets. The results are displayed in the tables below.

Table 4. Idle State, Power Measurements

State	Maximum (mA)	Minimum (mA)
Idle	0.627	0.186

Table 5. Search State, Power Measurements

State	Maximum (mA)	Average (mA)
Searching	5.375	1.104

Table 6. Operational State, Power Measurements.

Number of Motes (excluding mote under test)	Maximum (mA)	Average (mA)
0	0.591	0.034
1	0.619	0.036
2	0.652	0.041
3	0.594	0.041
4	0.777	0.040

Table 7. Transmit State, Power Measurements

State	Transmit Power (dBm)	Maximum (mA)
Transmit	0	4.963
	+8	12.334

Table 8 Receive state, Power Measurements.

State	Maximum (mA)	Average (mA)	Minimum (mA)
Receive	4.722	4.663	4.537

3.3.2 Analysis of Power Measurements

As seen in the tables above, some of the various states require power slightly exceeding the estimated power requirement. However, it is important to note that the commercially available nodes were not designed to be powered via power harvesting and additional power management techniques can be applied through custom designed electronics. For example, the commercially available nodes control only the RF circuitry (i.e. the radio that sends the wireless signals) to minimize power consumption. Overall, the estimated power budget appears to be appropriate, but additional engineering is needed to enable the reality of a wireless sensor node that is powered solely from environmental energy.

3.4 OPERATION DURING NORMAL AND "OFF-NORMAL" CONDITIONS

ORNL/TM-2012/442 [14] examined power harvesting methods utilizing kinetic, thermal, and radiant sources. These power harvesting technologies were characterized by the individual power density available, physical reliability in harsh environments, and feasibility of powering various sensors. Additionally ORNL/TM-2012/442 [14] assessed how well these various technologies, at their current maturity levels, met the requirements of a wireless sensor network in a NPP. When all factors were considered, the harvesting of thermal energy was deemed the most viable option. Therefore the remaining paragraphs in this section (Section 3.4) assume thermal energy as the environment energy source.

The majority of thermal harvesting devices feature no moving parts and, if they are not subjected to

severe environmental stresses, relatively long effective life spans. The maximum achievable efficiency for any thermodynamic device is limited to its theoretical Carnot efficiency which is determined by the difference in temperatures of the heat source and the heat sink. Greater temperature differentials yield greater theoretical efficiencies. Thermoelectric generators (TEGs) utilize the Seebeck effect to extract electrical energy from a temperature difference between two surfaces. Semiconductor thermocouples, consisting of one p-type material and one n-type material, are usually used in thermoelectric harvesters. Bismuth telluride (Bi2Te3) is the most often used material.

The second primary thermal harvesting devices use pyroelectric materials that derive energy from cyclical changes in temperature. Different temperatures create different degrees of spontaneous polarization in the bulk of the material. When material temperature changes, the amount of charge collected by metal electrodes on opposite surfaces of the material changes also, producing an AC electrical current. Some pyroelectric materials maintain their properties at temperatures beyond 1200°C, providing the potential to reach high Carnot efficiencies. While pyroelectric devices do compare favorably with the more mature TEG technologies, additional research is needed before pyroelectric devices can be deployed on a wide scale.

During normal operations it is anticipated that sufficient environmental energy can be harvested so that one transmission of data from each node every 10 seconds can be realized. Additionally, each wireless sensor node will transmit several relays of data from other nodes every second. Since there are many sources of thermal energy with sufficient temperature differential, it is anticipated that steady state operation will not be an issue. During startup, there may be periods of time where there is not a sufficient temperature differential to transmit the data from each node every 10 seconds. In these cases, the time between transmissions will need to be adjusted or an alternative power source, such as a small rechargeable battery, will be needed.

Because nodes scavenging certain types of energy, such as thermal, could continue to operate during "off-normal" conditions such as extended station blackouts (SBOs) and during periods when operation of the plant's internal power distribution system has been disrupted, measurements identified as critical to accident management should be among the first targeted for implementation. The availability of this data would be invaluable not only to operators trying to manage an accident situation, but also to the teams responsible for post-incident analyses. Self-powered WSNs and the networks that tie them together will provide an opportunity to make substantial improvements in the reliability and safety of modern NPPs. It should be noted that while TEG technologies do depend on temperature differentials that will eventually disappear during a severe accident scenario, wireless sensor nodes powered by TEG technology would be powered during the progression of a severe accident – arguably the most important period. Once environmental power is inadequate to support data transmission from each node every 10 seconds, the affected nodes could increase the time between transmissions so that this vital data could continue to be transmitted. Even if the time between transmissions increased to minutes or hours, this information would be extremely valuable to personnel trying to mitigate a severe accident situation.

4. NEXT STEPS

As documented in ORNL/TM-2012/442 and this report, wireless sensor networks can become a reality with a moderate level of R&D. Over the past eighteen months, it has been shown that wireless sensor networks are less expensive, more flexible, and more reliable in industrial settings when compared to their wired counterparts. In a NPP environment, environmental energy sources are readily available. Based on the maturity level of the various technologies, thermal energy harvesting seems to be the best solution.

The values for environmental parameters listed below were obtained from a preliminary safety analysis for a U.S.-designed commercial reactor [15] and seem consistent with other values found in literature. Values shown in Table 9 are the maximum expected for worst-case locations in primary and secondary containment. Because we are interested in the effects on enclosed electronics, radiation levels for only gamma and neutrons are of interest. Obviously, any wireless sensor system being considered for deployment should be tested in environments at least as harsh as the sensor network is designed to operate. For example, if the wireless sensor network is desired to operate during a severe accident scenario, the harsher environment should be tested based on the location of the sensor nodes.

Table 9**. Ambient environmental for normal and accidennt scenarios.**

Operating Scenario	**Primary Containment**				**Secondary Containment**			
	Temp	RH	γ	N	Temp	RH	γ	N
	°C	%	Gy/h	$cm^{-2} \cdot s^{-1}$	°C	%	Gy/h	$cm^{-2} \cdot s^{-2}$
Normal operations	65	90	0.2	6E4	60	90	0.02	low
Shutdown, pumps operating	65	90	0.2	low	60	90	0.02	low
Cladding, RPV, and pipe rupture	170	steam	2×10^5	low	142	steam	2×10^3	low

While it has been shown that wireless sensor networks are possible, there are many R&D activities that need to be accomplished prior to realizing wireless sensor networks in a NPP environment. These necessary activities can be grouped into two complementary areas, each being three years in duration. The first group is to design and develop individual wireless sensor nodes and the supporting technologies. The second group is the development of a demonstration system with a draft technology transition plan.

The first three years would focus on 1) developing requirements for the various sensors with an emulator for these sensors to expedite testing, 2) develop, design, and fabricate power efficient solid-state devices, and 3) conceptual system design capable of surviving in the intended environment which includes wireless security verification. While it has been shown that wireless sensor networks are possible, there are many R&D activities that need to be accomplished prior to realizing wireless sensor networks in a NPP environment. It is also envisioned that testing to the environments shown in Table 9 can be cost-shared with other NEET projects to help minimize costs during either the first or second phase.

