HIGH-SPEED MOBILE NETWORKS FOR MODERN FARMING AND AGRICULTURAL SYSTEMS

A Thesis
presented to
the Faculty of California Polytechnic State University,
San Luis Obispo

In Partial Fulfillment
of the Requirements for the Degree
Master of Science in Electrical Engineering

by
J. Santos Najar-Ramirez
March, 2014

COMMITTEE MEMBERSHIP

TITLE:	High-Speed Mobile Networks for Modern Farming and Agricultural Systems
AUTHOR:	J. Santos Najar-Ramirez
DATE SUBMITTED:	March 2014
COMMITTEE CHAIR:	Albert Liddicoat, PhD Professor of Electrical Engineering
COMMITTEE MEMBER:	Taufik, PhD Professor of Electrical Engineering
COMMITTEE MEMBER:	Samuel Agbo, PhD Professor of Electrical Engineering

ABSTRACT

High-Speed Mobile Networks for Modern Farming and Agricultural Systems

J. Santos Najar-Ramirez

High-speed mobile networks are necessary for agriculture to inventory **individual plant health, maximize yield and minimize the resources applied**. More specifically, real-time information on individual plant status is critical to decisions regarding the management of resources reserved and expended. This necessity can be met by the availability of environmental sensors (such as humidity, temperature, and pH) whose data is kept on storage servers connected to static and mobile local area networks. These static and mobile local area networks are connected to cellular, core and satellite networks. For instance, agricultural experts remotely working on vast acreage farms from business offices or while traveling can easily connect their notebook computers and other portable devices to these networks in order to check farm status, send email, read industry news or arrange a visit to neighbor farms or suppliers. Today, several mobile phone companies offer broadband service with 2Mbps downlink in rural and dense urban areas, however, they do not typically exist in farm areas. Although these networks (such as 802.11ac/n, 3G, 4G, etc) are significant achievements, they do not meet the projected needs of the agricultural industry. The present use model of high-speed networks for email and multimedia content, together with agriculture's expected intensive use of real-time plant and environmental condition monitoring, with statistics/plots and real-time high resolution video, necessitates a highly integrated and highly available networked system. For agricultural experts, attentive to market needs, seamless high-speed wireless communication 'anywhere, anytime at any speed' is critical to enhancing their productivity and crop yields.

ACKNOWLEDGMENTS

Access routers provided by the Hewlett-Packard Company.

I wish to express my gratitude for the patience and understanding to my mother, father, teachers, professors, advisors and my entire family at school and at home.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

I INTRODUCTION

Problem Statement

The necessity of constant access to information in the agricultural environment is more and more essential for agricultural resource management. Specifically, real-time environmental sensor information such as pH balance, humidity, soil, plant and ambient temperature, gathered and processed, is beneficial to the agricultural industry to determine immediate resource application. One example that illustrates the use of the information is determining **individual plant health and yield**. If a plant is not on target to meet the expected harvest volume for the applied resources, then the site high resolution video images of the plant, together with sensor data would notify the operator of a variance in the humidity, pH balance, soil temperature and ambient temperature. These metrics would show that either the soil content should be adjusted for the proper nutrients, increased or decreased irrigation and or removal of weeds. Finally, in some cases, an on site visit to the field with the sensor information on a handheld device would show that this plant would not benefit from future resource allocations. The affected plant would then be removed so that the soil remains dormant for the remainder of the current crop season.

Today, access to this type of information is possible “anyplace, anytime at vehicle speeds” via mobile portable devices with advanced communication features. However, as the aggregate data from agriculture, together with the current use of mobile devices for multimedia content, increases access to the network for voice, data and video applications, will strain the data traffic demands on service providers. This demand for real-time access is a significant challenge.

To meet the expected growth in demand, network service providers must be able to reduce bottlenecks by upgrading their 3rd Generation (3G) and 4th Generation (4G) - Global System for Mobile Communications (GSM) and Code Division Multiple Access (CDMA) networks with highly-mobile wireless local area network technology. Specifically, the new highly-mobile networks will use the same wireless encoding schemes, such as Orthogonal Frequency Division Multiplexing (OFDM), as 802.11 networks. Therefore, once the highly-mobile, high-speed networks are integrated into the 3G and 4G networks, mobile users with high-reliability, high-bandwidth needs, will be able to seamlessly connect their mobile devices to these high-speed networks.

Analysis of Requirements

The requirements for the High-Speed Mobile Networks for Modern Farming and Agricultural Systems are:

- Scalable farming adopted easily to optimize all farm types: family run to enterprise farms
- System applicable to farms that require extensive coverage with limited resources and labor
- Fast alerts notify operator immediately if conditions are unfavorable for expected yield on a per plant basis
- Control over irrigation in dynamically changing environmental conditions
- Grow plants that adapt and sustain their own nutrient and irrigation needs and without the need for intervention
- Constant monitoring for zero-loss and optimal expenditures in farm to market
- Highly available and fault tolerant system for data monitoring and gathering
- Predict market conditions using data from the field
- Defaults to manual operation when necessary
- Secure data transmission
- Encoding of data and network resources to be optimized for efficient ‘data harvesting’

Proposed Solution

In this thesis Mobile IP Version 6 (MIPv6) on Wireless LAN (WLAN) (i.e. 802.11a/b/g/n/ac) networks is used to solve the "seamless, anywhere, anytime, any speed" network challenge.

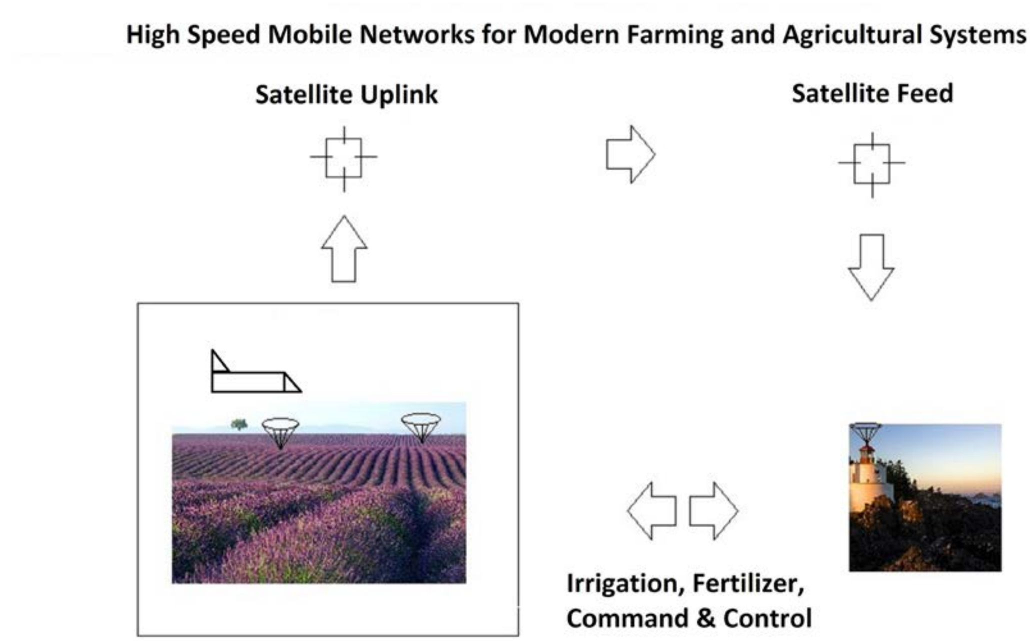

Figure 1 High-Speed Mobile Networks for Modern Farming and Agricultural Systems

Figure 1 shows the proposed solution [1]. The blocked off section represents the scope of this thesis, building a high-speed, highly-mobile and highly available wireless network in the large-scale farm environment. The arrows indicate the real-time data flow. Data originates from sensors on the farm. Sensor data is stored on the local Access Routers [with satellite antennas] and periodically sent to the uplink satellite under normal operation. The data is handed over to the subscriber from the satellite feed. The same real-time information can be retrieved, **at regular intervals or on-demand**, from the ground-based networks denoted by the bi-directional arrows. Command and control of the irrigation and fertilizer systems is operated from the ground-based networks.

Figure 2 shows the key components of the modern farm with an integrated high-speed mobile network as noted in the blocked off section of Figure 1. The Mobile IPv6 Node sends radio scan requests and receives radio probe responses, shown as circles highlighted in orange, in Figure 2. These responses reach the Mobile IPv6 Node "data harvester" at different intervals from which it selects an Access Router for a brief moment to retrieve the sensor and image data.

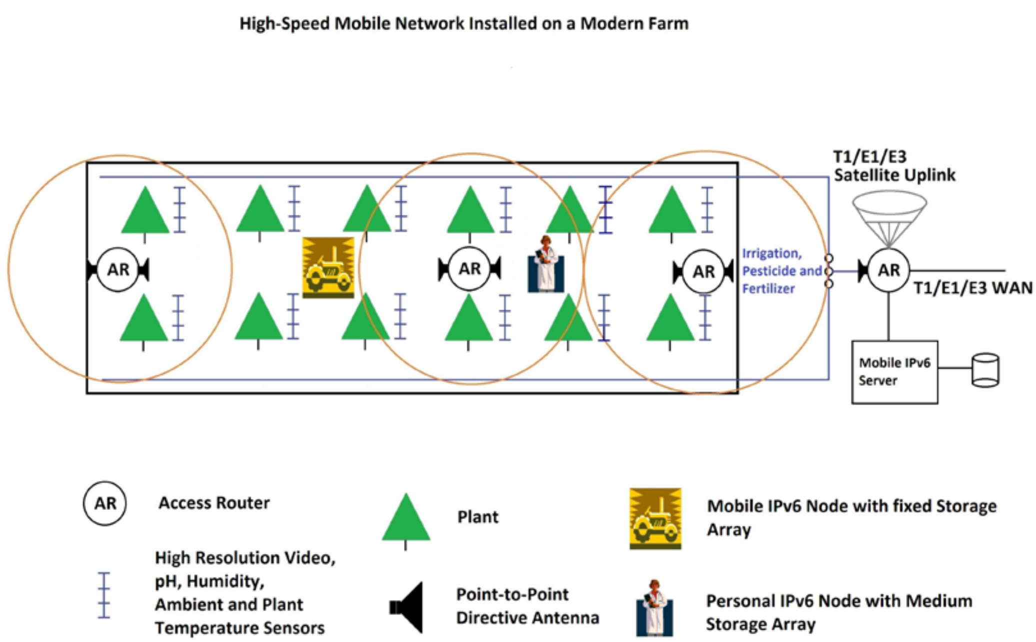

Figure 2 High-Speed Mobile Network Installed on a Modern Farm

The key components of a Modern Farm with a Mobile High Speed Network are:

1. Access Routers with multiple network interfaces noted as follows:

    - 802.11 a/b/g/n/ac Network Interface [broadcast circles shown]

- Point-to-Point Directive Antenna for 802.11 a/b/g/n/ac Mesh
- T1/E1/E3 Satellite Uplink Network Interface [right most Router]

2. High-Speed Mobile IPv6 Server with fixed Storage Array
3. High-Speed Mobile IPv6 Node with onboard Storage Array [airplane in Figure 1]
4. Per plant High Resolution Video, Humidity, Soil pH, Ambient and Soil Temperature Sensors
5. Per plant Drip Irrigation, Pesticide and Fertilizer system controlled electronically
6. Vehicle or Tractor with Mobile IPv6 Node and onboard Storage Array
7. Personal Handheld Mobile IPv6 Node with medium capacity Storage Array
8. On-site Control Valves for Irrigation, Pesticide and Fertilizer with Manual Shutoff

The sensor and image data from each plant can be 'harvested' from the Access Routers in several ways:

- Regular Satellite Transmission Feed [satellite antennas shown]
- Ground-Based On-Demand Request via 802.11 and Cellular to Telecommunications Network
- Fly-Over Field On-Demand in an Airplane with 802.11 antennas [Mobile IPv6 Node shown in Figure 1]

The first approach is possible using geostationary satellite subscriptions. The second and third approaches require a build-out of the 802.11 network on the farm. Figure 2 depicts four access routers, a tractor and personal "data harvesters" as the Mobile IPv6 nodes enabled with onboard storage capacity for roaming from one router to the next.

802.11 networks are already widely deployed in hot spots, campus, metropolitan environments. The next logical step, therefore, would be to extend the reach of 802.11 wireless networks to the rural areas that would benefit from real-time access to agricultural conditions on a per plant basis. Rural and farm access, together with access in high commuter traffic areas and on high-speed corridors such as major interstates, highways and airways would provide a highly available network. Furthermore, the deployment of an 802.11 network to connect rural and commuter paths would ensure the seamless integration of wireless to fixed wired networks.

Current Solutions

Two high-speed mobile networks are available today. They are FLASH Orthogonal Frequency Division Multiplexing (OFDM) and iBurst, also known as 802.20 or Mobile Broadband Wireless Access (MBWA). Both provide mechanisms for seamless access. The data rate and maximum velocity [155-200mph] supported are adequate for most mobile users and has potential for application in the agricultural environment. For greater reliability and capacity, these high-velocity and high data rate networks will still rely on 802.11 LAN networks to propagate data traffic to the cellular and core networks.

WiMAX High-Speed Mobile Network

Most users have adopted the mobile version of WiMAX (802.16m) for low-velocity access. WiMAX operates at a maximum range of 31miles with a data rate of up to 1Gpbs at 1km. It uses scalable orthogonal frequency division multiple access (SOFDMA) encoding to achieve high data rates [2].

Flash-OFDM High-Speed Mobile Network

Flash-Orthogonal Frequency Division Multiplexing (F-OFDM) was adopted in Norwegian countries, and Ireland. However, most users have migrated to the mobile version of WiMAX (802.16m) for low-speed access [3].

Flash-OFDM	**Values**
Application	Mobile internet access solution
Encoding	Flash-OFDM Encoding
Downstream Data Rates (Mbps)	5.3, 10.6, 15.9
Upstream Data Rate (Mbps)	1.8, 3.6, 5.4
Operational Cell Radius	18 miles
Extended Cell Radius	34 miles
Maximum Mobility Speed	200 Miles/hour

Table 1 Flash-OFDM High-Speed Mobile Network

Mobile Broadband Wireless Access High-Speed Network

iBurst or Mobile Broadband Wireless Access (802.20) was more widely adopted than Flash-OFDM and marketed under the iBurst name. It is available in Eastern Europe, Ireland, South Africa, the US and Australia [3].

iBurst	**Values**
Application	Mobile internet access solution
IEEE Standard	802.20
Encoding	HC-SDMA/TDD/MIMO
Downstream (Mbps)	95
Upstream (Mbps)	36
Operational Cell Radius	2 miles
Extended Cell Radius	7 miles
Maximum Mobility Speed	155 Miles/hour

Table 2 Mobile Broadband Wireless Access High-Speed Network

II History of 802.11 Networks

The history of wireless computer networks has its origins in the University of Hawaii's ALOHA network. This first wireless computer network was developed under the leadership of Dr. Norman Abramson [4] in the 1970's. The ALOHA network was a natural fit for the Hawaiian Islands due to the difficulty of installing regular phone lines in a landscape full of mountains and active volcanoes [4].

Subsequently, in the 1980's, the Federal Communications Commission (FCC) announced the availability of experimental industrial, scientific and medical (ISM) bands available for commercial application of spread spectrum applications. During this time, several companies were able to deploy computer networks using proprietary wireless modems with data rates reaching ~1Mbps [4].

The 1990's saw the emergence of wireless standards bodies, such as the IEEE 802.11 Work Group, as it is known today. The individuals of the work groups were tasked with evaluating the different technologies and proposing standards to meet the commercial demands of hospitals, schools and stock exchanges [4].

802.11 Network Standards

The standardization and implementation of the 802.11[a, b, g, n, ac] networks are not sequential but instead are implemented based on adoption and necessity. For example, although 802.11a networks were defined early, adoption of 802.11b networks was more prevalent [5].

The following describes the current 802.11 standards:

802.11

- The original WLAN Standard. Supports 1 Mbps to 2 Mbps [4].
- This is the original definition of 802.11 networks
- Networks of this type are the precursor networks and experimental in nature.
- Adhoc networks were common.

802.11a

- High speed WLAN standard for 5 GHz band. Supports up to 54 Mbps.
- Networks with 802.11a support have the advantage being outside the more common 2.4GHz range and therefore outside the range of interference from nearby Bluetooth or microwave devices.
- 20 MHz channels
- 52 channels are available for use in the 5GHz range
- Common in corporate environments [6]

802.11b

- WLAN standard for 2.4 GHz band. Supports 11 Mbps.
- 802.11b was the first common implementation of Wireless LAN networks.
- 20 MHz channels
- Networks that support this standard have the advantage of support at increased distances relative to 802.11a networks.
- 11 channels are available for use in the 2.4GHz range
- Non-overlapping channels: 1, 6 and 11 [7]

802.11d

- International roaming – automatically configures devices to meet local RF regulations
- Defines the country specific requirements for channels available and channel hopping [8].

802.11e

- Addresses quality of service requirements for all IEEE WLAN radio interfaces
- Defines network medium access control using Hybrid Coordination Function which is a combination of Distributed Coordination Function (DCF) and Point Control Function (PCF) and for collision avoidance on the medium.
- Stations typically use the DCF for medium access (transmissions) [9]

802.11f

- Defines inter-access point communications to facilitate multiple vendor-distributed WLAN networks.
- Defines the Extended Service Set (ESS) for inter-access point communication.
- Commonly referred to as the network name which is broadcast for roaming purposes [10].

802.11g

- Establishes an additional modulation technique for 2.4 GHz band. Supports speeds up to 54 Mbps.
- Enhanced implementation of the 802.11b standard with use of Orthogonal Frequency-Division Multiplexing (OFDM) for throughput of 54Mbps
- Matches the speeds of an 802.11a network in the 2.4GHz band
- Supports 802.11b devices [11]

802.11h

- Defines the spectrum management of the 5 GHz band.
- Defines Dynamic Frequency Selection (DFS) and Transmit Power Control (TPC)
- Eliminates interference with satellites and radar in the 5GHz band
- Adopted for 802.11a networks [12]

802.11i

- Addresses the current security weaknesses for both authentication and encryption protocols. This standard encompasses 802.1X, TKIP and AES protocols.

- WiFi Protected Access (WPA) and WPA2 are the most common secure methods
- WPA2-PSK uses a pre-shared key (PSK) in its simplest implementation
- WPA2 with Temporal Key Integrity Protocol (TKIP) and Advanced Encryption Standard (AES) are used in corporate environments [13]
- Together with 802.1X defines use of Remote Authentication Dial-In User (RADIUS) server to provide centralized authentication, authorization and accounting (AAA) management instead of a built-in user database [14].

802.11n

- Provides higher throughput improvements. Intended to provide speeds up to 248 Mbps.
- Higher throughput with multiple antennas.
- Up to 4 transmit antennas and 4 receive antennas (4x4:4Streams)
- Up to four (4) streams of multiple input and multiple output (MIMO)
- Operates in the 2.4GHz and 5GHz range
- Uses 40MHz channels (equivalent of two 20MHz channels in 802.11a or 802.11g) [15]

802.11ac

- Uses 2.4 GHz and 5GHz bands to attain speeds up to 6.77Gbps with multiple antennas
- Higher throughput with multiple antennas
- Up to 4 transmit antennas and 4 receive antennas (4x4:8Streams)

- Up to eight (8) streams of multiple input and multiple output (MIMO)
- Operates in the 2.4GHz and 5GHz range
- Uses 160MHz channels (equivalent of four 20MHz channels in 802.11a or 802.11g) [16]

802.11 Wireless Networks

The following table summarizes the 802.11 wireless network standards and their date of adoption [17].

