A System for Performance Measurement of PicoRadio Network Protocols

DeLynn Bettencourt

California State University, Fresno
Electrical Engineering

Summer Undergraduate Program in Engineering Research at Berkeley (SUPERB 2002)

Faculty Mentor:
Dr. Jan Rabaey

Berkeley Wireless Research Center Staff Mentor:
Fred Burghardt

College of Engineering
University of California, Berkeley

A System for Performance Measurement of PicoRadio Network Protocols

DeLynn Bettencourt
California State University, Fresno
DeLynn@ieee.org

Abstract

Knowledge of network performance should serve as a basis for critical design decisions in the Berkeley Wireless Research Center's PicoRadio project. Decisions on, for example, how much memory is needed or how much reconfigurability is needed in the individual PicoRadio network devices, PicoNodes, require knowledge on how the network will react when these parameters vary.

Measuring the performance of a wireless network by no means trivial because a traditional network monitoring device cannot simply be plugged into the network medium as it can be with, for example, an Ethernet network. Instead, requests for network protocol information are sent from a remote console and the PicoNodes gather the requested information and send it back to the console.

1. Introduction

This project is part of a larger research effort at the Berkeley Wireless Research Center (BWRC). The BWRC is a U.C. Berkeley graduate research facility located in downtown Berkeley. The focus of research at BWRC is wireless communication; one of the main research drivers is the PicoRadio project.

PicoRadio is a low power wireless ad-hoc network. The network consists of a large number of communicating PicoNodes, which are small, inexpensive, low power wireless devices [1]. Wiring a dense network of sensors is a costly endeavor— for example, wiring just a single sensor in a commercial building costs $200, on average [1]. PicoNodes are powered from environmental sources such as light and vibration and hence must consume very little power. Because of the requirement for low power**,** PicoRadio networks must be dense so that transmission ranges can be kept short. For reasons of cost, this high density reinforces the need for wireless installation.

The vision for the PicoRadio project is for it to become a truly ubiquitous technology. PicoNodes could be incorporated into building materials, allowing the PicoRadio network to form as a building is constructed [1].

The PicoRadio project is implemented in several phases. The first phase, the TestBed, is one that allows PicoRadio researchers to test their algorithms in the "real-world" in detail. Many aspects of the PicoNode design will be influenced by the knowledge of the environment in which the PicoNodes operate. For instance, information about the way the PicoRadio network operates can help researchers decide how much programmability to plan for and how much memory will be needed.

The fact that the PicoRadio network is wireless makes monitoring its performance a difficult task; a network monitoring device cannot be simply plugged into the network medium as it can in, for example, an Ethernet network. In order to find out information about the network traffic, one must design a system that remotely sends out a request to the PicoNodes for information and then receives and processes the responses from the PicoNodes.

1.1 Applications of PicoRadio Networks

Commercial Buildings

There are a number of applications for the PicoRadio network. One such application is managing the lighting and air conditioning in a large office building [1]. Typically only a few sensors are monitoring the office environment and the temperature, airflow, and lighting vary dramatically between sensors. A network of wireless sensors can allow for the micromanagement of the office environment and can insure that a comfortable work environment is maintained over the entire office building. The opportunity for energy savings is great—particularly in light of the California energy crisis. Even a minimal change in energy consumption can result in great savings; the power required to run large buildings in the United States is about 40% of the nations total [3]. It is predicted that the use of distributed monitoring and controlling systems for maintaining building environments will be able to reduce energy consumption, resulting in an estimated savings of $55 billion per year and 35 million metric tons of reduced carbon emissions in the United States alone [1].

Precision Agriculture

PicoRadio may be an ideal application for agriculture and agricultural research with the growth of precision agriculture. A PicoRadio network could easily be used to monitor and control drip irrigation. However, the installation overhead that drip irrigation requires makes it feasible only for greenhouses, small testplots, vineyards, orchards and other crops that remain in the field for several years. Flood irrigation and siphon irrigation are more challenging to monitor because the water flow is largely uncontrolled. With flood and siphon irrigation, the majority of irrigation time is spent merely monitoring the progress of the water flow so that one can disconnect the water source at the appropriate time. This process could be more automated with a wireless network of moisture sensors; PicoNodes could be mounted on top of stakes that the farmer would put in the field prior to irrigation. A monitoring system could employ a graphical user interface (GUI) that displays a detailed diagram of the water flow on a virtual field.

2. Design Goals

There are four main design goals that define PicoRadio:

- Low cost
- Small size
- Low energy consumption
- Self-configuration

The first three goals, low cost, small size, and low energy consumption, are related through the high level of integration found on a single PicoRadio chip. Virtually all other aspects of the PicoRadio network architecture are derived from these goals.

2.1 Low Cost

The design goal of low cost was adopted because PicoRadio requires a large number of nodes in its dense network; the cost of each individual node must be small so that the cost of the entire network is low enough to become a ubiquitous technology. Reducing the cost of the PicoNode is primarily achieved by integrating more and more components of the PicoRadio design onto a single chip. Adding more components on a chip reduces the manufacturing cost of the PicoNode by reducing the amount of packaging and assembly needed to make a PicoNode. Increased integration also decreases the cost to operate a

PicoRadio network in that the chip will consume less power so that it will become possible to operate the chip solely on energy provided by energy scavenging.

2.2 Small Size

If PicoRadio is to become truly ubiquitous then it must be small in size in order, for example, to be incorporated into building materials such as ceiling tiles. It is clear that size and cost are closely related; the small size of the node is also achieved by integrating more blocks onto a single chip.

2.3 Low Energy Consumption

Since every PicoRadio network is comprised of a large number of PicoNodes, battery-powered nodes are not practical because replacing the batteries of hundreds or thousands of nodes would be too tedious and costly. Ideally, the PicoNodes should be powered with energy collected from their environment. Two sources of energy likely to be available in a node's environment are light and vibrations. Solar cells can power the node using any sunlight or any other indoor lighting that the node encounters. Micro-Electro-Mechanical Systems (MEMS) devices designed to extract energy from vibrations allow the PicoNode to power itself from, for example, vibrations resulting from a footstep or a truck driving by. If the PicoNodes are to be powered by energy scavenging methods, then the PicoNodes must be low-powered in order to be self-sufficient on the small amount of energy that they are able to derive from their environment.

