Remote Access and Service Discovery for a Vehicular Public Safety Cognitive Radio

Rohit D. Rangnekar

Thesis submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Master of Science
in
Electrical Engineering

Dr. Charles W. Bostian, Chair
Dr. Michael S. Hsiao
Dr. Yaling Yang

July 9th, 2009
Blacksburg, Virginia

Keywords: public safety, cognitive radio, software defined radio, portable handheld device, service discovery, vehicular network, personal digital assistant, OLSR

Remote Access and Service Discovery for a Vehicular Public Safety Cognitive Radio

Rohit D. Rangnekar

ABSTRACT

The Virginia Tech Center for Wireless Telecommunications' (CWT) Public Safety Cognitive Radio (PSCR) addresses the radio interoperability issues that plague many of the existing public safety radios – disparate frequency bands, incompatible modulation schemes and lack of active channel detection features. The PSCR allows the operator to scan for active channels, classify the detected channels, connect to any of the recognized waveforms and begin analog audio communication as well as bridge two incompatible waveforms together.

The PSCR, although very useful, unfortunately is not portable enough to be used by public safety officials. The power requirement, processing requirement and equipment is respectively large, hungry and bulky. In this thesis, a possible solution to the portability problem is addressed by installing the PSCR in a public safety vehicle and using a Personal Digital Assistant (PDA) for remote access. The PDA allows the user to remotely scan, classify, talk, and bridge waveforms similar in operation to the PSCR. An ergonomically designed interface masks the channel and modulation selection procedure. This architecture can be extended to offer service to any remotely connected device.

In the second part of this thesis, the concept of remote access is extended to a wide-area wireless public safety network. A public safety network consisting of heterogeneous devices is proposed utilizing a small number of backbone nodes. The major research focus of this section is the algorithm for distributing services across the network. Service discovery is optimized to reduce the overhead of service messages and multiple service distribution techniques are utilized depending on the location of the services. Simulation is performed to evaluate the performance of the service discovery protocol in terms of overhead, dissemination time and scalability. The proposed protocol is determined to be superior to the competition in the overhead and scalability tests.

Dedication

This thesis is dedicated to my mother, Dr. Amita Rangnekar,
you were here when this began,
and you will always be there in our memories forever

and to my family

for their constant support, love and encouragement

Acknowledgements

I would like to express my gratitude for my family who has been very supportive of my studies and graduate education. I could not have completed this work without the love and faith of my parents – Dr. Amita and Dr. Dilip Rangnekar, my brother – Neel and grandparents – Dr. Kamal and Dr. Arvind Patankar.

I would like to sincerely thank my advisor Dr. Bostian for giving me an opportunity to work at the CWT. I have enjoyed his encouragement for new ideas and continuous challenges that a new graduate student experiences and I am positive this is the best place to find them. Thank you for your faith in me and I am extremely privileged to be a student under your guidance. I am honored to have met Mrs. Bostian who has so warmly welcomed me into her wonderful home.

I would like to thank my committee members Dr. Hsiao and Dr. Yang for their valuable advice over the past year and suggestions for betterment of my work and for their very helpful discussions during the semester.

I had the pleasure of working with some of the students at the lab - Aravind, Qinqin, Al (Fayez), Alex, Ying, Sujit, Gladstone, Terry, Mark and Andrew (Ge). I would like to take this opportunity to thank the students at the CWT which whom I have developed a long lasting friendship beyond work and cognitive radios! You guys are amazing and one big family. Mark and Andrew, thank you for involving me in your research, more so because I had the opportunity to interact with you better - I enjoyed every minute of it and have learned a lot from just observing the two of you! I had a great time this past year and I wish you all the best of luck for the future.

I would like to thank Judy Hood who has incessantly answered emails and arranged appointments for us with immense patience. Thank you for all the help and tireless dedication.

I take this opportunity to thank Gyu Hyun Kwon for his design of the PDA layout, Xueqi Cheng and Nannan He for their software testing of the PDA code and Scott Bates for the design of his specialized antennas for the VPSCR demonstration.

I would like to thank my friends and roommates in Blacksburg who have made the past two years unforgettable. Life here just wouldn’t be the same without you guys!

Grant Information

This project is supported by the National Institute of Justice, Office of Justice Programs; US Department of Justice award No. 2005-IJ-CX-K01. The opinions, results, and conclusions or recommendations expressed in this thesis/related publications/exhibitions/demonstrations are those of the author(s) and do not necessarily reflect the views of the Department of Justice.

This thesis is based upon research also supported by the National Science Foundation under Grant No. CNS-0519959. Any opinions, results and conclusions or recommendations expressed in this material and related work are those of the author(s) and do not necessarily reflect the views of the National Science Foundation (NSF).

Table of Contents

List of Figures

List of Tables

Chapter 1 : Introduction

1.1 Motivation

> "*A portable handheld device providing all the features of an existing public safety cognitive radio developed at the Virginia Tech Center for Wireless Telecommunications (CWT) and addressing the usability and portability problems of current devices.*"

Cognitive Radio is a research area which has generated immense interest in recent years due to the advancements in software defined signal processing capabilities and affordable RF hardware. Among the various implementations, the majority is based on platforms with general purpose processors present in laptops or desktops as the processing backend. The lack of portability has always plagued its wide spread deployment in public safety scenarios, where such a system can be used to save lives and provide emergency responders with a means to communicate with other agencies. This work is a solution to the portability issues which in the future, may be succeeded by powerful and portable complete cognitive radios utilizing a similar form factor. Emergency responders at the scene of a disaster can now be equipped with a handheld portable device that can harness the power of a cognitive radio and conceal the technical aspects of frequency and waveform selection with a specially designed graphical interface. This concept was originally suggested by Alex Young, a graduate student at the CWT, at an internal review session on 30th September 2008.

Imagine this scenario - an emergency responder requires time-critical assistance from other public safety agencies such as bomb squads and paramedics at an unspecified location. The responder is able to instantly pull out the portable cognitive radio and discover other public safety networks in proximity and immediately communicate with them in ad-hoc fashion. Using the portable radio the responder can also request assistance from agencies across the city. A public safety network incorporating service discovery can achieve exactly this goal. This is the motivation for the latter portion of this work.

1.2 Contributions and Related Publications

In the thesis, a remote control and service access scheme is developed and implemented for the existing system used in the Virginia Tech Center for Wireless Telecommunications (CWT) Public Safety Cognitive Radio (PSCR). The Vehicular PSCR (VPSCR) is introduced as a vehicle based modification of the PSCR, consisting of specially designed hardware for a public safety vehicle.

First, the remote control scheme is developed as an extension to the existing hardware and software code. The remote control scheme is implemented on a personal digital assistant (PDA) that wirelessly connects to the VPSCR. Second, a method for authentication, state maintenance and accessing hardware features is developed. On the PDA side, code is written to access the audio I/O hardware along with a feedback-control scheme for VPSCR control. Bi-directional Push-to-Talk digital audio processing code is incorporated into the VPSCR. An ergonomically designed graphical interface is designed on the PDA for ease of use.

In the second part of the thesis, the concept of public safety networks is extended to include a wide area network comprised of multiple VPSCR and PDA nodes. A network architecture is proposed that uses a small number of backbone nodes to interconnect principally self-contained public safety zones. A novel service discovery scheme is developed for optimized distribution of services among the nodes in the neighborhood as well as for remote locations. This integrated service discovery and routing protocol, based on optimized link state routing (OLSR), is designed to minimize overhead and to use multiple independent mechanisms for proximity based dissemination of services. The design of such a service discovery protocol is the primary contribution of the second half of this research work. The service discovery protocol is simulated in a discrete event network simulator and the results are discussed and analyzed. The simulation validates the effectiveness of the protocol against other simulated service discovery methods.