Protocol	Release Date	Op. Frequency	Throughput (Typical)	Data Rate (Max)	Range (Indoor)	Range (Outdoor)
Legacy	1997	2.4-2.5 GHz	0.7 Mbit/s	2 Mbps	~Varies	~75 m
802.11a	1999	5.15-5.25/5.25-5.35/5.725-5.875 GHz	23 Mbit/s	54 Mbps	~30 m	~100 m
802.11b	1999	2.4-2.5 GHz	4 Mbit/s	11 Mbps	~35 m	~110 m
802.11g	2003	2.4-2.5 GHz	19 Mbit/s	54 Mbps	~35 m	~110 m
802.11n	2008	2.4 GHz and/or 5 GHz	74 Mbit/s	248 Mbps = 2x2 ant	~70 m	~160 m
802.11ac	2015	2.4 GHz and 5GHz	433Mbit/s	433Mbps	~70m	~160 m

Table 3 802.11 Network Standards

Current 802.11ac Network Standard

The recent 802.11ac standard is suited for applications with streaming video or high throughput data connections. A summary of the proposed use of 802.11ac [16] standard is outlined in Table below.

Scenario	**Typical Client Form Factor**	**PHY Link Rate**	**Aggregate Capacity (Speed)**
1-antenna AP, 1-antenna STA, 80 MHz	Handheld	433 Mbit/s	433 Mbit/s
2-antenna AP, 2-antenna STA, 80 MHz	Tablet, Laptop	867 Mbit/s	867 Mbit/s
1-antenna AP, 1-antenna STA, 160 MHz	Handheld	867 Mbit/s	867 Mbit/s
2-antenna AP, 2-antenna STA, 160 MHz	Tablet, Laptop	1.69 Gbit/s	1.69 Gbit/s
4-antenna AP, four 1-antenna STAs, 160 MHz (MU-MIMO)	Handheld	867 Mbit/s to each STA	3.39 Gbit/s
8-antenna AP, 160 MHz (MU-MIMO) -- one 4-antenna STA -- one 2-antenna STA -- two 1-antenna STAs	Digital TV, Set-top Box, Tablet, Laptop, PC, Handheld	3.39 Gbit/s to 4-antenna STA 1.69 Gbit/s to 2-antenna STA 867 Mbit/s to each 1-antenna STA	6.77 Gbit/s
8-antenna AP, four 2-antenna STAs, 160 MHz (MU-MIMO)	Digital TV, Tablet, Laptop, PC	1.69 Gbit/s to each STA	6.77 Gbit/s

Table 4 802.11ac Network Standard

Source: http://en.wikipedia.org/wiki/IEEE_802.11ac [16]

III OPNET Modeler Simulation of Proposed Solution

In order to validate the proposed deployment of MIPv6 in an 802.11 network infrastructure, as a solution to future mobile wireless networks, OPNET Simulation software was used to model a roadside 802.11g network. The results obtained show that it is possible to deploy a roadside MIPv6 802.11g network, consisting of a MIPv6 Node, two MIPv6 access routers and a MIPV6 core, to serve the needs of high-bandwidth mobile users traveling up to ~70 miles per hour. Specifically, with OPNET Simulation software, 8-10Mbits/second data throughput was measured for file of 4MB-10MB on access routers 4kilometers apart.

Mobile IP Network Topology

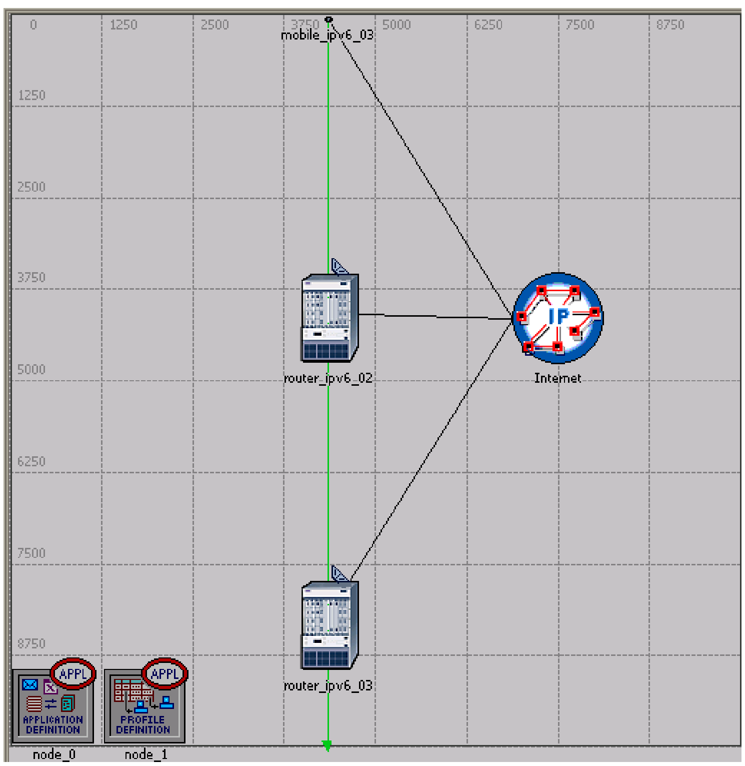

Figure 3 Network Simulation Model (10kilometers x 10kilometers)

The OPNET Modeler configuration in **Figure 3** shows the three access routers [**router_ipv6_01 (hidden)**, **router_ipv6_02**, **router_ipv6_03**] and the mobile node [**mobile_ipv6_03**] in the expanded view shown in

Figure 4 Mobile IPv6 Node Moving South (Green Line) and Server

[**server_ipv6_04**]. The mobile node [**mobile_ipv6_03**] moves south along the Green path and roams between Access routers [router**_ipv6_01**, **router_ipv6_02**, **router_ipv6_03**]. The server remains stationary and connected to the Access Router [**router_ipv6_01**].

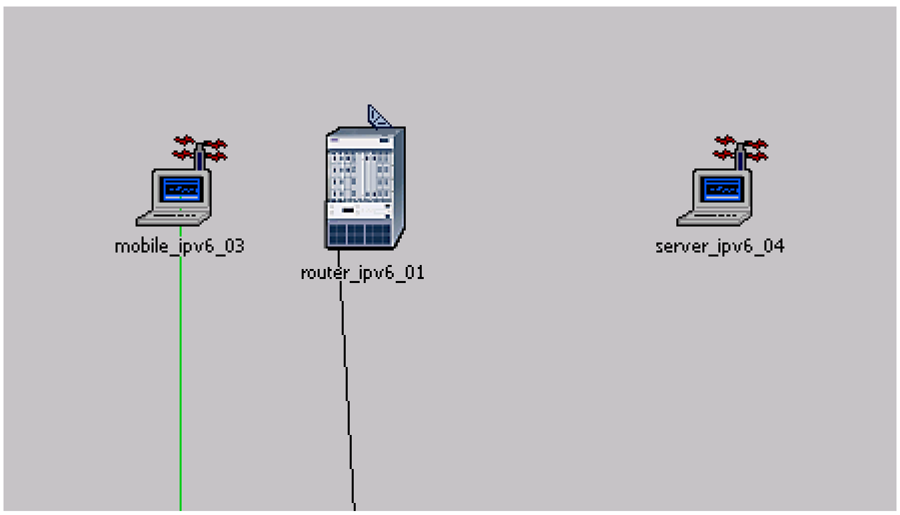

Figure 4 Mobile IPv6 Node Moving South (Green Line) and Server

OPNET Modeler Network Configuration

• Mobile Node Speed:	69.9 miles/hour (31.25m/s)
• Mobile Network Test Span:	10000 meters
• Mobile Network Simulation Duration:	15 minutes
• Mobile IP Version:	IPv6
• Mobile Node Delay (Stop) Time:	100 seconds

• Mobile Node Transit Time:	320 seconds
• Application Type:	File Transfer Protocol (FTP)
• Application Profile:	Serial, Unlimited Sessions
• Application Interval:	20 seconds per session
• WLAN Transmit Power (All Access Routers):	50mWatts
• WLAN Access Routers:	Fixed
• WLAN Total Access Routers:	Three (3)
• WLAN Access Routers Separation:	4100meters
• WLAN Encoding Method:	802.11g
• WLAN Frequency:	2.42 GHz
• Internet Cloud Links:	PPP over SONET OC3
• Internet Cloud Latency:	0 .0001 sec

OPNET Simulation Results for Highway Speeds

OPNET Mobile IPv6 Network Simulation Report

The results from the Mobile IPv6 network simulation using OPNET Modeler 11.0 are shown in Figure 5. The graph shows that as the Mobile IPv6 node moves due south at a fixed rate of 69.9 miles/hour it maintains the FTP sessions to the Mobile IPv6 server and is able to handle sessions of various duration. The graph below summarizes the throughput rate as a function of file size during the mobile node's movement.

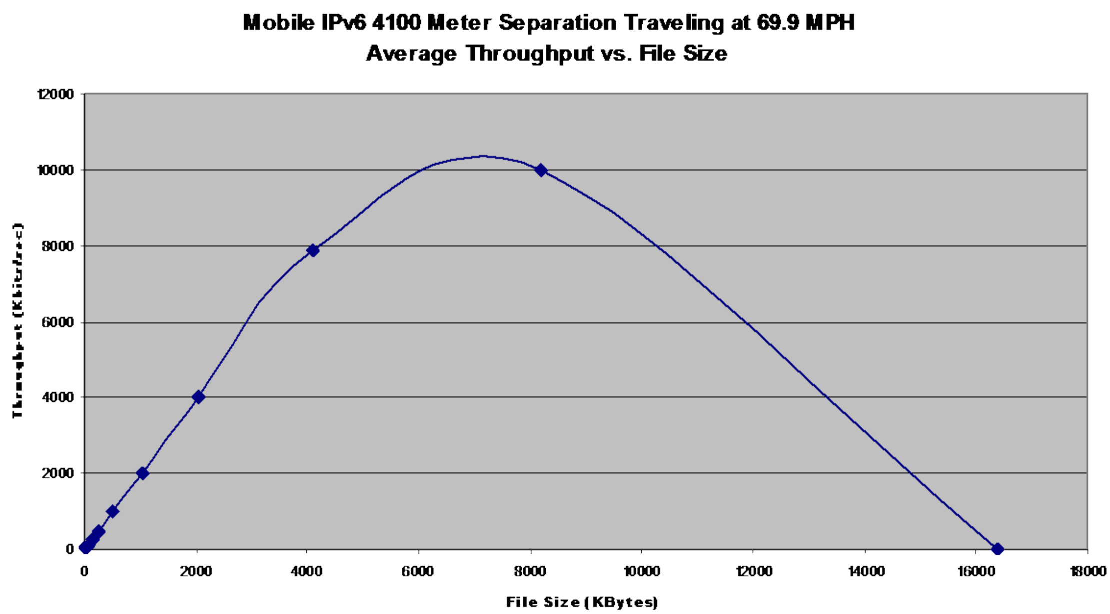

Figure 5 Mobile IPv6 Workstation Throughput (Kbps) vs File Size (Kbytes)

OPNET Modeler OSI Layers and Mobile IPv6 Comparison

The OPNET Modeler results show that small file size transfers of 64KB and below result in high overhead use due to the message headers at the different layers of the Open Systems Interconnect (OSI) or the Transmission Control Protocol (TCP)/Internet Protocol (IP) stack. Large file sizes result in retransmissions due to access point handover latency and therefore a lower throughput data rate.

Table 5 below summarizes the OSI and TCP/IP layers applicable to this thesis [18].

OSI Reference Model for FTP Application Using Mobile IPv6

OSI Layers for Mobile IPv6	**OSI Layers for Mobile IPv6 Application**
Layer 5-7 Session, Presentation, Application	File Transfer Protocol (FTP)
Layer 4 – Transport Layer	Transmission Control Protocol (TCP) V6 over TCPV4 Tunnels
Layer 3 - Network Layer	Internet Protocol (IP) V6 over IPV4 Tunnels
Layer 2 - Medium Access Control	802.11 Wireless LAN Standard
Layer 1 - Physical Medium	Air – Radio Frequency Encoding – Orthogonal Frequency Division Multiplexing (OFDM)

Table 5 OSI Reference Model for FTP Application Using Mobile IPv6

For this thesis the optimal file size for high throughput and low data loss was determined to be between 4-10MB. Minimum data loss and maximum throughput was measured with 7-8MB file sizes. The throughput for this range is considered to be ideal because the data file sizes are within the available capacity of 54 Mbps. File sizes in excess of 10MB result in TCP protocol data unit (PDU) segments not delivered on a roam from one access router to the next. A summary of the results from the OPNET Modeler follow in the table below.

OPNET Modeler FTP File Size vs Throughput Rate Comparison

File Size (KBytes)	**Throughput**	**Medium Utilization**	**Overhead**
< 64	Low	Low	Excessive Message headers
64-1000	Low	Low	Excessive Message headers
1000-4000	Medium	Moderate	Minimum overhead
4000-10000	High	Efficient	Minimum overhead
10000-16000	Low	Efficient	TCP Protocol Data Unit (PDU) Retransmissions

Table 6 FTP File Size vs Throughput Rate Comparison

OPNET Modeler Conclusions

All file size transfers up to 4MB would underutilize the available bandwidth. An FTP file size greater than 10MB would result in extended transmission time of more than five (5) seconds and the connection could span multiple Access Routers. The hand-off from one Access Router to another currently is about four (4) seconds. This brief transition time will result in a retransmission of several TCP packets. Improvements in hardware and software design and radio frequency optimization will minimize this hand-off time to be near wire speeds or better.

IV Mobile IPv6 Network

Mobile IPv6 Antennas

Mobile IPv6 Aluminum Foil Antenna

The Mobile IPv6 network prototype for this thesis uses antennas suitable for 802.11a/b/g standards. The "Aluminum Foil" antenna is one such low-gain antenna that was designed for a small scale environment to provide connectivity up to 10 meters. It has the following specifications:

- Aluminum foil wrapped on a folded aluminum foil pan
- Operates at: 2.4 GHz
- Application:
    1. 10 meter range
    2. Low Gain
    3. Access Router
- Ultra-miniature coaxial U.FL terminated cable connected to the mini-PCI radio card
- Received Power @20ft (6.09meters): -57dBm to -61dBm
- Received Power @~0ft (0.00meters): -34dBm
- Not calibrated or tuned
- Radiation pattern tests not available
- Replicated easily from any flattened aluminum material 8 inches by 7 inches

Figure 6 Mobile IPv6 Aluminum Foil Antenna

The typical use of this antenna would be an Access Router mounted inside or outside a small office, network/server room, vehicle or airplane. Users connected to the Access Router with the “Aluminum Foil” antenna would be temporary by design.

The Mobile IPv6 server and Mobile IPv6 Node in Figure 2 can be connected to the Access Router with this “Aluminum Foil” antenna in place of the fixed wired connection. The antenna serves as the wireless connection point to the Access Router and is intended for network administrators who may need temporary access to the Mobile IPv6 server to test Mobile IPv6 account authorization and initialization.

This antenna is intended to be low cost and readily available for use in an environment where the connection to the server is temporary in nature.

Mobile IPv6 Etenna Edgewave EE5801 Dual-Mode Antenna

The Etenna Edgewave 5801 (EE5801) dual-mode antenna was specified for the Mobile IPv6 network. It is an 802.11a/b/g/n/ac antenna intended for Mobile IPv6 nodes that roam from one Access Router to the next. The EE5801 has a low-profile form factor that fits inside the liquid crystal display (LCD) of a notebook computer. The EE5801 antenna requires a ground plane the size of the notebook LCD (9inches by 15 inches) display in order to operate according to the specifications outlined in the manufacturer datasheet. It has the following specifications:

- Etenna Edgewave EE5801 Mounted on an LCD with an aluminum foil ground plane
- Operates at: 2.4 GHz and 5 GHz
- Application:
    1. Very Low Profile
    2. 5dB Gain
    3. Access Routers
    4. Mobile IPv6 Server
    5. Mobile IPv6 Node
- Ultra-miniature coaxial U.FL terminated cable connected to the mini-PCI radio card
- Received Power @20ft (6.09meters): -62dBm
- Received Power @~0ft (0.00meters): -32dBm
- Easily replicated with the antenna mounted on an aluminum foil ground plane 15 inches by 9 inches

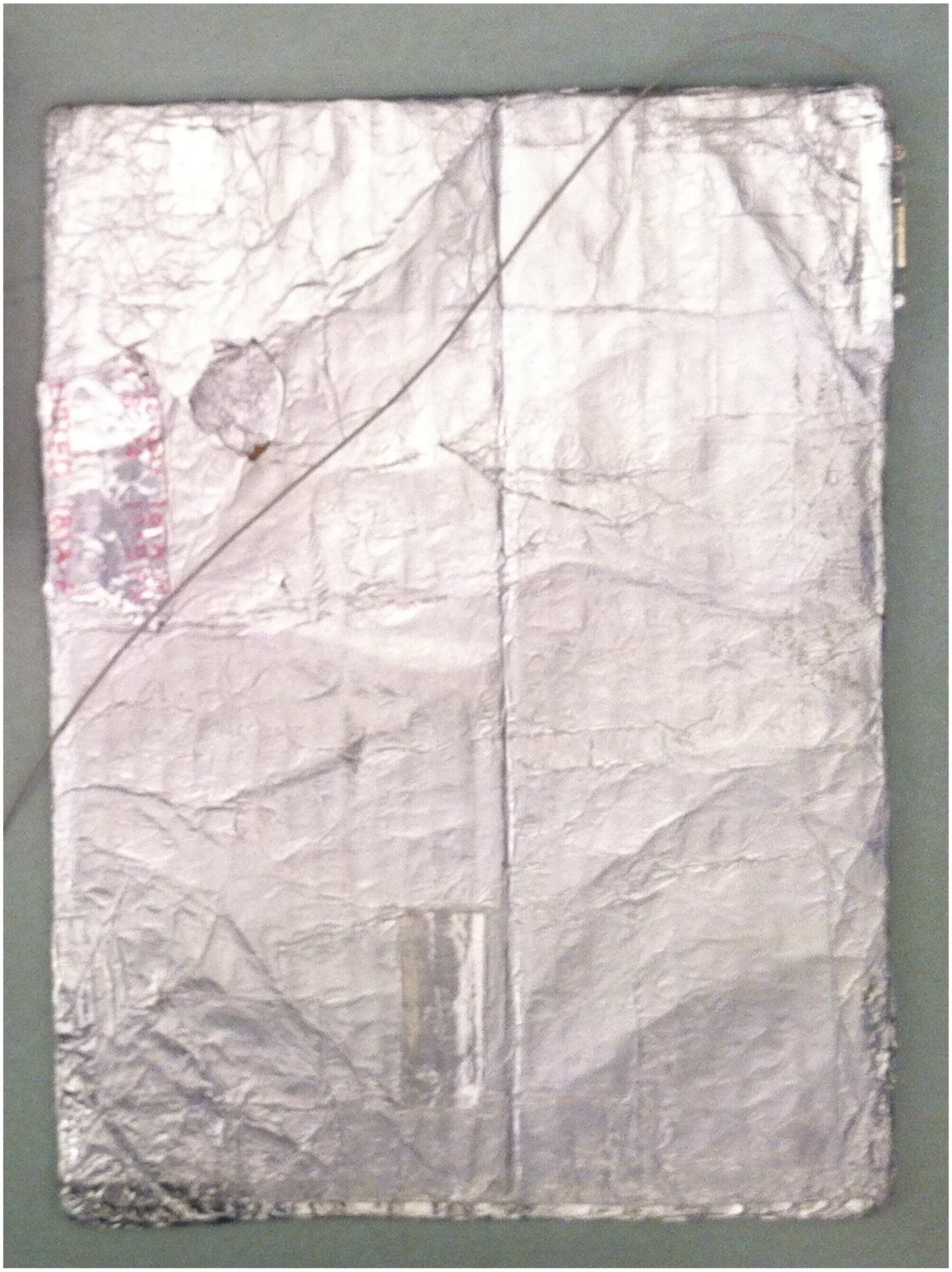

Figure 7 Mobile IPv6 Etenna Edgewave EE5801 Dual-Mode Antenna

The EE5801 dual-mode antenna, when properly tuned, provides a 5dB gain. The typical use of this antenna is servers, notebook computers, personal data assistants, Access Routers or other devices which require a small-profile high gain antenna for operation in the 2.4GHz or 5GHz range.

