

## **EcoMesh: A Scalable Outdoor Heterogeneous Wireless Infrastructure for Real-Time Application Development and Testing**

Thomas McCabe, Jacqueline Stewart, John Allen, Sean Kennedy, Robert Stewart.

Dept. of Electronics, Computer and Software Engineering, The Wireless Research Networking Group,  
Athlone Institute of Technology, Athlone, Co. Westmeath, Ireland.

---

**Abstract:** This paper firstly outlines the development, deployment and application testing of an integrated Wireless Mesh Network (WMN). The challenges for Wireless Sensor Network (WSN) middleware is then presented and includes a discussion on new software languages that are proposed. As real-time services such as critical sensing and real time location applications become ubiquitous extra provisioning is required in order to ensure that these services can operate correctly under such strict delay bounds attached. Many issues have arisen including integration issues, bandwidth and energy constraints for wireless sensor networks, simultaneous multi-point application management for wireless sensor networks and real-time service delivery for these new technologies. The objective of this work is to evaluate the delay and throughput boundaries experienced with heterogeneous wireless networking and in related research to evaluate the challenges faced in current Wireless Sensor Network middleware offerings. Currently some of the most common sensor nodes run on an operating system called TinyOS, the base language for which is nesC. However nesC has limitations in that it is not efficient in the amount of programming required to implement a given task. Also as nesC was originally designed as a stop gap solution for application development. A new alternative is now required to overcome its design limitations and the initial research carried out in this area by the Wireless Networking group is explored.

**Key words:**

---

### **INTRODUCTION**

#### ***1.1 Background and Introduction into current research:***

With the increasing demand for real time applications that can operate over current Wi-Fi standards and offer a broad range of services the Athlone Institute of Technology EcoMesh network has been developed. With modern communication systems having to operate effectively in a range of different locations as well as different climates; the challenge was set to develop a scalable heterogeneous wireless infrastructure that can handle multiple types of traffic (Data, Voice, Video ) from a range of different sources. When developing such a system optimization and scalability are imperative to ensure that overall system performance is able to handle real time applications.

Originally this research was conducted in Athlone Institute of Technology (AIT) by developing a mobile campus based test-bed that contained a two-tier wireless mesh network. The first tier would allow client traffic to be transmitted through the mesh network into a back-haul network. By using wireless mesh access points from Motorola the initial wireless mesh infrastructure was developed. The next step was to incorporate a wireless sensor deployment that would operate in a hostile communication environment surrounded by dense vegetation. After this deployment was successfully installed into the wireless infrastructure web services would be incorporated in order to allow information gathered from the heterogeneous network to be viewed by users in real time.

#### ***Motivation for Research:***

By developing a scalable wireless mesh network with multiple wireless sensor networks operating throughout it the information gained from such a system can be used in many forms of research from climate monitoring to farming. As this system will operate in conjunction with multiple web services it will allow colleagues, research groups and industry to collaborate effectively. Empirical results from the EcoMesh testbed are currently being used to inform the development of more theoretical based studies.

Currently the AIT based test bed operates by using a wireless mesh network that is controlled by a central Motorola RFS 6000 Switch that hosts management software allowing for the network statistics to be easily accessed. The wireless sensor networks that are in place currently operate on the TinyOS based sensor platform while the hardware comes from a range of vendors (Crossbow/Memsic, Lantronix). Currently web services are being engineered to allow users access to the network so that results can be collected and viewed in real-time or historically over a long period of time.

---

**Corresponding Author:** Thomas McCabe, Dept. of Electronics, Computer and Software Engineering, The Wireless Research Networking Group, Athlone Institute of Technology, Athlone, Co. Westmeath, Ireland.

### **Technical Challenges:**

Currently the technical challenges can be divided into the following four areas:

- Provisioning of the WLAN in dense vegetation.
- Optimization of the WLAN to handle multiple traffic types.
- WSN tuning for real-time service delivery.
- Optimization of the software engineering techniques associated with WSN development.

Section 2 illustrates the literature review. Section 3 gives an overview of the provisioning and optimization of the WLAN in the AIT heterogeneous test-bed. Section 4 details the optimization of the current software technologies associated with sensor networks. Section 5 discusses further research within the Wireless Networking Group in Athlone Institute of Technology.

### **Related Work:**

During the literature review process, a number of indoor test beds have been observed. In (Karl, H. and A. Willig, 2007; Tachtazis, C., 2010) WiSar lab developed an indoor office based WSN test-bed that would facilitate the development and analysis of hardware, MAC and routing protocols. The Centre for Adaptive Wireless Systems incorporates a test bed in order to test energy management systems (Spinar, R., *et al.*, 2009). Werner-Allen *et al* (2005) from Harvard University, developed a test-bed using Crossbow Mica Z motes together with MIB600 gateway access boards ("Wireless Sensor Networks, 2011), while the above test-bed deployments mentioned were solely of an indoor and fixed nature, the AIT test bed will operate on a scalable outdoor deployment with its main focus being towards real time service provisioning. The deployment of an outdoor test-bed facilitates parameters such as range, throughput and connectivity to be evaluated throughout the year under varying environmental conditions.. The majority of the aforementioned test beds have only been developed to grant access to internal users. Although Werner-Allen *et al* (2005) proposed granting access externally via the use of an Open Source web interface that operates on the purpose of job scheduling.