The PDA-VPSCR extension is described in:

R. Rangnekar, F. Ge, A. Young, M. Silvius, A. Fayez and C. W. Bostian, "A Remote Control and Service Access Scheme for a Vehicular Public Safety Cognitive Radio," in *2009 IEEE Vehicular Technology Conference (IEEE-VTC 2009)*, Anchorage, Alaska, Sept 2009. (Accepted for publication)

1.3 Thesis Organization

In Chapter 2, the background of the PSCR, hardware and its existing public safety features is described along with a short introduction to cognitive radio and its specific implementation at the CWT. The lack of portability of the PSCR is described and a solution to address this issue - namely the remote control scheme. Next, a description of the remote control and service access scheme is explained along with the implementation details describing the control logic and hardware interface blocks. The audio processing details are thoroughly examined for both the VPSCR and the PDA. Some screenshots of the PDA depicting the finalized product are compared to the original Java GUI of the PSCR and the operations are detailed in the conclusion.

In Chapter 3 of this thesis, the architecture for a public safety network utilizing the VPSCR and PDA devices is proposed. An introduction to MANETs and existing service discovery schemes is given and their inability to be used in public safety networks is detailed. A public safety network and service discovery scheme to remotely and efficiently access these services is presented. Finally, the service discovery implementation and algorithm is described as part of the simulation model and the optimizations are highlighted. Results of the simulation are given at the end of the chapter and the evaluation in favor and against the various schemes is discussed in light of service discovery in public safety networks.

Chapter 2 : Portability for Cognitive Radio

2.1 Introduction and Background

This section will review some of the research that is being conducted at the Center for Wireless Telecommunications (CWT) at Virginia Tech. An introduction to Cognitive Radio and the Public Safety Cognitive Radio, including the hardware, is presented. Some of the aspects of the existing system are discussed as well as their limitations.

2.2 Cognitive Radio

The concept of software defined radio (SDR) was introduced by Joseph Mitola III in his 1992 paper [1] to describe a radio capable of performing traditional hardware functions like mixing, modulation and filtering completely in software. Since then, due to the advances and recent ubiquity of general purpose processors, software defined radio has become a reality [2]. A typical SDR platform consists of RF front-end hardware to capture and transmit samples and a personal computer for digital signal processing in software. The term SDR is generally used for a system with software processing capabilities though with limited reasoning and decision making.

In 1999, Joseph Mitola III and Gerald Maguire, Jr. [3] introduced the term 'Cognitive Radio' (CR) to describe an evolved SDR capable of sensing the environment and fully reconfiguring itself when interacting with other users in the environment and acting on a users requirements. At CWT we previously defined a CR in [4] as :

> "*A transceiver that is aware of its environment and can combine this awareness with knowledge of the user's priorities, needs, operational procedures, and governing regulatory rules. It adapts to its environment and configures itself in an optimal fashion. The radio learns through*

experience and is capable of generating solutions for communications problems unforeseen by its designers."

Since the explosive growth of cognitive radio research, many applied research interests offering an application of cognitive radio functionality including – Spectrum Sensing, Spectrum Management and Allocation and Spectrum Sharing [5] to make efficient use of spectrum resources and prevent interference have spawned [6].

2.2.1 Cognitive Engine

At the heart of a CR, lies the cognition cycle which was first described by Mitola [3]. At CWT we defined a Cognitive Engine (CE) loop based on Mitola's original definition of the cycle and modified the process for practical implementation in hardware. Our CE loop [4] intended for radio implementation consists of "observe", "orient", "plan", "decide", "learn" and "act" states as shown in Figure 1.

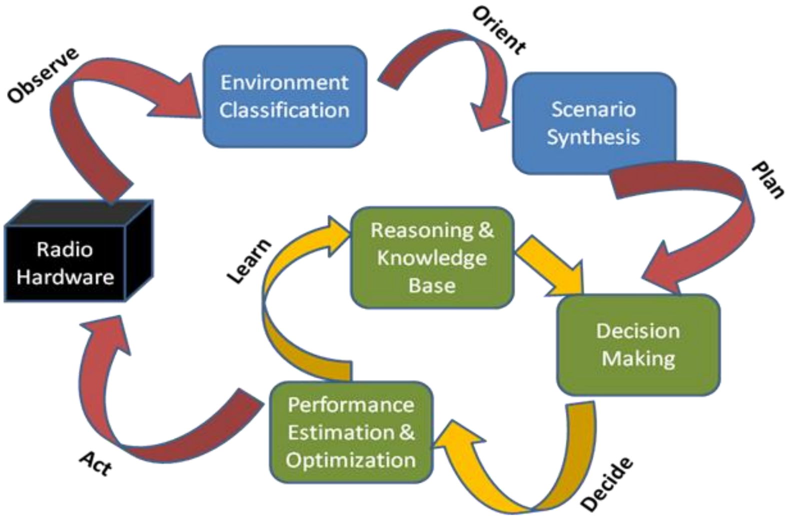

Observe
Environment Classification
Orient
Scenario Synthesis
Plan
Radio Hardware
Learn
Reasoning & Knowledge Base
Decision Making
Performance Estimation & Optimization
Act
Decide

Figure 1. CWTs Cognitive Loop

In our CE implementation [4], the cognitive controller manages the associated function states by issuing commands to connected components thereby traversing the cognition loop. For the hardware implementation, the commands are issued using network sockets and standard XML files.

- **Environmental Classification:** Sensors observe the external environment and internal states through the radio hardware and provide an external input to the loop.

- **Scenario synthesis:** Perform scenario synthesis and estimate the resultant preliminary solution using existing knowledge.
- **Decision Making:** Decide the course of action using the preliminary solution.
- **Optimization:** The multi-objective wireless system genetic algorithm (WSGA) [4] performs link configuration optimization using the generated solution from the decision maker. The optimizer produces an updated solution.
- **Reasoning and Knowledge Base:** The policy verifier and the decision maker verify the optimized and updated result and modify the knowledge base with the optimal settings.
- **Radio Hardware:** The radio hardware receives the optimal settings and reconfigures itself for the updated scenario modifying parameters such as transmit power, modulation, coding, symbol rate and spectrum shaping, to achieve a required user setting. More details are available in [7].

Therefore, we describe the cognitive engine as automatically “reading the meters” (i.e. sensing the environment) and “turning the knobs” (i.e. radio reconfiguration) in its complete operation around the cognition loop.

2.2.2 CR Application: Radio Interoperability Problem

Today’s Public Safety agencies and first responders range from the police and emergency rescue to fire departments and the armed forces. A significant problem in radio systems deployed by such agencies is the lack of interoperability among them [8]. Public Safety interoperability is expected to be one of the biggest applications of CRs [9]. The lack or the loss of infrastructure is a huge challenge to first responders when dealing with situations such as September 11, 2001 [10] and Hurricane Katrina in 2005. The US Federal government has proposed the SAFECOM and RapidCom programs [11] which endeavor to enforce public safety interoperation among agencies, however the proposals seek to integrate interoperability within the devices themselves which leaves currently deployed devices in the lurch. This problem is a major research focus of the CWT.

2.3 Public Safety Cognitive Radio

Public safety interoperability is one of the foremost applications of CR which will equip first responders at the scene of a disaster, with the tools to communicate with other first responders and possibly save lives and property. The prototype Public Safety Cognitive Radio (PSCR) is an application of CWT's CE technology in the public safety domain [12]. Briefly, the objective includes – the ability to scan and classify the public safety spectrum, multiple frequencies and multiple waveforms and to reconfigure its operation to communicate with these waveforms. The CE technology provides the waveform recognition abilities as well as access to previous decisions using existing data from the waveform knowledge database. The PSCR aims at providing universal interoperable communication of voice and data, and in the future possibly video, to solve the wide spread and severe interoperability problem: that various incompatible public safety waveforms cannot communicate with each other. The PSCR is designed to scan and indentify existing public safety waveforms and networks, and then to be reconfigured to talk to any detected channel on user command within defined operational limits of time. Further, it serves as a gateway to bridge incompatible waveforms, different frequency bands, and networks to each other.

It can be concluded that the PSCR is a unique application of cognitive radio technology and is an indispensible tool in the hands of public safety agencies and its widespread adoption is in the public interest.

2.3.1 PSCR Architecture

The PSCR is designed using a modular architecture where individual components exchange data using sockets or standard XML definitions [13]. The original PSCR modular architecture is shown in Figure 2, and is discussed thoroughly in [12].