2.4 A Self-Configuring Network

The vision for PicoRadio is of a wireless network that will run without any type of maintenance in either supplying energy (changing batteries) or configuring the network. The PicoRadio network is ad-hoc, which means that the collection of nodes in the network is dynamic; nodes may come in and out of the network at any given time due to node failures, radio interference, or node mobility. The network must be able to reconfigure itself accordingly; if a node fails, the network will still work because the network will reconfigure itself to exclude the failed node, and information will be routed around the failed node. In other words, the network has no single point of failure.

3. Requirements Derived from Primary Design Goals

3.1 Network Density

A dense network is ideal for sensor applications because a large number of sensors increases the resolution of the data collected from the sensor network. Also, if one assumes that conditions do not change much between two sensors that are close to each other, then not much information is lost if one of the two sensors fails.

The density of the PicoRadio network also requires that the nodes be self-configurable because with such a large number of nodes, some nodes are bound to fail, and the network must able to tolerate this and keep on running; as mentioned previously, there must be no one single point of failure in the network.

3.2 Multiple Hops

Because the energy consumption of the PicoNodes must be low, the energy required to transmit information should be reduced as much as possible. The required transmission energy increases with the cube of the distance to the destination—so, as required by the design goal of low energy consumption, the transmission distance must be short (about 2-3 meters). Rather than trying to transmit across a large building directly, data is passed along in multiple hops. Each hop is the transmission of data from one node to another node. After several hops, the data will reach its destination. In effect, the transmission energy requirement will increase linearly as the distance increases because the overall source to destination transmission energy requirement is the sum of the individual hop transmission energies. The limited hop distance is another reason why PicoRadio networks are dense; the average hop distance is only 2-3 meters and several hops are required to convey information over a large area, so to be able to relay information reliably, the network must be dense.

4. The Three Phases of PicoRadio

There are three phases of the PicoRadio project. The project is implemented in several phases so that different aspects of the overall PicoRadio design can be tested and developed independently in the initial phases and then be later combined in the final phase. The physical size of the PicoNode decreases by an order of magnitude in each phase.

4.1 PicoRadio I

PicoRadio I, also known as the TestBed, consists of PicoNodes that are made from off-the-shelf components (Figure 1). Although the TestBed PicoNodes are much larger and higher powered than the final PicoNodes will be, the nodes do allow PicoRadio researchers to test PicoRadio algorithms in a real environment. Reconfigurability of the TestBed hardware design is possible; PicoRadio researchers can load their hardware designs onto a Xilinx 4020 Field Programmable Gate Array (FPGA). ‘C’ programs can be loaded onto the StrongArm 1100 200 MHz microprocessor.

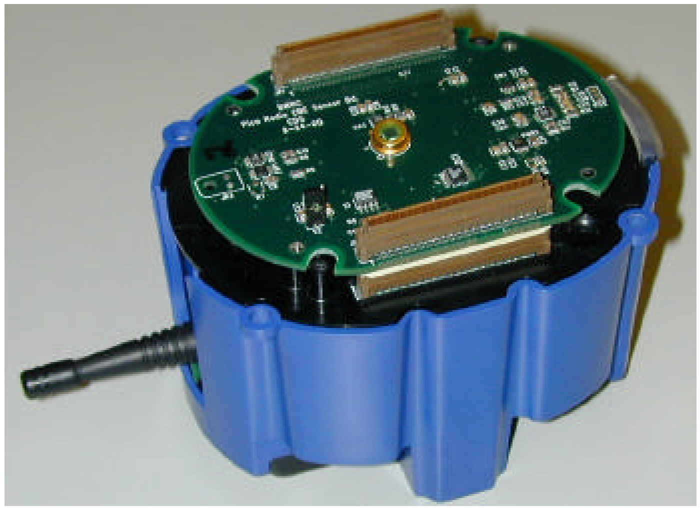


Figure 1: A TestBed PicoNode

4.2 PicoRadio II

PicoRadio II is a two-chip intercom system. The purpose of the second phase is to investigate a design flow for PicoRadio Phase III. The digital portion of the PicoNode is implemented on two integrated circuits and the radio and other analog portions of the PicoNode II consist of off-the-shelf components.

4.3 PicoRadio III

PicoRadio III is even more integrated than the two previous phases (Figure 3). Like PicoRadio II, PicoRadio III also has two chips. However, in phase II integration covered only the digital circuits. In phase III, the radio and A/D converters are integrated onto a custom chip and the two phase II digital chips are combined into one. Energy consumption is of greater concern in the third phase. To address this, the PicoNode III contains a system supervisor that will shut off portions of the PicoNode when they are not in use in order to save energy. PicoRadio III will be powered by solar cells.

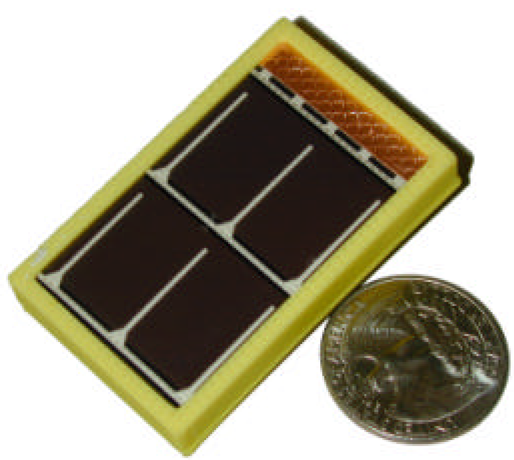


Figure 2: PicoNode III Prototype

5. Communication Protocol

Communication in the PicoRadio network includes both inter-PicoNode and intra-PicoNode communication. Intra-PicoNode communication involves communicating within the six functional blocks of the PicoNode design. Inter-PicoNode communication involves sending information out over a PicoRadio network.

5.1 The Packet

A packet is a piece of information that is conveyed across the PicoRadio network. Each packet originates at a source node and moves through the network to reach its final destination node. The packet is divided up into two sections: the header and the payload (Figure 3). The header contains information to be used by a particular layer of the PicoNode design. Each layer has its own header that becomes part of the payload of the lower layers. When a packet is received, the header for a given layer is stripped off before that layer forwards the packet to the next higher layer. The payload contains the actual data that was requested as well as headers that are used by other, higher layers of the PicoNode design.