### **Provisioning and Optimization of Wireless Mesh:**

#### **Wireless Mesh Networks:**

A Wireless Mesh Network refers to a range of wireless technologies that route traffic between nodes in collaborative topology. By using this network architecture redundancy can be easily added to the network so that single node failure will not result in a network breakdown. Wireless Mesh Networks can be either of an ad-hoc design, where each node collaborates with the others although no pre-defined structural infrastructure exists, or they can exist as part of a larger infrastructure in order to allow for connections to other existing networks or to the internet. A number of characteristics that are associated with Wireless Mesh Networks are the ability to self-form, self-heal in case of a node or multi-node failure and self-organization. Mobile nodes, similar to the ones used in the AIT EcoMesh deployment support multiple types of network traffic (Data, Voice, Video), although power constraints depend on the type of mesh node used (Akyildiz, I.F. and X. Wang, 2009; Madden, S. and P. Levis, 2008; Ishmael, J., *et al.*, 2008). Previous research into creating a campus wide wireless mesh network undertaken by Brito *et al.*, (2009) developed a three node wireless mesh network that operated with two mobile nodes. The positioning of these nodes was found by using Motorola's mesh planner software.

#### **3.2: Wireless Sensor Networks:**

A wireless sensor network (WSN) can be defined as a collection of nodes that updates a database structure via a gateway using ZIGBEE 802.15.4. The information collected via these sensor nodes can then be analysed by the end user and presented using a graphing program (Moteview), or just raw data from each sensor made available..

Sensor networks are now evolving to include actuators to instigate some functions such as moisture, temperature and fertiliser management in agriculture.

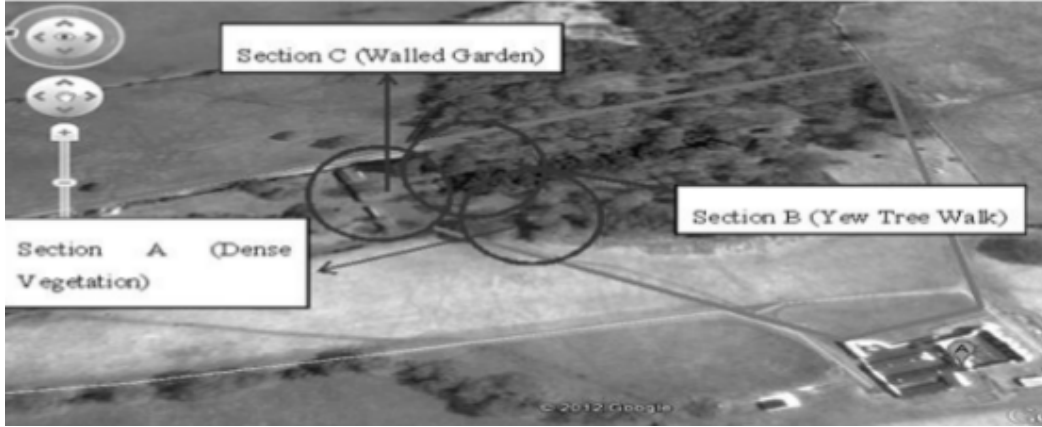
A sensor network generally conveys data over a relatively small distance by the use of energy efficient nodes. The communication between nodes is governed by using the closest node thus allowing traffic to travel through the deployment to the gateway node. By using this approach it is possible to cover a large area with fewer nodes as they are able to piggy-back across the deployment in order to send information back.

The main sensor types used in the EcoMesh deployment are Iris Motes running the current TinyOS 2.X stable release. This allows for the deployment to operate with the current software applications from Crossbow/Memisc and initial experiments to use other software language variants such as Ceu instead of the standard nesC programming language have been conducted. Ceu shall be discussed in further detail later in this paper.

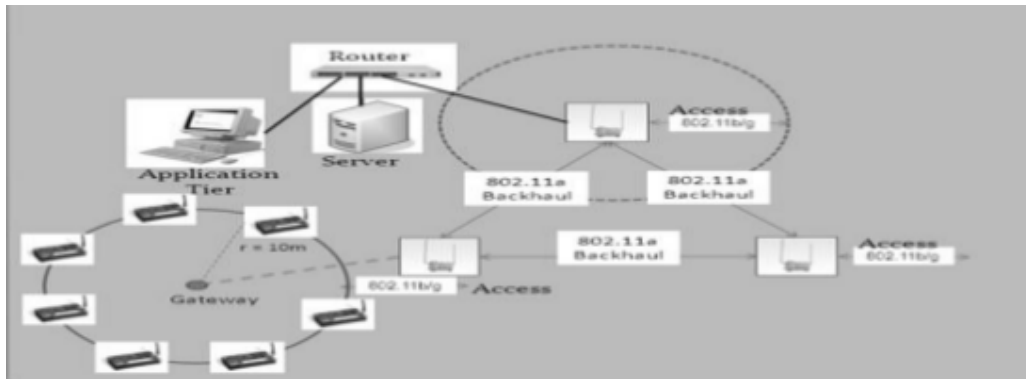
#### **3.2 Installation and Provisioning of the WLAN:**