The Radio hardware platform chosen for the PSCR is the ubiquitous Universal Software Radio Peripheral (USRPv1) [14]. The USRP is connected to the personal computer and is operated using GNU Radio [15] drivers and libraries with additional code for CR functionality. The spectrum sweeper utilizes the GNU Radio flowgraphs to scan the spectrum in the specified frequency range and then isolate individual channels using a power spectral density (PSD) sensor and a hierarchical fast Fourier transform (FFT). Signal classification uses a feature based pattern recognition scheme with a k-nearest neighbor cluster after obtaining phase lock [16]. Once the waveform is recognized, its parameters which include frequency, bandwidth, modulation, received power and channel ID among others are stored in the knowledge base for future use.

The case-based waveform solution maker interfaces with the knowledge base and the control and graphical user interface (GUI) to produce an estimated waveform solution for the user requirements. The solution is refined by the wireless system genetic algorithm [17] process and the solution XML files [13] (waveform and radio configuration) are processed by the GNU Radio based radio interface. The radio interface automatically reconfigures itself with the parameters specified in the XML files.

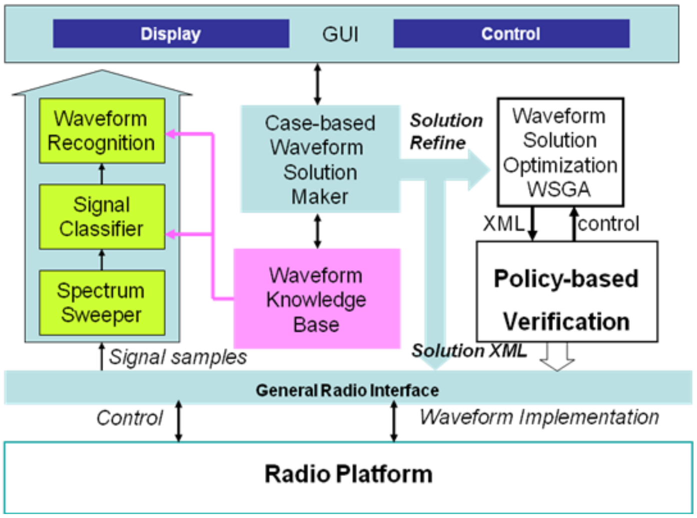


Figure 2. CWTs Original PSCR Architecture

The GUI is Java based and the code for the signal sweeper and classifier is written in machine independent C++. The default frequency values and waveform configuration XML file names are contained in a MySQL database. The architecture is platform independent, however for the prototype it was implemented and tested on a laptop running Ubuntu Linux 7.04 to 8.04, gcc/g++ version 4.04 to 4.3.0, GNU Radio 3.1.1, MySQL 5.1 and Java SE 6u10.

2.3.2 PSCR Modes and Features

There are three modes of operation of the PSCR which provide access to public safety features.

1. **Scan Mode:** The PSCR detects any active waveform in a specified frequency range and classifies its waveform parameters and modulation type.
2. **Talk Mode:** The PSCR can establish an audio or data connection with any standard detected public safety waveform and automatically reconfigure itself using the waveform parameters.
   a. **Analog Talk:** An analog audio link between the PSCR and the public safety waveform is established using an analog modulation scheme such as narrowband FM (NBFM).
   b. **Digital Talk:** A digital data link between the PSCR and the public safety waveform using digital modulation such as BPSK, QPSK or MPSK is established. The PSCR can also been used to communicate with P25 [18] waveforms as well as other PSCRs.
3. **Gateway Mode:** The PSCR can bridge two public safety waveforms, with incompatible modulations and different frequencies, together to allow radio interoperation.

2.3.3 PSCR Hardware Components

The PSCR hardware consists of a desktop or laptop computer running the software, the USRP and two daughterboards with multiband antennas and USB connection cable. In a cognitive radio, all signal and sample processing occurs in software on the general purpose processor which provides the needed flexibility.

The following sections briefly describe the USRP hardware and the removable daughterboards used in the prototype hardware.

2.3.3.1 Universal Software Radio Peripheral

The USRP is a popular RF hardware front end available from Ettus Research [14] and is used extensively for software defined functionality in the research community [19]. The USRP provides raw samples (two I-Q data channels each for input and output) through the USB 2.0 interface to a connected host device. The host is usually a personal computer with a general purpose processor. The USRP contains an Altera Cyclone series FPGA that performs signal processing for transmit and receive paths. The receive path uses a mixing and decimation unit and the transmit path has an interpolation unit. The AD9862 mixed signal chips perform ADC and DAC conversions. The USRP supports two RF daughterboards which are available in a variety of flavors from DC to 5.9 GHz. The 32MBytes/s USB interface coupled with the 64 Msamples/s and 16-bit signed I-Q integer samples provide around 8 MHz in Nyquist spectral bandwidth.

2.3.3.2 USRP Daughterboards

The daughterboards in the PSCR use the RFX400 board on Side A and a modified RFX900 on Side B. (Side A supports 400-500 MHz and Side B supports 703-973 MHz). The prototype PSCR therefore supports interoperation with family radio service (FRS) and general mobile radio service (GMRS) handheld FM radios in the UHF band. In the 700/800MHz band, the PSCR was successfully tested for interoperation with the EF-Johnson 5100 series radios [20].

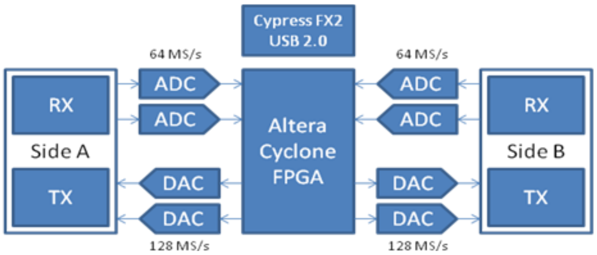


Figure 3. USRP Simplified Architecture

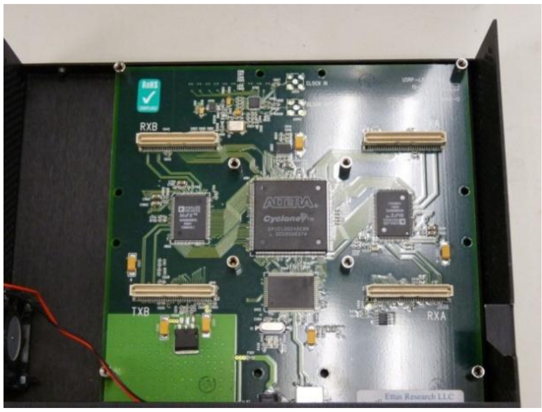


Figure 5. USRP Hardware

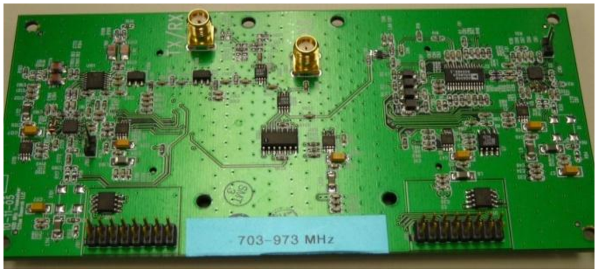


Figure 4. USRP Daughterboard

2.4 Remote Access for the PSCR

2.4.1 Problem Statement - Lack of Portability of the PSCR

Ability to interoperate with other users can provide universal cooperation among public safety agencies which has been a subject of discussion recently [21] and [22]. The PSCR described in the earlier section enables a first responder to exploit the advances in cognitive radio technology where it counts the most – to save lives.

A public safety cognitive radio system, as discussed earlier, has a considerable computational overhead which includes not only digital signal processing (DSP) in a personal computer, but also the computational support responsible for cognition, reasoning, decision-making, networking, database functions, etc. [23]. The computational requirement of such a system using software based DSP is demanding [24] and power intensive. Power consumption and battery capacity are traditionally the significant factors that decide the limit on computing speed. However, the rapid increase in computing speed, steadily following Moore's law, has far outpaced generational advances in battery technology [25]. Therefore, currently available software defined radios (SDRs) which support full cognitive radio development for public safety lack mobility because of battery and power constraints. Further, because they are saddled with a personal computer backend, they are also limited by physical size which is much larger than any hand-held devices [26].