The second three years would focus on designing the demonstration system, fabricating the demonstration system and developing a draft Technology Transition Plan.

While a large scale implementation is not included in this phase, it is envisioned that once the demonstration system is operational a large scale implementation could arise through a cost sharing arrangement, especially as this wireless alternative is demonstrated to be more cost effective than hardwired cabling.

5. CONCLUSIONS

Many industries are beginning to utilize mesh networks to replace conventional point-to-point wiring, reaping the cost savings associated with eliminating the communications cabling. In addition to these cost savings, these mesh networks open the potential for greater expansion in instrumentation in the plant that could augment human performance, provide additional data on plant equipment and component status, and facilitate online assessment of the material condition of plants. By combining wireless communications with environmental power harvesting, truly wireless sensor networks become possible. These truly wireless sensor networks would benefit all new reactor designs and fuel cycle facilities (e.g., enrichment facilities, mixed oxide fuel fabrication facilities, and used fuel deposition facilities) by helping to reduce maintenance and operating costs associated with installing wiring for replacement or temporary diagnostic sensors.

Robust digital instrumentation communication techniques and architectures are essential to address the potential for greater expansion in instrumentation in industrial environments that could augment human performance, provide additional data on plant equipment and component status, and facilitate online assessment of conditions. To develop wireless alternatives to costly hardwired-cabling for real-time, online monitoring, demonstration of a high-reliability, secure wireless communications system for continuous data transmission is necessary. Wireless sensor networks would benefit all new reactor designs and fuel cycle facilities (e.g., enrichment facilities, mixed oxide fuel fabrication facilities, and used fuel deposition facilities) by helping to reduce maintenance and operating costs associated with installing wiring for replacement or temporary diagnostic sensors.

This report has examined the communication requirements for such a wireless sensor network as part of the instrumentation infrastructure at a NPP. These wireless sensor networks would not operate in a stand-alone mode. Instead these wireless sensor networks would interface with the existing communications (wired and wireless) infrastructure. While physical security to the sensor nodes will provide some level of protection, it will still be necessary to address the possibility of both passive and active cyber-attacks since the packets are transmitted through the air where anyone could theoretically eavesdrop. In a passive attack, the attacker does not insert any information into the network, but listens, and attempts to retrieve vulnerable information. In active attacks, messages are inserted and as a result the operation is disrupted or some nodes may be harmed – impersonation and spoofing are examples of active attacks. An attacker may also attempt to disrupt the operation of the network by causing a large amount of control packets that can cause overloading of wireless links and render the network unavailable.

This report also examined the current state-of-the-art of for sensor networks and the associated security solutions which will be applicable to wireless sensor networks powered by environmental energy, such as thermal energy in a NPP. Some hands-on experimentation was conducted to verify power consumption estimates and to demonstrate the ease at which sensor nodes could be added or removed. Finally, this report examined a general concept of operation including “normal” and off-normal conditions and how wireless sensor networks could be utilized to enhance the affordability, safety, and reliability of nuclear power.

6. REFERENCES

[1] K. Korsah and e. al., “Instrumentation and Controls in Nuclear Power Plants: An Emerging Technologies Update, CR-6992,” Nuclear Regulatory Commission, Washington, D.C., 2009.

[2] B. Kaldenbach and e. al., “Assessment of Wireless Technologies and Their Application an Nuclear Facilities, CR-6882,” Nuclear Regulatory Commission, Washington, D.C., 2006.

[3] R. Allan, “Energy Harvesting Powers Industrial Wireless Sensor Networks,” *Electronic Design*, pp. 22 - 29, 20 September 2012.

[4] Peryton's Network Visibility, "Multi Channel 802.15 ZigBee, RF4CD, 6LoWPAN Protocol Analyzer for 2.4 GHz Networks," 2013. [Online]. Available: http://www.perytons.com/products/general-wireless-page/peryton-m/. [Accessed 05 August 2013].

[5] S. Deering and R. Hinden, "RFC 2460, Internet Protocol, Version 6 (IPv6) Specifications," p. 1, December 1998.

[6] S. Kent and R. Atkinson, "RFC 2401, Security Architecture for the Internet Protocol," 1998.

[7] 8.-2. IEEE Standard, "Wireless Medium Access Control and Physical Layer Specifications for Low-rate Wireless Personal Area Networks," *ISBN 0-7381-3677-5*, May 2003.

[8] D. Gascon, "Security in 802.15.4 and ZigBee Networks," *http://www.sensor-networks.org/index.php?page=0903503549*, 5 February 2009.

[9] N. Sastry and D. Wagner, "Security Considerations for IEEE 802.15.4 Networks," *WSE '04*, 1 October 2004.

[10] H. Kim, "Protection Against Packet Fragmentation Attacks at the 6LoWPAN Adaption Layer," *IEEE*, 2008.

[11] S. Raza, S. Duquennoy, T. Chung, D. Yazar, T. Voigt and U. Roedig, Securing Communication in 6LoWPAN with Compressed IPsec.

[12] ZigBee Alliance, ZigBee Specification, Document 053474r17, San Ramon, California: ZigBee Standards Organization, 2008.

[13] Linear Technology, "Wireless Sensor Networks - SmartMesh IP," 2013. [Online]. Available: http://www.linear.com/products/smartmesh_ip. [Accessed 1 August 2013].

[14] D. A. Clayton, W. H. Andrews and R. Lenarduzzi, "Power Harvesting Practices and Technology Gaps for Sensor Networks," ORNL/TM-2012/442, 2012.

[15] Taiwan Power Company, "Lungmen Units #1 and #2 Preliminary Safety Analysis Report," Taiwan Nuclear Energy Council, New Taipel City, 2007.

[16] Linear Technology, "SmartMesh IP Quick Start Guide," [Online]. Available: http://www.linear.com/products/smartmesh_ip. [Accessed 01 August 2013].

[17] Linear Technology, "SmartMesh IP Tools Guide," [Online]. Available: http://www.linear.com/products/smartmesh_ip. [Accessed 01 August 2013].

APPENDIX A. EXPERIENCE WITH SMARTMESH IP

This appendix summarizes the experience ORNL staff had with the SmartMesh IP from Dust Networks with sufficient instructions to allow others to duplicate the hands-on experience.

A.1. GETTING STARTED

This section will cover important topics such as how to use the Command Line Interface (CLI) and Application Programming Interface (API) for the manager and mote, how to connect and interact with the Low Power Border Router (LBR), and general information about the system and various applications in the SDK. The test setup consisted of the Dust Networks Development Kit which contains the following: one Manager board, five mote boards, and two interface cards. The first task is to follow this link www.linear.com/starterkits and download the following packages: Serial Mux, for connecting multiple applications to the manager, and the SmartMesh SDK Software Development Kit. The LBR package will also need to be downloaded and will be covered in a later section of this appendix.