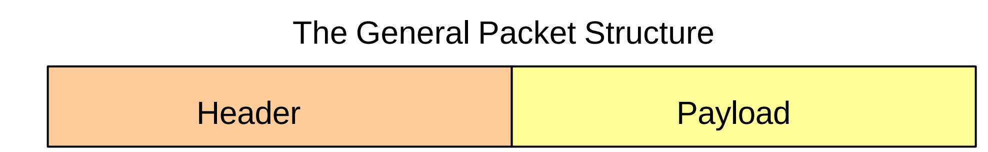


Figure 3: The Packet Structure

5.2 Location-Based Addressing

Each node in a PicoRadio network is assigned an address based on its location. A node determines its location by measuring the distance from a number of "anchor" nodes with known geographical positions. Location-based addressing is ideal for sensor networks because the data returned by the sensors is usually location dependent, and most requests for information are also based on location—for example, one might want the temperature in the SE corner of room 500. Locationing packets are periodically sent out over the network to determine or update a node's location. A locationing packet can be a broadcast from an anchor, or it can be a packet that is sent to all of a node's neighbors when a node updates its location [2].

5.3 Classes and Types of Packets

There are three classes of packets: directed broadcast (flooding), unicast, and locationing. This project is primarily concerned with broadcast and unicast packets.

Broadcast packets are sent to all nodes, and are of three types: interest, statistics/maintenance request, and route trace request. A unicast packet is sent to a specific node as specified by the X, Y, Z coordinates in the network header. Unicast packets are always generated in response to broadcast packets.

A critical distinction between broadcast and unicast packets is that a broadcast packet is sent only once, and the sender has no way of knowing whether it was received or not, while a unicast packet is potentially sent multiple times in a procedure known as a *session*. In a session, a dialog is set up between sender and receiver, and the receiver will acknowledge receipt of data. If no acknowledgement is received in a specified time period, another data packet will be sent. This retransmission can happen several times. So, a unicast data has a better overall chance of being received.

In comparison to true broadcasts which are intended for any receiver within range, directed broadcast packets are sent by a single node to a specific group of nodes. This group of nodes is determined by two sets of X, Y, Z coordinates specified in the network header. The two sets of X, Y, Z coordinates specify opposite corners of a three-dimensional box. Any node that is within this area is considered to be in the area of interest and should respond to the packet accordingly. An interest packet is a request for data that is observed by the sensor. A statistics/maintenance request packet is a request for information on the performance of the network protocol.

5.4 Packet Composition

A PicoRadio packet consists of a header and a payload. As the packet is being composed each software layer adds its own header to the packet. The network packet header is of primary concern for measuring the performance of the PicoRadio network protocols. The network packet header gives information on the source address, the destination address, the payload length, and what type of data the payload contains.

The payload contains the actual data that will be used by the end application. This data is packed into integers by shifting and masking operations and then is later extracted by more shifting and masking operations that decode the information in the fields of the payload.

5.5 Layers

The PicoRadio communication protocol is divided up into several layers that work together to transmit and receive packets, determine their content, collect data and respond to requests, compose requests, and route packets (Figure 4). The layers are defined in a hierarchy of abstraction, with the lowest layer being the physical layer and the highest layer being the application layer.

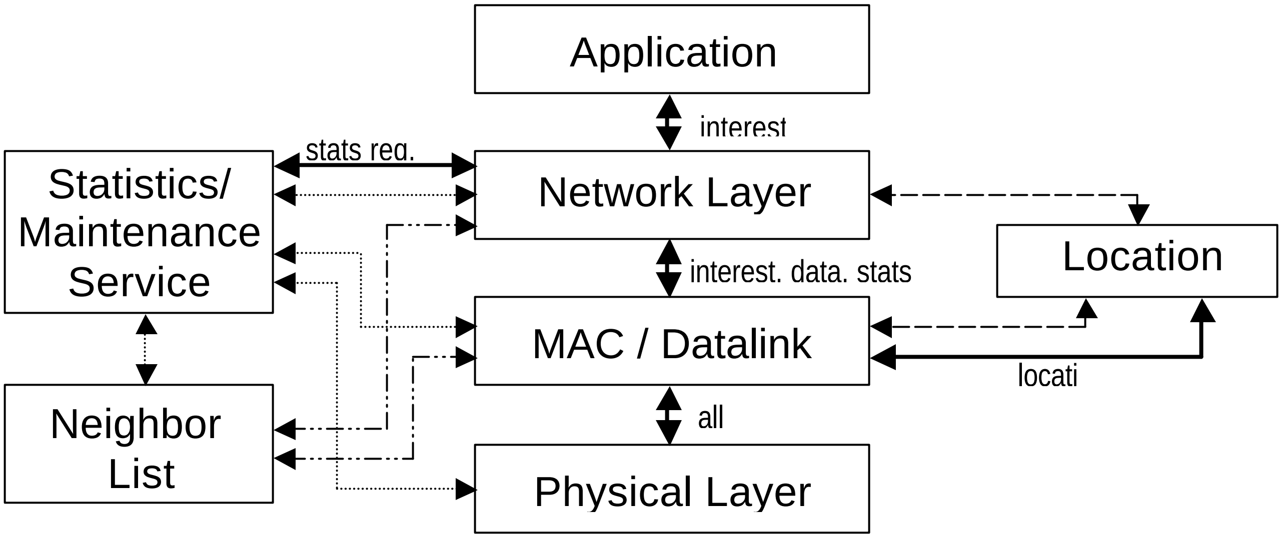


Figure 4: The layers of the PicoNode design

The Physical Layer

The physical layer is composed of the actual hardware itself—in a PicoRadio Phase I node, the physical layer includes the Bluetooth radio and a circuit within the Xilinx FPGA. Commands such as "turn the radio on" are executed by the Physical Layer.

The MAC/Datalink Layer

The Media Access Control (MAC)/Datalink layer is responsible for node-to-node communication. The purpose of the MAC/Datalink layer is to control access to the radio channel; it tells the physical layer when to transmit and what channel to use. This layer handles data received from and sent to the physical layer as well. The MAC/Datalink is concerned with a single node-to-node connection; it does not have a concept of the original packet source and ultimate destination. In the PicoRadio TestBed, the interface between the Network and MAC/Datalink layers is where the hardware meets the software.