In handheld mobile devices, system designers have to increase the processing power of microprocessors to handle the high data volume and number crunching that today's applications demand, simultaneously keeping heat generation low, and minimizing the power consumption [27]. As a result of this trend, current generation mobile processors operate at lower CPU clock operating frequencies compared to processors used in personal computers. As an example, next generation mobile processors are expected to provide greater computing performance at lower power consumption – ARM's Cortex-A8 [28] delivers up to 600MHz-1GHz and 2000 DMIPS

(Dhrystone Millions of Instructions Per Second) with a 300mW power budget. In the future, the ubiquity of such devices will ensure full cognitive radio development on handheld devices; however, at present a suitable medium between portability and performance is needed.

In the following section, CWTs cognitive radio technology incorporated in the PSCR [23] is applied to public safety communications for vehicles and a prototype Vehicular Public Safety Cognitive Radio (VPSCR) that can scan and classify the public safety spectrum (multiple bands and multiple waveforms, multiple modulations) is described. It can also configure itself to interoperate with any public safety waveform that it finds, within operational limits, of determining that a signal is present. The VPSCR aims at providing universal interoperable communication of voice and data, exactly similar in operation to the PSCR, to solve the radio incompatibility problem [29] through hardware designed for a public safety vehicle. This provides a solution to the accessibility as well as power consumption requirements.

However, like other SDR platforms, the SDR based VPSCR's power consumption is very demanding – and this has not solved the portability problem as yet. A solution to this problem is proposed:

> "*To extend cognitive radio functions for public safety responders in mobile environments, an extension to the VPSCR is designed so that a handheld mobile device like a PDA can discover and remotely access its services wirelessly or across a network.*"

Such extensions have previously been designed to extend operating system functionality to PDAs [30] however extending functionality for cognitive radio is a relatively new idea. The lack of portability and processing requirements of a public safety cognitive radio system were described in the previous section. The immediate requirement is a scheme in which portability can be achieved without compromising on the capabilities of a full-featured cognitive radio in the public safety domain.

2.4.1.1 Motivation

Until the ability of handheld devices achieves a significant threshold for the processing requirements of an SDR platform, the cognitive radio community must rely on

GPP based platforms. Second, a majority of development in the software domain for signal and sample processing is not suitable for the mobile environment. The secondary reasons which factor into this are – lack of reconfiguration, limited software API libraries, minimalistic Input/Output (I/O) interfaces and manufacturer dependent hardware features. Third, battery drain is directly proportional to the computing power of the device; this leads to a trade-off between battery capacity and processing power. It can therefore be concluded that building a standalone mobile public safety cognitive radio, that supports the features of existing systems is, at present, a huge challenge.

This chapter is divided into the following sections – Architecture, State Diagram and Control flow, Network access description, Audio framework Implementation and a description of the GUI layout for the proposed scheme. These sections will describe the changes to the system from the existing PSCR as well as modifications and upgrades to current features. The audio framework will detail the method and protocols used for accessing low level audio from the devices and processing the samples. The final section will compare the usability of the GUI layout with the existing PSCR Java GUI and will highlight some of the ergonomic and usability features of the design.

2.4.2 Approach - PDA based remote access

Remote access over a network requires a client and a server implementation for a user to request features from a connected system. This implementation assures the user of access to the PSCR system over a widely deployed wireless network such as a campus 802.11 (Wi-Fi) system spread over a large area. Alternatively, the capability to establish a wireless ad-hoc network is also provided. In fact, any IP based network between the end points is acceptable for remote access. Multi-hop networks which do not perform port blocking or network address translation (NAT) [31] for unicast UDP along the segments, are also supported.

Henceforth, the PSCR will be referred to as the Vehicular PSCR (VPSCR) to indicate the changes to the original hardware and its ability to allow remote access. The VPSCR is specifically designed to be installed in the trunk of a public safety vehicle and consists of battery backup and optional inverter to power the box.

2.4.3 Vehicular PSCR hardware

The public safety officer is equipped with a PDA capable of remotely accessing the VPSCR. The VPSCR is situated in the public safety vehicle and is powered by the vehicular power supply – the laptop is fixed firmly on the box shown in Figure 6. There are external connections for the power supply, dual antenna connectors and laptop cables. For external viewing of the laptop screen, a VGA cable could be used to connect the external monitor output to a miniature LCD screen placed near the dashboard. A small form factor USB keyboard may be connected to the laptop for command input. It should be noted that operating the PDA remotely does not require any input or screen display from the vehicle. This provides the public safety officer with the ability to interoperate with public safety waveforms using the PDA, as depicted in Figure 6 and Figure 7.

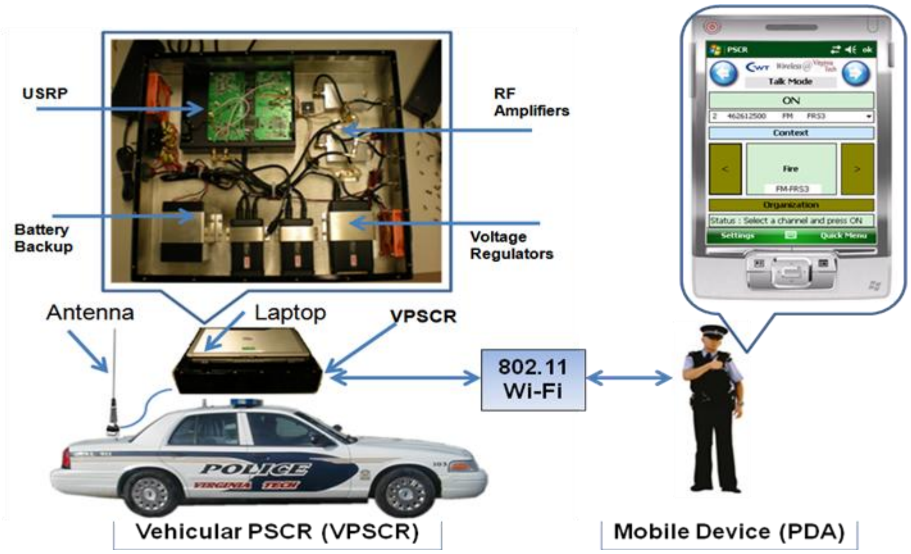


Figure 6. VPSCR and PDA Scenario

The USRP is connected to the laptop using USB and has connections to the antennas on the external side of the box. The two multi-band antennas are specifically designed for the dual frequency range supported by the individual daughterboards. The operating frequency range of the VPSCR depends on the installed daughterboards.

Network access is achieved by the inbuilt 802.11b/g Wi-Fi and thereby ensuring that both the laptop and the mobile device support similar wireless standards. Potentially the system can support a wireless link using Bluetooth (802.15.1) or a mutually agreed on physical and data link model that is capable of IP based networking. The wireless link types tested on the prototype device included access point based (single-hop infrastructure), wide area infrastructure (multi-hop) and ad-hoc mode (in single-hop). Data link encryption using WPA/WPA2 security prevents sniffing and eavesdropping, simultaneously alleviating the overhead of encryption from the application layer. This describes the structure of a single PDA-VPSCR connection. The Remote Access Architecture supports inbuilt routing, in order to forward control and audio streams from a PDA through multiple VPSCRs.