A.2. COMMAND LINE INTERFACE (CLI)

First, connect the Manager/Mote to one of the DC9006 interface cards and then connect to the PC via a USB interface. The drivers for the devices should be downloaded automatically; if they do not, follow the link above to download the required FTDI drivers. Eight total virtual COM ports should be added to the computer once everything has finished, four COM ports for the Manager and the four are for the mote. The COM ports will be numbered COM1, COM2, COM3, and COM4, and the CLI COM port will be the third of the four, in this case COM3. The fourth COM port is reserved for the API of the Manager/Mote. Next, connect to the Manager/Mote using a serial terminal program such as PuTTy, Hyperterminal, or TeraTerm. The information needed to connect to the devices is provided below [16].

Manager:	Mote:
• Baud Rate: 57600	Baud Rate: 9600
• Data Bits: 8	Data Bits: 8
• Parity: None	Parity: None
• Stop bits: 1	Stop bits: 1
• Flow Control: None	Flow Control: None

The Manager CLI is different from the Mote CLI because it requires users to be logged in to use it. The Manager CLI has two default views, user and viewer, that can be chosen and each has its own command set. To login to the Manager CLI use either of the following login commands “login user” or “login viewer.” Use the help command at any time during the session to get a list of commands available for the device. Refer to the Manager/Mote CLI guides provided in the SmartMesh documentation for more information on what commands are available and how to use them.

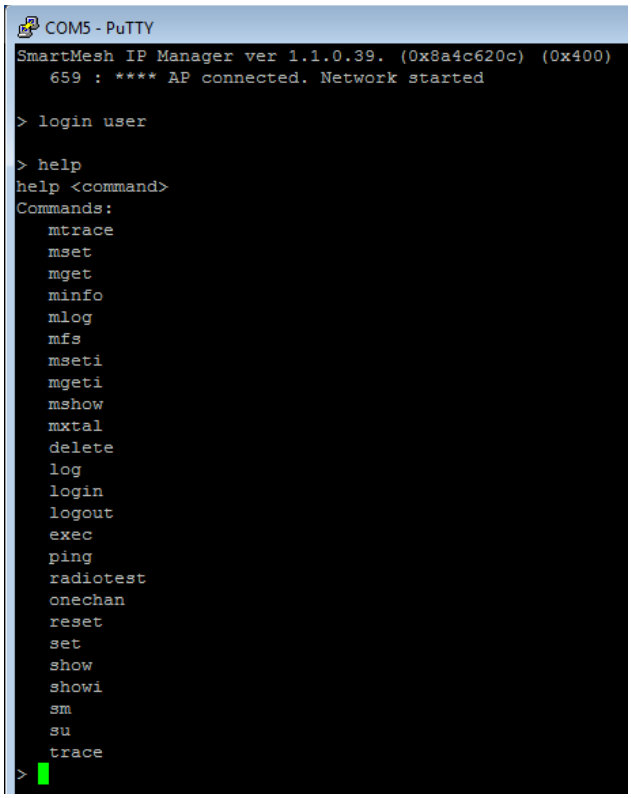

Fig. 14. Manager Command Line Interface (CLI)

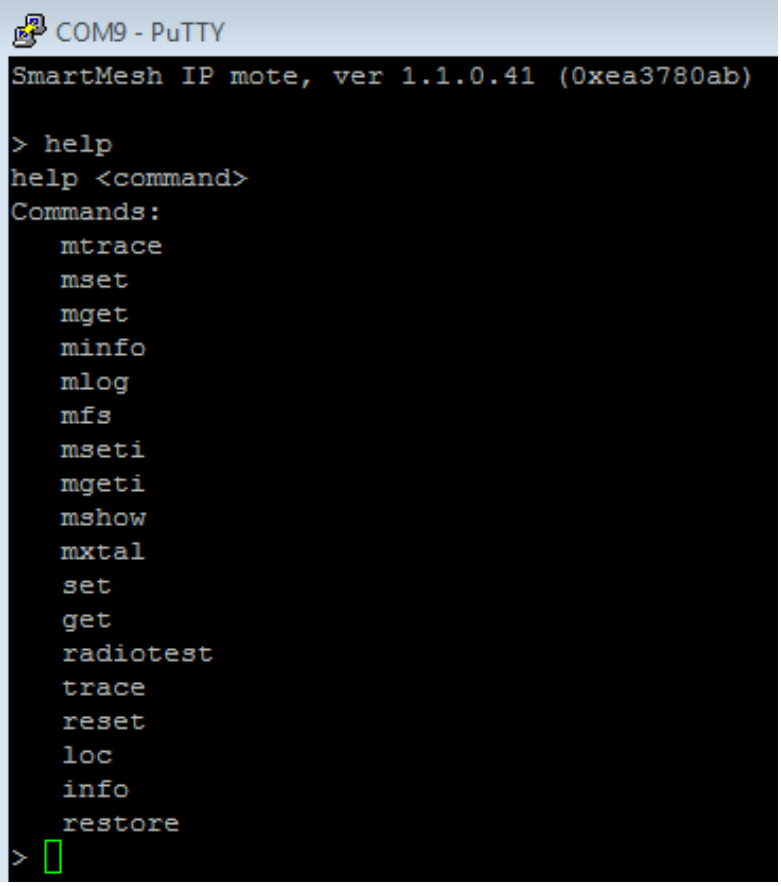

Fig. 15. Mote Command Line Interface

A.3. API

The API is different from the CLI because it is designed for machine-to-machine interactions whereas the CLI is designed for human-to-machine interactions. However, there are a few applications provided which allow the user to interact with the Manager/Mote devices. Python will need to be installed in order to run these applications; version 2.7.2 was the current version when the applications were written. The applications are stored in the win directory of the SmartMesh SDK folder. The two most important applications are the APIExplorer, which allows the user to interact with the Manager/Mote using the API commands, and the LBRConnection, which allows an interaction between the mesh network and the Low Power Border Router (LBR). Refer to the LBR section of this appendix for more information on how to use the LBR and LBRConnection application. The API commands are very similar to the CLI commands but offer slightly more functionality.