The Network Layer

The network layer is where functions concerning network routing are executed. The network layer is concerned with endpoint-to-endpoint communication that involves the overall packet source and destination. When the network layer receives an incoming packet, it determines whether the packet is for itself and also determines whether the packet should be forwarded to other nodes. For an outgoing packet, the network layer determines which nodes the packet should be sent to in order to reach the destination, which can be several hops away.

The Application Layer

The application layer is where the data in the payload is actually produced and consumed. Two applications, controller and sensor, are currently implemented in the PicoRadio TestBed.

The Controller Application

> The controller application layer composes packets that request information from the sensors. When receiving packets in response to the requests, the controller application extracts the information from the payload, processes this information, and writes the results to a log file.

The Sensor Application

> The sensor application produces data in response to request packets it receives from a controller. To produce the requested data, the sensor application records sensor data and puts the requested data into response packets.

5.6 Services

In addition to the layers, three types of services, the Statistics/Maintenance Service (SMS), the Neighbor List Service, and the Locationing Service are also functional blocks within the PicoNode system.

5.7 Sending and Receiving Packets

The MAC protocol, a subpart of the overall network protocol, is a set of rules for node-to-node communication that the PicoNodes follow in order to send packets over the PicoRadio network. Requests from the controller are broadcast packets and responses from the sensor are unicast packets. Packets sent from node to node during the hops are also unicast packets. In order to ensure that the unicast packet reaches its destination, the session setup protocol (a subpart of the MAC protocol) is followed.

To send a unicast packet a session must be set up, so the sending node sends a session setup packet to the receiving node on a channel that all nodes listen to, then switches to a channel that is private to the receiver (Figure 5). The advantage of a unicast session is that it takes place on a separate channel, protected from contention on the shared 'broadcast' channel. The receiving node responds to the session setup packet with a ready to receive packet on its private channel. When the sending node receives the ready to receive packet, the sending node then responds with a data transmit packet on the private channel, which contains the actual data of interest. When the sending node no longer receives any ready to receive packets it knows that the data has arrived at the receiver, so it will stop sending data transmit packets, and the session has ended.