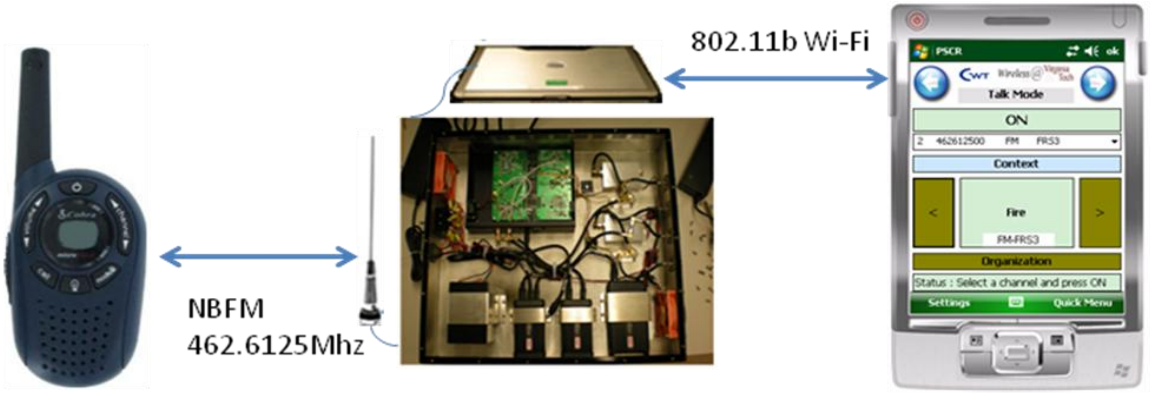


Figure 7. PDA Interoperation

2.4.4 Remote Access Architecture

The PSCR Cognitive technology is based on the open-source GNU Radio framework. The framework serves as an extensible platform to build processing and control logic and also contains a number of predefined blocks that may be utilized. All processing occurs using C++ code blocks; control logic is in Python and user interaction, GUI display, and as network services are coded in Java. The extended architecture for remote access is shown in Figure 8 and with modifications to the GUI, separation of the mode control functions, addition of the remote agent and service discovery blocks as well as new audio processing functions for processing digital audio.

GUI and Control logic: A majority of modifications have occurred in the GUI and control logic. The new components for providing remote access and service discovery (RAS) are separate threads which run in the background waiting for remote commands. The GUI supports simultaneous manual operation and remote access, although the hardware features cannot be used simultaneously. The results and operations from the remote commands can be displayed on the GUI or can be disabled. Visual additions to the GUI for the PSCR include a new screen to display the status of the PDA and the IP and port information of a connected device.

Audio Services: The separate Audio Service Thread executes independently and is controlled by the RAS. The optional vocoder allows for audio compression and decompression of raw RCM audio. The received audio can be raw PCM or may be vocoded using an implementation of G711a/u (a C# implementation such as [32] can be adapted for the PDA). The audio thread contains an implementation of a UDP client and server which acts as a client and source for PCM audio. The raw audio is received by the server, de-packetized and parsed through the vocoder which then sends expanded audio to the GNU Radio General Interface flowgraph where the actual audio processing occurs.

Control Blocks: The mode control block maintains a state machine of the current mode (modes discussed in the next section). Modes are simultaneously operated between the physical user at the GUI as well as the remote user. Currently, the supported modes are – Scan, Gateway and Talk, each with remote access and local capabilities. The RAS block manages the control interface for network access and provides state updates to the PDA.

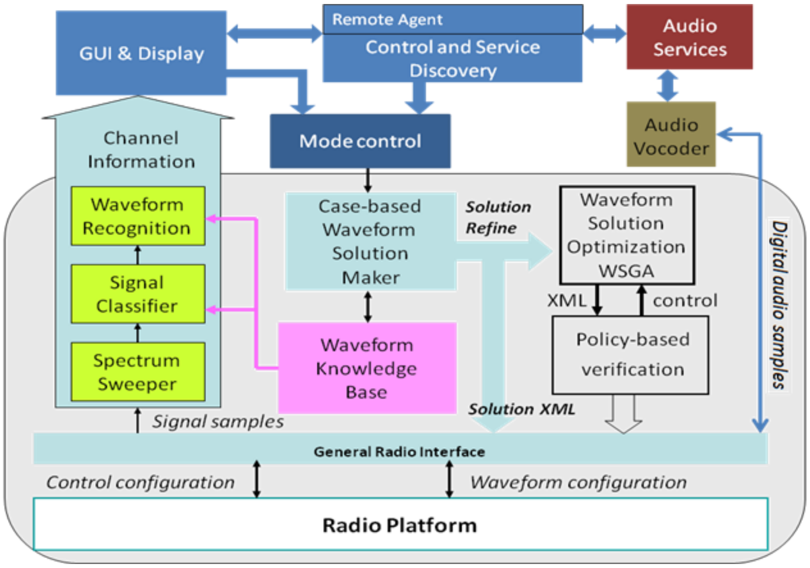


Figure 8. Extended VPSCR Architecture for Remote Access and Service Discovery

2.4.5 Control and State Sequence

A typical service request and audio talk scheme is shown in Figure 9. The user, through a PDA, initiates a service request from the VPSCR. The VPSCR responds with the list of services that it offers. Any VPSCR can offer this feature without prejudice. In the latter half of this thesis, it will be demonstrated how services could be automatically disseminated along the network along with integrated routing. If the services offered by the VPSCR satisfy the user requirement, the PDA can initiate an association request. The association request is designed to use password or other simple authentication methods. However, for this single link implementation prototype we use an IP table lookup based authentication method; although unsecure, it is designed for a single hop link which is encrypted by 802.11 data-link security. On successful authentication, the PDA is ‘Associated’ with the VPSCR. This association procedure allows the both the PDA and

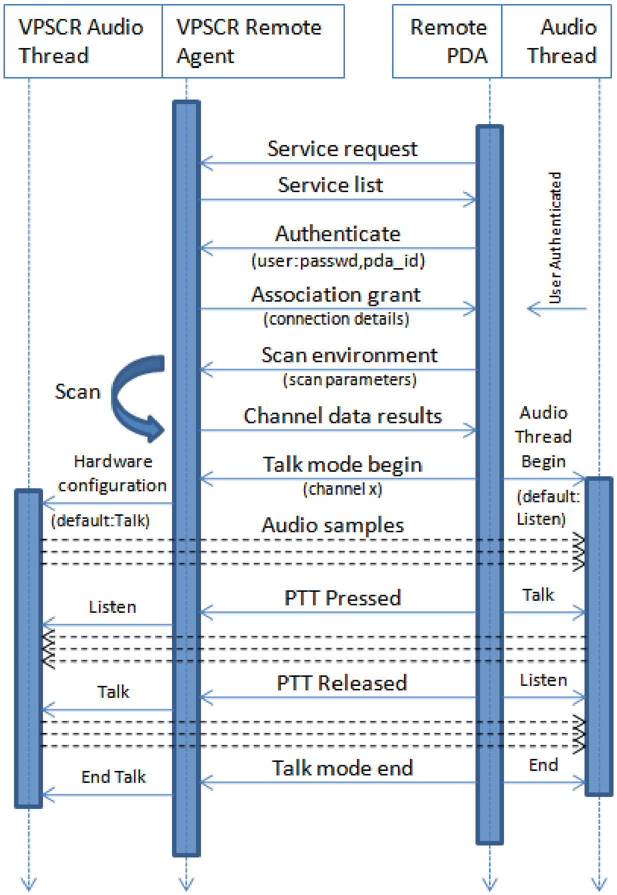


Figure 9. Typical Network Access Sequence

the VPSCR to initiate the ‘Association’ procedure and either device to accept or deny the connection. In most cases, the PDA will initiate the association procedure.

The noteworthy feature of the service request mechanism, in general, is that it is device initiated rather than advertised. This ‘pull’ based method is designed for a single hop direct link and for more complicated topologies; a hybrid push based scheme is developed in the latter section. This architecture also provides the VPSCR the ability to negotiate with another VPSCR for services that it may not offer.

Association involves authentication as well as mutual exchange of required port numbers – control ports and audio streaming UDP ports for each device. After association the VPSCR waits for commands from the PDA. Commands are described in Table 1. Commands and audio streams are processed separately by different threads. In the audio talk mode, the PDA supervises the transmission direction (i.e. listen/talk).

The Figure 10 shows the state machine involved in processing remote commands at the VPSCR. The VPSCR needs to be in the associated state before it can process any hardware related functions. Each command proceeds through a complete processing loop before returning data to the remote user. Audio streams are handled by the existing Framework developed in GNU Radio with several features added for processing of digital audio.