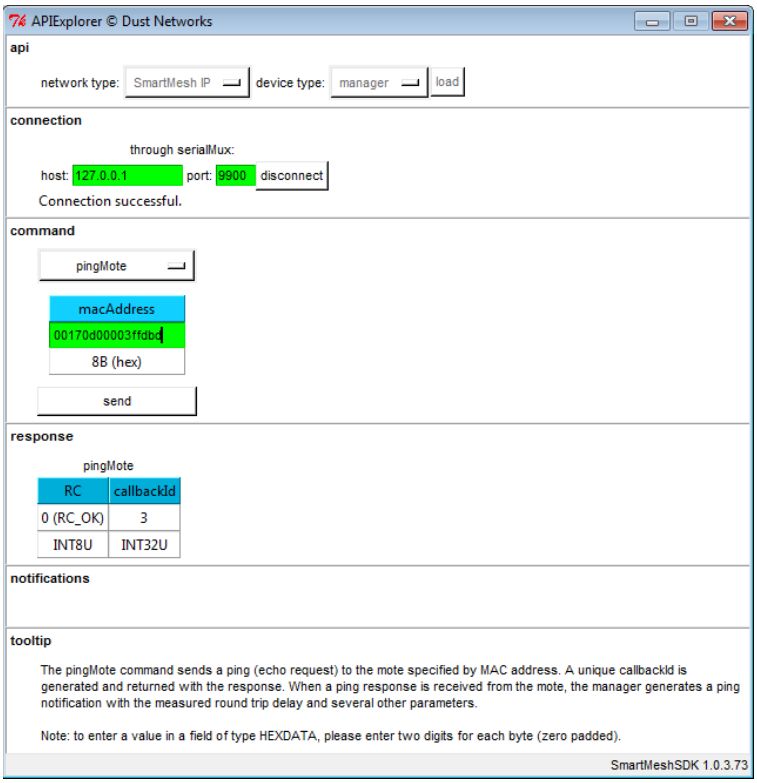

Fig. 16. Manager API User Interface

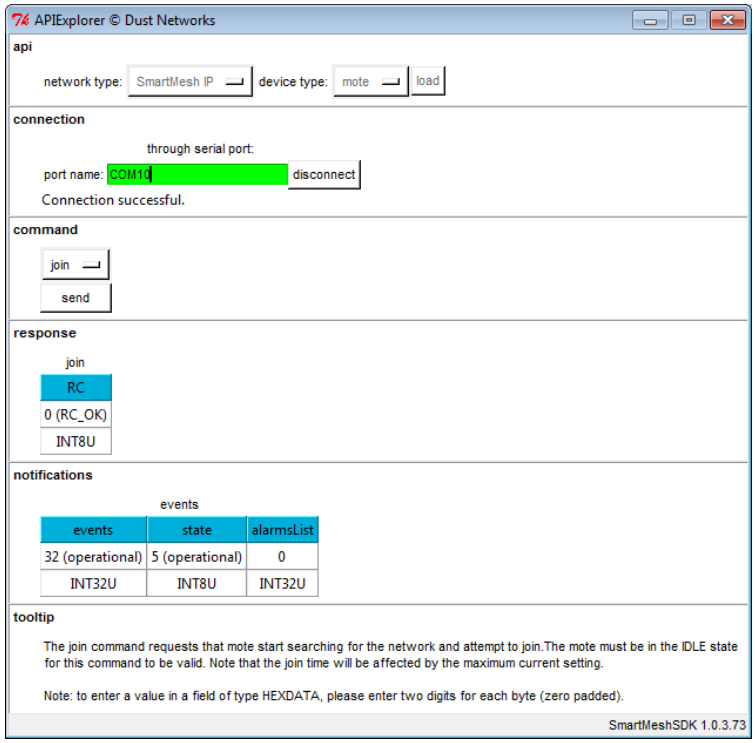

Fig. 17. Mote API User Interface

A.4. CONNECTING TO THE MANAGER/MOTE API

Enter the fourth COM port number for the Manager, created when the Manager was first connected to the PC, into the serial port box of the API user interface. If the application is unable to connect using this COM port try installing and then connecting using the Serial Mux application. In order to connect to the Mote API user interface, the mote must first be set into "Slave" mode. The mote can be put into this mode by using the CLI and entering the following commands "set mode slave" and "reset". It is possible to have both the CLI and the API running at the same time because they are communicating using two different serial ports. This is very useful because important information can be seen on the CLI while interacting with the devices using the API.

Besides the API User Interface, Dust also offers simple start up scripts, SimpleMgr and SimpleMote, so the user can write scripts using the general API. These scripts take care of all the pre-setup so that the user can easily create their own application without the initial overhead. The general API for the system can be found by using the "index.html" file located in the SmartMeshSDK -> doc -> html directory. It is all setup in a documentation generator known as Doxygen so the user should have no trouble finding the information that they need to create their application. [17]

A.5. LOW POWER BORDER ROUTER (LBR)

Most of the information in this section can be found in the SmartMesh IP Tools Guide found in the documentation. This is a brief overview of the purpose of the LBR and also includes testing information performed using the LBR. The LBR sits between the Manager and the Internet. It converts the packet format of the Internet to the 6LoWPAN packet format of the SmartMesh IP network; this enables communication from hosts on the internet to the motes in the network. The figure below shows a typical network setup [17].

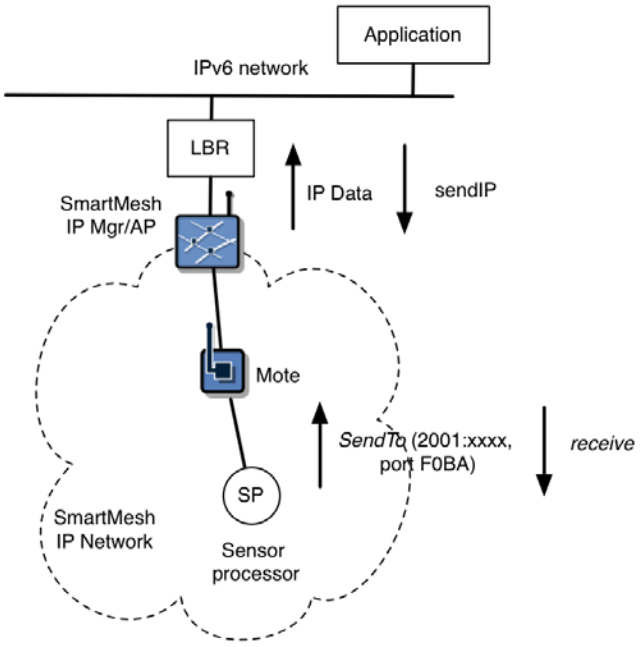

Fig. 18. Typical SmartMesh Network Setup

The LBR performs the following functions:

1. Connectivity – It allows SmartMesh IP Motes to send data to servers on the Internet and vice versa.
2. Compression – Its compression/decompression engine translates IPv6 into 6LoWPAN as data flows between the Internet and the SmartMesh IP network.
3. Addressing – It manages a pool of IPv6 addresses, thereby configuring each SmartMesh IP Mote with a unique, globally reachable IPv6 address.

The LBR can be setup in either the standalone or server modes. In standalone mode the LBR will run as an application on a computer and handles a single SmartMesh IP network. Server mode allows the LBR to run on a server which can be located anywhere on the Internet and is able to manage multiple SmartMesh IP networks. The LBR application is a Python script and can be downloaded from Linear's website. The LBR will also need to run on a computer with a version of Linux. The testing information found in this appendix used the LBR in standalone mode on a computer running Ubuntu version 13.04.

A.6. TESTING COMMUNICATION BETWEEN THE LBR AND THE MESH

The LBR is a vital part of the system because it performs the conversion between IPv6 packets and the 6LoWPAN packets of the mesh network. It is vital to have the network communicating successfully with the LBR and vice versa. The following are simple experiments to test this communication between the network and the LBR.

The test consists of using the Mote APIExplore application to first open and bind a socket, then send the data upstream to the Manager where it will be forwarded to the LBR. The data can then be viewed using either the netcat utility on the Linux machine or by looking at the LBR data log. Below are the steps performed during the experiment.

1. On the LBR/Host machine run the dustlbr.py script as root; this sets up the LBR. Here is the command: "sudo python dustlbr.py aaaa:0000:0000". The last part of the command is the prefix for the LBR connection.