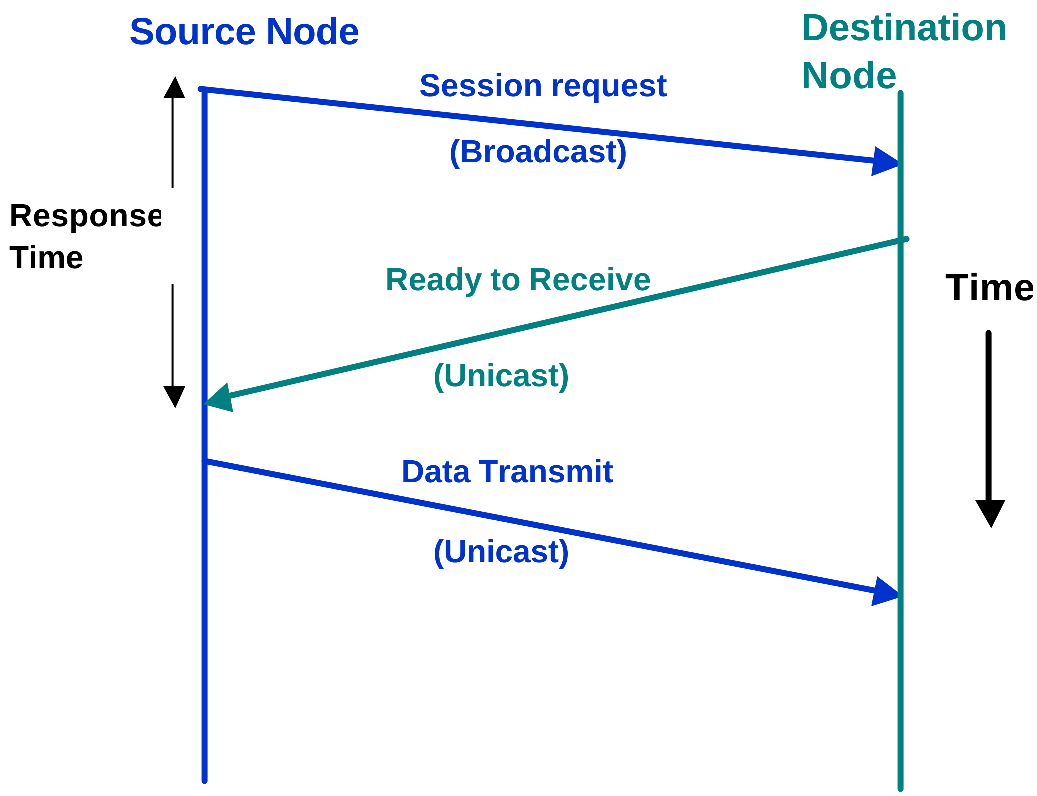


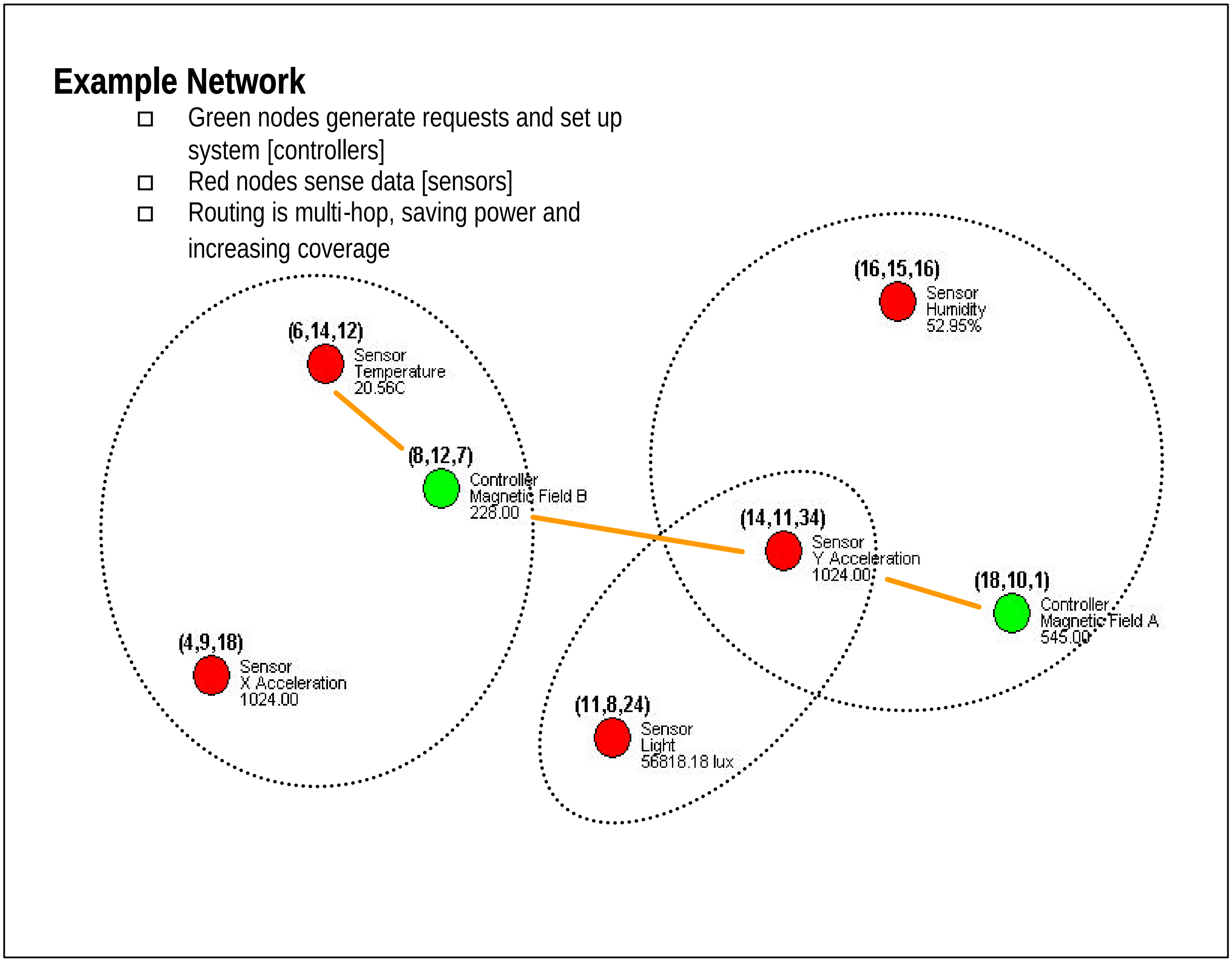
Figure 5: The protocol involved in a session

5.8 Timing

To ensure that the unicast mode is reasonably reliable the network protocol is designed to allow for the retransmission of packets by allowing a specified amount of time in which the packet can be retransmitted in each step. Timing is important in this protocol; the session can time out at any step of the protocol. If the specified amount of time passes and no ready to receive signal is received by the sending node, then the session has failed and is terminated. Also, if the specified amount of time passes and the receiving node does not receive a data transmit signal, then the session also ends.

5.9 The Neighborhood

Each node keeps a list of its neighbors, those nodes with which it can communicate directly. The neighbors are said to be within a one-hop radius and comprise a neighborhood (Figure 6). In the neighbor list, the (X, Y, Z) location of each node is associated with a neighbor ID. The private channel for a node used in unicast is related to its ID, so each node must have an ID that is unique from its own neighbors and its neighbors' neighbors so that collisions do not occur on unicast channels. The fact that the neighbor ID is assigned dynamically allows the PicoRadio network to be self-configurable.



Nodes periodically send out ping packets to notify neighbors of their presence. The time between pings is recorded, and if a set amount of time has gone by and no pings are received from a particular node, then the node has timed out and is removed from the neighbor list. This allows the network to recover from removed or failed nodes.
If a new node comes into the network, the node will select a unique neighbor ID after arbitration with its neighbors, and the node's neighbors will receive the new node's pings and add its ID to their neighbor lists.

5.10 Routing

The routing of a PicoRadio packet is determined by the information in the network payload of the packet, namely the source address and the destination address. For broadcast packets, the node determines whether its own location is between the source node and the destination node and forwards the packet if it lies in the path to the destination. As the broadcast packet travels from source to destination, its route is remembered by the nodes it passes through. The response to a broadcast request packet is a unicast response packet. When the response packet is returned it follows the same route that the request packet took.

5.11 Request/Response Packets

A controller sends requests to the sensors for information. The information that the controller requests can be of several types: information about the environment that the sensor records, information about the network's performance, and information about the node's neighbor list. A request packet is always a broadcast packet, and because of forwarding, the destination node(s) may receive the packet more than once. To avoid processing the same packet twice, each request packet is assigned a sequence number and the receiving node checks the sequence number of all incoming packets. An incoming packet will be dropped if the sequence number is repeated.

When a request is received, all the useful information in the packet is saved so that it can be used in the outgoing packet. This useful information is the source address (the source address of the request packet becomes the destination address of the response packet), the packet type (the packet type indicates what kind of information the sensor should respond with) and subtype (if present), and the payload length. Once the useful information has been saved, the requested information is gathered and the response packet is constructed.

6. The Statistics and Maintenance Service (SMS)

This paper addresses the performance measurement of network protocols using the PicoRadio I TestBed as the testing medium. The bulk of this project has been devoted to designing, implementing, and testing the service that does the performance measurement, called the Statistics and Maintenance Service (SMS).

Many features of the PicoRadio network protocol must work together in order for the network to function properly. The network must also be robust and the robustness of the network must be verified. The SMS monitors and records network performance at every step of the protocol to allow PicoRadio researches to test the robustness of their algorithms and to give them detailed data for debugging purposes. As mentioned before, knowledge of how the PicoRadio network performs can be used to make critical design decisions in the future phases of the PicoRadio project.

6.1 SMS Subtypes

The first step in designing the SMS was to determine what information would deliver the most insight into how the network is functioning. The information was then divided up into several groups, and each group represents a different SMS packet subtype.

The SMS subtypes are:

- Neighbor List
- Network
- Session
- Maintenance
- Bit Error Rate (BER)
- Physical

Neighbor List

The Neighbor List subtype returns the number of neighbors in a node's neighbor list. The contents of the neighbor list, which associates the neighbor ID with the node's location, is also returned.

Network

The Network subtype gathers information about the packets that a node has routed.