The Wireless Mesh Network that operates inside the EcoMesh was deployed and tested over a ten month

period. The main factors that had to be taken into account was the signal strength and throughput statistics that would ensure the mesh network would be operating correctly. For a greater area of coverage the two mobile access points were extended to the canopy roof to allow for a greater line of sight. By doing this the problems associated with overcoming scattering, signal loss, interference and reflection through dense vegetation are all factors which need to be considered.. The figure below shows the placement of each access point in the topology (Fig 1). A general hardware topology view of the test-bed network can be seen in Figure 2.

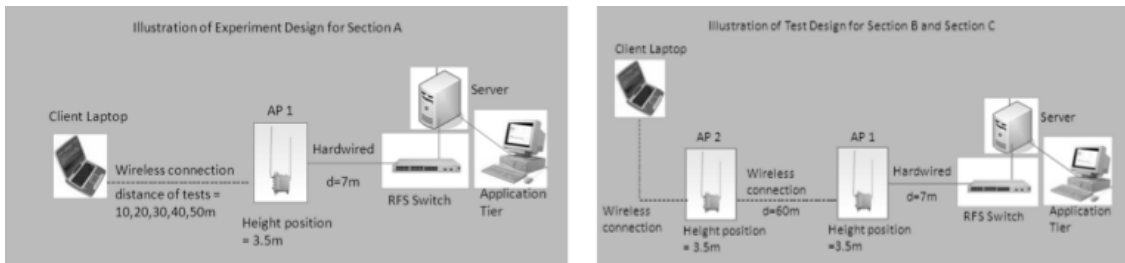


**Fig. 1:** This shows the topology placement for the EcoMesh Wireless Mesh Network.



**Fig. 2:** This is the hardware topology of the entire EcoMesh network.

When this topology was installed and deployed correctly a testing phase was carried out using IxChariot network testing software. These tests focused around TCP and UDP throughput statistics operating over one and two hops of the network. The deployment's mesh backhaul architecture operates on 802.11a 5GHz technology. A range of traffic types were used from a distance of ten meters, then moving back until the connection was eventually lost. The topology used when doing these tests can be seen below in Figure 3 (a) and Figure 3 (b).



**Fig. 3:** (a) and (b) illustrates the experimental designs for testing over one hop and two hops

By using the R&S FSH 4/8 spectrum analyser the RSSI values were recorded. This allowed for a bandwidth calculation to be formed as well as the signal power in dBm's using the Root Mean Square (RMS) as well as the Signal to Noise Ratio (SNR). By doing this the experimental results could thus be compared to the theoretical results.

In theory the throughput of a network can be determined by using the following equation.

$$\text{Throughput} = \frac{1}{n}$$

Where n is the number of hops that the network is utilizing when tested.

The standard deviation was obtained from some of the throughput results in to evaluate the impact of traffic type, environmental factors, distance and operational loading in the network. Typically TCP type traffic is referred to as 'Bursty' traffic which can simply be described as traffic variability in terms of the amount of traffic generated in a given time frame. Studies has illustrated that even when aggregated applications are applied traffic smooths out as more application traffic is multiplexed (Tinnakornsrisuphap, P., W. Feng and I. Philp, 2000).

**Table 4:** (a) and 4(b) Throughput, Response Time and Standard Deviation from the EcoMesh Network

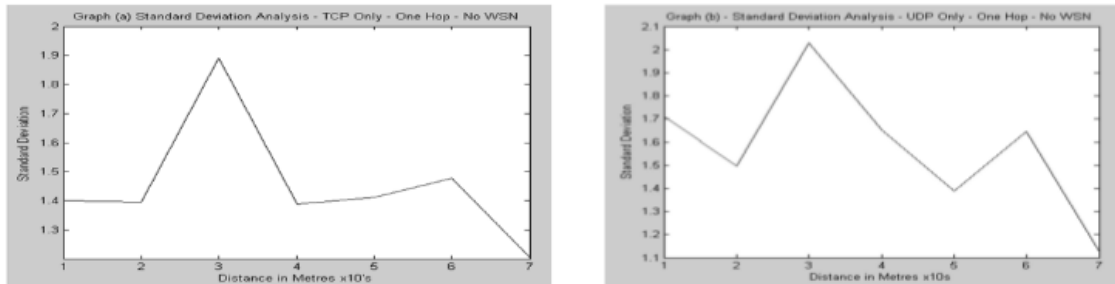
Exp no./ Test No/ Summer	Network Utilization At 2.472 GHz – channel 13	No. Of hops And height of Access Point	Distance From Client in meters	Nic Card Data Rate Mbits/ sec	Traffic Type	Average Throughput	Average Response Time in seconds	RSSI Value (RMS) In dBms
01	WMN only	1 Hop-3.5 m (H)	10	48	TCP only	6.036	0.132	-53.7
02	WMN only	1 Hop-3.5 m (H)	20	48	TCP only	5.621	0.142	-50.8
03	WMN only	1 Hop-3.5 m (H)	30	5.56	TCP only	2.008	0.398	-----
04	WMN only	1 Hop-3.5 m (H)	40	5.5 – 5.5	TCP Only	0.530	1.508	-51.6
05	WMN only	1 Hop-3.5 m (H)	50	5.5 – 5.5	TCP only	1.782	0.448	-44.2