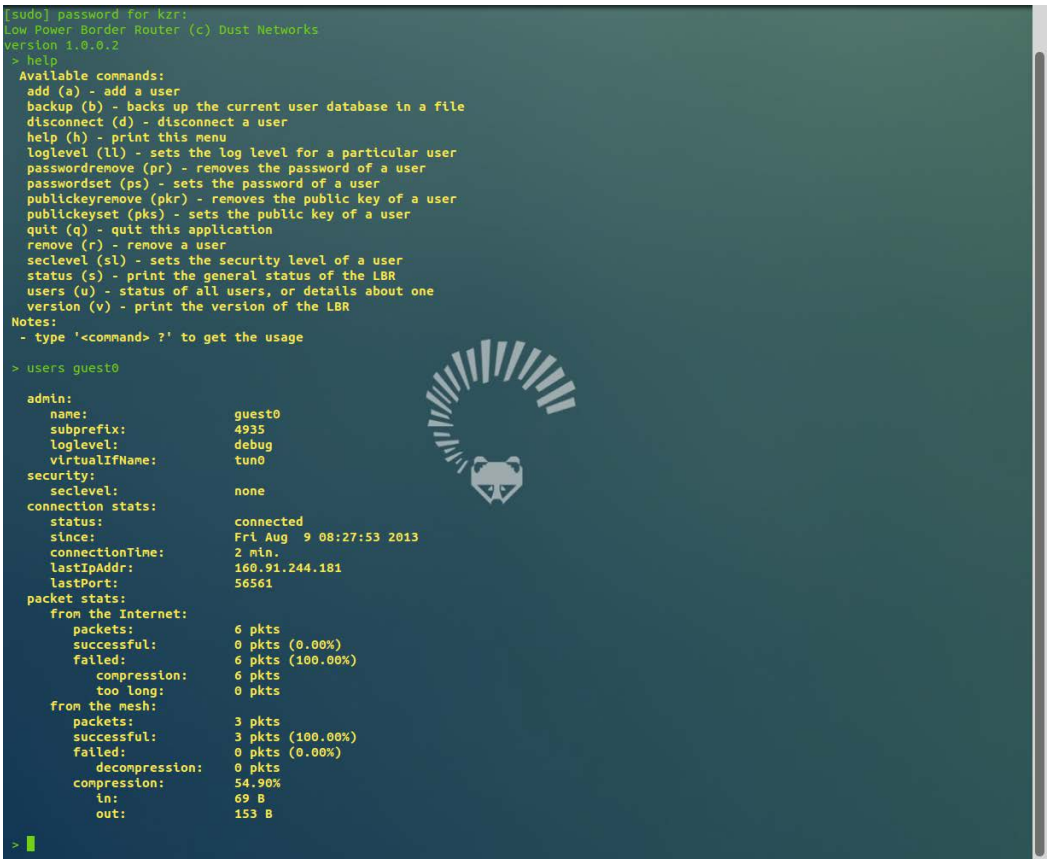

Fig. 19. LBR Command Line Application running on Ubuntu 13.04

2. To connect to the LBR it is necessary to create an LBR authentication file (lbrauth) with the following information: IP address of the LBR machine, the TCP port to connect to the LBR, the username, and the security level of the user. The figure below shows an example guest authentication file [17].

```
# the IPv4 address of the LBR
lbrAddr  = 160.91.245.111

# the TCP port of the LBR
lbrPort  = 80

# username
username = guest

# security level
seclevel = 0
```

Figure 12 – Guest LBR authentication file (guest.lbrauth)

3. On the client machine (connected to the Manager) run the LBRConnection application and connect to the LBR using the lbrauth file. Also connect to the Manager using this application.

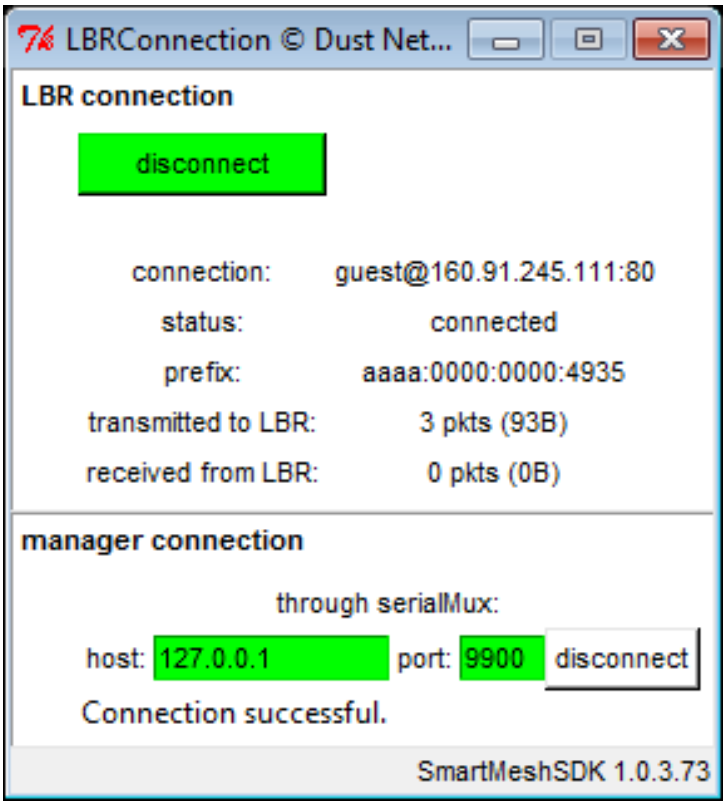

Fig. 20. LBRConnection application connected to the LBR machine

4. Open two APIExplorer applications, one for the manager and one for the mote. Subscribe to all notifications on the manager by using the subscribe command and passing ffffffff and 00000000 into the filter and unack filter spots respectively.
5. On the Mote API use the open socket command to prepare a socket, then use the bind socket command to bind the socket to a port, port 61625 was used in this test.
6. Use the Mote API sendTo command with the following information to send data to the LBR machine: Socket ID (22), IPv6 address of the LBR machine (fe80::1 in this case), destination port (61625), bandwidth as the service type, medium priority, packet ID (1), and a payload in hex.

