Abstract— There are a wide number of applications in which there are few, if any, in-place measurements taken. Presently, these applications rely on observed measurements or examination to discover anomalous situations. This requires experience and face-to-face negotiation to realize the abnormal situations and take action.

This paper describes how we developed a simple measurement and control device and examined its use in implementing several control strategies in various wireless network topologies. This initial development will be used to observe the architecture requirements for such a device and investigate the issues that arise. What is learned from this and any concerns that appear during testing may be used to develop a more complete solution at a later date.

The final section of this paper describes some of the elements industrial devices like those described here may have.

Index Terms—mesh networks, wireless control, measurement

1 INTRODUCTION

Measurement and control devices can be used to simplify many real-world applications. Current measurement and control mechanisms require infrastructure support: power, communication, configuration, and support. This support infrastructure is usually unavailable or costly to use.

The revolution in low-cost low-power wireless communication devices can be used to provide very low cost measurement and control devices called Limpets. A Limpet is a small intelligent device that is portable or attached non-evasively to an object. These devices could provide information on a process, certain equipment, the environment, or the health of the target of interest. More importantly, Limpets could be used in an ad-hoc collection to provide more complex measurements and calculations in order to provide some level of feedback and control (i.e. valves, gates, lights, etc.).

Brandon Hieb is a Software Engineer with Emerson Process Management, Austin, TX 78759. Email: Brandon.Hieb@EmersonProcess.com
some action to occur? Is it possible for race conditions to develop across the mesh? How much information can be sent in a single request? Can commands be synchronized? What are the practical limitations? In order to be practical, can the price of a single node cost less than $50?

The objective of this report was to create a small network of Limpets that could execute Boolean ladder logic instructions. This network would work together to perform common monitoring and control tasks. Basic observations about the dynamics of a Limpet system were made and analyzed for use in future Limpet designs, specifically to the questions asked in the previous paragraph.

2 LIMPETS

A Limpet is a small device that transforms input states into output states using a control strategy. The inputs can come from any number of sources: analog or digital values, or communication protocols like Ethernet or MODBUS. The control strategy uses these values along with previous values, called feedback, to determine the state of the outputs. The outputs can be used to activate or deactivate some physical value.

Even the simplest Limpet must have four properties: inputs, control, feedback, and outputs. The simplest inputs are discreet Boolean values in which there are only two possible states, on or off. The simplest output is also a Boolean value. Because the inputs and outputs are Boolean values, it is reasonable to assume that Boolean values can be used for feedback. In a Limpet network, each Limpet must be able to read values from other Limpets.

Each Limpet can be configured to run its own control strategy. A Limpet network is a true distributed control network.

The control strategy must be configurable and customizable for each Limpet. There must be a system for users to design a control strategy and download it into a Limpet. The control strategy then executes on a periodic schedule that the user chooses.

2.1 Boolean Ladder Logic

One of the simplest forms of a control algorithm is Boolean ladder logic [2]. Ladder logic (sometimes called Boolean Mnemonics) was originally used in the electrical industry where relays are used to control equipment.

Boolean Mnemonic instructions revolve around the use of “rungs”. Each rung represents a sequence of execution. Rungs can be branched or combined and each rung ends in an output. Each rung contains inputs, called contacts (CON), that act as on/off switches. When a switch is activated (on), the circuit is closed and the output (COIL) is energized (on).

A simple ladder logic algorithm looks like this in graphical format:

Figure 2. Simple Ladder Logic

The Boolean Mnemonic for this graphical ladder would be:

```
STR CON0
OUT COIL0
END
```

When CON0 is “on”, the ladder run is energized and OUT is “on”. Each Ladder algorithm has an END statement. Any instructions (or rungs) placed after the END statement will be ignored.

Contacts can be placed in series to have the same effect of an AND statement:

```
STR CON0
AND CON1
OUT COIL0
END
```

In this diagram, both CON1 and CON2 will have to be “on” in order for COIL0 to be “on”.

Commands can be branched to have an output in the middle of an instruction:

Figure 3. Contacts in Series

```
STR CON0
AND CON1
OUT COIL0
END
```

Figure 4. Mid-Line Outputs

Rungs can be branched to allow OR commands to be formed:
To allow users to create Boolean ladder logic algorithms and program Limpet devices with this logic, a Windows-based application called LogicParser was created. LogicParser takes Boolean ladder logic instructions in the form of “Boolean Mnemonics” to create a control strategy, which it presents in the form of a C programming language file. This C file was then included in the build for the Limpet firmware.