Exp no./ Test No/ Winter	Network Utilization At 2.472 GHz – channel 1	No. Of hops And height of Access Point	Distance From Client in meters	Nic Card Data Rate Mbits/ sec	Traffic Type	Average Throughput In Mbps	Average Response Time in seconds	STD of each expert. (Avg 1000 samples)
6	WMN only	2 Hops-3.5 m (H)	10	54	TCP only	5.695	0.140	1.7302
7	WMN only	2 Hops-3.5 m (H)	20	48	TCP only	5.729	0.139	1.5994
8	WMN only	2 Hops-3.5 m (H)	30	36	TCP only	5.604	0.142	1.6587
9	WMN only	2 Hops-3.5 m (H)	40	24	TCP only	5.177	0.154	1.5368
10	WMN only	2 Hops-3.5 m (H)	50	5.5	TCP only	4.203	0.190	1.2962

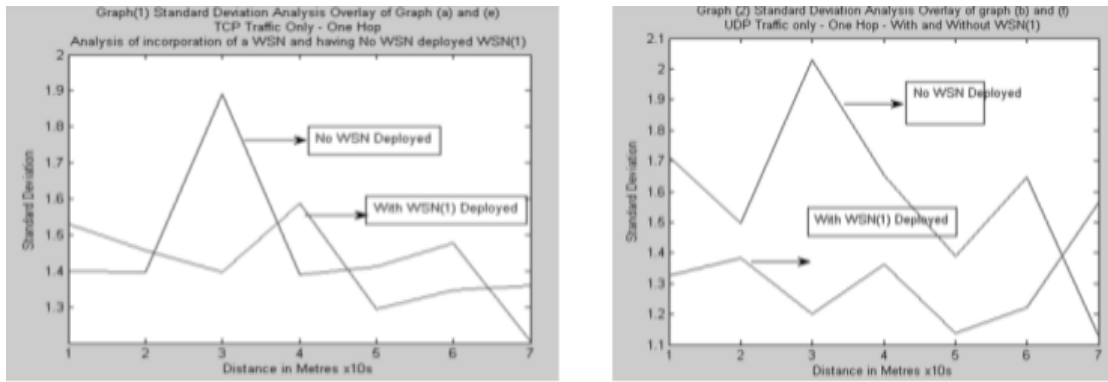
From the tests carried out on the live network the following results were obtained. The impact of foliage density was noticed in the winter months to those found during the summer. The variance in these results can be put down to the following factors: Vegetation density during the summer months being much higher than that of the winter. Weather conditions can both hamper or improve signal strength in terms of scattering and wave propagation. The tables below Table 4(a) and 4(b) show this variance in results depending on the time of year.

These tests were carried out over a period of 2 minutes with the client antenna positioned 0.5-1.0 meters from ground level with AP antenna at 5 meters from the ground. In order to validate these tests for a two minute duration a single test was ran for over a one hour period and returned a value of 5.502Mbps which was comparable to the 2 minute test of 5.987Mbps, giving both tests a deviation of 0.458Mbps. The performance results for both TCP traffic and UDP traffic gained from the IxChariot software can be seen in Figure 5 (a) and (b). Both these tests were performed over a single hop starting at a 10 meter distance before progressing to 70 meters.



**Fig. 5(a):** TCP over one hop in winter, Figure 5 (b): UDP over one hop in winter.

After these tests a Wireless Sensor Network was deployed using the MIB600 interface board that allows for traffic coming from the WSN to be sent through the Wireless Mesh back-haul network. The effects that the WSN deployment had on the throughput calculations can be seen in Figure 6 (a) and Figure 6 (b) below.



**Fig. 6 (a):** This shows the TCP throughput statistics for one hop on the network.

**Fig. 6 (b):** This is the UDP throughput for a single hop through the network

Again during the winter period both when a WSN is deployed and also without the deployment for comparison. This throughput test was again calculated by starting at 10 meters and working back to 70 meters until the connection to the network was severed. This statistic was taken during the same period as the TCP throughput shown in Figure 6 (a) and conforming to the same distance parameters.

**3.3 Discussion of Results from Wireless Mesh Testing:**

Based on the results collected from testing the throughput of the wireless mesh network under different parameters and different traffic types the following conclusions can be made. When testing for the following parameters: Throughput (TCP and UDP), Signal to Noise Ratio (SNR) and Received Signal Strength Indicator (RSSI); it can be seen that there is a difference in results depending on the time of year. This is due to the dense vegetation surrounding the canopy in summer time as opposed to Winter. In the Winter period there is an increase of 65% in the overall network performance. These results can be viewed in Figure 4 (a) and 4 (b) above.