For this thesis, the Mobile IPv6 Nodes in Figure 2 (personal data assistant or the notebook computer mounted on the tractor) connect to the Access Router with this antenna. It is intended for access at distances of up to 110 meters. In the farm application in Figure 2, it serves as the gathering antenna for the sensors and high resolution images. The data is retrieved at intervals specified either in the Mobile IPv6 Server or immediately after the Mobile IPv6 Node connects (associates) to the Access Router.

Radiation Test Pattern for Etenna Edgewave EE5801 Dual Mode Antenna

The Etenna Edgewave 5801 (EE5801) dual-mode wireless data antenna was tested in the Anechoic Chamber at California Polytechnic (Cal Poly) San Luis Obispo during Winter 2005.

The Anechoic Chamber at Cal Poly is a three (3) meter chamber with a Scientific Atlanta controller for radiation pattern tests. It was built and installed by graduate students at Cal Poly San Luis Obispo. The main RF source is the Hewlett-Packard (HP) 8720C Vector Network Analyzer. The equipment is controlled by custom LabView software installed on a personal computer and written by graduate students at Cal Poly. The information below summarizes the radiation pattern test setup and the results for the EE5801 antenna [23].

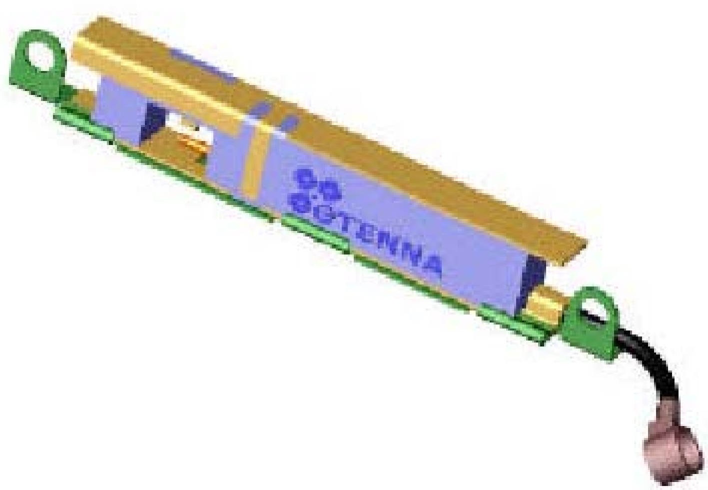

Figure 8 Etenna Edgewave EE5801 Dual-Mode Antenna

The process of verifying radiation pattern data is a process which may require assistance from a technical representative at the antenna manufacturer. In the case of the

EE5801 antenna radiation pattern tests, several exchanges of information was necessary to finally determine what the proper test fixture and ground plane was needed to match those in the datasheet [23].

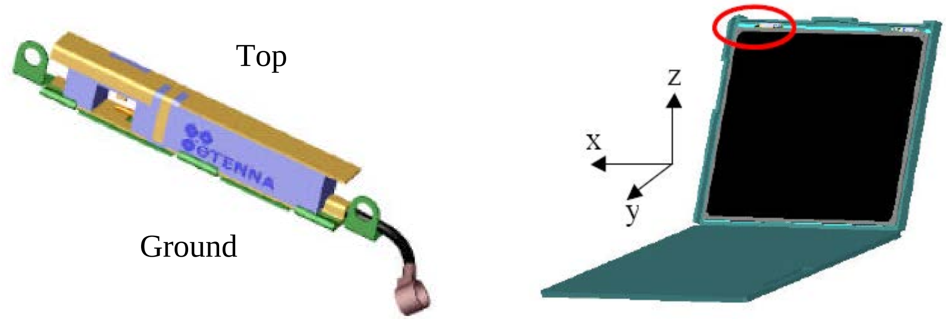

Figure 9 Etenna Edgewave EE5801 Dual-Mode Antenna Test Setup

Specifications

Frequency Range	2.4–2.48 and 4.9–5.9 GHz
Efficiency	50% min., 70% typical
Peak Gain	5.2 dBi nominal (2.4–2.48); +5 dBi (4.9–5.9)
-3 dB Efficiency Bandwidth	250 MHz (2.4–2.48); 1.4 GHz (4.9–5.9)
Typical 10 dB Return Loss Bandwidth	250 MHz (2.4–2.48); 1.5 GHz (4.9–5.9) nominal
Polarization	Linear
Power Handling	2 watt CW
Feed Point Impedance	50 ohms unbalanced
Temperature	-35°C to 85°C
Size (W x L x H)	3 mm x 30 mm x 5 mm
Weight	1.00 g max. (no bracket or cable)
Mounting	Screw, clipping or snap-on
Antenna Termination	Ultra-miniature coaxial connector (U.FL-LP-006 or equivalent)
Antenna Cable	Golden Bridge model number RF-MF5016
Shock	200 g, 3 ms, half size for all 6 axes (preliminary)
Vibration	Per IEC 68-2-6 Environmental Testing Part 2, Random Vibration

Table 7 Etenna Edgewave EE5801 Dual-Mode Antenna Specifications

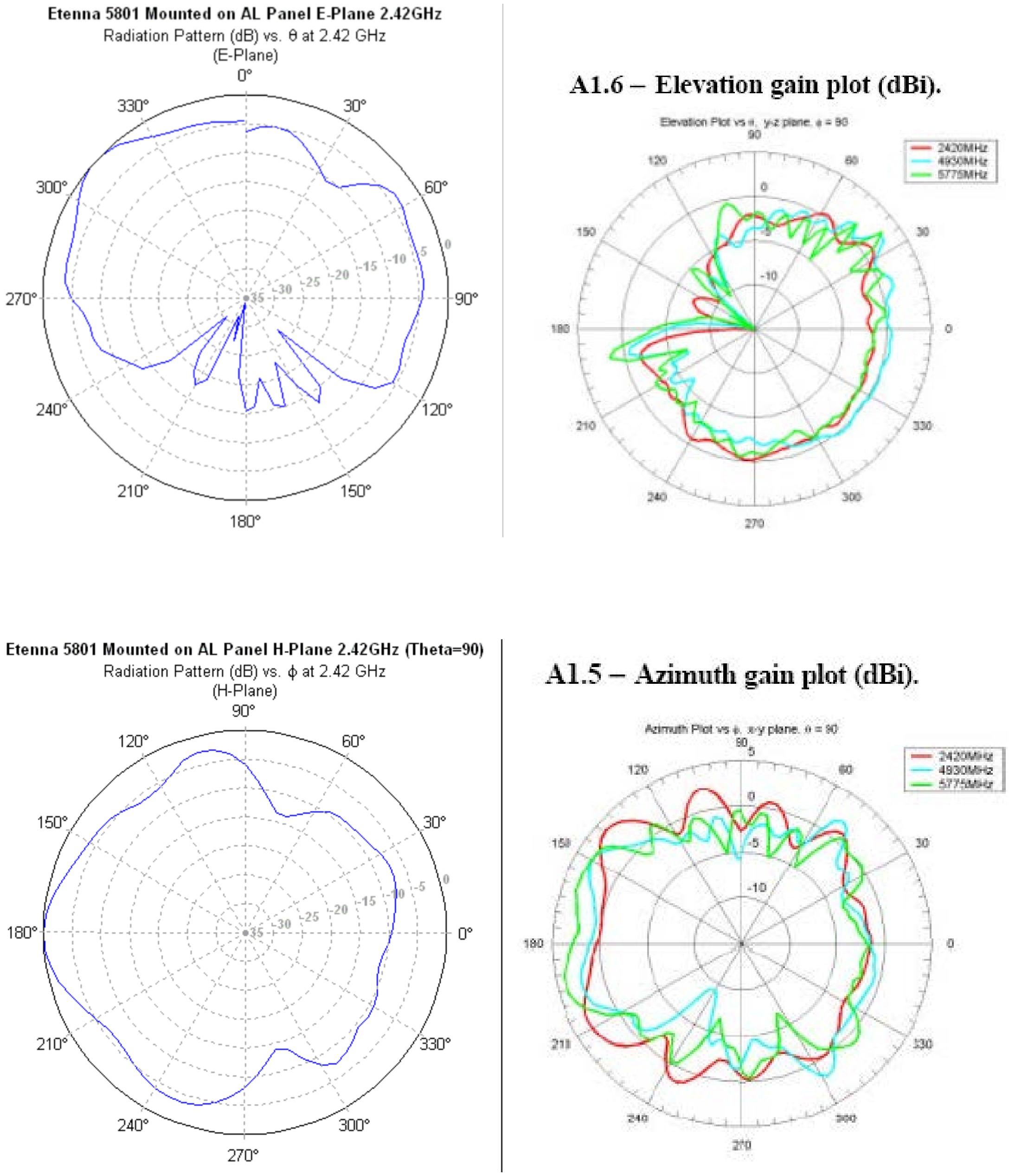

Figure 10 Etenna Edgewave EE5801 Dual-Mode Antenna Radiation Patterns

Mobile IPv6 Etenna Edgewave EE5801 Dual-Mode Antenna Summary

The radiation pattern test results at the frequency of 2.42GHz obtained from the Cal Poly Anechoic chamber match the general outline provided by the manufacturer. Other observations include:

- The plots for Cal Poly are in the opposite direction to those provided by the manufacturer. Changes to the Cal Poly Matlab files would reconcile these differences.
- The Cal Poly radiation pattern from 120 degrees to 180 degrees (240 to 180 on data sheet) needs further investigation because on this interval the antenna exhibits three nulls when compared to the data sheet plots.
- The results confirm that the antenna mounted on an aluminum ground plane replicates the laptop or notebook environment. This antenna and a rectangular ground plane could be deployed exactly as designed and tested to a server or access router.
- The Etenna Edgewave EE5801 radiation pattern test results obtained at the Anechoic Chamber at Cal Poly provide one more data point to fine tune and improve the design and implementation of antenna test and verification procedures.

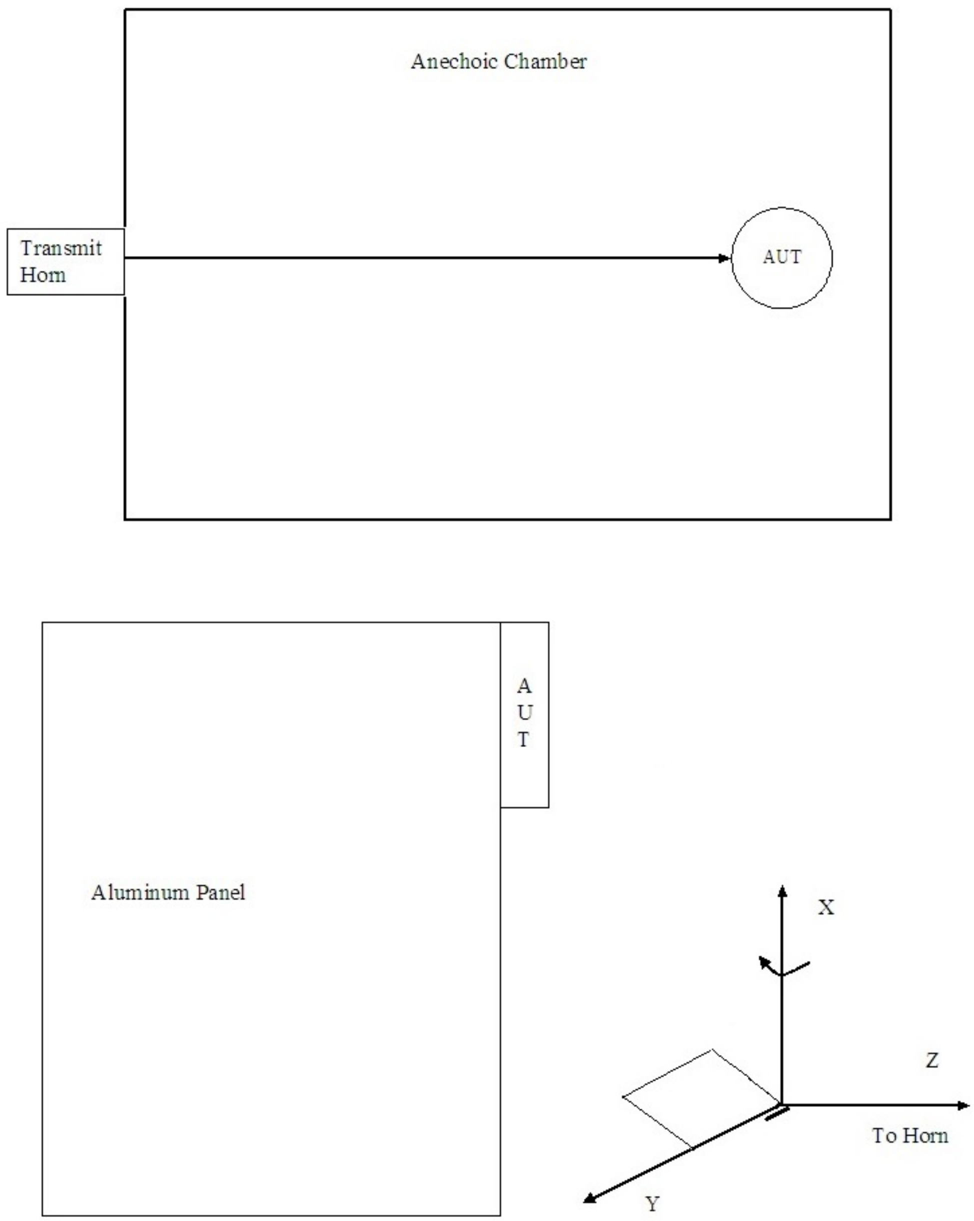

Figure 11 Anechoic Chamber Antenna Under Test (AUT)

Theoretical Calculations for Planar Waves

The power from the transmitter to the receiver is best estimated with a simplified view of the transmitter and receiver in the Z-plane as shown in Figure 12. The figure shows an incident wave traveling in the Z-plane from the transmitter to the receiver.

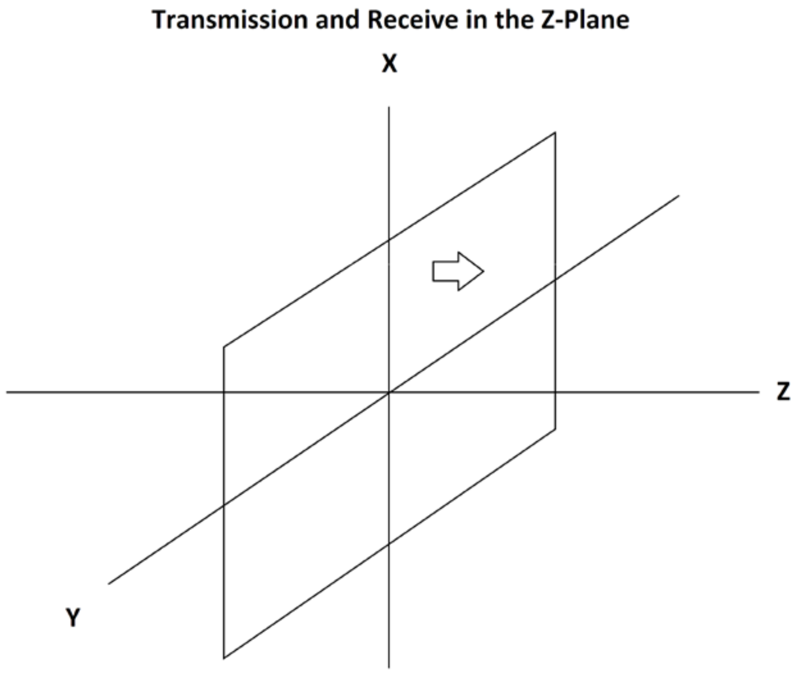

Figure 12 Planar Wave Transmit and Receive in the Z-Plane

Transmitted Power for Planar Wave

The transmitted power from the Access Routers is 19mWatts as measured from the radio driver.

The generic equation for transmitted or received power [19] over a given volume is:

$$\mathbf{Pt} = \int(\mathbf{E} \cdot \mathbf{J})dv \textbf{ Watts}$$

where **E** is the electric field intensity in Volts/meter and **J** is the current density in Amperes/meters square.

This reduces to the simplified equation for planar wave transmission in the Z-plane [19]:

$$Pt = \iiint (E \cdot J) dz dy dx \text{ Watts}$$

Given that **E** and **J** are specified as follows [19]:

$J = \sigma E$ and $E = T(E_o)\exp(-2\gamma z)$ A/m^2 where

$$\gamma = \alpha + j\beta$$

Where **T** is the transmission coefficient and γ is the propagation constant, α is the attenuation constant and β is the phase constant (wave number) [21].

For a good conductor [21]:

$\gamma = \alpha + j\beta = (1+j)\sqrt{\omega\mu\sigma/2}$ where $\sigma = \sigma_o = \omega\varepsilon_o$

Where $\omega = 2\pi f$ for frequency **f** and so,

$$\alpha = \sqrt{(2\pi f\mu\sigma/2)}$$

For free space, $\mu = \mu_o = 4\pi \cdot 10^{-7}$ **Henrys/meter** and

$\varepsilon = \varepsilon_o$ is $8.854 \cdot 10^{-12}$ **Farads/meter.**

For real power **Pt** evaluates as follows for a planar wave with x=0 to x=1 meter and y=0 to y=1 meter and for the change in z=0 to z=z1 [19]:

$$|Pt| = |\sigma E_o^2 T^2[\exp(-2\alpha z1)-1]/(4\alpha)| \text{ Watts}$$

Planar Wave Power Received

The power received **|Pr|** is given in units of dBm which is defined as follows [20]

$$\mathbf{dBm = 10 \cdot \log (Pr/1mW)}$$

$$\mathbf{|Pr| = 10^{[dBm/10]} \cdot 1mW \quad Watts}$$

Etenna Edgewave EE5801 Power Received

For the receiving non-isotropic Etenna Edgewave EE5801 antenna the following equation applies.

$|Pr| = (Gr \cdot Gt \cdot Pt)/(4\pi R^2)$ **Watts** where **Gr** is the Etenna EE5801 gain of 5.2. The distance in meters is measured at **R**. **Pt** is the power transmitted from the "Aluminum Foil" antenna. However, without the measured **Gt** value from the system starting from the amplifier stage to the "Aluminum Foil" Antenna gain plot for the 2.42 GHz frequency this equation cannot be readily utilized [22].

V Mobile IPv6 Network Prototype

Mobile IPv6 Contributions

To confirm the OPNET Modeler results, a prototype of the high-speed network for a rural farm application was specified with simulated data [i.e. sensor modules were not tested]. The following diagram outlines this network.