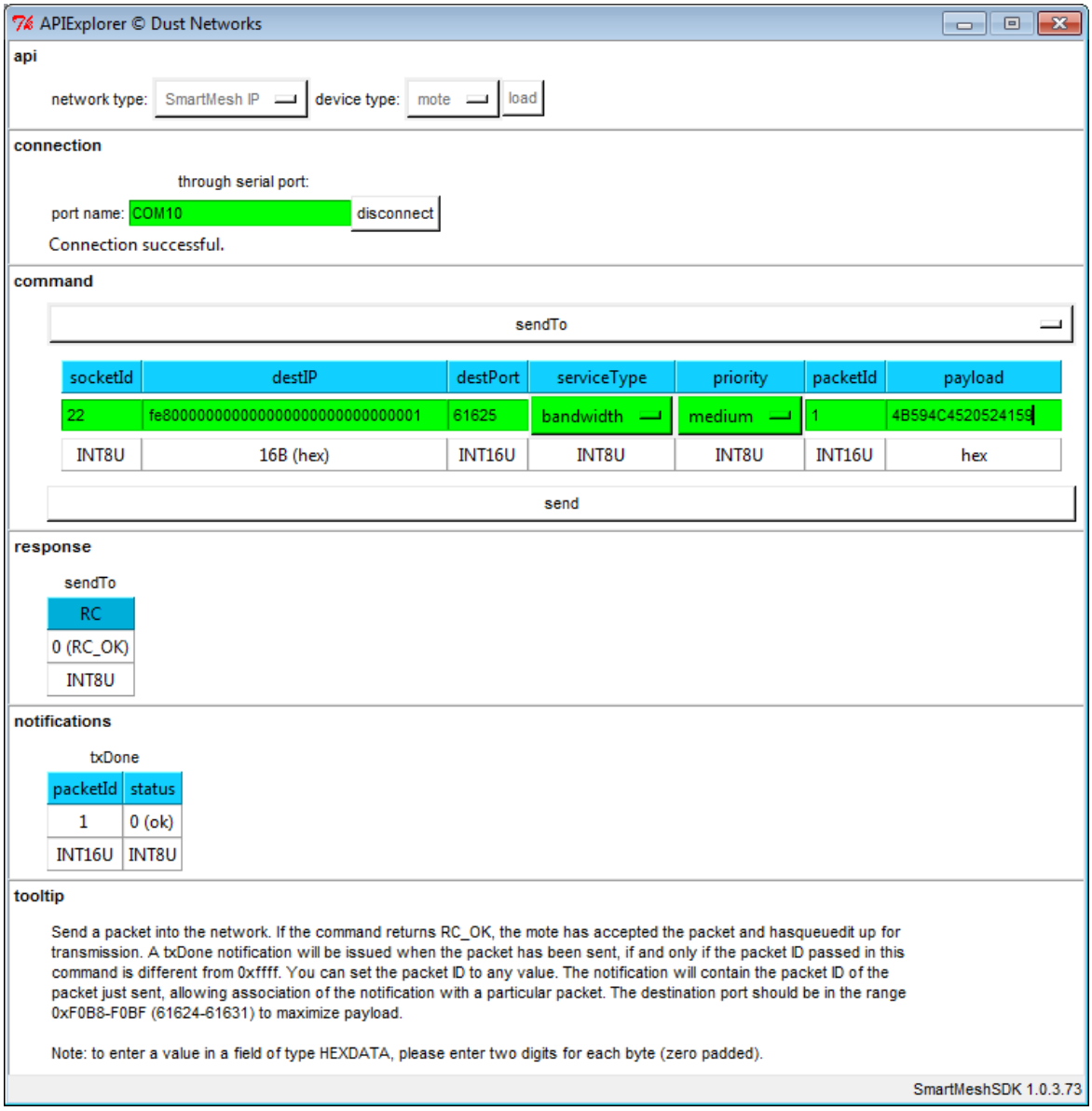

Fig. 21. Example input for the sendTo command

7. To view the data sent to the LBR use the netcat utility or look at the user logs in the logs folder. The netcat command is “nc -6lu <port to listen on>”. In the test above, the payload “4B 59 4C 45 20 52 41 59” which represents the string of ASCII characters “KYLE RAY” was sent over to the LBR. When viewing the user log on the LBR machine it is possible to see information about the packet and also the contents of the payload.

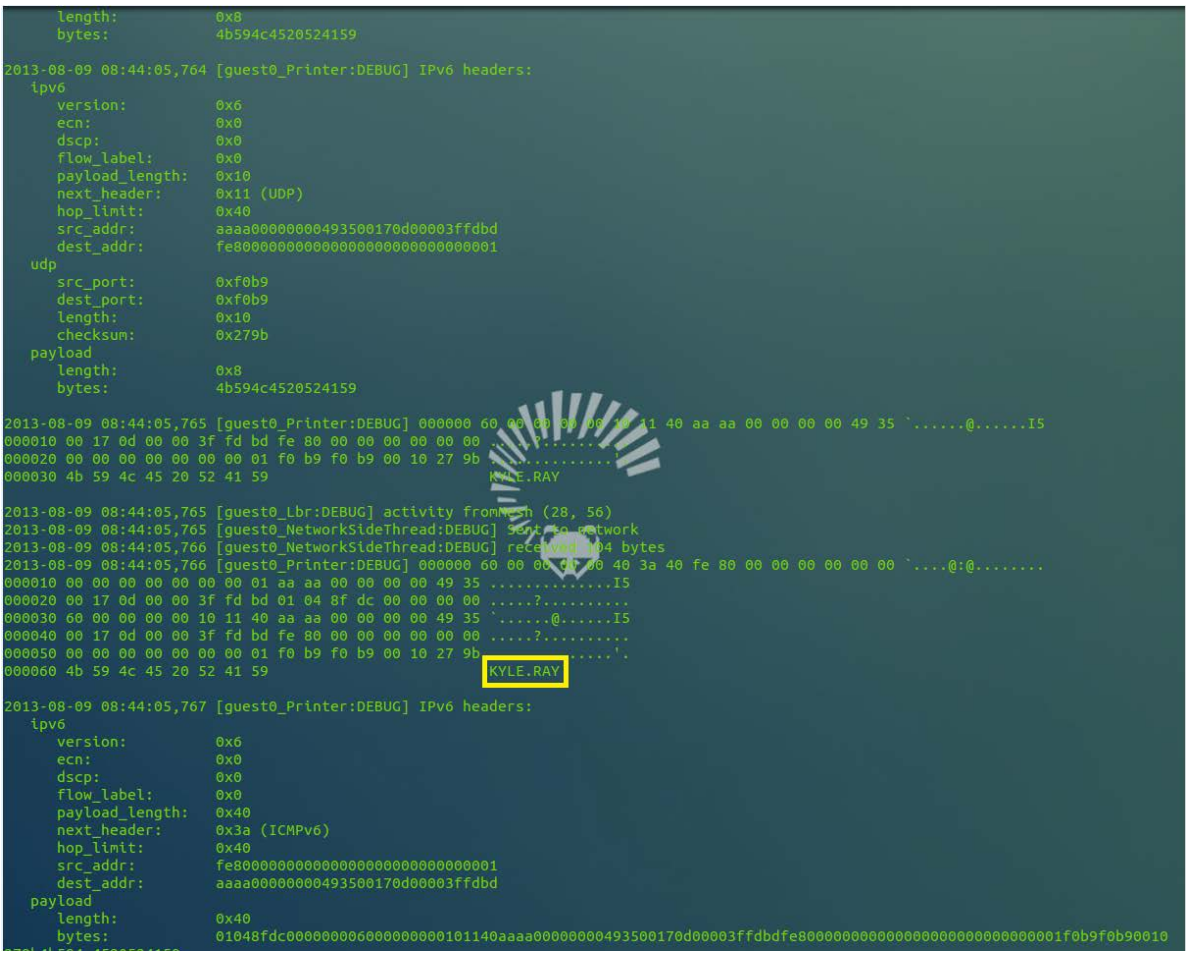

Fig. 22. LBR Guest User Log

The next test consists of sending data from the LBR machine to a mote in the mesh. First get the MAC address of the mote by using the getParameter.macAddress in the mote API. To create the IPv6 address of the mote concatenate the prefix of the LBR, in the case above the prefix was aaaa:0000:0000:4935, and the MAC address of the mote, 0017:0d00:003f:fdbd. To send the data to the mote use the netcat utility command “nc -6u <IPv6 address of the mote> <Port Number>” i.e. nc -6u aaaa:0000:0000:4935:0017:0d00:003f:fdbd 61625; type a string and press enter. When performing this test, a receive notification should come up in the Mote API. However, when we attempted this nothing happened. Watching the mote CLI when the packet was sent we noticed that a receive error was generated, NET rx err=7. The explanation for this error has yet to be found but it could be because the mote was having trouble reading the format of the packet [17].