**4 Current Wireless Sensor Network Middleware Challenges:**

One of the methods currently used for generating and viewing information from wireless sensor networks is to include a third party application such as those supplied by Crossbow and Memsic.

Moteview is an application that uses the Cygwin terminal to retrieve information from sensors that are TinyOS based. Moteview provides a graphical front end that allows users to view and monitor local and remote based sensors. The information gained from the wireless sensor network is then converted from standard hexadecimal data into readable data and then can be stored using a PostgreSQL database. Each packet containing the hexadecimal data is converted depending on the users specifications for the sensor and interface board that is being used. This information can then be viewed through graphs and tables. The wireless sensor network topology is generated by using the ParentID of each sensor which allows for a topology to be designed based on their location to the gateway.

Legacy middleware to view and model data also exists through the use of TinyOS based tools such as TinyDB and the use of the Java based serial forwarder application. Although these tools allow for information to be generated from the sensor network they don't provide a generally good end-user experience when it comes to viewing data and storing it. TinyDB has since been discontinued on the TinyOS toolchain and currently only works with sensors that use the TinyOS version one firmware. No updated version of TinyDB is expected as a complete re-write would be required to ensure that it functions correctly.

Some problems that exist using these methods for viewing the information sent and received by a wireless sensor network is that data can only be viewed from one single source at a time. For example if two wireless sensor networks are functioning simultaneously Moteview has the ability to see that both networks are present but can only read and store data from the selected network. This can be very troublesome if you wish to monitor many different networks from a single location as it requires the use of multiple hosts all running either the Moteview or Ekoview software platforms.

Another reason that Moteview causes problems for end-users is the fact that it is unsupported on current operating systems. It currently doesn't work on Microsoft's Windows 8, 7 or Vista; There is currently no support planned for Mac OS X or any other UNIX based operating system.

**4.1 Web services and Platform development:**

In order to solve this problem a web service (Kwang-il Hwang, *et al.*, 2003) is proposed for viewing and

monitoring wireless sensor networks.. This front-end service will exist on the cloud and will allow data from multiple regions to be viewed and displayed visually through the Internet. This system will exist on top of the LAMP (Linux, Apache, MySQL, PHP/Python/Perl) software stack. This will allow for a front end to be displayed using the Apache web server, all data shall be stored using MySQL and all programming shall be done using a mixture of PHP and Python. The main reason that Linux was selected as the default development operating system was for the following reasons.

- It provides a cost effective approach when developing such a system
- The finished system can then be ported easily to ARM based single board computers (Rasberry PI, Pandaboard). This will further lower the end-users costs when implementing such a system.

In order to view information from two sensor networks using the proposed approach multiple TCP sockets can be opened using PHP or Python, the raw information coming from the wireless sensor network can then be converted into hexadecimal so that it can then be converted into engineering units using the formulae provided with TinyOS based sensors. For users that require the use of multiple serial interface boards i.e. MIB520 and MIB510 a second script shall be allocated for performing this function. This program will open multiple connections to the various wireless sensor networks and convert the information in a similar fashion to the TCP based function. This will allow for a seamless connection between multiple TinyOS based sensors. The next step is to create a functional database that can interact and store the data coming from multiple locations based on the users specifications. A model of this database can be seen in Figure 7 below.

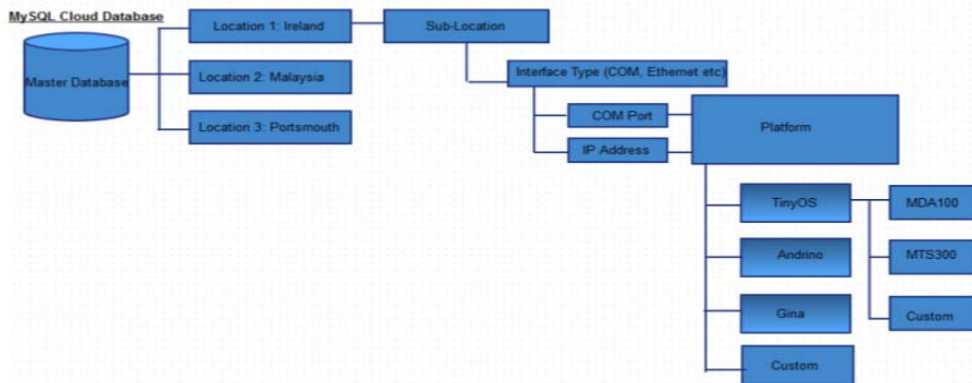


Fig. 7: This is the proposed database structure of the new web based platform.