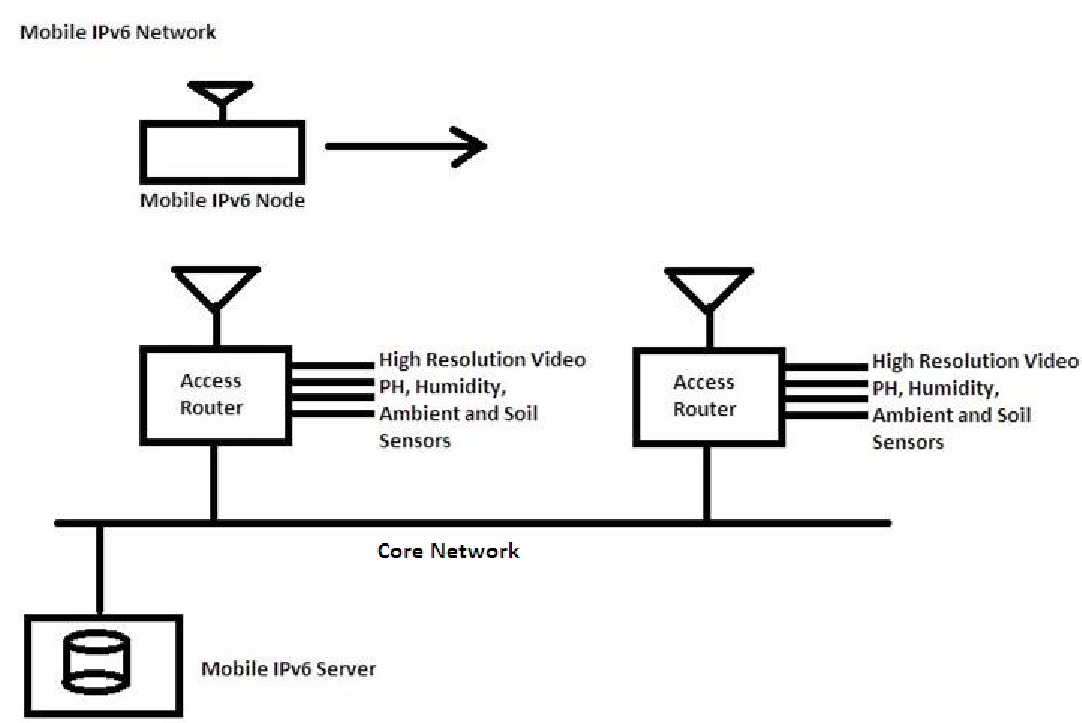

Figure 13 Mobile IPv6 Prototype Network

Figure 13 shows the two access routers which connects the Mobile IPv6 Node to the core network and provides access to the Mobile IPv6 Server. In the scenario above, the Mobile IPv6 Node roams from one Access Router to the next router when it reads a signal of -60dBm on its radio receiver. If a better signal is not present it remains with the current Access Router until it is out of range. Upon connection to either of the Access Routers the Mobile IPv6 Node retrieves the sensor data from the router's primary storage.

Mobile IPv6 Experimental Results

For this thesis and the Mobile IPv6 prototype network system of Figure 13, the measured and experimental data is tabulated below.

Mobile IPv6 Experimental Data	**Value**
Transmit Power	|Pt| = 19.0 mWatts (13dBm)
Received Power	@20 Feet |Pr| = .000631 μWatts (-62dBm)
Received Power	@~0 Feet |Pr| = .794328 μWatts (-31dBm)
Distance Between Transmitter and Receiver	20 Feet (6.09 Meters)
Electric Field	$\|E_o\|$ = 6.36 Volts/meter
Transmit Antenna	Aluminum Foil
Receive Antenna	Etenna Edgewave EE5801 mounted on Aluminum Foil Ground Plane with Gain=5.2 @2.42 Gigahertz (GHz)
Operating Frequency (f)	2.42 Gigahertz (GHz)
Operating Channel Bandwidth	20 Megahertz (MHz)
Operating Channel Numbers	Channel 6 (2.437GHz) and Channel 11 (2.462GHz)
Attenuation Constant (α) in free space	35.8636
Wave Number (β) in free space	35.8636
Impedance (η) in free space	377 Ω
Reflection Coefficient (Γ) in free space	0
Transmission Coefficient (T) in free space	1
Conductivity (σ) in free space	.1346278 Siemens/meter @2.42 Gigahertz (GHz)

Table 8 Mobile IPv6 Measured Experimental Data

The information for the experimental data in the table above can be summarized as follows:

- Transmit Power – The transmit power is the power output from the Access Router used to communicate with the associated Mobile IPv6 Nodes. This value meets the expected requirements for the antennas tested.
- Received Power – The received power is the power received at the Mobile Node from a distance of twenty (20) feet. This value meets the expected requirements for a Mobile IPv6 Node that communicates to the Mobile IPv6 Server via the Access Router.
- Distance between Transmitter and Receiver – The maximum distance between the transmitter and receiver is dependent on the type of antenna used. For this prototype network the distance of 20 feet met the expected transmit and receive data rates in the experimental network.
- Electric Field – The transmitted electric field value was calculated [19] from the Access Router's transmit power.
- Transmit Antenna – The Access Router's aluminum foil antenna serves as a substitute for a manufactured antenna. It meets the requirements as the source antenna for measurement purposes.
- Receive Antenna – The Access Router's Etenna Edgewave antenna mounted on an aluminum foil panel is a prototype for a notebook computer antenna. It also meets the Access Router's requirements as the receive antenna for measurement purposes.
- Operating Frequency – The carrier frequency specified for the Mobile IPv6 network for all nodes. For this prototype network the expected and measured value of 2.42 GHz met the requirements.

- Operating Channel Bandwidth – The specific channel width in the allotted channel space used to transmit and receive data within the carrier frequency. The bandwidth of 20MHz is the same for all supported Mobile IPv6 channels.
- Operating Channel Number – The channel numbers 6 and 11 were used to provide a network with minimal interference on a fast roam. The channel numbers 1, 6 and 11 are non-overlapping channels.
- Attenuation Constant (α), Wave Number (β), Impedance (η), Reflection Coefficient (Γ), Transmission Coefficient (T) and Conductivity (σ) are calculated values for free space. These values were not measured but are expected to be accurate for this experimental network.

Mobile IPv6 Sensor Data Structure

On a roam to an Access Router the Mobile IPv6 has a finite amount of time to read the data from the Access Router's primary storage. Therefore, to be efficient, data for each plant must have a fixed data structure. The following data structure is proposed for N number of plants:

Data Format for Sensor and High Resolution Image

Plant ID [32Bits]	**pH [8Bits]**	**Plant Temperature Celsius [8Bits]**	**Soil Temperature Celsius [8Bits]**	**Ambient Temperature Celsius [8Bits]**	**Soil Humidity Percent [8Bits]**	**High Resolution Image/Video [512MBytes]**
1	2.0	58	57	75	90	Plant#1 Image/Video
2	2.1	59	58	75	90	Plant#2 Image/Video
3	1.8	57	56	75	90	Plant#3 Image/Video
...	…	…	…	…	…	…
N=32	1.8	58	58	77	85	Plant#32 Image/Video

Table 9 Data Format for Sensor and High Resolution Image

The table above shows the per plant data structure (id, pH, plant temperature, soil temperature, ambient temperature, soil humidity, high resolution image/video). The "Plant ID" field is a bitmap of an individual plant, however, if this should require more than N plants (i.e. N=32) then this field can be re-interpreted as an enumerated value.

Mobile IPv6 System Software

The system components that are part of the Mobile IPv6 network are listed as follows:

- Mobile IPv6 Access Routers
- Mobile IPv6 Server
- Mobile IPv6 Node

Each of the above components has a number of software installations which are installed into the base system. The installed software modules were obtained from open source repositories. A description of each of the systems listed above follows.

Mobile IPv6 Access Routers

Mobile IPv6 Access Routers are the main entry points for the wireless to core network infrastructure. Each Access Router contains a base system and these additional software components. They are:

- Base System: Buildroot version 2
- Kernel version: Linux 2.6.11.12
- Mobile IPv6 Version 2.0
- Routing Advertising Daemon version 0.9
- Madwifi IEEE 802.11 Wireless LAN driver version madwifi-trunk-r1259-20051111
- Wireless Tools version 28

Two types of Access Routers are specified for this thesis, they are:

Router Type	**Description**
Access Router	Deployed in the field, it generally only has two (2) interfaces: a broadcast and land-based infrastructure/core network connection
Base Router	Deployed in the field and at the headquarters of the farm operation, it has three (3) interfaces: a broadcast, land-based infrastructure/core and satellite network connection

Table 10 Mobile IPv6 Access Routers

Access Routers interfaces are described as follows:

- Wireless 802.11 network interface for Mobile IPv6 Node access
- Land-based infrastructure/core network connections such as point-to-point Wireless 802.11, Ethernet, ATM, etc. is used for Binding update and acknowledgements, the Mobile IPv6 Server has access to the infrastructure/core network via this connection
- Satellite network connection for remote infrastructure access

A Mobile IPv6 Node associates to this system using the network name also known as the service set identifier (SSID). Upon association the Mobile IPv6 Node connects to the Mobile IPv6 Server via an IPv6 to IPv4 tunnel.

Mobile IPv6 Server

The Mobile IPv6 Server is the primary contact point for Mobile IPv6 Nodes. The purpose of this server is to receive Home Agent (HA) Binding updates (BU) from Mobile IPv6 Nodes that have left their home network. The Mobile IPv6 Server returns a Binding acknowledgement (BA) with the care-of-address for future correspondence to this Mobile IPv6 Node. Upon receipt of the Binding acknowledgement the Mobile IPv6 Node creates a secure tunnel to the Mobile IPv6 Server. The server also sends and receives router advertisements from the access routers. For this thesis, the Mobile IPv6 Server sends and receives Binding Updates and router advertisements on the Ethernet interface; it represents access to the core network.

The software components that are part of the base Mobile IPv6 Server are as follows:

- Base System: Slackware Distribution 7 version 10.2.0
- Kernel version: Linux 2.6.15
- Mobile IPv6 Version 2.0.1
- Intel ProWireless 2200 Driver for Linux v1.1.0
- Routing Advertising Daemon version 0.9
- IEEE 802.11 Wireless LAN driver version 1.1.13
- Wireless Tools version 29

Mobile IPv6 Node

The Mobile IPv6 Node for this thesis is a mobile workstation that functions as the primary data gathering system. In a fully deployed system Mobile IPv6 Node takes the form of an airplane, tractor and person with a tablet or PDA that roams from one Access

Router to the next in order to “harvest data” [see Figure 1 and Figure 2]. Upon association, it will trigger a read from the Access Router’s primary storage.

The software components that are part of the base Mobile IPv6 Node are as follows:

- Base System: Slackware Distribution 7 version 10.2.0
- Kernel version: Linux 2.6.15
- Mobile IPv6 Version 2.0.1
- Intel ProWireless 2200 Driver for Linux v1.1.0
- IEEE 802.11 Wireless LAN driver version 1.1.13
- Wireless Tools version 29
- Xine Multimedia Engine version 0.99.4

Mobile IPv6 Handoff

A Mobile IPv6 handoff sequence is best described as a set of steps that the Mobile IPv6 Node must pass through. Once associated to an Access Router, a handoff occurs only when the Mobile IPv6 Node has a deteriorating signal to the current Access Router.

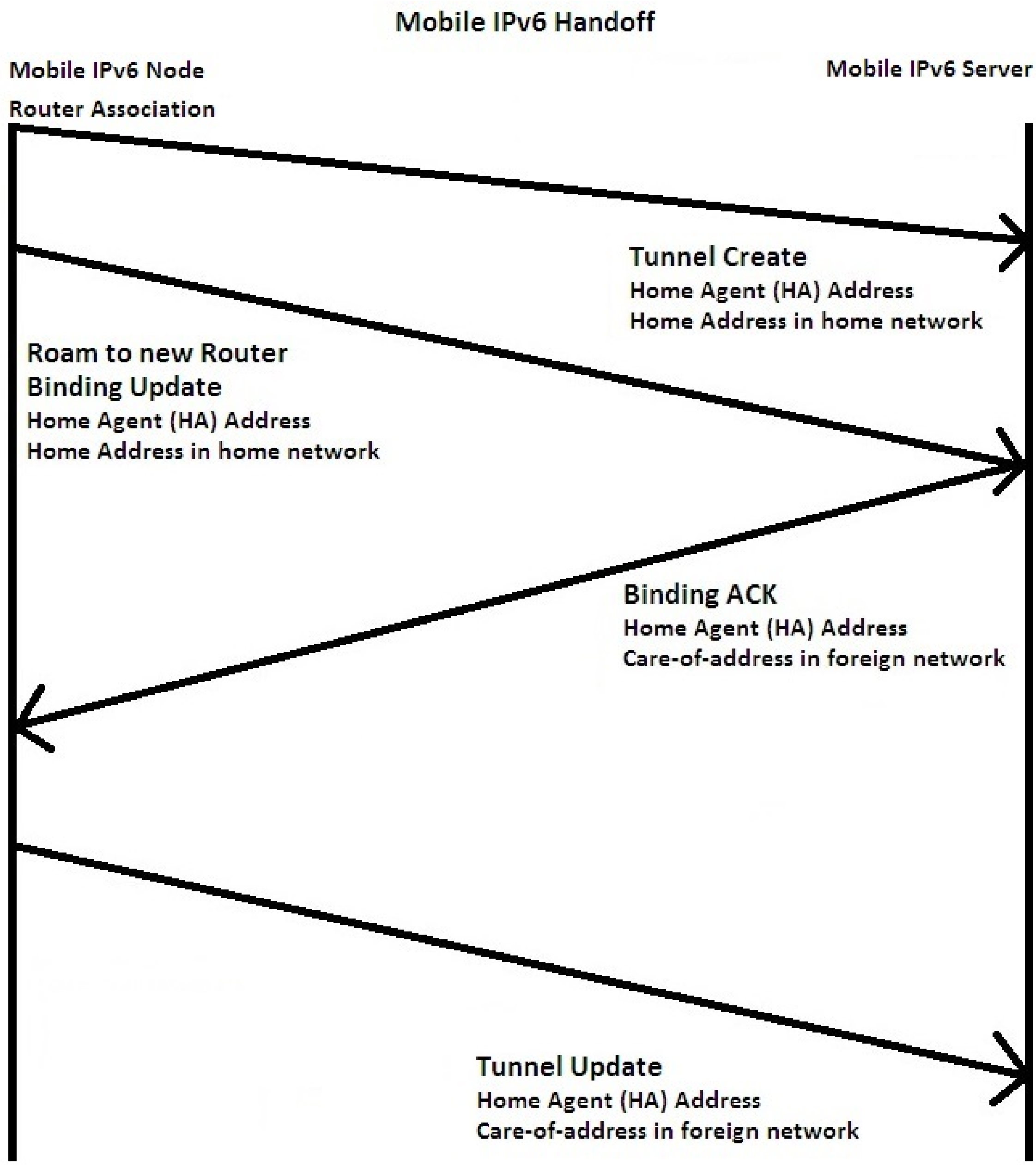

Figure 14 Mobile IPv6 Node Handoff on a Roam to a new Access Router

The time lapse diagram in the figure above shows a typical handoff from one Access Router to the next.

Router Association

The initial step is the "Router Association". During the router association, the Mobile IPv6 Node must authenticate to the Access Router using the service set (SSID) and any other credentials. For this prototype network, the following SSID and credentials were applied:

Service Set ID/Network Name:	**mipasn**
Credentials Type/Encoding:	**N/A**
Medium Access Control (MAC) Authentication:	**Access Control Lists (ACLs)**

As a security enhancement, beacons are disabled. Instead 802.11 probe requests and probe responses serve to gauge the signal strength of all the Access Routers that respond. Also, the main authentication mechanism is the MAC address on the Mobile IPv6 Node.

Tunnel Create

After the Access Router adopts the Mobile IPv6 Node into the foreign network the Mobile IPv6 Node creates an IPv6 over IPv4 tunnel to the Access Router using the router configured address. This tunnel serves to add or remove Access Routers advertised while it is in the foreign network. This tunnel is removed when the Mobile IPv6 Node roams to a new router at which point it creates a new tunnel.

If IP Security (IPSec) is enabled and a Home Agent (HA) is configured, this router tunnel is migrated in the "**Tunnel Update**" step to an IPSec tunnel to the Home Agent.

Roam to New Router

Roam Threshold: -60dBm

Scan Threshold: -60dBm

Measured Transition Time: 4 seconds

As soon as the radio connection goes 10dBm below the threshold of -60dBm the Mobile IPv6 workstation starts to scan the nearby Access Routers. To find the next available Access Router, the Mobile IPv6 Node will use an average of three radio received signal strength indication (RSSI) values to determine if the Access Router is suitable for association at its current position.

Binding Update

After the Mobile IPv6 Node associates to the new Access Router it sends a Binding Update to the Mobile IPv6 Server over the newly created tunnel. The Binding Update serves to notify the server who it's Home Agent (HA) is and the current address configured on its interface.

Binding Acknowledgement

Upon receipt of a Binding Update request from the Mobile IPv6 Node, the server replies with a Binding Acknowledgement. The Binding Acknowledgement confirms that the specified care-of-address and the Home Agent address (on the Mobile IPv6 Server) are valid.

Tunnel Update

After the binding update and acknowledgement are complete, a Tunnel Update may occur if IP Security (IPSec) is enabled. With IPSec enabled the original router re-configuration tunnel is replaced with this secure tunnel to the Home Agent on the Mobile IPv6 Server.

VI Mobile IPv6 Software Enhancements

Access Router Enhancements

IEEE 802.11 Driver Changes

Hidden Network Name

Two enhancements were made to the Access Router's 802.11 driver. The first enhancement improves security by hiding the network name from any potential devices that attempt to access any available network.

```
File: ieee80211.c
#define MIPASN 1
#ifdef MIPASN
vap->iv_flags |= IEEE80211_F_HIDESSID;
#endif /*MIPASN*/
```

Figure 15 Access Router Hidden Network Name

Access Control Lists

The second enhancement enhances security and facilitates access to the sensor data and images authentication so that it is efficient. The approach used in this network is inline Medium Access Control (MAC) address access control lists (ACLs). ACLs filter out unwanted traffic so that only those sensors and stations configured into the network will be able to associate to the Access Router. The following code sample defines the allowed MAC addresses for this prototype network:

```
File: ieee80211_input.c
#define MIPASN 1
#ifdef MIPASN
const char  mipasn_ha[] = {0x00, 0x12, 0xf0, 0x47, 0xf2, 0xd3};
const char  mipasn_hp[] = {0x00, 0x12, 0xf0, 0x47, 0x18, 0xb5};
const char  mipasn_hn[] = {0x00, 0x15, 0x00, 0x57, 0x45, 0x22};
const char  mipasn_mn[] = {0x00, 0x15, 0x00, 0x57, 0xc2, 0xa9};
#endif /*MIPASN*/

#ifdef MIPASN
        if (!IEEE80211_ADDR_EQ(mipasn_ha, wh->i_addr2) &&
                !IEEE80211_ADDR_EQ(mipasn_hp, wh->i_addr2) &&
                !IEEE80211_ADDR_EQ(mipasn_hn, wh->i_addr2) &&
                !IEEE80211_ADDR_EQ(mipasn_mn, wh->i_addr2))
        {
                goto out;  /* Reject Mobile Node/Sensors */
        }
#endif /* MIPASN */
```

Figure 16 Access Router ACL Enhancement

This code sample specifies that only traffic from MAC addresses “mipasn_ha”, “mipasn_hp”, “mipasn_hn” and “mipasn_mn” are accepted as valid traffic. All other traffic is rejected. This inline code, although secure and efficient does not scale for a high number of Mobile Nodes and sensors. To scale for a high number of MAC addresses, the ACL list would have to be read in at initialization.