A.7. OVERVIEW OF SMARTMESH HELPER APPLICATIONS

The SmartMesh SDK also includes various other applications that, while not as useful as the APIExplore, still serve a purpose. The first is a LEDPing application which allows the user to toggle the LED on any mote in the network. This is useful for checking communication between the Manager and the motes in the network.

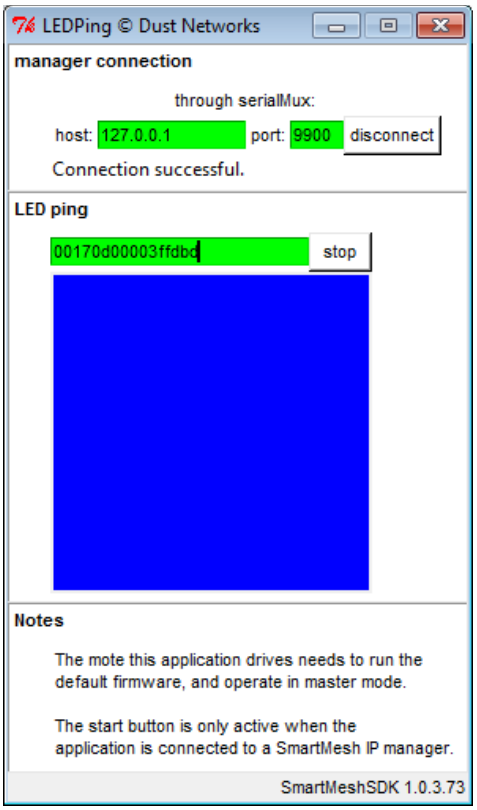

Fig. 23. LEDPing Application User Interface

Another useful application is the MgrListener or Manager Listener, which displays the name of the notification and how many were received by the Manager. This application is simple and easy to use but the APIExplore has a better method of doing this. Using the subscribe command in the APIExplore one can see the notification and the information associated with it, instead of just the name and how many were received.

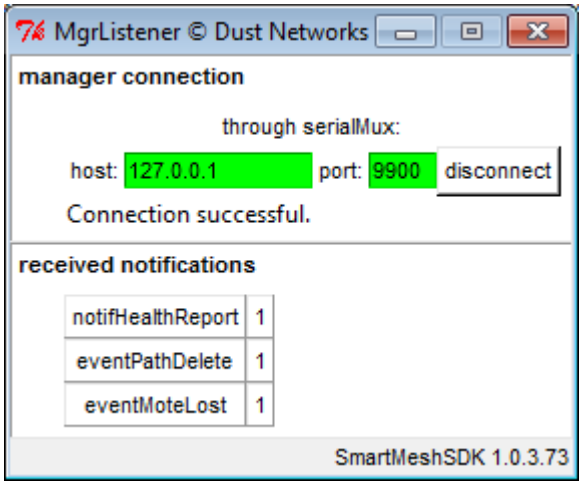

Fig. 24. MgrListener Application User Interface

The PkGen (Packet Generator) will allow the user to specify the number of packets, the rate in milliseconds at which the packets are sent, and the size in MB of the data packets to be sent. It will then proceed to send packets to the mote specified with the parameters set by the user. This is useful when checking packet statistics and network reliability.

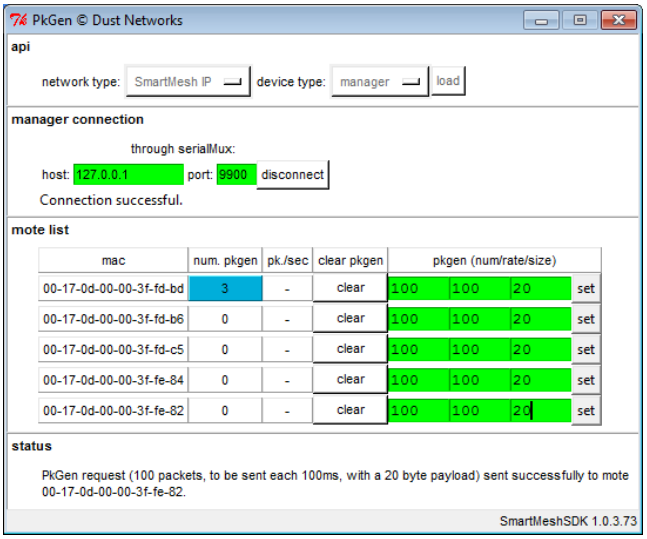

Fig. 25. PkGen Application User Interface

The SensorDataReceiver and Upstream applications work together. The Upstream application, shown in Figure 17, connects to the mote and drives it through its various states. It then opens and binds a socket so the mote can transmit data upstream to the manager or host application. The user can then move the slider to any number of their choosing and send that value to the manager or a host on the Internet by providing its IPv6 address. Before sending the data to the manager make sure that the SensorDataReceiver application is connected either via Serial Mux or serial port connection.

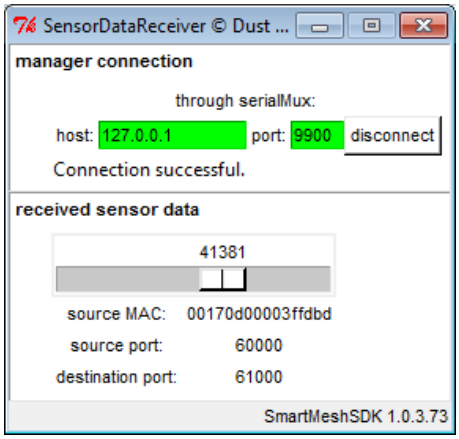

Fig. 26. SensorDataReceiver Application User Interface

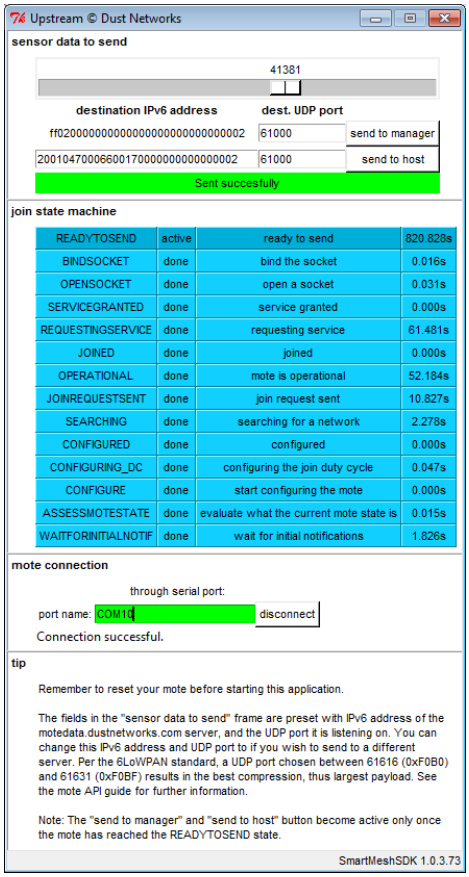

Fig. 27. Upstream Application User Interface

TempMonitor will ping all of the available motes in the network and display the information in an easy to read table. The actual application connects to the manager and queries the motes individually to acquire information such as notification counts, various latencies, and temperature. The user has the ability to set the information publish rate by inserting a time in milliseconds into the rate field.