**Sensor Development:**

While developing a new web service driven approach to monitoring and administrating wireless sensor networks the software engineering techniques used when it comes to developing applications to run on WSN's has too also be taken into account. One modern programming language that is used in wireless sensor network programming is nesC. NesC has been designed as an alternative to C and other embedded languages such C++ .. NesC is said to take a modular approach to programming as it requires the use of interfaces, components and wiring as shown in Figure 8. This approach means that there are more than one file that needs to be compiled at run time. This has been said to be a good approach as it allows for source code of a nesC application to be easily understood..

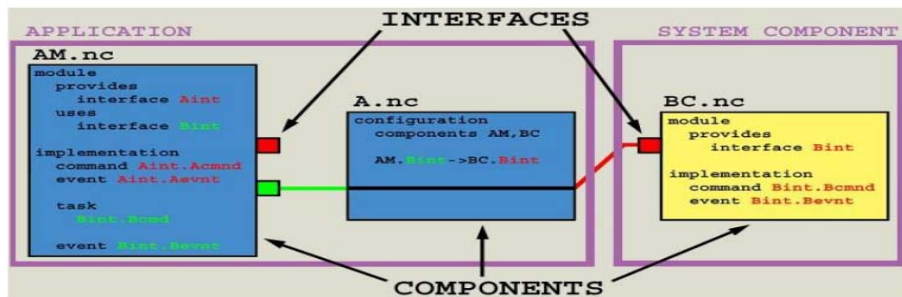
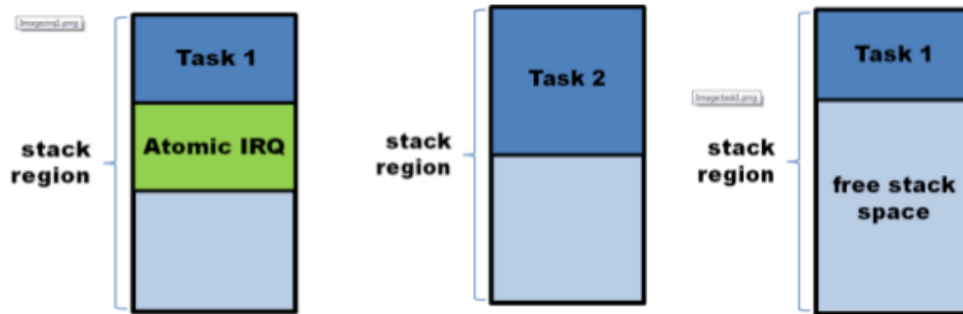


Fig. 8: This is the wiring process that is used extensively in nesC development.

One major problem that has been noted with nesC development for wireless sensor networks is that to newer users the code footprint required to complete a simple task can be rather long and not efficient.. This is

due to the single-stack design that it operates on. By developing applications for this single stack design the programmer has no knowledge of how much memory is being taken up on each sensor node, thus making designing a program to fall within these memory constraints more difficult. The single stack design can be seen in Figure 9.



**Fig. 9:** This shows the single stack design and how tasks are passed through in a queued method.

An alternative to the nesC language may be to use a scripting language. One reason to use a scripting language for a wireless sensor network deployment is because unlike virtual machine code, a script can be interpreted by the nodes without the need for a compiler. Although this interpreter can in some cases be too large for the sensor to handle. However many lightweight forms of scripting languages have been implemented by many different vendors in order to solve this issue.

Céu is a programming language that was developed to be used with embedded systems and its main objective is to offer a higher level and safer alternative to C or nesC. Céu contains bindings for TinyOS and Arduino platforms, allowing it to be used as a scripting language for both. One additional aspect of the Céu approach to programming sensor networks is that the code size becomes more manageable and easier to comprehend to new users. This can be seen by the blink application, which is used to blink the LED's (Light Emitting Diodes) on a mote. Written through Céu for TinyOS based wireless sensor networks it reduces the code size significantly when compared to the nesC alternative and only requires one file to be developed in order to run the blink program on a mote.

From this we can see that the use of scripting languages is applicable to Wireless Sensor Network application engineering as it allows the engineer to spend less time designing and writing the application code. It also allows for a more simplistic approach to be taken when developing a program. This can also be useful when teaching embedded programming concepts in a classroom environment.

On a UNIX based system the same command syntax can be used to send a Céu script as a nesC script. This allows for an easy transition to be made after Céu has been installed along with the users TinyOS configuration. Another reason that Céu can be used as a programming language alternative for wireless sensor networks is that it can be easily used with simulation. This is vital in wireless sensor networks as it allows for the following:

- Not having to deploy the application on motes every time to ensure that it is working correctly.
- It allows for easier debugging of the application
- Can be used to test large scale networks without a large deployment of motes being needed.

**Discussion:**

Firstly this paper has discussed the unique challenges faced with deploying a heterogeneous wireless mesh network in a harsh external environment such as those proposed for many Agri-tech industries. With important detail being taken to the provisioning and distribution of both the mesh back-bone infrastructure and the wireless sensor network the overall success of the EcoMesh design can be seen in the various tests carried out over a ten month period. Whilst there is variance between the different seasons (Winter and Summer) the overall network was more than able to keep working despite the changing weather conditions. When deploying such a network these seasonal factors must be taken into account in order to ensure that the network is capable of sustaining real-time services over a topology with parameters fluctuating continuously. The results of these tests should benefit researchers and industry developing intelligent management systems for such networks.