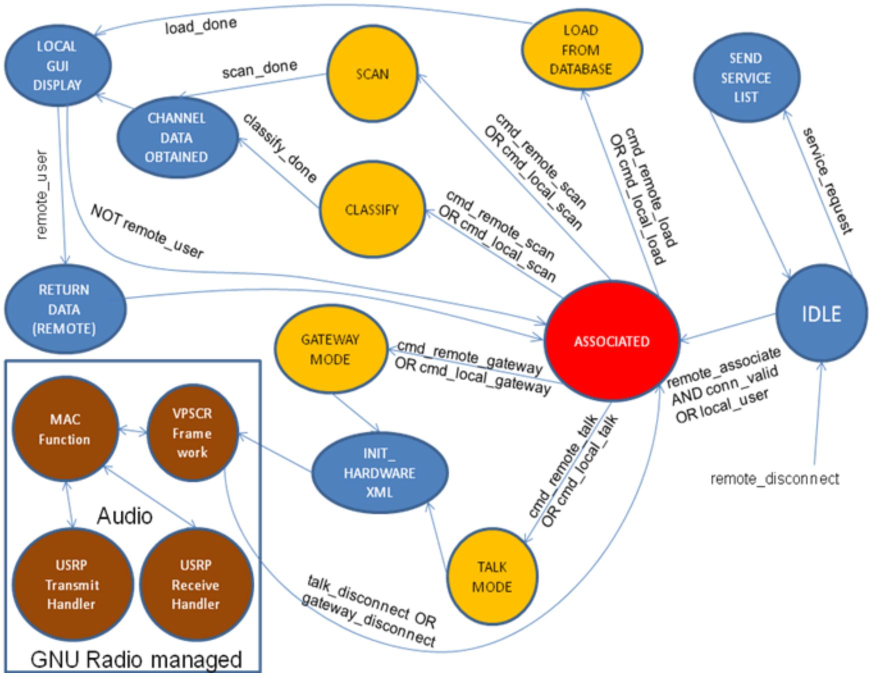


Figure 10. VPSCR State Machine

PDA commands		VPSCR response(s)	
Command	**Function**	**Response**	**Function**
sweep	Scan for active waveforms	sweepdata, *2, *3	Returns the results and channel data
classify	Classifies the found waveforms	classifydata, *2, *3	Returns the results and channel data
sweep_classify	Local channels from database	sweepclassifydata, *2, *3	Returns the results and channel data
gateway	Bridge waveforms	*1,*2,*4	
gateway_disconnect	Disconnect bridge	*1,*2,*4	
talk_connect	Connect to waveform	*1,*2,*4	
talk_disconnect	Disconnect talk	*1,*2,*4	
talk_ptt_pressed	PTT pressed on PDA	*1,*2,*4	
talk_ptt_released	PDA PTT released	*1,*2,*4	
pscr_standby	Stop all operational modes		No acknowledgement
restart_pscr	Restart the software processes		No acknowledgement
ping_pscr	Check connection status (alive) and parameters	ping_response. *3	Returns connection parameters
associate_pscr	Initiate association from PDA and exchange parameters	A1,A2, *3	Returns association status and connection details

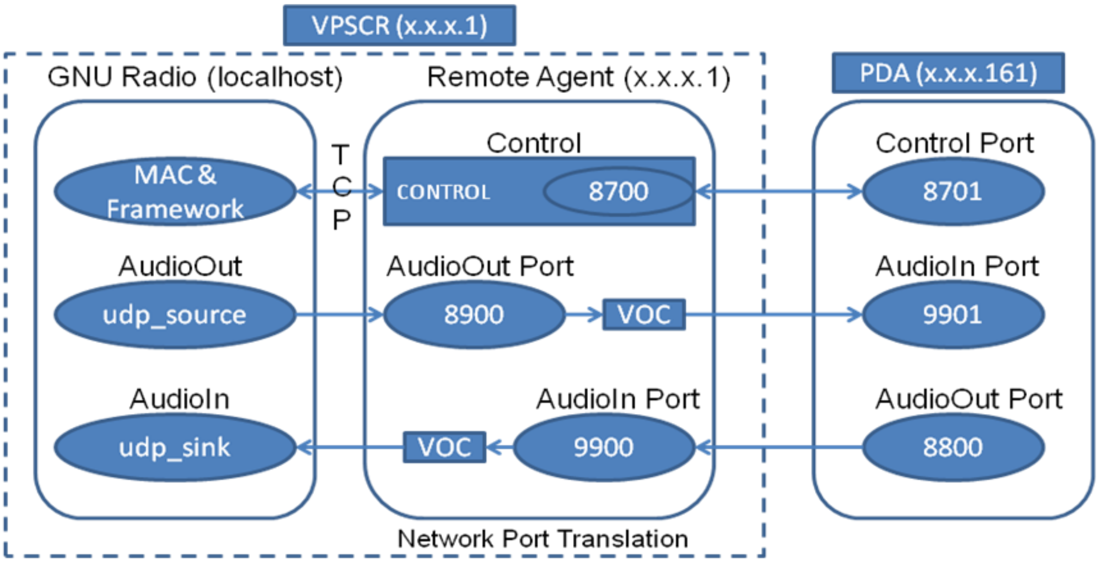
For description of the packet structure refer to Appendix B	*1 - success : Success on command *2 - usrp_error : Hardware error or disconnected / Failure on command A1 - associated : VPSCR is now associated A2 - currently_associated : VPSCR is already associated *3 - No response : PDA will timeout *4 - No response : Currently unhandled

Table 1. Network Access Commands and Responses from the PDA and VPSCR

2.4.6 Network Access

The VPSCR is connected externally by two radio interfaces – 802.11 Wi-Fi and the USRP RF frontend. Control and digital data transmission occur over the 802.11 interface and public safety interfaces connect through the USRP.

All communication between independent processes within the VPSCR occurs using either TCP or UDP sockets shown in Figure 11. TCP is used where reliable communication is required and communication is not delay sensitive, such as inter-block communication among the MAC and Framework processes and the General Radio Interface as control commands. The commands can be externally issued using the TELNET [33] functionality already existing in the framework.



(All communication shown is over UDP unless indicated otherwise) VOC = Pass-through vocoder

Figure 11. Network Port Diagram and Flow

UDP communication is used over lossy and unreliable links, and for real-time communication which is tolerant to moderate packet loss such as audio streaming. UDP control channel link over 802.11 is used for reduced overhead and quick response at the expense of reliability since end to end reliability is provided by the inbuilt state machine in the RAS.

The port diagram describes the communications framework which uses bi-directional port mapping, a term described as network port translation (NPT). The Remote Agent (RAS) acts as an intermediary between the GNU Radio and the PDA. The RAS performs port translation which re-maps the port values from packets and forwards them to the appropriate destination. The vocoder block lies at the RAS and performs pass-through-vocoding (on-the-fly) on the raw audio from GNU Radio and the packets are passed to the PDA AudioIn Port. In the reverse direction, audio from the PDA is pass-through-vocoded and sent to GNU Radio as raw audio. Control commands are interpreted and appropriate instructions are issued to the framework for configuring the RF hardware.

2.4.7 Audio Framework Implementation

As with any bi-directional real-time streaming audio implementation, the PDA-VPSCR audio was a challenge from the beginning, especially since the components of the system used multiple audio buffer agents and programming languages. The use of various platforms (C++, Python, Java, VB.NET CF 2005) implies that the audio processing must strive to be as language independent as possible. For this reason raw PCM audio at standard sample rates and UDP packetization is used as the default scheme. It is noted that this scheme can be easily generated using any of the modern high-level platform languages. The default scheme for audio is shown in Table 2, with each set of parameters used automatically for the type of source, destination devices.

Table 2. Audio Generation Scheme for (source, destination) pairs

Source Device	**Destination Device**	**Source Sampling**	**Quantiz-ation**	**Min Payload**	**Max Payload**
		(Hz)	(bits)	(bytes)	(bytes)
PDA	VPSCR/ Waveform	22,050*	8	44	1024**
VPSCR/ Waveform	PDA	16,000	8	44	512**
PDA	VPSCR/Local	16,000	8	N/A (flexible)	N/A(flexible)
VPSCR/Local	PDA	16,000	8	N/A(flexible)	N/A(flexible)

*resampling to standard integral sample rates is not possible, minor sampling mismatch is observed

**Practically half the maximum payload size is used

2.4.7.1 Audio Framework for the VPSCR

The audio framework is implemented as a combination of Java audio server blocks, vocoder, network port translation code, and GNU Radio based audio processing. The general sequence for audio received by the VPSCR is show in Figure 12. The reverse structure is implemented for audio sent the other direction.