To prevent MAC address spoofing, factory radio hardware with pre-programmed MAC addresses, hardware encrypted radios and trusted signed radio software and drivers can be used. A device can still eavesdrop on the connection but without the hardware and software decryption keys it will be unable to decipher any of the intercepted content.

Mobile Node and Server Enhancements

IEEE 802.11 Driver Changes

Hidden Network Name

This enhancement complements the Access Router's hidden network name for probe requests and responses.

```
#define MIPASN 1
#ifdef MIPASN
const char  mipasn_ap_hr[] = {0x00, 0x90, 0x4b, 0xcc, 0x74, 0x55};
const char  mipasn_ap_ar[] = {0x00, 0x90, 0x4b, 0xcc, 0x73, 0xd5};

#endif /*MIPASN*/
#ifndef MIPASN
#else
        if(!memcmp(network->bssid, mipasn_ap_hr, ETH_ALEN))
        {
                network->ssid_len = 6;
                memcpy(network->ssid, "mipasn", 6);
                network->ssid[network->ssid_len] = 0;
        }
        else if(!memcmp(network->bssid, mipasn_ap_ar, ETH_ALEN))
        {
                network->ssid_len = 6;
                memcpy(network->ssid, "mipasn", 6);
                network->ssid[network->ssid_len] = 0;
        } else {
                /* AP is not on the allowed list */
                return 1;
        }
#endif /*MIPASN*/
```

Figure 17 Mobile Node and Server ACL and Network Name Enhancements

The code to restore the hidden network name involves a check of the access control list. If this traffic originates from an allowed Access Router then the network name is restored.

Intel 802.11b/g Radio Driver Enhancements

The Intel 802.11b/g radio driver was enhanced with ACLs and a signal strength history for each Access Router after a probe request and response. The following code shows both enhancements.

```
#define MIPASN 1
#ifdef MIPASN
#define MIPASN_RSSI_ROAM_THRESH_MIN 10
#define MIPASN_RSSI_ROAM_THRESH_MAX 10
#define MIPASN_RSSI_OPTIMUM_SIG_VAL -60 /* dBm */
#define MIPASN_RSSI_FILTER 3
const char  mipasn_ap_hr[] = {0x00, 0x90, 0x4b, 0xcc, 0x74, 0x55};
const char  mipasn_ap_hp[] = {0x00, 0x12, 0xf0, 0x9d, 0x18, 0xb5};
const char  mipasn_ap_hn[] = {0x00, 0x15, 0x00, 0x3a, 0x45, 0x22};
const char  mipasn_ap_mn[] = {0x00, 0x15, 0x00, 0x23, 0xc2, 0xa9};
const char  mipasn_ap_ar[] = {0x00, 0x90, 0x4b, 0xcc, 0x73, 0xd5};
static int  hr_index;
static int  mipasn_ap_hr_avg_rssi = 0xffffffff;
static int  mipasn_ap_hr_rssi[]   = {0, 0, 0};
static int  ar_index;
static int  mipasn_ap_ar_avg_rssi = 0;
static int  mipasn_ap_ar_rssi[]   = {0, 0, 0};
#endif /*MIPASN*/

#ifdef MIPASN
#else
       if (match->network && !memcmp(network->bssid, mipasn_ap_hr, ETH_ALEN) &&
               (average_value(&priv->average_rssi) + MIPASN_RSSI_ROAM_THRESH_MAX) >
mipasn_ap_hr_avg_rssi) {
```

```
		return 0;
	}
	else
	if (match->network && !memcmp(network->bssid, mipasn_ap_ar, ETH_ALEN) &&
		(average_value(&priv->average_rssi) + MIPASN_RSSI_ROAM_THRESH_MAX) >
mipasn_ap_ar_avg_rssi) {
		return 0;
	}
#endif /* MIPASN */

#ifdef MIPASN
	if (!memcmp(header->addr3, mipasn_ap_hr, ETH_ALEN))
	{
		mipasn_ap_hr_rssi[hr_index] = stats.rssi;
		mipasn_ap_hr_avg_rssi =
(mipasn_ap_hr_rssi[0]+mipasn_ap_hr_rssi[1]+mipasn_ap_hr_rssi[2])/MIPASN_RSSI_FILTER;
		if (hr_index >= MIPASN_RSSI_FILTER)
		{
			hr_index = 0;
		}
		else
		{
			hr_index++;
		}
	} else
	if (!memcmp(header->addr3, mipasn_ap_ar, ETH_ALEN))
	{
		mipasn_ap_ar_rssi[ar_index] = stats.rssi;
		mipasn_ap_ar_avg_rssi =
(mipasn_ap_ar_rssi[0]+mipasn_ap_ar_rssi[1]+mipasn_ap_ar_rssi[2])/MIPASN_RSSI_FILTER;

		if (ar_index >= MIPASN_RSSI_FILTER)
		{
			ar_index = 0;
		}
		else
```

```
		{
			ar_index++;
		}
	}
	else
	{
		/* Continue */
	}

	if (((average_value(&priv->average_rssi) + MIPASN_RSSI_ROAM_THRESH_MIN) <
MIPASN_RSSI_OPTIMUM_SIG_VAL))
	{
		if ((priv->assoc_network != NULL) && (priv->ieee != NULL))
		{
			struct ieee80211_network *network = NULL;
			unsigned long flags;
			struct ipw_network_match match = {
				.network = priv->assoc_network
			};

			spin_lock_irqsave(&priv->ieee->lock, flags);
			list_for_each_entry(network, &priv->ieee->network_list, list) {
				if (network != priv->assoc_network)
				{
					ipw_best_network(priv, &match, network, 0);
				}
			}
			spin_unlock_irqrestore(&priv->ieee->lock, flags);

			if (match.network == priv->assoc_network)
			{
				/* Clear the scanning status bits */
				queue_work(priv->workqueue, &priv->disassociate);
				queue_work(priv->workqueue, &priv->request_scan);
				queue_work(priv->workqueue, &priv->associate);
			}
```

```
                        else
                        if (match.network != NULL)
                        {
                                if (priv->status & (STATUS_ASSOCIATED))
                                {
                                        ipw_link_down(priv);

                                        priv->status &= ~(STATUS_ASSOCIATING |
STATUS_ASSOCIATED);
                                        priv->assoc_network = match.network;
                                        /* Set up AP to this network */
                                        ipw_compatible_rates(priv, priv->assoc_network,
&match.rates);
                                        queue_work(priv->workqueue, &priv->associate);
                                        break;
                                }
                        }
                }
        }
#endif /* MIPASN */
```

Figure 18 Mobile Node and Server 802.11b/g ACL and RSSI History Enhancement

VII MIPv6 System Performance

MIPv6 File Transfer Sizes

To find a reference point for the effectiveness of the High-Speed Mobile IPv6 Network, the IPERF application was used to measure throughput with different file sizes. High-Speed Mobile velocity conditions were not measured in this thesis. The measurements below are for an indoor application.

MIPv6 Data Rates Near Access Router on 802.11g Channel 6

Data rates were measured for the following conditions on the Mobile IPv6 Network prototype:

Wireless Standard	802.11g
Wireless Channel:	Channel 6 (2.437 GHz)
Access Router Instance:	1
Access Router Antenna:	Aluminum Foil
Received Signal Indication:	-42 dBm
Configuration:	Access Router #1 and Mobile IPv6 Node
Separation Distance:	Two (2) feet apart

The data tabulated for file sizes: 49901424 (49.9MBytes), 9671806 (9.7MBytes), 5119222 (5.1MBytes) and 1763847 (1.8MBytes) follows.

Channel 6 Data Rate for a 49.9MByte File Size

Test Run	Duration (Seconds)	Data Rate (Mbps)
1	19.5	20.5
2	18.9	21.1
3	19.4	20.5
4	18.9	21.1
5	19.0	21.0
6	19.0	21.1
7	18.8	21.2
8	19.1	20.9
9	19.0	21.0
10	19.2	20.8

Table 11 Channel 6 Data Rate for a 49.9MByte File Size

Channel 6 Data Rate for a 9.7MByte File Size

Test Run	Duration (Seconds)	Data Rate (Mbps)
1	4.3	18.1
2	4.1	19.0
3	3.7	20.8
4	3.7	21.2
5	3.8	20.6
6	3.7	21.2
7	3.7	21.2
8	3.7	21.2
9	3.6	21.4
10	3.7	21.1

Table 12 Channel 6 Data Rate for a 9.7 MByte File Size

Channel 6 Data Rate for a 5.1MByte File Size

Test Run	Duration (Seconds)	Data Rate (Mbps)
1	2.4	17.4
2	2.0	21.3
3	2.0	21.3
4	1.9	21.5
5	2.0	21.3
6	2.0	21.2
7	2.0	20.7
8	2.0	21.3
9	2.0	21.2
10	2.0	21.3

Table 13 Channel 6 Data Rate for a 5.1 MByte File Size

Channel 6 Data Rate for a 1.7 MByte File Size

Test Run	Duration (Seconds)	Data Rate (Mbps)
1	1.2	12.5
2	0.7	21.4
3	0.7	21.2
4	0.7	22.0
5	0.7	21.2
6	0.7	21.0
7	0.7	20.4
8	0.7	21.0
9	0.7	20.4
10	1.0	15.4

Table 14 Channel 6 Data Rate for a 1.7 MByte File Size

MIPv6 Data Rates Near Access Router on 802.11g Channel 11

Data rates were measured for the following conditions on the Mobile IPv6 Network prototype:

Wireless Standard:	802.11g
Wireless Channel:	Channel 11 (2.462 GHz)
Access Router Instance:	2
Access Router Antenna:	Etenna EE5801 on an Aluminum Panel
Received Signal Indication:	-20 dBm
Configuration:	Access Router #2 and Mobile IPv6 Node
Separation Distance:	0 feet apart

The data tabulated for file sizes: 49901424 (49.9MBytes), 9671806 (9.7MBytes), 5119222 (5.1MBytes) and 1763847 (1.8MBytes) follows.

Channel 11 Data Rate for a 49.9MByte File Size

Test Run	Duration (Seconds)	Data Rate (Mbps)
1	20.1	19.8
2	20.1	19.9
3	20.1	19.9
4	Dropped Connection	Dropped Connection
5	19.6	20.3
6	19.8	20.2
7	Dropped Connection	Dropped Connection
8	20.1	19.9
9	19.8	20.1
10	19.8	20.2

Table 15 Channel 11 Data Rate for a 49.9MByte File Size

Note that the “Dropped Connetions” as indicated in Table 15 above were caused by collisions from other network traffic on the same channel.

Channel 11 Data Rate for a 9.7MByte File Size

Test Run	Duration (Seconds)	Data Rate (Mbps)
1	3.9	19.9
2	3.7	21.0
3	3.9	19.9
4	3.7	20.8
5	4.1	18.7
6	3.7	20.9
7	4.2	18.6
8	3.7	21.0
9	3.9	19.7
10	3.9	19.9

Table 16 Channel 11 Data Rate for a 9.7 MByte File Size

Channel 11 Data Rate for a 5.1MByte File Size

Test Run	Duration (Seconds)	Data Rate (Mbps)
1	2.0	20.6
2	2.0	21.1
3	2.0	21.4
4	2.0	21.1
5	2.0	21.2
6	2.0	21.1
7	2.0	21.3
8	2.0	21.3
9	2.0	21.1
10	2.0	21.1

Table 17 Channel 11 Data Rate for a 5.1 MByte File Size

Channel 11 Data Rate for a 1.7 MByte File Size

Test Run	**Duration (Seconds)**	**Data Rate (Mbps)**
1	0.7	21.4
2	0.7	21.6
3	0.7	21.4
4	0.7	21.8
5	0.7	22.0
6	0.7	21.5
7	0.9	15.9
8	0.7	21.7
9	0.7	21.8
10	0.7	21.3

Table 18 Channel 11 Data Rate for a 1.7 MByte File Size

MIPv6 Measured Data Rates Conclusion

In conclusion the data rates measured with IPERF match almost identically to the typical data of 19Mbit/s specified by the 802.11g standard. Refer to Table 3 802.11 Network Standards for more information.

The system performance of the Mobile IPv6 prototype network under static conditions for different file sizes is summarized in the table below.

File Size (Bytes)	Duration (Seconds)	Data Rate (Mbps)
49901424	21.3	18.7
9671806	4.1	18.9
5119222	2.6	16.3
1763847	1.2	12.5

Table 19 Mobile IPv6 Network FTP Transfer Data Rate for Different File Sizes

The results show that under a High-Speed mobile environment only the file sizes of 9671806, 5119222 and 1763847 would transfer successfully within the 4 seconds allowed before handoff would occur.

To measure the power requirements during a file transfer, a file of size of 49.9MBytes was used under non-mobile conditions. The results are tabulated below.

The results show a twenty second (20) average transfer rate would span multiple nodes under mobile conditions. For this experiment only non-mobile conditions were used for data rate measurements.

VIII MIPv6 Hardware

The hardware platforms for the Access Routers use the MontaVista Board Support package whose architecture is outlined below:

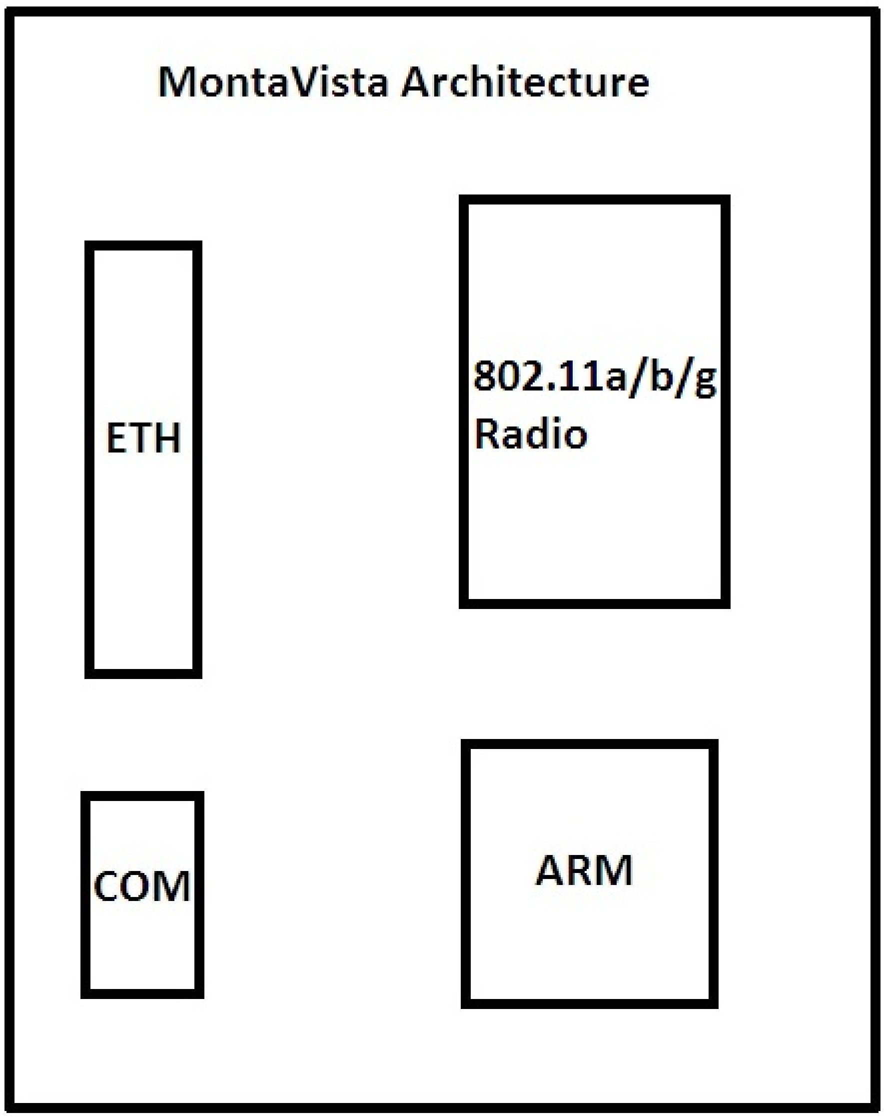

Figure 19 MontaVista Board Support Architecture

The MontaVista board is a development board with support for the following:

- ARM Processor

- Five Port Ethernet
- Serial COM1 Port
- Two (2) Mini-PCI Slots
- Replaceable Boot Flash
- USB Port

This board was chosen because it was made available by Hewlett-Packard. Another possible choice was the Linksys WRT54G wireless router. The Linksys WRT54G router was not chosen because it is not debug-ready for development purposes. To make it debug ready requires serial port access. The MontaVista board, in addition to being debug ready, has the added advantage of having telephone, USB, Ethernet and mini-PCI connection points. Another very beneficial and useful feature is that the boot flash is not soldered on but instead is a replaceable part. If the boot flash should fail or is inadvertently erased it is easily replaced. Otherwise, soldering is required to remove and restore the part and/or the use of an external flash programming tool.

Access Router #1

Access Router: MontaVista Board Support with ARM Processor

Radio & Antenna: Atheros Radio 802.11a/b/g with aluminum foil antenna

Figure 20 Mobile IPv6 Access Router #1

Figure 20 shows Mobile IPv6 Access Router #1 with an aluminum foil antenna attached to an ultra-miniature coaxial U.FL terminated cable and connected to the mini-PCI radio card. The ribbon cable with the DB9 connector at the end enables serial port terminal access to the system. The five (5) port Ethernet 10/100 Mbps metal strip at the bottom edge of the router serves as the core network for this prototype instead of the previously referenced Access Router Point-to-Point directive antenna in Figure 2. A substitute network using directive point-to-point antennas is known as a wireless distribution system (WDS) and it enables wireless mesh networks. This substitute WDS network was not tested for this thesis.