Session

The SMS Session packet returns information on the total number of sessions, session requests, ready-to-receives, and data transmits that occurred in a given amount of time. Ideally there would only be one session request, ready-to-receive, and data transmit packet per session, but in practice that number is greater than one because in a session, the protocol allows packets to be sent multiple times until the appropriate response is received or that particular part of the session has timed out. The lower the ratio of session request, ready-to-receive, and data transmit signals to the number of sessions, the better the network is working because it means that fewer packets had to be retransmitted.

Maintenance

The SMS maintenance packet allows system managers to modify internal parameters such as timeout values from a remote console.

Physical

The Physical SMS packet subtype will return information about the number of CRC (Cyclic Redundancy Check) failures, the total number of packets processed, and the bit error rate. This information will help PicoRadio researchers determine the cause of a packet failure.

6.2 Using the SMS

The SMS allows the user to specify a class of information and the area of interest by means of a graphical user interface at a control console. The SMS then sends out a request for the information. The returned data is written to a log file and can be easily read by a spreadsheet program such as Excel for further analysis.

6.3 Implementation

The SMS is primarily implemented with C code that runs on the StrongArm microprocessor. Two SMS modules were written: one for the sensor and one for the controller. For gathering session data, the FPGA design was modified to add session packet counters and control logic for the counters.

The SMS Controller Module

The SMS controller module takes the information from the control console and forms an SMS request packet with that information. The controller module also decodes the information in the SMS response packets that it receives and writes that information to a log file.

The SMS Sensor Module

The SMS Sensor Module receives SMS request packets and decodes the information in the packet to determine what type of information is requested. The module then gathers the requested information and composes a SMS response packet to be sent back to the controller.

6.4 The SMS Packet

There are two types of SMS packets: SMS request packets and SMS response packets (Figure 7). The SMS request packet payload contains the SMS subtype so that the receiving node knows what type of information is requested. The SMS response packet contains the requested information and the SMS subtype. The fields and size of the payload of the SMS response packet vary with the SMS subtype.

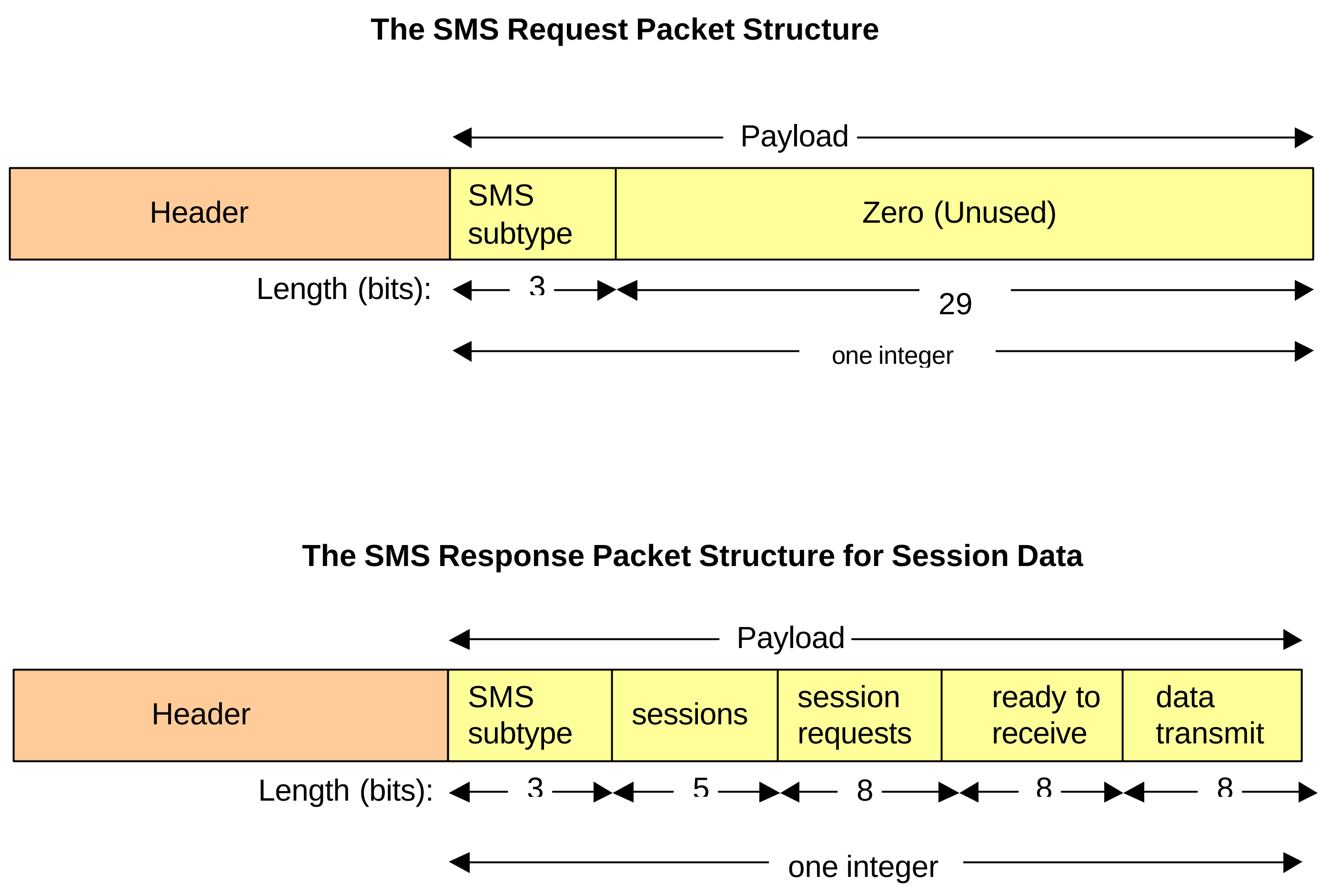


Figure 7: The SMS Request and Response Packets

6.5 Gathering Information

The method the SMS module uses to gather the requested information depends on the type of information that is requested. In the case of an SMS Neighbor List packet, gathering the information can be done in software alone. In the case of an SMS Session packet, packet counters in the Xilinx FPGA must be read to retrieve the needed information.

When gathering information involves hardware, the PicoNode program must be interrupted to gather the information that resides in hardware. In the case of the SMS Session packet subtype, four counters had to be added to the PicoNode design. A maximum number to which these counters count is set in the software, and is written to a Xilinx port. A comparator is used to compare the maximum count number with the values of the counters. If the value of the session request, ready to receive, or data transmit counter reaches the maximum count value, the PicoNode program is interrupted and the interrupt service routine (ISR) reads the values of the counters and saves them as variables in the PicoNode program. The counters are then cleared and the interrupt service routine returns control to the main program.