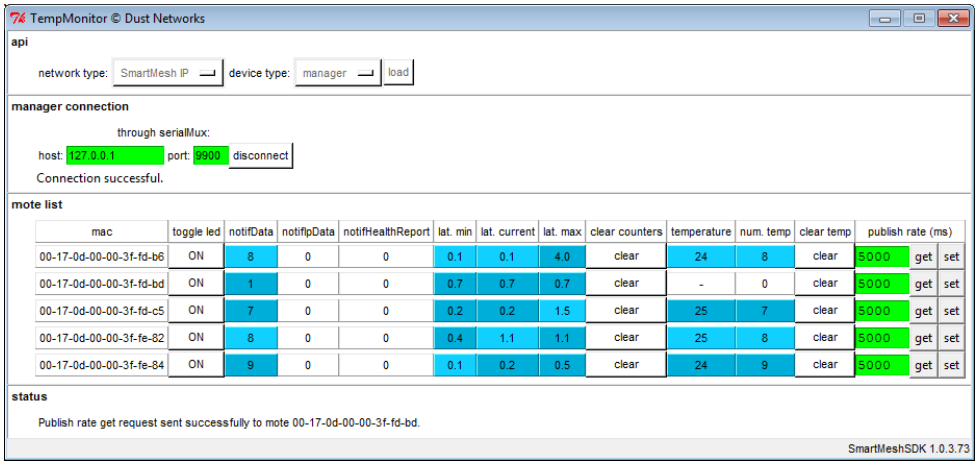

Fig. 28. TempMonitor Application User Interface

This was a brief overview of the applications within the SmartMesh SDK. For more information please refer to the following documents located in the SmartMesh documentation: Manager API Guide, Mote API Guide, SmartMesh Application Notes, and SmartMesh Tools Guide.

A.8. STARGAZER

Stargazer is an application that gives the user a graphical representation of the mesh network.

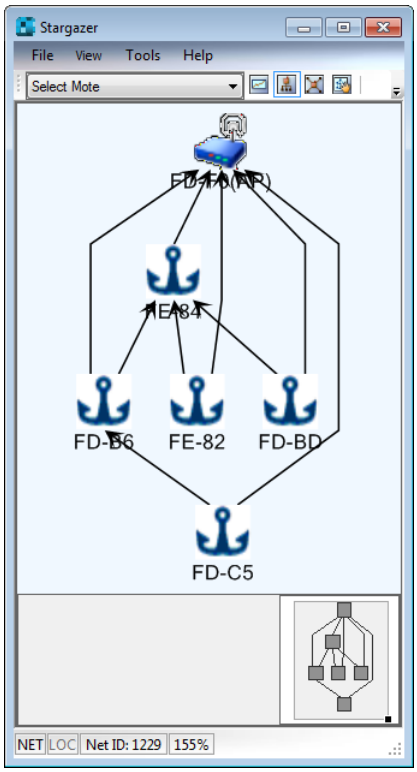

Fig. 29. Hierarchical View

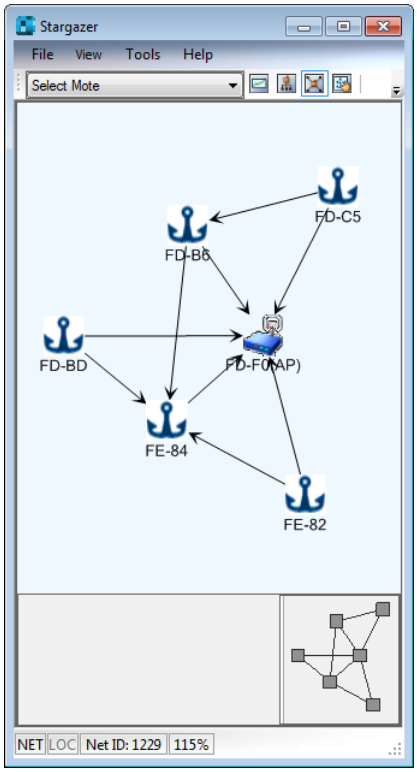

Fig. 30. Radio Space View

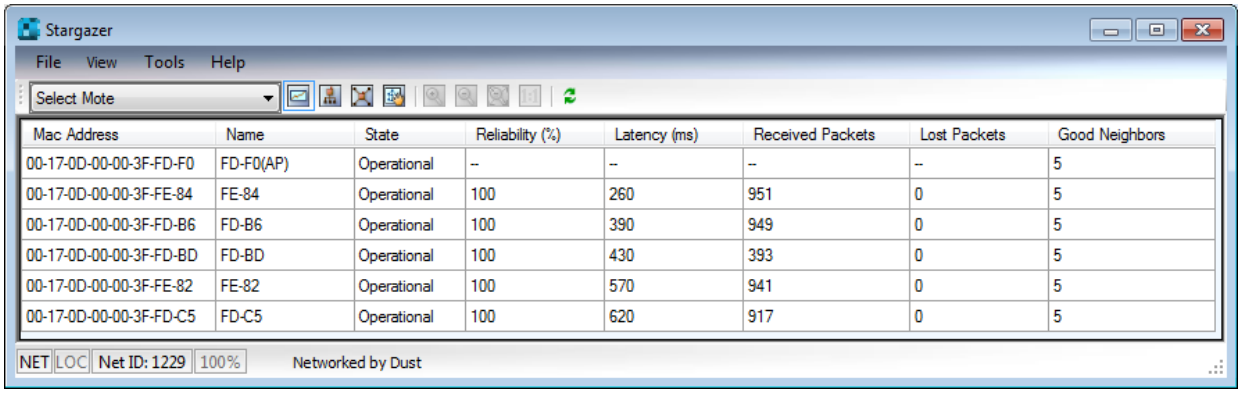

Fig. 31. Tabular View

This allows the user to see information about the motes as well as the actual path connections in the mesh. Having a graphical representation like this is useful because one can determine how many hops there will be between any mote and the manager and therefore have more control over the network layout. A tutorial on how to use Stargazer can be found in the SmartMesh Tools Guide and can be downloaded from the following site www.linear.com/starterkits [17].