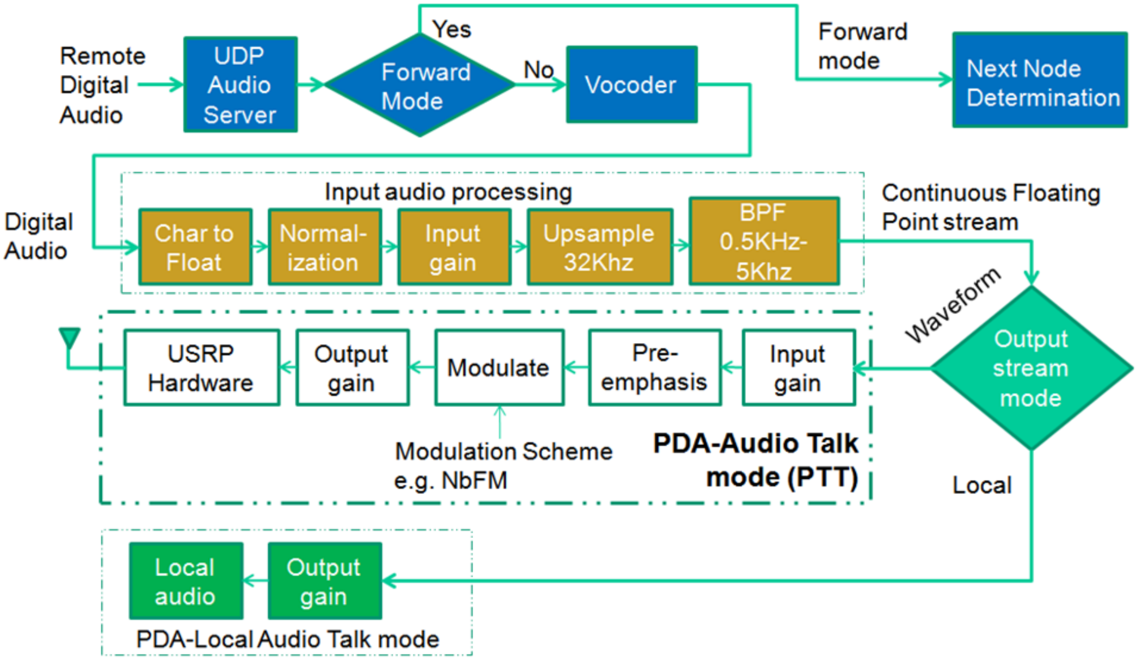


Figure 12. Processing Digital Audio on VPSCR

The remote audio in the UDP structure defined above is received by the UDP audio server at the RAS. If the VPSCR is in forwarding mode, the audio packet is forwarded to another VPSCR for processing, if not, the audio packet is sent to the pass-through vocoder which expands the packet. The expanded packet which consists of digital audio is now port translated to the GNU Radio flowgraph pda_analog_transmit_path.py. The char to float conversion separates the audio samples, taking care of the endianness (byte ordering), and converts individual samples into a stream of integer values (stored in float data structure) between 0.0 and 255.0. The normalization process converts this stream into pure floats and scales the individual samples between (±1.0). Gain is adjusted by a previously determined multiplication factor. Now upsampling is performed on the samples (ref Table 2) to obtain a standard rate of 32 KHz to match the varying input rate with that of the sound system and also achieve a higher time resolution. Resulting samples are filtered in the voice frequency range. Now the bit stream is a continuous sequence of floating point audio samples.

The output stream mode flag decides if the audio is to be transmitted to the radio waveform through the USRP or played back on the local VPSCR sound system. Audio sent to the USRP is passed through a series of blocks which are automatically configured to use the appropriate frequency, modulation, bandwidth and gain characteristics. This configuration is automatically generated through XML depending on the remote user's requirement at the initialization of the audio mode.

If audio is played back locally, the onboard sound system of the VPSCR acts as an audio termination device providing playback and recording. In this mode, the VPSCR laptop can be used for audio communication with the PDA bypassing the public safety interoperation and the USRP.

Table 3. GNU Radio Implementation Blocks

#	Function Block	GNU Radio Implementation Block
1	UDP Source	gr.udp_source
2	Char to Float	gr.uchar_to_float
3	Normalization	gr.multiply_const_ff, gr.add_const_ff
4	Input Gain (Stage 1)	gr.multiply_const_ff
5	Upsampler	blks.rational_resampler_fff
6	Band Pass Filter and Audio Interpolator	gr.firdes.band_pass, gr.fir_filter_ccf, gr.interp_fir_filter_fff, gr.firdes.low_pass
7	Input Gain (Stage 2)	gr.multiply_const_ff
8	Pre-emphasis (for NBFM)	fm_preemph
9	Modulation (for NBFM)	gr.frequency_modulator_fc
10	Output Gain (Stage 2)	gr.multiply_const_cc
11	USRP Hardware	usrp.sink_c
12	Local Audio	audio.sink

2.4.7.2 WaveIn/WaveOut Flowchart for PDA

The Microsoft Wave API is a standard set of application layer function calls on Windows operating systems and Mobile platforms [34] that allows low level access to audio input/output devices. This section will briefly describe the implementation of the wave structure and the flowchart for playback and recording. The audio sequence was designed by referring to the Microsoft Platform/Invoke Library Samples [35]. The buffer structures defined here are not specific to this implementation and could be used for designing the audio processing software on any supported mobile device.

The audio playback flowchart for the PDA is presented in Figure 13. The methods shown in yellow are Microsoft waveOut methods for playback on any Windows operating system. The waveOutGetNumDevs will return a list of devices and a check is performed to determine if any device supports the playback of audio. If not, the audio mode is terminated. On successfully locating a supported WaveOut device, the buffers for handling audio data at specified sample rates and quantization levels are prepared.

The buffer structure for playback consists of 4 circular buffers which are continuously being filled with incoming data over UDP. The UDP server present on the PDA, accepts the UDP audio from the VPSCR and extracts the samples to fill up the buffers in sequence. After the fourth buffer is full, writing begins at the first buffer. A message from the buffer handler indicates when each buffer is full. After each buffer is full, the buffer is copied to the waveOut buffer for playback using the waveOutWrite method. A mismatch in the playback rate and the incoming audio rate will cause silence to appear in between buffer blocks. This is can be overcome using compression of audio. When the user initiates PTT, the listen mode is paused using waveOutPause and reinitialized using waveOutRestart. The processing limitations of the PDA constrain the full duplex operation using the WaveIn/WaveOut method.