Access Router #2

Access Router: MontaVista Board Support with ARM Processor

Radio & Antenna: Atheros Radio 802.11a/b/g with ETENNA EdgeWave EE5801 dual-mode antenna on aluminum ground plane

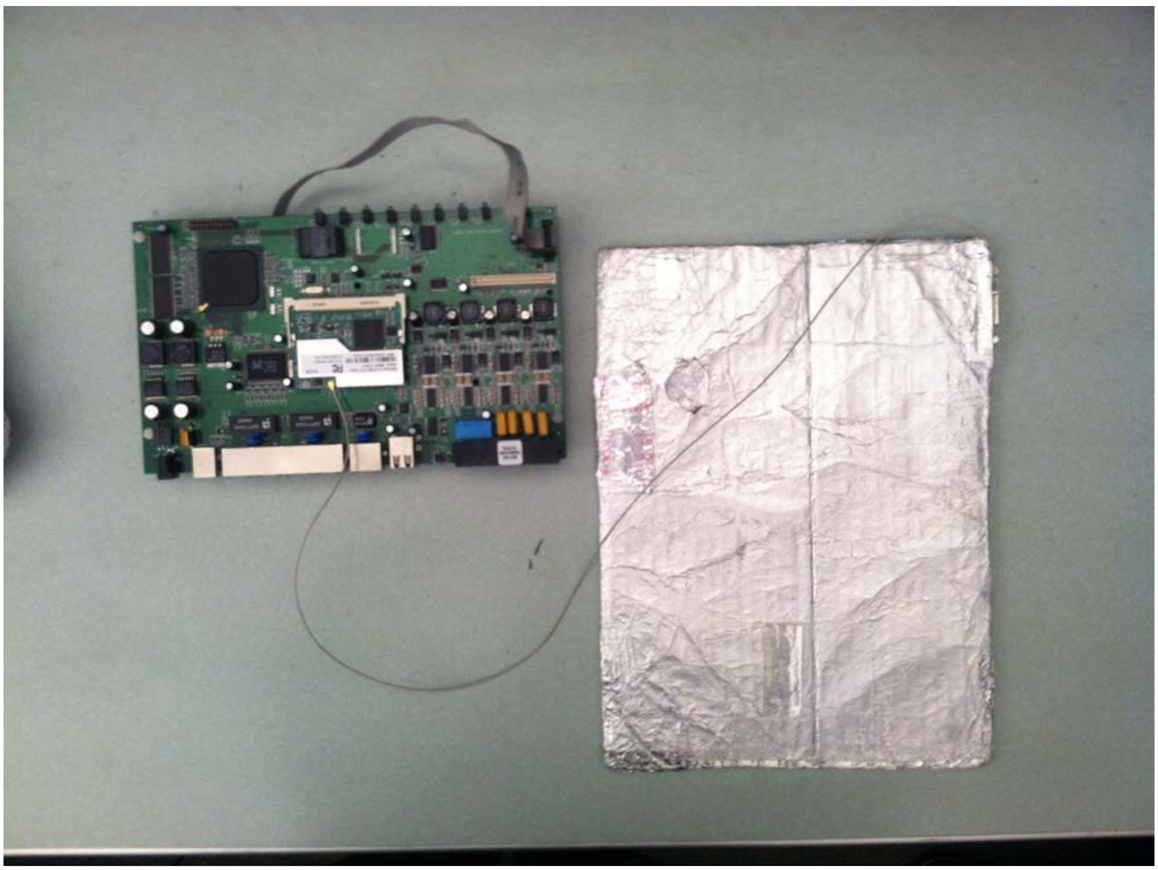

Figure 21 Mobile IPv6 Access Router #2

Figure 21 shows Mobile IPv6 Access Router #2 with the Etenna Edgewave EE5801 dual-mode antenna attached to an ultra-miniature coaxial U.FL terminated cable connected to the mini-PCI radio card. The Edgewave EE5801 antenna is at the top right-hand corner of the aluminum ground plane. The rectangular aluminum ground plane simulates a typical grounding mechanism used in notebook computers.

Core Network

Not shown connected is the core Ethernet network. For this prototype the core network is an Ethernet cable attached from Access Router #1 Ethernet Port 1 to Access Router #2 Ethernet Port 2. The core network consists of the following components:

- Core Network: 100Mbps (Fast) Ethernet
- Mobile IPv6 Server: HP Pavilion DV4000 Notebook
  1.7 GHz Intel Centrino Processor
  1GB RAM
  40GB Hard Disk Drive
  Intel ProWireless 2200 b/g Radio
  Realtek 8139/810x Fast Ethernet connected to Access Router #1
- Mobile IPv6 Node: HP Pavilion DV4000 Notebook
  2.0 GHz Intel Centrino Processor
  1GB RAM
  40GB Hard Disk Drive
  Intel ProWireless 2200 b/g Radio

The hardware components selected were based on availability at the time of development and potential for code re-use. For an efficient development process two identical systems with the only difference being processor speed and disk space is preferred.

Antennas Verified

Table 20 shows antenna power measurements at 20 Feet from the 19mW Access Router Source to the Mobile IPv6 Node.

Antenna Model	Power at 20 Feet (dBm) at Cal Poly Rm 20-143
No Antenna	-70 to -75
Aluminum Foil Antenna	-57 to -61
Etenna Edgewave EE5801 antenna on aluminum foil Ground Plane	-62
WL-ANT151 9" Monopole	-44
Gateway AP 12" Monopole	-41

Table 20 Antenna Power Measurements at 20 Feet from 19mW Access Router Source

The table shows that the Access Router, without an antenna attached to the mini-PCI radio card is able to detect an 802.11g probe request and send a response from the Access Router. If an aluminum foil antenna is used the probe request is received and the probe response is sent, but it is susceptible to interference as demonstrated by the variable power intensity readings.

The Etenna Edgewave EE5801 antenna mounted on an aluminum foil panel, the WL-ANT151 9 inch antenna and the Gateway AP 12 inch antenna all have characteristics that provide for a consistent signal and thus have a good reception at greater distances.

MIPv6 Power Requirements

To determine the power requirements for the Mobile IPv6 installations an Instek GP8212 AC Power Meter was setup in series with the Mobile IPv6 network components. The configurations tested are noted in the figures below.

MIPv6 Power Measurements – Server

The figures and tables that follow summarize the power requirements for a typical Mobile IPv6 Server installation.

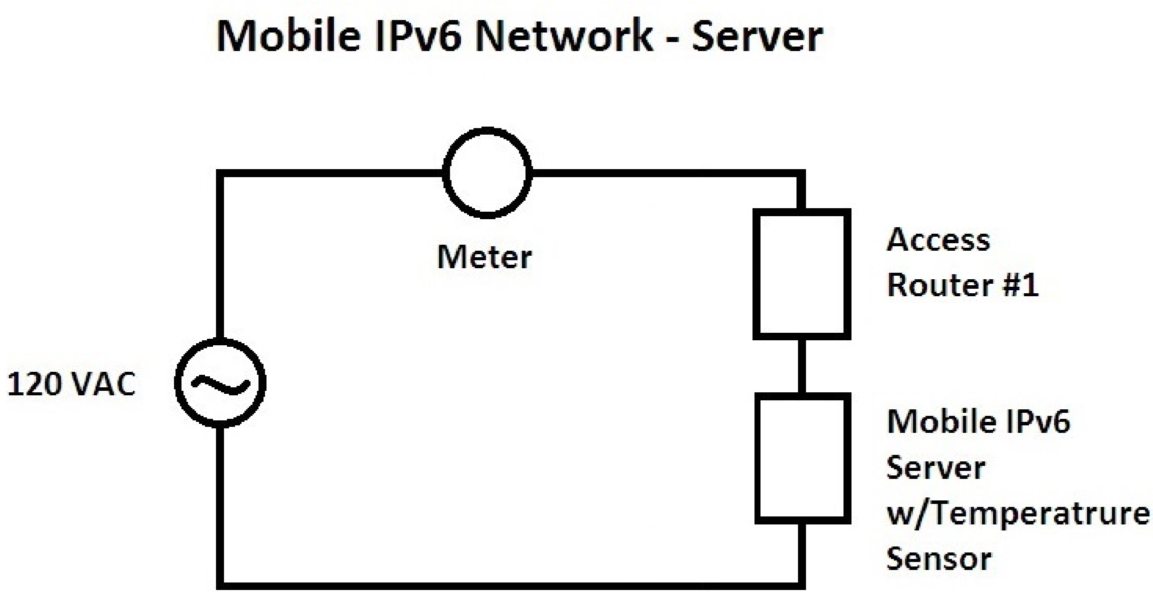

Figure 22 Power Meter Setup for Mobile IPv6 Network - Server

The figure above shows a 120Volt AC source, a Watt Meter, Access Router #1 and the Mobile IPv6 Server with a temperature sensor.

For the Mobile IPv6 Server installation the measured power with traffic from one Mobile IPv6 Node is shown below.

Mobile IPv6 Configuration	**Power without Mobile IPv6 Node Traffic**	**Power with Mobile IPv6 Node Traffic**	**Power Per IPv6 Node Traffic**
Access Router #1 Mobile IPv6 Server w/Sensor	53.0 Watts	54.7 Watts	1.7 Watts

Table 21 Power Measurements for Mobile IPv6 Network - Server

Given that each Mobile IPv6 node takes approximately 1.7 Watts of power, the expected power requirements for 32 Mobile IPv6 Nodes all simultaneously active are as follows:

Mobile IPv6 Server System Peak Power = 53.0 Watts + 1.7 Watts/Node * 32 Nodes = 107.4 Watts

Therefore, a Mobile IPv6 Server installation with the maximum number of nodes all actively passing traffic to and from the server requires 107.4 Watts of total peak power.

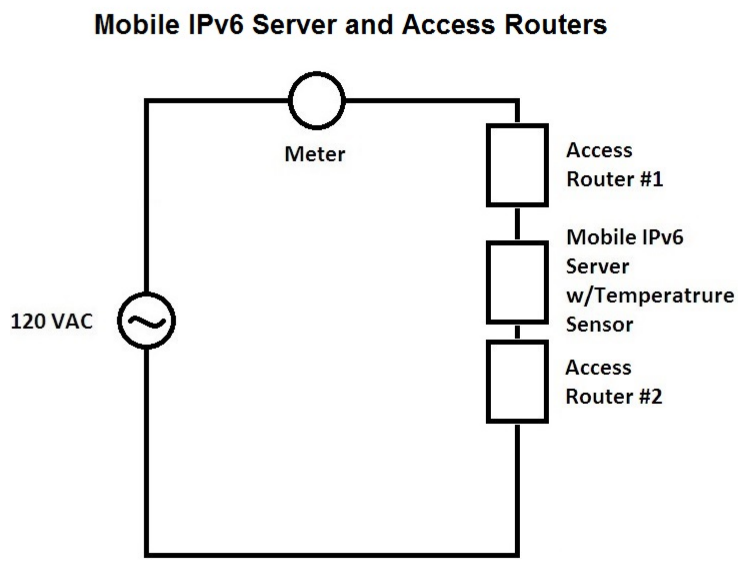

Figure 23 Mobile IPv6 Server and Access Routers

For the setup in Figure 23, the power required for a single router with one sensor was measured as follows.

Steady State Mobile IPv6 Server Setup with Access Router #1 and #2 = 64 Watts

From Figure 22, the Steady Sate Mobile IPv6 Server with Access Router #1 and one sensor as measured as follows.

Steady State Mobile IPv6 Server Setup with Access Router #1 = 53 Watts

Therefore, from these two measurements we can estimate the Access Router power requirements as follows.

Access Router Steady State Power = Steady State Mobile IPv6 Server Setup with Access Router #1 and #2 - Steady State Mobile IPv6 Server Setup without Access Router #2

Access Router Steady State Power = 64.0 Watts - 53 Watts = 11 Watts per Access Router

MIPv6 Power Measurements – Prototype Network

The figures and tables that follow summarize the power requirements for the Mobile IPv6 Prototype of this thesis. This is not a typical installation and only serves as a reference for quantifying the power requirements for a single sensor installation.

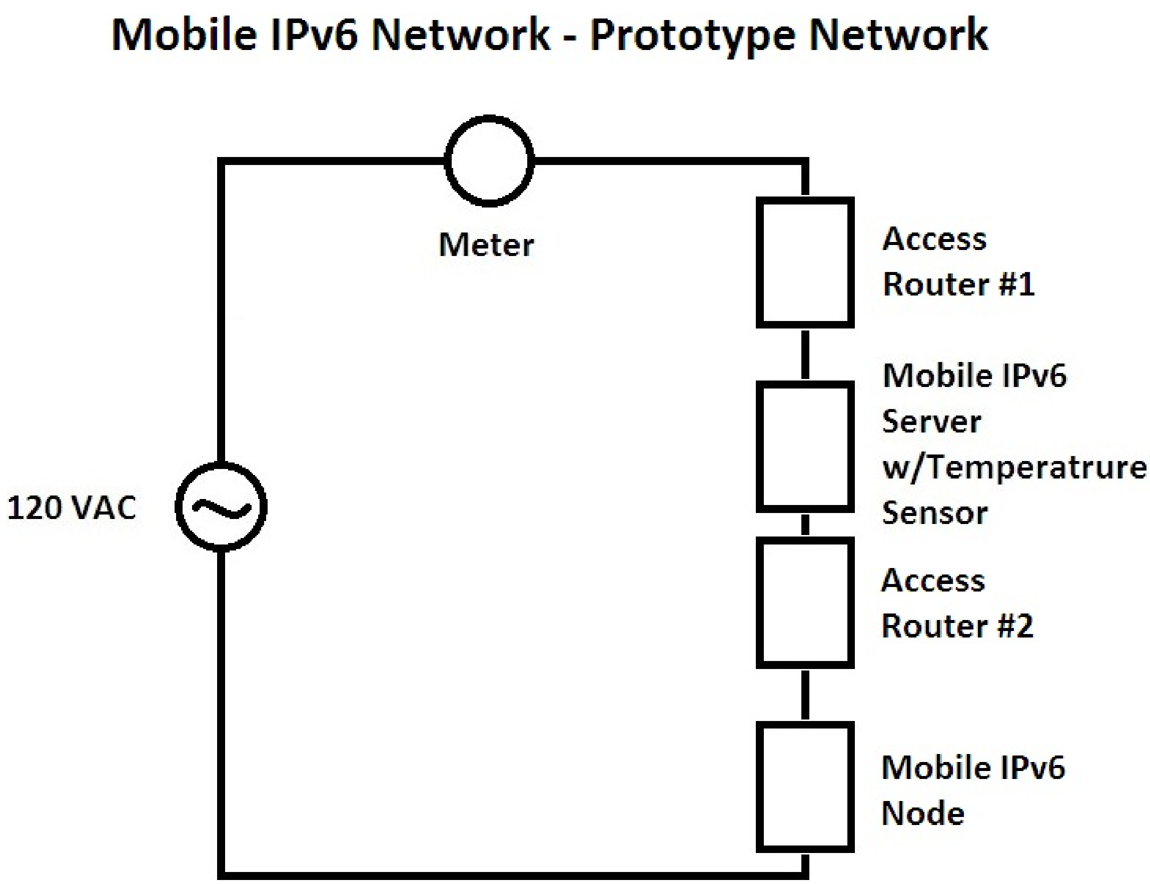

Figure 24 Power Meter Setup for Mobile IPv6 Prototype Network

The figure shows the setup to measure power consumption of the Mobile IPv6 Prototype network described in this thesis. A one sensor install consists of Access Router #1 and Mobile IPv6 Server setup indoors. Access Router #2 with a temperature sensor attached via USB and Mobile IPv6 are installed out on the field. The Mobile IPv6 Node, which is a portable device that runs on batteries, was accounted for in these power measurements using its AC power supply converter.

Mobile IPv6 Configuration	**Power without Mobile IPv6 Node Traffic**	**Power with Mobile IPv6 Node Traffic**	**Power Per IPv6 Node Traffic**
Full IPv6 Prototype Network	86.4 Watts	88.0 Watts	1.6 Watts

Table 22 Power Measurements for Mobile IPv6 Prototype Network

The table shows that the entire prototype network consumes 86.4 Watts. A 1.6 Watt requirement to pass MIPv6 Node traffic is almost identical to the 1.7 Watts required to pass traffic in the Mobile IPv6 Server setup. From these measurements, a typical field Access Router installation with one sensor and a single Mobile IPv6 Node actively passing traffic can be approximated as follows:

Mobile IPv6 Access Router Peak Power with MIPv6 Node and Sensor = 88.0 Watts for Full Network – 54.7 Watts per Mobile IPv6 Server with Access Router #1 = 33.3 Watts

This single field Access Router installation with the Mobile IPv6 Node power derived from the solar cell requires 33.3 Watts of power with active traffic from the Mobile IPv6 Node to the Mobile IPv6 Server.

In conclusion, for a single sensor Access Router field installation, the calculated steady state power required is determined to be **11 Watts**. For an MIPv6 Server with Access Router #1 and one sensor, power was measured at **54.7 Watts.** For an MIPv6 Node with Access Router #2 and one sensor, power was calculated to be at **33.3 Watts**. These two Access Router installations (Mobile IPv6 Server and Mobile IPv6 Node) have power requirements, one measured, the other calculated, that are substantially different on what should be nearly identical values. The differences can be attributed to the efficiency rating of the DC Power Supply and the sensitivity of the Instek GP8212 AC Power Meter to the DC source and load. For the **self-contained prototype network implemented in this thesis**, a total of **88.0 Watts** was the power utilized by all the components of the network. Finally, a theoretical calculated value of **107.4 Watts** is required for the **proposed Access Router with 32 sensors** powered in the field.

MIPv6 Power Specifications

The power measurements are as follows to support the following MIPv6 installations.

Mobile IPv6 Server	**107.4 Watts for 32 Mobile IPv6 Nodes**
Mobile IPv6 Access Router & Node	**33.3 Watts for Single Sensor Installation**

MIPv6 Server Power Specification

Given the **107.4 Watts** required in power measurements from the previous section and the maximum peak power rating of the AC to DC power supply converter, a MIPv6 Server installation can be supported by a single Miasole MS 150GG-03 150 Watt solar cell.

Access Router #1	25 Watts Maximum Peak Power Converter Rating
Mobile IPv6 Server:	90 Watts Maximum Peak Power Converter Rating
Proposed Solar Panel:	Miasole MS 150GG-03 150 Watt Continuous [24]

Figure 25 Miasole MS 150GG-03 150 Watt Solar Panel

MIPv6 Access Router Power Specification

The **33.3 Watts** measured for the field Access Router of the previous section and the maximum peak power of the power supply converter of the Mobile IPv6 Node show that a single Kaneka U-SA110 110 Watt solar panel is sufficient for an Access Router field installation.

Access Router #2	25 Watts Maximum Peak Converter Rating
Mobile IPv6 Node:	70 Watts Maximum Peak Converter Rating
Proposed Solar Panel:	Kaneka U-SA110 110 Watt Continuous [25]

Figure 26 Kaneka U-SA110 110 Watt Solar Panel

The proposed solar panels are for lab tested equipment. Actual requirements for an outdoor application have not been verified.

IX Cost

The cost for this prototype High-Speed Mobile IPv6 Network for a farming application as implemented with an Ethernet core network is:

Mobile IPv6 Server	$1500
Mobile IPv6 Node	$1500
Access Router #1 (MontaVista)	$4000
Access Router #2 (MontaVista)	$4000
Arduino UNO Controller	$ 25
Maxim Temperature Sensor	$ 25
USB Cables	$ 25
Ethernet Cables	$ 25
Edgewave EE5801 Antenna	$ 20
EP.FL Antenna Cables	$ 100
Total Cost	$7220

X Benefits

The benefits derived from applying a high-speed, highly-mobile and highly adaptable Mobile IPv6 network are:

- Reduce resource usage:
    1. Water
    2. Fertilizer
    3. Pesticide
    4. Labor
- Improve plant production and yield at harvest time
- Utilize space better
- Reduces heavy equipment use for tilling soil
- Improves efficiency and recovery of soil by keeping weeds out
- Improves labor efficiency
- Improves communication with neighbor farms and cooperatives
- Improves plant response and adaptability to extreme conditions

XI Future Work

Further work is required to investigate the application of solar cells in the farming environment. This modern network for the farm environment requires constant access for optimum use. If the sun is not out on a particular set of days, it would require batteries to keep the system functioning. However caution and containment must be considered since the use of batteries pose a serious health hazard exposes dangerous chemicals to the farm and public. Alternatively, use of other locally available renewable energy sources such as wind and hydro-electric power may be suitable for this application.

Solar cells are valid if downtime for a few days to complete shutdown is acceptable for the farm application or an alternate energy source can be used during poor solar conditions. The cost of such a flexible system is more economical.

XII Contributions

This thesis has focused exclusively on creating the high-speed mobile network necessary to sustain the data needs of a modern farm. Topics that have been researched and resulted in meaningful contributions are:

High Speed Mobile Network System Architecture Proposed

The proposed High-Speed Mobile IPv6 Network for a modern farm is intended to build a framework based on the principle on-demand data is necessary to manage resources effectively. With a proper data monitors in place, any expenditures that do not yield acceptable crops should be easily detected early in the plant's existence with a highly-available network.

This proposed High-Speed Mobile IPv6 network provided higher bandwidth that is currently available via cellular data networks. In the table below the EV-DO option summarizes the maximum data rate available domestically for cellular data networks. The other evolving network solutions listed in Table 23 below provide potential higher bandwidth networks that could be considered for this proposed network architecture.