6.6 Testing

To measure network protocol performance, two types of tests were performed: those involving a single sensor and controller (single-sensor test) and those with multiple sensors and a single controller (multi-sensor test). Performance is measured in the number of tries it takes to reach the receiving nodes, as well as the number of failed packets. Better performance means fewer failed packets and taking fewer tries to reach the receiving node.

In the single-sensor test the sensor and controller are able to communicate without contention from other nodes. In this context, contention occurs when two nodes transmit at the same time and on the same channel. Without contention, no broadcast packets should be lost due to collisions with broadcast packets from other nodes. From these observations, one would expect the performance of the broadcast and unicast channels to be about the same in a network comprised of only two nodes. The single-node tests verify that the performance of the broadcast and unicast channels is comparable (Figure 8).

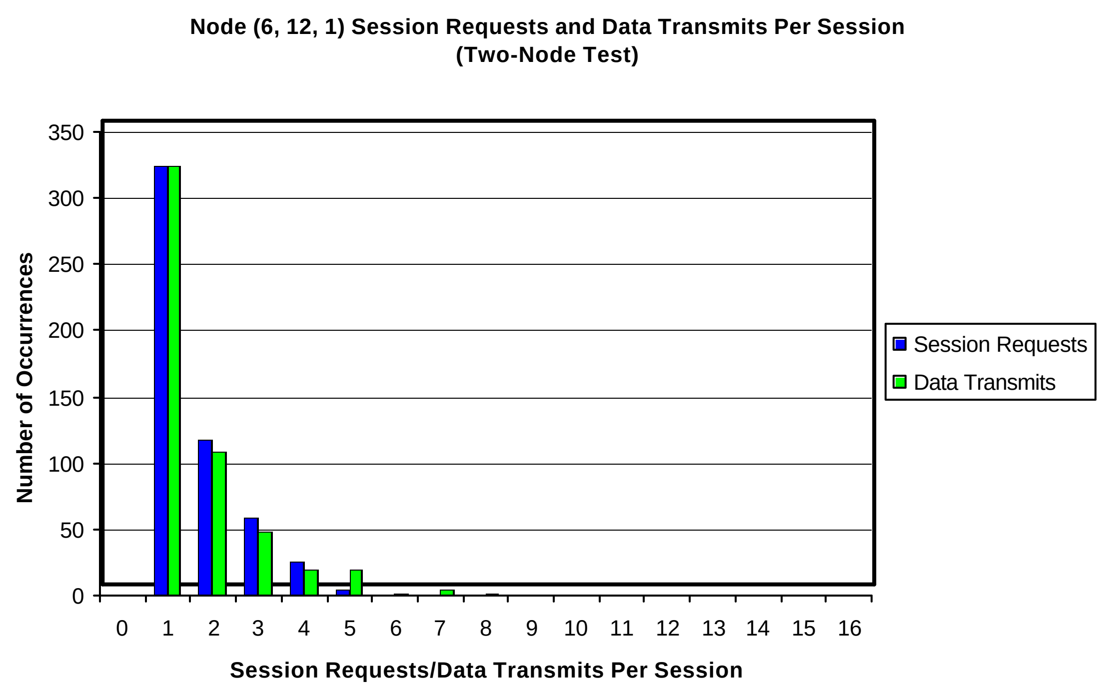


Figure 8: Single Sensor Test

Significant differences between the unicast and broadcast channels are seen in the multi-sensor test, where the unicast channel clearly outperforms the broadcast channel in terms of the number of tries it takes the packet to reach the receiving node. (Figure 9). This is because the performance of the broadcast channel is greatly reduced due to packet collisions while the packets on the unicast channel do not experience collisions.

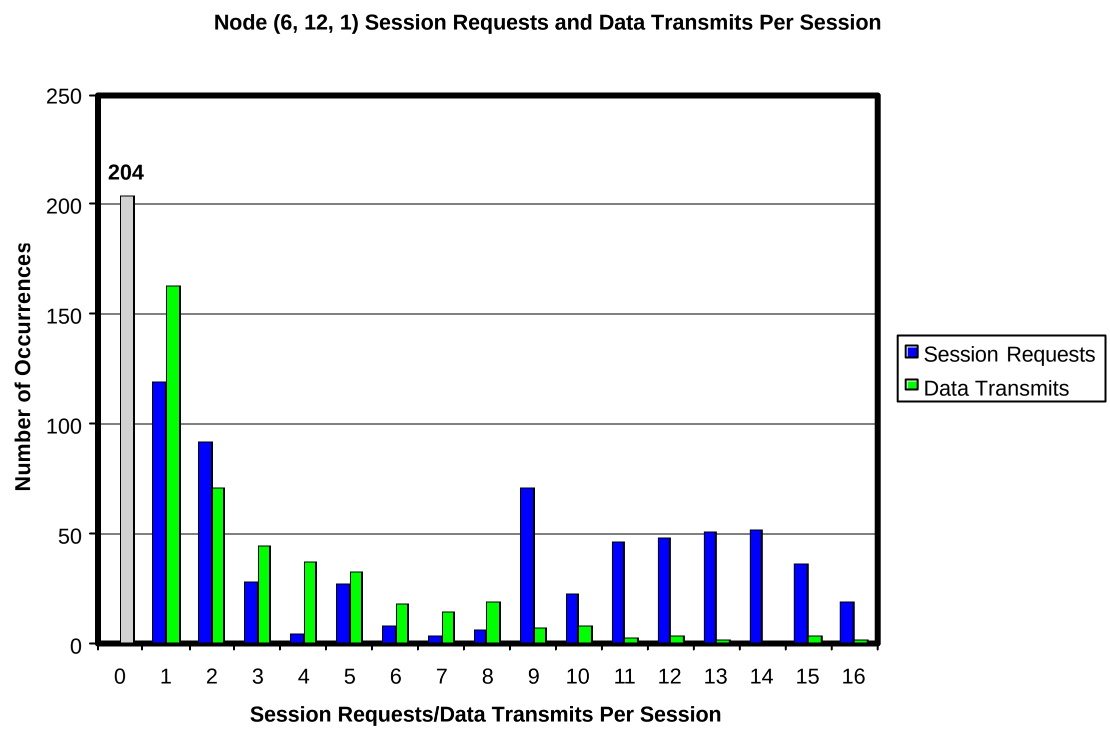


Figure 9: Multi-Sensor Test

Increased Performance Due to Packet Forwarding

Although the number of collisions on the broadcast channel increased in the multi-sensor test, the number of responses generated per request packet also increased dramatically. In the two-node test, 35.6% of the request packets received responses, while the nodes in the multi-sensor test responded to, on average, 59.3% of the request packets (Figure 10). The increase in the response-to-request ratio can be attributed to broadcast packets being forwarded by the receiving nodes, allowing the nodes in the area of interest several chances of receiving the broadcast packet.