3 EXPERIMENTAL DESIGN

Three Limpets and two repeater devices were available for the experiment. Three different motor-control strategies of various complexities were used in each of two mesh topologies. The first topology simply involved two different Limpets communicating over a direct link. The second topology made use of three repeaters to observe the behavior of the mesh network when repeaters failed.

The contacts from multiple limpets were used in a single control strategy by communicating their values to nodes which hosted the final coils. Three different control strategies for a device that can raise or lower an object were to control motor up functionality, motor down functionality, and motor on/off logic.

The motor direction was set to UP if:
• the handheld (manual operation) switch is closed and the handheld up/down switch is closed, or
• the AUTO (automatic operation) switch is closed and the motor up switch is closed, or
• the remote MANUAL mode switch is closed and the motor up switch is closed, or
• the power up mode switch is closed and the motor up switch is closed.

This results in the following Boolean Mnemonic algorithm:

```
STR CON4
AND CON5
STR CON6
AND CON7
ORSTR
STR REMCON0
AND CON7
```

The motor direction is set to DOWN if:
• The handheld (manual operation) switch is closed and the handheld up/down switch, which is normally closed, is set to UP, or
• The AUTO (automatic operation) switch is closed and the motor down switch is closed, or
• The remote MANUAL mode switch is closed and the motor down switch is closed, and
• The device is at the top of its range. This is observed by the limit up switched being closed, and
• The Motor On/Off switch must be on.

This results in the following algorithm:

```
STR CON3
STR CON5
NOTSTR
ANDSTR
STR CON8
AND CON9
ORSTR
STR REMCON8
AND CON9
ORSTR
AND CON12
AND CON13
OUT COIL3
END
```

Two different Limpets were placed at a distance from each other outside their expected range. Two repeater nodes were placed in between to relay data. Control was interrupt for several seconds when one repeater was powered down, forcing the mesh routing algorithms to switch to the other functional repeater.

The network latency times on a system with two repeaters were less than one half second. In the application for which Limpets were first envisioned, these times were acceptable. These delays may be more than wired protocols, but because Limpets are a true distributed network, most high-speed control should be performed at the node itself.

4 FUTURE WORK

This study covered the initial investigation of wireless monitoring and control devices, but there is much more to learn and many improvements that can be made before Limpets can become a commercially viable product. Many simplifications were made to this initial system. There is still work to be done and more to learn from more complex and useful Limpets.
4.1 Low Power Limpets

The domain of Limpets would include several instances where the node must be able to run with a low-power. It would be advantageous to be able to run a Limpet off battery or solar power therefore eliminating the need for any wiring. The current prototype requires approximately 7 Watts of power – mostly because off-the-shelf hardware was used to speed up development. The power requirements will probably be more than that of so-called “Smart Dust Motes” [3] because they are designed as sensor-only platforms and do not have the need to drive voltages to output devices.

4.2 Large Network Testing

It would be desirable to determine the amount of network loading and the reconfiguration times of a network of hundreds of Limpets. In addition, the effect of Limpet density on the fastest possible remote coil update times on a finite bandwidth network will need to be examined. For example, how many Limpets can be placed within a small area before the bandwidth limitations of the medium become too great to have the update rates required for a system?

4.3 Downloading Control Algorithms

In a real world environment, it is not practical to program remote nodes at the node’s physical location using a physical interface (i.e. USB or serial). A much more practical method is to download the control strategy to the Limpet using the device’s radio interface. This presents several issues ranging from security to reliability. How can the image be verified to be from a reputable source? What happens if the image is only partially downloaded before the transfer is interrupted?

4.4 Exception-Driven Data Reporting

Because wireless channels are a shared medium there are hard limits on the amount of bandwidth each channel can carry. In addition, each channel may be sharing bandwidth with a variety of third-party devices that can further reduce the amount of data that can be transferred. Because of this, Limpet applications should avoid data polling and push data whenever possible. A Limpet requiring data from another node would request that data be sent. Losing data messages sent from the source to the destination is disastrous as subsequent states may depend on the previous state. If a data message were to be lost, the previous state may be invalid. A combination of acknowledgement messages and heartbeat messages can be used to provide more reliable transmission, but their use must be balanced with the need to conserve scarce bandwidth resources.