The range of comparable network solutions, including the Mobile IPv6 network proposed in this thesis, are summarized in Table 23 below [3].

High-Speed Network	**Encoding**	**Downstream Peak (Mbps)**	**Upstream Peak (Mbps)**
WIMAX	802.16m	110, 183, 219, 365	70, 188, 140, 376
Flash-OFDM	Flash-OFDM	5.3, 10.6, 15.9	1.8, 3.6, 5.4
IBURST	802.20	95	36
802.11ac	802.11ac	433	433
Mobile IPv6	802.11g	20	20
EV-DO	CDMA2000	2.45, 3.1	0.15, 1.8

Table 23 Summary of High-Speed Mobile Networks

Device Driver Software Improvements for Secure and Efficient Roam/Handoff

Device driver improvements in the Access Router's Atheros radio driver and the Mobile IPv6 Server/Node with the Intel Pro Wireless 2200 802.11b/g driver were required for efficient transmission. Specifically, an access control list (ACL) was applied to the Access Routers and Mobile IPv6 Server and Node such that only nodes added into the list are accepted into the network. Another enhancement is removal of broadcast beacons in the Mobile IPv6 network. Specifically, the Access Routers do not send out beacons. Also, probe requests are not replied to if the proper network name is not provided. This prevents other mobile devices that scan for open networks from entering the Mobile IPv6 network.

Antenna Design and Testing

Antenna design and test was required for this prototype network. The antennas designed and tested in on the Access Routers were a) an Aluminum foil antenna b) an Edgewave EE5801 antenna. The Aluminum foil antenna on the Access Router serves to provide access to the Mobile IPv6 network for stations that are near the Mobile IPv6 Server's installation. This initial network access serves to test out connectivity before being sent out into the field. The second antenna tested was the Edgewave EE5801 antenna. It is used in the field of Access Routers where the sensors are constantly updating the pH, image/video, humidity and ambient and plant temperatures. The Edgewave EE5801 has other uses as an 802.11a/b/g wireless antenna for a notebook workstation. It is suitable as a notebook antenna due to Etenna Corporation's dual-mode 802.11bg and 802.11a (2.4 GHz and 5.0GHz) and it's low-profile design.

Both the Mobile IPv6 Server and Mobile IPv6 Node can connect to the Mobile IPv6 network with a wireless connection. However, for this application, only the Mobile IPv6 Node is connected via the 802.11g radio. The Mobile IPv6 Server is expected to be a high-throughput fixed wired connection.

OPNET Model of Proposed High-Speed Mobile Network

The OPNET modeler software enabled the test of the prototype network before any expenditure was made. Specifically, a test of core networks such as 802.11a/b/g, Asynchronous Transfer Mode (ATM) and Ethernet can easily be verified with this software. The OPNET modeler confirmed that the Mobile IPv6 network was possible with an application such as FTP running in the background while a Mobile IPv6 Node was being handed-off from one Access Router to the next. The OPNET modeler results are for an ATM core network. However, since Ethernet technology has caught up with ATM speeds of 100Mbps and 150Mbps either technology is suitable for the core network. The wireless mesh network with point-to-point horn antennas was not tested, but a system such as the MontaVista board with two radios could serve as the mesh network.

System Level Integration and Testing of a Natural and Event Driven Roam

The integration of the Mobile IPv6 Network encapsulated in both Ethernet and 802.11g frames was accomplished with drivers and software modules that were either custom configured and/or modified for this application. The adoption of the open source Mobile IPv6 stack was implemented both in the TCP and IP layers such that it was transparent to the Layer 1 (CSMA and OFDM) and Layer 2 (Ethernet and 802.11 frames). Together with the changes to the Access Router and Mobile IPv6 Server and Node resulted in a roam that takes four (4) seconds instead of the previously measured eight (8) seconds. Further improvements in the Mobile IPv6 software can result in a faster roam.

XIII Conclusions

In conclusion, this project and its prototype of a network meets the requirements of the proposed high-speed mobile farm network via a seamless and efficient roam from one Access Router to the next. Improvements in the Mobile IPv6 software roam sequence and disassociation/association in the Intel ProWireless 2200b/g device driver changes will enable faster roams. Future work with mobile devices such as tablets and personal digital assistants (PDA) is the next adoption cycle for this Mobile IPv6 network.

Also, it is not expected that the sensors currently in the market today will meet the cost requirements to the scale envisioned in this system. However, with modern advances in sensor and image gathering technology, the cost will be significantly reduced to justify its implementation of the proposed automated farm system.

The expectation of this project is to provide the high speed mobile network necessary to develop the complete system. This project tested an MAXIM DB18B20 temperature sensor controlled by Arduino UNO module. This sensor however does not have a wireless technology support and instead relies on a USB connection to the Arduino UNO processor. Future sensors will be 802.11a/b/g/n/ac compliant and will be able to broadcast their information.

In the present design, every instance of corrective action, for a plant in distress, human intervention is required before a corrective action is applied. Alerts are the means for notification that a plant is in distress.

The benefits derived from this project are not easily quantifiable since the cost today prohibits a full scale system. However, at some point cost will reduce and the full scale implementation will be possible. The goal of its implementation is to adapt our human interest in food production and to aid our efforts in plant cultivation to meet the demands of the earth's population.

Plants monitored for maximum resource conservation, yield and efficient decision making require a system that is robust enough to withstand any environment. Therefore, immediate access to the data on any plant at any given moment is necessary from a network that meets the requirements of 'any speed, any place and any time'.

BIBLIOGRAPHY

[1] Microsoft Windows 7 Home Premium Sample Pictures Library

[2] http://en.wikipedia.org/wiki/WiMAX

[3] http://en.wikipedia.org/wiki/4G

[4] http://en.wikipedia.org/wiki/Wireless_LAN

[5] http://www.intel.com/standards/case/case_802_11.htm

[6] http://en.wikipedia.org/wiki/IEEE_802.11a-1999

[7] http://en.wikipedia.org/wiki/IEEE_802.11b-1999

[8] http://en.wikipedia.org/wiki/IEEE_802.11d-2001

[9] http://en.wikipedia.org/wiki/IEEE_802.11e-2005

[10] http://en.wikipedia.org/wiki/Inter-Access_Point_Protocol

[11] http://en.wikipedia.org/wiki/IEEE_802.11g-2003

[12] http://en.wikipedia.org/wiki/IEEE_802.11h-2003

[13] http://en.wikipedia.org/wiki/IEEE_802.11i-2004

[14] http://en.wikipedia.org/wiki/RADIUS

[15] http://en.wikipedia.org/wiki/IEEE_802.11n-2009

[16] http://en.wikipedia.org/wiki/IEEE_802.11ac

[17] http://en.wikipedia.org/wiki/IEEE_802.11

[18] Data Communications, Computer Networks and OSI, Second Edition, 1988 by Fred HallSall

[19] Microwave Engineering Second Edition, 1998 by David M. Pozar page 35

[20] Microwave Engineering Second Edition, 1998 by David M. Pozar page 73

[21] Microwave Engineering Second Edition, 1998 by David M. Pozar page 21

[22] Antenna Theory and Design Second Edition, 1998 by Warren L. Stutzman and Gary A. Thiele page 79

[23] Etenna Edgewave EE5801 Wireless LAN Dual-Mode Antenna Datasheet

[24] http://www.miasole.com/products_ms

[25] http://www.gogreensolar.com/products/Kaneka-u-sa110-110-w-hybrid-thin-film-solar-panel

[26] http://www.ibm.com

Appendix A Access Router Configuration

The following initialization and configuration applies to the Access Routers which form the Mobile IPv6 Network.

Access Router Modules Initialization

Loadable modules must be installed before the interfaces are configured.

```
#!/bin/sh
echo "Building module dependencies";
echo "Installing ethernet drivers";
cd /lib/modules/*/kernel/drivers/net
insmod ixp400.ko
insmod ixp425_eth.ko
echo "Installing atheros drivers";
cd /lib/modules/*/kernel/drivers/net/wireless/net80211
insmod wlan.ko
cd /lib/modules/*/kernel/drivers/net/wireless/_ath_hal
insmod ath_hal.ko
cd /lib/modules/*/kernel/drivers/net/wireless/_ath_rate
insmod ath_rate_onoe.ko
cd /lib/modules/*/kernel/drivers/net/wireless/ath
insmod ath_pci.ko
echo "Installing wlan ap drivers";
cd /lib/modules/*/kernel/drivers/net/wireless/net80211
insmod wlan_scan_ap.ko
insmod wlan_scan_sta.ko
```

Figure 27 Ethernet and Wireless Loadable Module Initialization

Access Router #1 Interface Configuration

The following router configuration initializes the interfaces and starts the routing daemon on the Access Router #1.

```
#!/bin/sh
./mods
sleep 20
rm -f /tmp/radvd.pid
./mip6-hr
radvd
```

Figure 28 Access Router #1 Initialization Script

File: mip6-hr

```
#!/bin/sh
echo "Setting IPv6 for Mobile-IPv6"
wlanconfig ath0 create wlandev wifi0 wlanmode ap
iwconfig   ath0 essid "mipasn" enc off channel 6
ifconfig   eth0 up
ifconfig   eth1 up
ifconfig   ath0 up
ifconfig   eth0 2000:106:2300::7/64 up
ifconfig   ath0 2000:106:2700::1/64 up
echo "1" > /proc/sys/net/ipv6/conf/all/forwarding
echo "0" > /proc/sys/net/ipv6/conf/all/autoconf
echo "0" > /proc/sys/net/ipv6/conf/all/accept_ra
echo "0" > /proc/sys/net/ipv6/conf/all/accept_redirects
route -A inet6 add 2000:106:1100::/64 gw 2000:106:2300::5
```

Figure 29 Access Router #1 Interface Configuration

Access Router #1 Routing Advertising Daemon Configuration

Configuration to advertise routing prefix for Access Router #1.

File: /etc/radvd/radvd.conf

```
interface ath0
{
    AdvSendAdvert on;
    AdvIntervalOpt on;
    MinRtrAdvInterval 1;
    MaxRtrAdvInterval 2;
    AdvHomeAgentFlag off;
    prefix 2000:106:2700::/64
    {
        AdvOnLink on;
        AdvAutonomous on;
        AdvRouterAddr on;
    };
};
interface eth0
{
    AdvSendAdvert on;
    AdvIntervalOpt on;
    MinRtrAdvInterval 1;
    MaxRtrAdvInterval 2;
    AdvHomeAgentFlag off;
    prefix 2000:106:2300::/64
    {
        AdvOnLink on;
        AdvAutonomous on;
        AdvRouterAddr on;
    };
};
```

Figure 30 Access Router #1 Routing Advertising Daemon Configuration

Access Router #2 Interface Configuration

The following router configuration initializes the interfaces and starts the routing daemon on the Access Router #2.

```
#!/bin/sh
./mods
sleep 20
rm -f /tmp/radvd.pid
./mip6-sr
radvd
```

Figure 31 Access Router #2 Initialization Script

File: mip6-sr

```
#!/bin/sh
echo "Setting IPv6 for Mobile-IPv6"
wlanconfig ath0 create wlandev wifi0 wlanmode ap
iwconfig   ath0 essid "mipasn" enc off channel 11
ifconfig   eth0 up
ifconfig   ath0 up
ifconfig   eth0 2000:106:2300::5/64 up
ifconfig   ath0 2000:106:1100::1/64 up
echo "1" > /proc/sys/net/ipv6/conf/all/forwarding
echo "0" > /proc/sys/net/ipv6/conf/all/autoconf
echo "0" > /proc/sys/net/ipv6/conf/all/accept_ra
echo "0" > /proc/sys/net/ipv6/conf/all/accept_redirects
route -A inet6 add 2000:106:2700::/64 gw 2000:106:2300::7
```

Figure 32 Access Router #2 Interface Configuration

Access Router #2 Routing Advertising Daemon Configuration

The following configuration advertises the routing prefix for Access Router #2.

File: /etc/radvd/radvd.conf

```
interface ath0

{

    AdvSendAdvert on;

    AdvIntervalOpt on;

    MinRtrAdvInterval 3;

    MaxRtrAdvInterval 10;

    AdvHomeAgentFlag off;

    prefix 2000:106:1100::/64

    {

        AdvOnLink on;

        AdvAutonomous on;

        AdvRouterAddr on;

    };

};
```

Figure 33 Access Router #2 Routing Advertising Daemon Configuration

Appendix B Mobile IPv6 Server and Node Configuration

Mobile IPv6 Server Module and IPSec Initialization

The following configures the interfaces and starts the MIPv6 and routing advertising daemons on the Mobile IPv6 Server.

```
#!/bin/sh
export PATH=/usr/local/sbin:$PATH
cd tools
. wlan-ha start
. mip6-ha
radvd
```

Figure 34 Mobile IPv6 Server Initialization Script

Mobile IPv6 Server IPSec Configuration

IPSec configuration to allow the MIPv6 daemon to convert a regular autoconfigured tunnel to an IPSec tunnel.

```
# Usage: setkey -f sa.conf
# 2000:106:2700::4 is home address of MN
# and 2000:106:2300::8 is address of HA
# MN -> HA transport SA for BUs
add 2000:106:2700::4 2000:106:2300::8 esp 2000
        -m transport
        -E des-cbc "TAHITEST"
        -A hmac-sha1 "this is the test key" ;
# HA -> MN transport SA for BAs
add 2000:106:2300::8 2000:106:2700::4 esp 2001
        -m transport
        -E des-cbc "TAHITEST"
        -A hmac-sha1 "this is the test key" ;
# MN -> HA tunnel SA for HoTIs
add 2000:106:2700::4 2000:106:2300::8 esp 2004
        -m tunnel
        -E des-cbc "TAHITEST"
        -A hmac-sha1 "this is the test key" ;
# HA -> MN tunnel SA for HoTs
add 2000:106:2300::8 2000:106:2700::4 esp 2005
        -m tunnel
        -E des-cbc "TAHITEST"
        -A hmac-sha1 "this is the test key" ;
```

Figure 35 Mobile IPv6 Server IPSec Configuration

Mobile IPv6 Server Module IPSec Initialization

The following script initializes the modules and IPSec configuration.

File wlan-ha:

```
#!/bin/sh

# Main wlan start/stop/setup

case $1 in

  start)

  if [ -x /usr/src/ipw2200-1.1.0/load ]; then

      modprobe ipw2200

      sleep 10

      setkey -f sa.conf

  fi

  ;;

  stop)

  if [ -x /usr/src/ipw2200-1.1.0/unload ]; then

    cd /usr/src/ipw2200-1.1.0

    . unload

  fi

  ;;

  setup)

    echo "Setting default ap"

    iwconfig eth1 ap 00:90:4B:CC:73:D5

    iwconfig eth1 ap 00:90:4B:CC:74:55

  ;;

  *)

    echo "Usage: wlan_start <start|stop|setup>"

  ;;

esac
```

Figure 36 Mobile IPv6 Server Module and IPSec Initialization

Mobile IPv6 Server Interface Configuration

The following configures the server interfaces and starts the MIPv6 daemon on the Mobile IPv6 Server.

File: mip6-ha

```
#!/bin/sh
echo "Setting up Mobile-IPv6 for Home Agent"
ifconfig eth0 up
ifconfig eth0 inet6 add 2000:106:2300::8/64
ifconfig eth0 up
echo "1" > /proc/sys/net/ipv6/conf/all/forwarding
echo "0" > /proc/sys/net/ipv6/conf/all/autoconf
echo "0" > /proc/sys/net/ipv6/conf/all/accept_ra
echo "0" > /proc/sys/net/ipv6/conf/all/accept_redirects
ip route add ::/0 via 2000:106:2300::7
/usr/local/sbin/mip6d -d 10 -c /usr/local/etc/mip6d.conf
```

Figure 37 Mobile IPv6 Server Interface Configuration

Mobile IPv6 Server Daemon Configuration

The following configuration for the MIPv6 Home Agent allows it manage IPSec tunnels to the specified Mobile Nodes.

File mip6d.conf:

```
# This is an example of mip6d Home Agent configuration file
NodeConfig HA;
## If set to > 0, will not detach from tty
DebugLevel 10;
DoRouteOptimizationCN enabled;
## List of interfaces where we serve as Home Agent
Interface "eth0";
UseCnBuAck enabled;
##
## IPsec configuration
##
UseMnHaIPsec enabled;
# HaMaxBindingLife 10;
# MnMaxHaBindingLife 10;
# MnMaxCnBindingLife 10;
## Key Management Mobility Capability
#KeyMngMobCapability disabled;
IPsecPolicySet {
        HomeAgentAddress   2000:106:2300::8;
        HomeAddress        2000:106:2700::4/64;
        IPsecPolicy HomeRegBinding UseESP;
        IPsecPolicy MobPfxDisc UseESP;
        IPsecPolicy TunnelMh UseESP;
}
```

Figure 38 Mobile IPv6 Server Daemon Configuration

Mobile IPv6 Server Routing Advertising Daemon Configuration

The following configures the Routing Advertising daemon to send routing advertisements for the configured network prefixes and announces that it is the Home Agent.

File: /etc/radvd.conf

```
interface eth0
{
    AdvSendAdvert on;
    MinRtrAdvInterval 1;
    MaxRtrAdvInterval 2;
    AdvIntervalOpt off;
    AdvHomeAgentFlag on;
    HomeAgentLifetime 10000;
    HomeAgentPreference 20;
    AdvHomeAgentInfo on;
    prefix 2000:106:2300::8/64
    {
        AdvRouterAddr on;
        AdvOnLink on;
        AdvAutonomous on;
        # AdvPreferredLifetime 10;
        # AdvValidLifetime 20;
    };
};
```

Figure 39 Mobile IPv6 Server Routing Advertising Daemon Configuration

Mobile IPv6 Node Module and IPSec Initialization

The following configures the interfaces and starts the MIPv6 daemon on the Mobile IPv6 Node.

```
#!/bin/sh
export PATH=/usr/local/sbin:$PATH
cd tools
. wlan-mn start
. mip6-mn
```

Figure 40 Mobile IPv6 Node Initialization Script

Mobile IPv6 Node IPSec Configuration

IPSec configuration to allow the MIPv6 daemon to convert a regular autoconfigured tunnel to an IPSec tunnel.

```
# Usage: setkey -f sa.conf
# 2000:106:2700::4 is home address of MN
# and 2000:106:2300::8 is address of HA
# MN -> HA transport SA for BUs
add 2000:106:2700::4 2000:106:2300::8 esp 2000
        -m transport
        -E des-cbc "TAHITEST"
        -A hmac-sha1 "this is the test key" ;
# HA -> MN transport SA for BAs
add 2000:106:2300::8 2000:106:2700::4 esp 2001
        -m transport
        -E des-cbc "TAHITEST"
        -A hmac-sha1 "this is the test key" ;
# MN -> HA tunnel SA for HoTIs
add 2000:106:2700::4 2000:106:2300::8 esp 2004
        -m tunnel
        -E des-cbc "TAHITEST"
        -A hmac-sha1 "this is the test key" ;
# HA -> MN tunnel SA for HoTs
add 2000:106:2300::8 2000:106:2700::4 esp 2005
        -m tunnel
        -E des-cbc "TAHITEST"
        -A hmac-sha1 "this is the test key" ;
```

Figure 41 Mobile IPv6 Node IPSec Configuration

Mobile IPv6 Node Module and IPSec Initialization

The following script initializes the modules and IPSec configuration.

```
File: wlan-mn

#!/bin/sh

# Main wlan start/stop/setup

case $1 in

  start)

  if [ -x /usr/src/ipw2200-1.1.0/load ]; then

      modprobe ipw2200

      sleep 10

      setkey -f sa.conf

  fi

  ;;

  stop)

  if [ -x /usr/src/ipw2200-1.1.0/unload ]; then

    cd /usr/src/ipw2200-1.1.0

    . unload

  fi

  ;;

  setup)

    echo "Setting default ap"

    iwconfig eth1 ap 00:90:4B:CC:73:D5

    iwconfig eth1 ap 00:90:4B:CC:74:55

  ;;

  *)

    echo "Usage: wlan_start <start|stop|setup>"

  ;;

esac
```

Figure 42 Mobile IPv6 Node Module and IPSec Initialization

Mobile IPv6 Node Interface Configuration

The following configures the interfaces on the Mobile IPv6 Node.