Single-Sensor Test			
Node	Number of Request Packets	Number of Response Packets	Percent Responded (%)
(6, 12, 1)	1500	534	35.60

Multi-Sensor Test			
Node	Number of Request Packets	Number of Response Packets	Percent Responded (%)
(6, 1, 1)	1500	956	63.73
(12, 12, 1)	1500	944	62.93
(3, 12, 1)	1500	885	59.00
(6, 12, 1)	1500	865	57.67
(12, 1, 1)	1500	799	53.27

Figure 10: The ratio of response packets to request packets in the multi and single-sensor tests

Physical Characteristics of the Individual PicoNodes

The results of the multi-sensor test revealed that not all nodes have equal physical capabilities. Since a single session request per packet is ideal, comparing the number of packets with single session requests that were sent by each node should indicate the relative quality of the physical layer of each node, that is, the radio (Figure 11). From this data, once can see that node (6, 1, 1) is the best performing node in terms of its ability to send a successful packet in a single try. The number of failed sessions is also an indicator of physical capability; the best performing nodes have a lower number of failed sessions. Node (6, 1, 1) is also one of the most capable nodes because it has the fewest failed sessions and the most session requests that reached their destination on the first transmission.

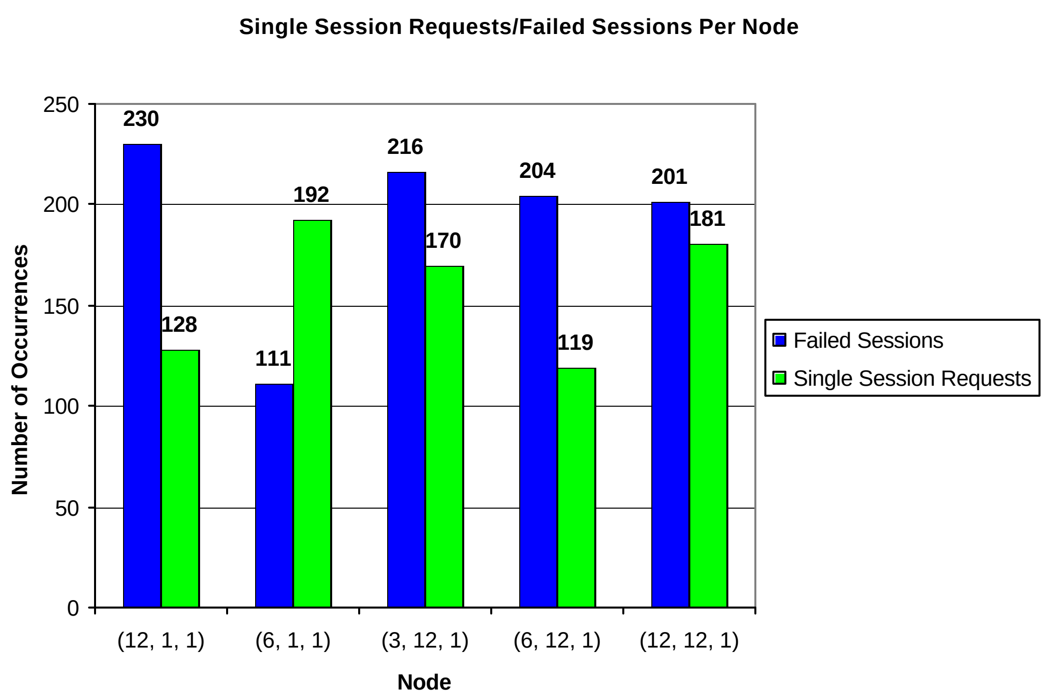


Figure 11: Relative Ability of Nodes in Terms of Failed Packets

7. Conclusion

A mechanism for acquiring PicoRadio network information, the SMS module, has been designed and implemented. Using this mechanism, we have been able to collect highly valuable information about the operation of the PicoRadio network. In some cases it has confirmed assumptions about the network. In others, it has pointed to areas where performance can be improved, such as the high broadcast loss rates in the single-sensor tests and the variability in relative quality of nodes.

The results of the single-sensor and multi-sensor tests show that multiple nodes increase the network performance due to packet forwarding. In a multiple sensor environment, unicast transmissions have been shown to be more reliable than broadcast transmissions due to the absence of packet collisions on the unicast channel. However, in the multiple-sensor environment, the lossy broadcast channel can be made up for by packet forwarding. Finally, the tests reveal that the nodes differ in their ability to communicate—knowledge of these differences can be used to normalize the results of future tests.

Work on the SMS module is a work in progress; more functions that will return different types of data will be added to the SMS module in the future.

8. Acknowledgements

I would like to thank Professor Jan Rabaey for giving me the chance to work at the Berkeley Wireless Research Center. I would also like to thank my mentor, Fred Burghardt; I have been extremely lucky to have him as a mentor this summer, and cannot imagine how he could have possibly done a better job at it. I thank the SUPERB staff for their help and for putting on a truly excellent program. Finally, I would like to thank my fellow SUPERBites their support and friendship. In a word, my summer has been SUPERB

9. References

[1] J.L. da Silva, Jr., et. al. *Design Methodology for PicoRadio Networks.* http://bwrc.eecs.berkeley.edu/Research/Pico_Radio/Publications.htm. 2001.

[2] Savarese, Chris. *Robust Positioning Algorithms for Distributed Ad Hoc Wireless Sensor Networks.* http://bwrc.eecs.berkeley.edu/Research/Pico_Radio/Publications.htm. 2000.

[3] Arens, Ed. Center for the Built Environment, Presentation at the January 2001 BWRC retreat.