Whilst the hardware network has developed further and is now at a point that the data coming through the mesh network via the MIB600 interface board design is viewable there are still some problems with the software and middleware designs of the WSN.

As the applications of wireless sensor networks continue to grow so does the underlying programming languages that are needed to improve their functionality. As nesC has been slow to develop from the original

release version it can be said that the once “stop gap” language has met its limits when it comes to programming for wireless sensor networks. Languages such as Ceu allows for a scripting approach to be taken when developing large applications in a fraction of the time that it takes using standard sensor languages such as C and nesC. As Ceu is currently still under development it does show great promise when it comes to overcoming the problems associated with nesC whilst allowing students to get a grasp of embedded languages.

Using a scripting language such as Ceu allows for less time to be taken up by tasks associated with nesC such as wiring. As Ceu continues to grow and can be developed on multiple embedded platforms such as Arduino and as its user base is constantly increasing further development from more sources will soon make it a viable alternative for embedded application development. When designing an application with Ceu the ability to test it for a large scale simulated wireless sensor network is both efficient and user friendly. This allows for a more efficient approach to be taken when testing an application using Ceu as it doesn't require the user to download and configure the application for a sensor network.

As the scope of the programming languages used in conjunction with wireless sensor networks grows so too must the application used for viewing and visualizing the wireless sensor network. Whilst tools such as Moteview and EkoView provide a somewhat decent graphical representation for single wireless sensor networks, an OpenSource alternative to these standards would provide users the ability to view and store data from multiple sensor networks running across different platforms. Having a single open system would allow for third party development to take place and for people to change the software to whatever they wanted.

As this system currently under development will be initially Linux based it will allow for it to be installed on a wide variety of platforms and will run perfectly on older hardware. By using a web based system for monitoring, viewing and storing information generated from multiple wireless sensor networks it will be constantly accessible to a wide range of users, researchers and technology hobbyists. One application that has been proposed when the finished system is fully developed would be to monitor climate change in multiple locations, starting with Ireland, the United Kingdom and Malaysia. As the sensor network would be taking readings based on the temperature, humidity and light penetration over a period of time (+5 Years) a viable relationship between climate change and these factors would be determined. This would allow for studies to be undertaken into the effect of climate change with multiple entry points on the globe.

#### ***Future Work:***

On-going testing will be carried out on the EcoMesh and further analysis of the coverage and performance of the Crossbow wireless sensors. The addition of an Industrial standard environmental monitoring camera with infra-red capabilities to the test-bed is near completion. Testing and analysis under network performance parameters such as those discussed in this research will commence upon full installation and re-evaluated during the summer months in order to provide a yearly analysis. The gathering of empirical experimental performance measurements will form the bases of future work in relation to an Admission Control algorithm to quantify parameters for a guaranteed QoS delivery to the end user with further research developed using an OPNET™ simulation model.

Also the proposed software platform will continue to be further developed. So that it can monitor the entire EcoMesh based sensor network and store all the environmental data coming from it that should provide an invaluable source of information to be used by researchers using techniques such as data mining. More programming languages similar to Ceu that may be suitable for the development of sensor networks will also be investigated as it will allow for a wider range of software research and application development to be carried out.

#### ***Discussion:***

Firstly this paper has discussed the unique challenges faced with deploying a heterogeneous wireless mesh network in a harsh external environment such as those proposed for many Agri-tech industries. With important detail being taken to the provisioning and distribution of both the mesh back-bone infrastructure and the wireless sensor network the overall success of the EcoMesh design can be seen in the various tests carried out over a ten month period. Whilst there is variance between the different seasons (Winter and Summer) the overall network was more than able to keep working despite the changing weather conditions. When deploying such a network these seasonal factors must be taken into account in order to ensure that the network is capable of sustaining a working topology.

Whilst the hardware network has developed further and is now at a point that the data coming through the mesh network via the MIB600 interface board design is viewable there are still some problems with the software and middleware designs of the WSN.

As the applications of wireless sensor networks continue to grow so too does the underlying programming languages that are needed to improve their functionality. As nesC has been slow to develop from the original release version it can be said that the once “stop gap” language has met its limits when it comes to programming for wireless sensor networks. Languages such as Ceu allows for a scripting approach to be taken when

developing large applications in a fraction of the time that it takes using standard sensor languages such as C and nesC. As Ceu is currently still under development it does show great promise when it comes to overcoming the problems associated with nesC whilst allowing students and beginners to get a grasp of embedded languages.

Using a scripting language such as Ceu allows for less time to be taken up by tasks associated with nesC such as wiring. As Ceu continues to grow and can be developed on multiple embedded platforms such as Arduino and as its user base is constantly increasing further development from more sources will soon make it a viable alternative for embedded application development. When designing an application with Ceu the ability to test it for a large scale simulated wireless sensor network is both efficient and user friendly. This allows for a more efficient approach to be taken when testing an application using Ceu as it doesn't require the user to download and configure the application for a sensor network.