4.5 Write Safety

Another data service issue that needs to be addressed is write safety. It would be advantageous for the writing node to have knowledge of the success or failure of a write to another node. The value on the destination node would not be allowed to change without a successful write. The success of the write could depend on many factors: communication disruptions, security settings, device states, etc.

4.6 Data Balancing on the Mesh

It would be desirable to take bandwidth loading into account in the mesh routing algorithms. If a node or frequency became overloaded, it should be possible to route one or more connections over a different path that is not heavily loaded.

4.7 Security

There are two phases of security for ad-hoc networks: data security and defense against outside attacks.

An important part of any future Limpet will be the ability to secure wireless commands. There are two levels of security that are possible: identity verification and encryption.

Verifying the identity of a sender is important for process security. Each node could use keyed MD5 (Message Digest 5) to guarantee their identity [4]. Each Limpet could be preconfigured to share a secret key. The sender then runs MD5 over the concatenation of the message and the secret key. If the message is denoted m and the key denoted as k, the sending node actually sends \( m + MD5(m+k) \). The receiver applies MD5 to the concatenation message body and the secret key and compares the result to the original secret key. If the results match, the sender’s identity is verified. The utility of using keyed MD5 instead of RSA is found in improved performance. MD5, especially when implemented in hardware, can be many orders of magnitude faster than RSA.

Monitoring data over a wireless network is much easier than monitoring data over a wired network. The network analyzer (sniffer) does not even have to have physical access to a site in order to examine data. The best defense against network sniffing is to encrypt sensitive data that is sent over the network.

One way to do this is to configure a secret key that must be the same on each Limpet on the network. Some ways to configure this key include using a physical connection such as a serial port, Ethernet, or USB or simply using hardware jumpers or switches on the device itself. Using a simple three-way handshake authentication protocol, each node can authenticate itself. They each agree on a random number to be a temporary secret key (the session key) to be used to encrypt data. Performance-intensive RSA algorithms may be used to perform the initial handshake and agree on a session key. All subsequent transmissions may be encrypted using the session key with Data Encryption Standard (DES) or MD5.

Because wireless nodes use a communication medium accessible to devices outside of the physically secure area, a denial of service attack is a possibility [5]. A denial of service attack may come in many different layers: physical, link, network and routing, or transport. Each of these must be taken into account if Limpets are to be used in critical industrial applications.
Physical layer attacks use jamming to attack the network by sending useless signals therefore making other nodes unable to communicate. There are four possible defenses against this type of attack: sleeping during the attack, using spread spectrum or frequency hopping, or rerouting traffic away from the affected area.

Link layer attacks attempt to disrupt parts of the transmission, causing collisions. This will cause repeated retransmission which may lower battery life in the nodes. One solution is to make the MAC admission control rate limiting in order to ignore excessive requests.

Attacks using the network and routing layer may attempt to forward messages along wrong paths or provide wrong route information. Some solutions to this problem are only allowing specific nodes to exchange routing information or having each node monitoring the behavior of their neighbors.

An attacker may send many connection establishment requests to a particular node, making that node run out of resources. To defend against this, the number of connections can be limited, but this may keep authentic nodes from establishing a connection. Another way to defend against this is to force the requesting node to solve a puzzle before accepting a connection.

5 Conclusion

One of the major goals of this project was to gain familiarity with wireless control and monitoring. To accomplish this, simple monitoring and control devices called limpets were developed and used to run several real-time control algorithms involving different network topologies.

We discovered that network reconfiguration times on a small network updating at one second intervals were noticeable – on the order of several seconds. The prototypes used did not allow for the observation of update intervals of less than one half second.

In order to make Limpets a practical solution, the cost should be kept down as low as possible. The hardware used in this prototype has a cost of around $425. It would be desirable to see a product with a final cost of under $50.

There are many areas to expand on the initial observations performed in this project to move toward a device capable of standard use thought industry, but there is nothing to suggest there are any insurmountable hurdles to keep these types of devices from becoming commonplace in the future.

6 References