File: mip6-mn

```
#!/bin/sh

echo "Setting up Mobile-IPv6 for Mobile Node"

iwconfig eth1 mode managed essid mipasn enc off

ifconfig eth1 inet6 add 2000:106:2700::4/64

ifconfig eth1 up

echo "0" > /proc/sys/net/ipv6/conf/eth1/forwarding

echo "1" > /proc/sys/net/ipv6/conf/eth1/autoconf

echo "1" > /proc/sys/net/ipv6/conf/eth1/accept_ra

echo "1" > /proc/sys/net/ipv6/conf/eth1/accept_redirects

#ip route add ::/0 via 2000:106:2700::1

/usr/local/sbin/mip6d -d 10 -c /usr/local/etc/mip6d.conf
```

Figure 43 Mobile IPv6 Node Interface Configuration

Mobile IPv6 Node Daemon Configuration

The following configuration for the MIPv6 Mobile Node allows it to establish a secure IPSec tunnel to the Home Agent.

File mip6d.conf:

```
# This is an example of mip6d Mobile Node configuration file
NodeConfig MN;
## If set to > 0, will not detach from tty
DebugLevel 0;
# MnDiscardHaParamProb enabled;
## Support route optimization with other MNs
DoRouteOptimizationMN enabled;
## Use route optimization with CNs
DoRouteOptimizationCN disabled;
MnMaxHaBindingLife 10;
MnMaxCnBindingLife 10;

# UseCnBuAck disabled;
# MnRouterProbesRA 1;
# MnRouterProbesLinkUp 0;

MnHomeLink "eth1" {
        HomeAgentAddress 2000:106:2300::8;
        HomeAddress 2000:106:2700::4/64;
}

##
## IPsec configuration
##

UseMnHaIPsec enabled;

## Key Management Mobility Capability
```

```
# KeyMngMobCapability disabled;

IPsecPolicySet {
        HomeAgentAddress   2000:106:2300::8;
        HomeAddress        2000:106:2700::4/64;

        IPsecPolicy HomeRegBinding UseESP;
        IPsecPolicy MobPfxDisc UseESP;
        IPsecPolicy TunnelMh UseESP;
}
```

Figure 44 Mobile IPv6 Node Daemon Configuration

Appendix C Mobile IPv6 Server and Node Application

The communication links between the Mobile Server and a Mobile Node are managed by the MIPv6 daemon via an auto-configured tunnel. Applications in turn use regular TCP and UDP connections to send and receive data. If the interface used by the applications changes, these TCP and UDP connections are re-routed by the MIPv6 daemon for seamless communication. The following are files are modified [as noted with #ifdef MIPASN] from those posted at IBM Corporation Linux Support [26].

Mobile IPv6 Server Application Socket

```
/*
 * 8.6.2 Programs Using AF_INET6 Sockets
 *
 * This section contains a client and a server program that use AF_INET6 sockets.
 * 8.6.2.1 Client Program

 * The following is a sample client program that you can build, compile and run
 * on your system. The program sends a request to and receives a response from
 * the system specified on the command line.
 */

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#include <netinet/in.h>
#include <sys/socket.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <sys/errno.h>
#include <sys/fcntl.h>
```

```
#define SERV_BACKLOG    1                   /* server backlog                   */
#define SERV_PORTNUM    12345               /* server port number               */
#define FILE_OPTION     "/tmp/matrix.mpg"
#define FILE_SIZE_MAX   33554432
#define SEND_SIZE_MAX   2048

int  main( int argc, char *argv[] );        /* server main                      */

int
main( int argc, char *argv[] )
{
    int fd, count = 0;
    int optval = 1;                         /* SO_REUSEADDR'S option value (on) */

    int conn_sockfd;                        /* connection socket descriptor     */
    int listen_sockfd;                      /* listen socket descriptor     */

    unsigned int client_addrlen;            /* returned length of client socket */
                                            /* address structure      */
    struct sockaddr_in6 client_addr;        /* client socket address structure  */
    struct sockaddr_in6 serv_addr;          /* server socket address structure  */

    // char port[1025];                        /* buffer to receive port number
NI_MAXHOST*/
    char addrbuf[INET6_ADDRSTRLEN];         /* buffer to receive host's address */
    int send_size = SEND_SIZE_MAX;
    if (argc == 2)
    {
        send_size = atoi(argv[1]);
    }

    char *buf = (char*) malloc(FILE_SIZE_MAX);  /* server data buffer      */
    if (!buf)
    {
        printf("ERROR: malloc file buffer failed\n");
```

```
        exit ( -1 );
    }

    printf("Data size: %d\n", send_size); /* output server's data buffer  */

begin:
    memset( &client_addr, 0, sizeof(client_addr) );

    memset( &serv_addr, 0, sizeof(serv_addr) );
    serv_addr.sin6_family      = AF_INET6;
    serv_addr.sin6_port        = htons( SERV_PORTNUM );
    serv_addr.sin6_addr        = in6addr_any;

    if ( (listen_sockfd = socket(AF_INET6, SOCK_STREAM, 0)) < 0 )
    {
    perror( "Failed to create socket" );
    exit( -1 );
    }

    if ( setsockopt(listen_sockfd,
      SOL_SOCKET, SO_REUSEADDR, &optval, sizeof(optval)) < 0 )
    {
    perror( "Failed to set socket option" );
    exit( -1 );
    }

    if ( bind(listen_sockfd,
       (struct sockaddr *) &serv_addr, sizeof(serv_addr)) < 0 )
    {
    perror( "Failed to bind socket" );
    exit( -1 );
    }

    if ( listen(listen_sockfd, SERV_BACKLOG) < 0 )
    {
    perror( "Failed to set socket passive" );
```

```
    exit( -1 );
    }

    printf( "Waiting for a client connection on port: %d\n",
        ntohs(serv_addr.sin6_port));

    client_addrlen = sizeof(client_addr);

    conn_sockfd = accept( listen_sockfd, (struct sockaddr *) &client_addr,
                          &client_addrlen);
    if ( conn_sockfd < 0 )
    {
    perror( "Failed to accept client connection" );
    exit( -1 );
    }

    memcpy(addrbuf, client_addr.sin6_addr.in6_u.u6_addr8, 16);
    printf("Accepted connection from host:
%02x:%02x:%02x:%02x:%02x:%02x:%02x:%02x:%02x:%02x:%02x:%02x:%02x:%02x:%02x:%02x, port:
%d\n",
        addrbuf[0],addrbuf[1],addrbuf[2],addrbuf[3],
        addrbuf[4],addrbuf[5],addrbuf[6],addrbuf[7],
        addrbuf[8],addrbuf[9],addrbuf[10],addrbuf[11],
        addrbuf[12],addrbuf[13],addrbuf[14],addrbuf[15], ntohs(client_addr.sin6_port));

    fd = open (FILE_OPTION, 00 | 04000);
    if (fd < 0)
    {
        printf( "Failed to open file: %s", FILE_OPTION);
        exit ( -1 );
    }

    do {
        if ((count = read (fd, buf, send_size)) < 0)
        {
        perror( "Failed to read data from source file" );
```

```
    exit( -1 );
    }

    if ((count = send(conn_sockfd, buf, count, 0)) < 0 )
    {
    perror( "Failed to write data to client connection" );
    exit( -1 );
    }

    // printf( "Data sent: %d\n", count );    /* output server's data buffer  */
} while (count);

if (close(fd) < 0)
{
perror( "Failed to close data file" );
}

if ( shutdown(conn_sockfd, 2) < 0 )
{
perror( "Failed to shutdown client connection" );
exit( -1 );
}

if ( close(conn_sockfd) < 0 )
{
perror( "Failed to close socket" );
exit( -1 );
}

if ( close(listen_sockfd) < 0 )
{
perror( "Failed to close socket" );
exit( -1 );
}

goto begin;
```

```
    exit( 0 );
}
```

Figure 45 Mobile IPv6 Server Application Socket

Mobile IPv6 Node Application Socket

```
/*
 * 8.6.2 Programs Using AF_INET6 Sockets
 *
 * This section contains a client and a server program that use AF_INET6 sockets.
 * 8.6.2.1 Client Program

 * The following is a sample client program that you can build, compile and run
 * on your system. The program sends a request to and receives a response from
 * the system specified on the command line.
 */

#include <netinet/in.h>
#include <sys/socket.h>                     /* define BSD 4.x socket api        */
#include <sys/types.h>
#include <sys/errno.h>
#include <sys/fcntl.h>

#define SERV_PORTNUM    12345               /* server port number               */
#define FILE_OPTION     "/tmp/matrix"
#define FILE_SIZE_MAX   1048576

int main( void );                           /* client main                      */

int
main( void )
{
    int  fd, count = 0;
    int  flags;
    int  sockfd;                            /* connection socket descriptor     */
    char buf[FILE_SIZE_MAX];                /* client data buffer               */
    char ipa[] = { 0x20,0,0x01,0x06,0x23,0,0,0,0,0,0,0,0,0,0,8 };

    struct sockaddr_in6 target;
```

```
memset( &target, 0, sizeof(target) );
target.sin6_family = AF_INET6;
target.sin6_port   = htons(SERV_PORTNUM);
memcpy(target.sin6_addr.in6_u.u6_addr8, ipa, 16);

/* Create FIFO */
umask(0); mknod(FILE_OPTION, 0010000 | 0666, 0); /* S_IFIFO = 0010000 */
fd = open(FILE_OPTION, 02); /* O_RDWR | O_NONBLOCK */

if (fd < 0)
{
printf("Failed to open FIFO file: %s", FILE_OPTION);
exit ( -1 );
}

if ( (sockfd = socket(AF_INET6, SOCK_STREAM, 0)) < 0 )
{
perror( "Failed to create socket" );
exit( -1 );
}

if (connect(sockfd, (struct sockaddr *)&target, sizeof(target)) < 0)
{
    perror( "Failed to connect to server" );
    exit( -1 );
}

// flags = fcntl(sockfd, F_GETFL);
// flags |= 04000; /* O_NONBLOCK */
// fcntl(sockfd, F_SETFL, flags);

printf( "Initiated connection to host: port: %d\n", target.sin6_port);

do {
    if ( (count = recv(sockfd, buf, FILE_SIZE_MAX, 0)) < 0 )
    {
```

```
            perror( "Failed to read data from server." );
            exit ( -1 );
        }

        printf( "Data received (bytes): %d\n", count);

        if (write(fd, buf, count) != count)
        {
            perror( "Failed to write data to FIFO file" );
            exit ( -1 );
        }

    } while (count);

    if ( shutdown(sockfd, 2) < 0 )
    {
        perror( "Failed to shutdown server connection" );
        exit( -1 );
    }

    if ( close(sockfd) < 0 )
    {
        perror( "Failed to close socket" );
        exit( -1 );
    }

    if (close(fd) < 0)
    {
        perror( "Failed to close FIFO file" );
        exit( -1 );
    }

    exit( 0 );
}

/*
```

```
 *  1. Function prototype for server host address/name translation function.
 *  2. Declares addrinfo structures.
 *  3. Clears the addrinfo structure and sets values for fields of the structure.
 *  4. Calls get_serv_addr() passing pointers to the input and output addrinfo
structures.
 *  5. Creates an AF_INET6 socket.
 *  6. Uses values from the output addrinfo structure for host name and port.
 *  7. Calls connect() using values from the output addrinfo structure.
 *  8. Retrieves the server host's address from the user and stores it in the
 *     addrinfo structure. The user can specify a server host by using any of the
following:
 *          * An IPv4 address in dotted-decimal notation
 *          * An IPv6 address in hexadecimal
 *          * An Ipv4-mapped IPv6 address in hexadecimal
 *          * A host domain name
 *  9. Calls getaddrinfo() to retrieve the server host's name or address.
 * 10. Calls gai_strerror() to convert one of the EAI_xx return values to a string
describing the error.
 */
```

Figure 46 Mobile IPv6 Node Application Socket

Appendix D Antenna Measurement System (AMS) Test Procedures

The following procedure outlines the steps necessary to obtain antenna measurements using the equipment in the Anechoic Chamber Building 4 Room 113 (please read the notes below before beginning the test session).

Scientific Atlanta Controller Notes

The Scientific Atlanta Controller is very old and occasionally experiences intermittent operation during an antenna measurement session. It may be necessary to turn it off and allow it to cool down for about 10-15 minutes before starting a new measurement.

How To Reset Controller to 0^o

Open the "Advanced Controls" from the "C:\Documents and Settings\chamberadmin\Desktop\AMS\Advanced Controls" folder and set to "CCW," "On" and 0^o. Press "SET PARAMETERS" to set the positioner. During the positioner movement, the controller front panel speed setting should switch to HIGH SPEED. If the controller is stuck in LOW SPEED (see controller front panel) close the "Advanced Controls" application, turn off the controller, then retry. If the rewind operation overshoots and crosses the 0^o (or 360^o) mark use the ""C:\Documents and Settings\chamberadmin\Desktop\AMS\Boresight Locator" application to reposition the positioner to the left or right of the 0^o mark.

NOTE: Do not use the knob on the Scientific Atlanta 4131 Controller to reset to 0^o.

Vector Network Analyzer Notes

The HP Vector Network Analyzer has one small quirk; it must warm up before the display will work. Turn it on and let it warm up for 10-15 minutes, and then press the "User Preset" button.

The Antenna Measurement System Software Notes

The antenna measurement software works best with the National Instruments (NI) Spy logging application is open in the foreground (not minimized) and running. Otherwise, the software may appear to lock up.

Radiation Pattern Measurements:

A) Test Equipment Preparation

1) Turn on the HP 8720C Vector Network Analyzer and allow it to warm up.
2) Turn on the PC and allow it to initialize.

B) Antenna Chamber Test Setup

1) Find the N-Type to SMA Connector to properly connect the antenna to the measurement (aqua) cable.
2) Mount the antenna onto the round table top mounted to the Scientific Atlanta positioner.
3) Connect the antenna to the aqua cable.

4) If it is necessary to re-enter the chamber to make adjustments, turn off the HP 8720C transmitter using the LOCAL>MENU>Trigger>Hold buttons.

C) Antenna Measurement Software Setup

1) Find the N-Type to SMA Connector to properly connect the antenna to the measurement (aqua) cable.
2) Turn on the PC with the AMS Software installed.
3) After the PC initializes, click “Start>All Programs>National Instruments>Measurement and Automation” to start the National Instruments Measurement and Automation Explorer.
4) Once the explorer appears on the screen, click “Devices and Interfaces” to display all GPIB devices installed on the PC.
5) Right click on the GPIB interface (GPIB0) and select “NI Spy” to initialize the log application for GPIB0.
6) Press the blue arrow to begin logging. To stop logging press “Stop.” All logging entries can be erased by pressing the big “X” once logging has been stopped.
7) Keep all windows open in the background.

D) HP 8720C Vector Network Analyzer Setup

1) Press the “LOCAL” button to return control to the user.
2) Press the “MEAS” button in the Response section of the analyzer.
3) Press the “Trans: FWD” button.

4) Set the starting frequency by pressing "START." Enter the frequency using the number keypad and the scale buttons (xG, xM, x1...) to set the proper frequency.
5) Similarly, set the stopping frequency by pressing "STOP."
6) Press the "MENU" button and select "Number of Points." Enter 401 and use the x1 scale button to set.
7) Press the "POWER" button and set it to 10dBm.
8) Press the "AVG" button, press the "IF BW" softkey and set to 100Hz.
9) Press the "SCALE REF" button to scale the display for measurements. Use the up/down arrows, "Reference Position" and "Scale/DIV" buttons to position the reference point and set the dB/Div.
10) Press the "CAL" > "Calibrate Menu" > "Response" > "THRU" buttons. Wait for the calibration to occur and then press "Done Response." This should zero out the reference Line.
11) To start transmitting use the "MENU" > "Trigger" >"Continuous". To stop transmitting use the "MENU" > "Trigger" >"Hold".

E) Running a "Single Stop and Go" Test

1) On the PC, open the "C:\Documents and Settings\chamberadmin\Desktop\AMS" folder to see the test suite.
2) Select the "Single Stop and Go" test.

3) Set the proper frequency on the application. NOTE: The start and stop frequencies on the analyzer and the "Single Stop and Go" frequency setting must match.

4) It is best to use the default values for the test, otherwise it may take longer to complete.

5) Verify the "LabView Data" folder exists. Rename any old "spsheet" files otherwise this test will append the results to this file.

6) Turn on the controller. Reset position to zero if necessary using the AMS "Advanced Controls" application.

7) Press the "Run Test" Button.

8) The azimuthal angle will sweep from 0^o to 360^o in 5^o increments. Approximate duration: 45min

9) Once the test is complete, close the "Single Stop and Go" window.

10) Rename the "spsheet" file and save.

11) Return the positioner to its original position with the "Advanced Controls" application from the "C:\Documents and Settings\chamberadmin\Desktop\AMS" folder.

12) In the "Advanced Controls" set to "CCW," "On" and 0^o. Press "SET PARAMETERS" to set the positioner.

13) Once the controller reaches zero, turn it off to allow the instrument to cool for at least 30 minutes.

14) On the analyzer, press "LOCAL" > "MENU" > "Trigger" > "Hold" to stop transmitting.

15) The chamber can now be accessed to make changes or adjustments

16) Stop, clear and restart NI Spy logging for the next test session.

Return Loss Measurement

A) Measuring Return Loss (Controller is not used)

1) Obtain the HP calibration kit and find the 50Ω, OPEN(female), SHORT(female), and LOAD/BROADBAND connectors.
2) On the network analyzer, press the "USER PRESET" button to clear the settings.
3) Press “MEAS” > “S22” to select the port the antenna under test (AUT) is connected to.
4) Press “FORMAT” > “Log Magnitude”
5) Set the Start and Stop frequencies. Use the antenna's BW specification to select the proper frequencies.
6) Set the number of points to 401
7) The power should be set to 10dBm
8) Press “CAL” > “Cal Kit” > “N50” for N-Type couplers.
9) Press Return
10) Press “Calibration Menu”
11) Press “S22 1 Port”
12) Enter chamber and connect the "OPEN" connector to the aqua cable.
13) Press “Open” > “Female” > “Done”

14) Enter chamber and connect the “SHORT” connector to the aqua cable.

15) Press “Short” > “Female” > “Done”

16) Enter chamber and connect the “LOAD/BROADBAND” connector to the aqua cable.

17) Press “Load” > “Broadband” > “Done”

18) Enter chamber and connect the antenna under test to the aqua cable. Point the antenna toward foam.

19) On the PC, start the “VNA” application from the AMS Folder.

20) Set the start and stop frequencies on the VNA application.

21) Verify application settings match HP 8720C settings. For example, Port is S22, 401 points, "Return Loss" (Log Magnitude) is selected. All other settings should not be changed.

22) Start the test: a plot should appear.

23) Rename the results file (spsheet) and save.