As the scope of the programming languages used in conjunction with wireless sensor networks grows so too must the application used for viewing and visualizing the wireless sensor network. Whilst tools such as Moteview and EkoView provide a somewhat decent graphical representation for single wireless sensor networks, an OpenSource alternative to these standards would provide users the ability to view and store data from multiple sensor networks running across different platforms. Having a single open system would allow for third party development to take place and for people to change the software to whatever they wanted.

As this currently under development system will be initially Linux based it will allow for it to be installed on a wide variety of platforms and will run perfectly on older hardware. By using a web based system for monitoring, viewing and storing information generated from multiple wireless sensor networks it will be constantly accessible to a wide range of users, researchers and technology hobbyists. One application that has been proposed when the finished system is fully developed would be to monitor climate change in multiple locations, starting with Ireland, the United Kingdom and Malaysia. As the sensor network would be taking readings based on the temperature, humidity and light penetration over a period of time (+5 Years) a viable relationship between climate change and these factors would be determined. This would allow for studies to be undertaken into the effect of climate change with multiple entry points on the globe.

#### **Future Work:**

On-going testing will be carried out on the ECOMESH and further analysis of the coverage and performance of the Crossbow wireless sensors. The addition of an Industrial standard environmental monitoring camera with infra-red capabilities to the test-bed is near completion. Testing and analysis under network performance parameters such as those discussed in this research will commence upon full installation and re-evaluated during the summer months in order to provide a yearly analysis. The gathering of empirical experimental performance measurements will form the bases of future work in relation to an Admission Control algorithm to quantify parameters for a guaranteed QoS delivery to the end user with further research developed using an OPNETTM simulation model.

Also the proposed software platform will continue to be further developed. So that it can monitor the entire EcoMesh based sensor network and store all information coming from it in a hope that it will be useful in future research. More programming languages similar to Ceu that may be suitable for the development of sensor networks will also be investigated as it will allow for a wider range of software research and application development to be carried out.

## **REFERENCES**

- “Wireless Sensor Networks, 2011.” Crossbow\_Technology, [Online]. Available:  
Akyildiz, I.F. and X. Wang, 2009. Wireless Mesh Networks, A John Wiley and Sons Ltd.  
Blumenthal, J., M. Handy, F. Golatowski, M. Haase, D. Timmermann, 2003. Wireless sensor networks - new challenges in software engineering. Emerging Technologies and Factory Automation, Proceedings. ETFA '03. IEEE Conference. 1(-): 551-556.  
Brito, E., R. Stewart, J. Allen and M. Hassan, 2009. “Performance Optimisation of the AIT Campus Wireless Mesh Network Deployment,” in 9th International Conference on Information Technology and Telecommunication, Dublin, Ireland.  
Hassan, M., R. Stewart, L. Murphy and S. Kennedy, 2011. “Analysis of GTS allocation mechanism in IEEE802.15.4 WSN for Real-Time Applications,” in International Conference on Telecommunication Systems, Modelling and Analysis (ICTSM), Prague, Czech Republic.  
<http://www.xbow.com/Home/wHomePage.aspx>.  
Ishmael, J., S. Bury, D. Pezaros and N. Race, 2008. “Deploying Rural Community Wireless Mesh Networks,” IEEE Internet Computing, 12(4): 22-29.  
Karl, H. and A. Willig, 2007. Protocols and Architectures for Wireless Sensor Networks, John Wiley & Sons, Ltd,

Kwang-il Hwang, Jeongsik In; NhoKyung Park ; Doo-Seop Eom. 2003. A design and implementation of wireless sensor gateway for efficient querying and managing through World Wide Web. Consumer Electronics, IEEE Transactions on. 49 issue 4(4): 1090-1097.

Madden, S. and P. Levis, 2008. "Mesh Networking - Research and Technology for Multihop Wireless Networks," IEEE Internet Computing, 12(4): 9-11.

Spinar, R., P. Muthukumaran, R. De Paz, D. Pesch, W. Song, S.A. Chaudhry, C.J. Sreenan, E. Jafer, B. O'Flynn, J. O'Donnell, A. Costa and M. Keane, 2009. "Demo Abstract: Efficient Building Management with IP-based Wireless Sensor Network," in in EWSN 2009 - 6th European Conference on Wireless Sensor Networks, Cork, Ireland.

Tachtazis, C., 2010. "WiSAR Lab Testbeds," [Online]. Available: <http://www.wisar.org>. [Accessed 7th December 2011].

Tinnakornsrisuphap, P., W. Feng and I. Philp, 2000. "On the Burstiness of the TCP Congestion Control Mechanism in Distributed Computing System," in The 20th International Conference on Distributed Computing Systems (ICDCS 2000), Taipei, Taiwan.

Werner-Allen, G., P. Swieskowski and M. Welsh, 2005. "MoteLab: A Wireless Sensor Network Testbed," in 4th International Conference on Information Processing in Sensor Networks (IPSN'05), Los Angeles, USA.