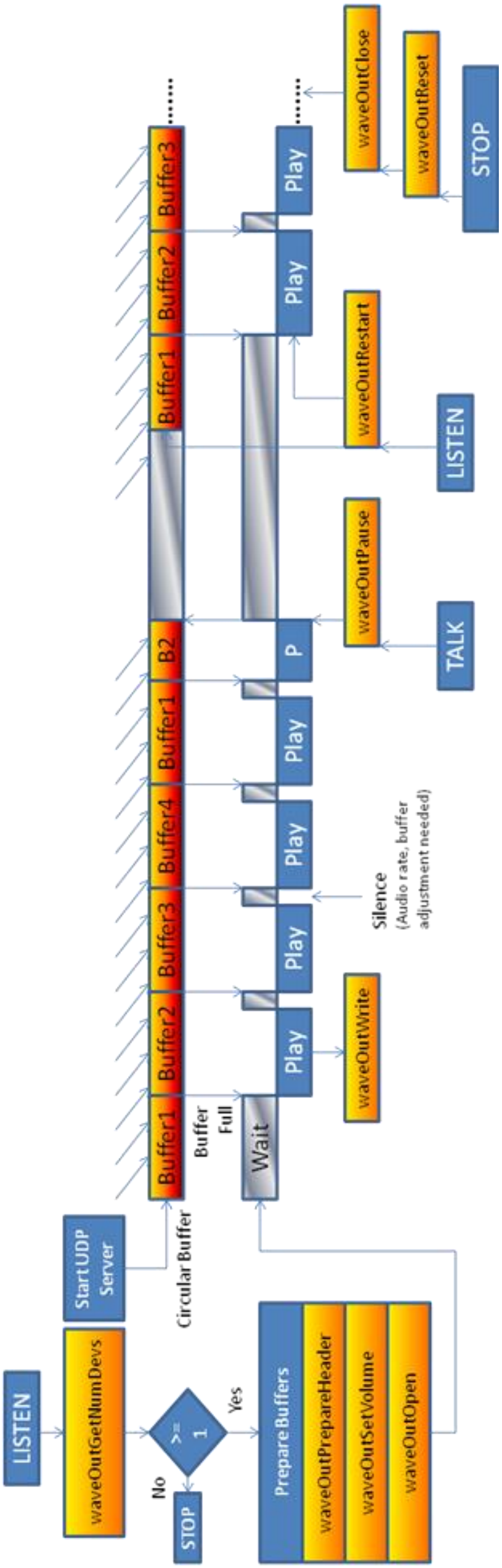


Figure 13. Listen and Playback of Audio on the PDA

The audio recording and talk flowchart is shown in Figure 14. The talk process is initialized only after the user presses the PTT button. The talk process interrupts the listening and begins recording. There is a short delay in initializing talk because 1) the VPSCR needs to suspend the current flowgraph and start another flowgraph, which happens once 2) the PDA needs to create buffers and initialize the headers using the waveIn methods.

The buffer structure for recording uses two circular buffers which are continuously filled with raw audio samples. After a buffer is full, the internal buffer handler informs the UDP sender method to packetize the full buffer and transmit the samples. The UDP sender method will fragment the packets depending on the number of samples stored in the buffer (based on buffer size). The circular buffer allows the PDA to record and send packets simultaneously.

When the PTT is released, the recording is completely terminated and control is passed to the listen handler. In contrast the waveOutPause method simply pauses the audio during listen mode. The maximum duration which a user can continuously talk is 30 sec, due to buffer size limitations, and the talk mode automatically switches to listen thereafter. The PTT can be re-initiated again after the 30 seconds is completed, an indefinite number of times.

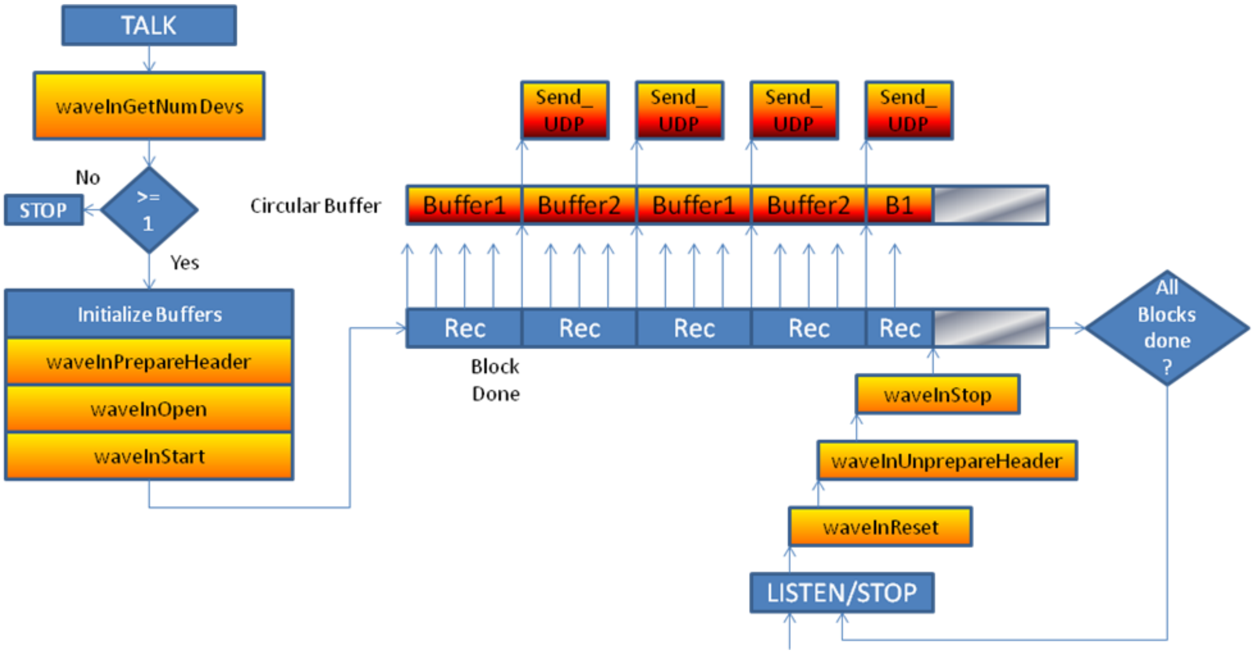


Figure 14. Talk and Recording on the PDA

2.4.7.3 Buffer structure for Wave API

The WaveIn buffer for recording uses a dual circular buffer system for parallel processing and transmission. The buffer size is selected in order to synchronize the recording rate with the playback rate on the VPSCR, accounting for the transmission delay and processing, thereby proving least number of discontinuities in the received audio. The UDP packet size selected as half the maximum size shown in Table 2.

The WaveOut buffer for playback uses four circular buffers for simultaneous playback and receiving digital audio. The buffer size and number of buffers is carefully selected to synchronize the rate of incoming audio with the playback rate and provide the smallest of silence gaps during playback.

The number of audio blocks (buffers) is given by Equation 1 for a single channel of audio, applicable for both playback and recording. The corresponding audio rate and buffer utilization is shown in Figure 15 and Figure 16.

$$numBlocks = \frac{duration * (samples / \sec) * (bits / sample)}{bufferSize}$$

Equation 1. Calculating the number of buffer blocks required

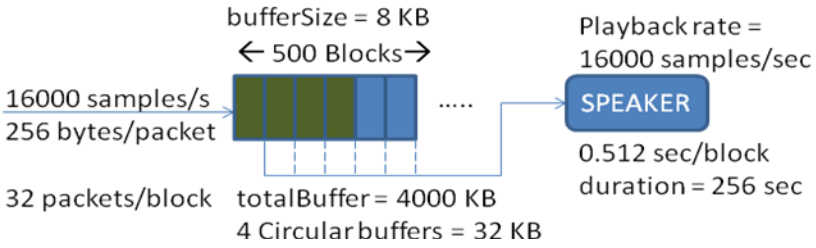


Figure 15. PDA Listen/WaveOut Audio Rate Calculation (500 blocks)

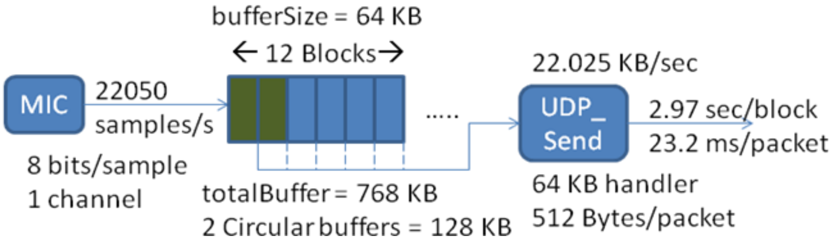


Figure 16. PDA Talk/WaveIn Audio Rate Calculation (30 sec duration)

2.4.8 PDA Implementation Results

The PDA used specifically for this project is an HP iPAQ 110 which is Windows Mobile (WM) 6.0 based and contains a Marvell PXA310 624 MHz processor and 64 MB of SD RAM. Potentially other PDAs based on WM 5.0/6.0/6.1 can be used since they support the 2005 .NET Compact Framework libraries. The PDA application was developed using VB.NET 2005 CF; but, it could be developed using any of the .NET languages such as VC# and VC++. The preliminary design and testing environment consisted of the Microsoft Device Emulator 3.0 which allowed emulation of the PDA on a Windows machine before deploying on a physical device.

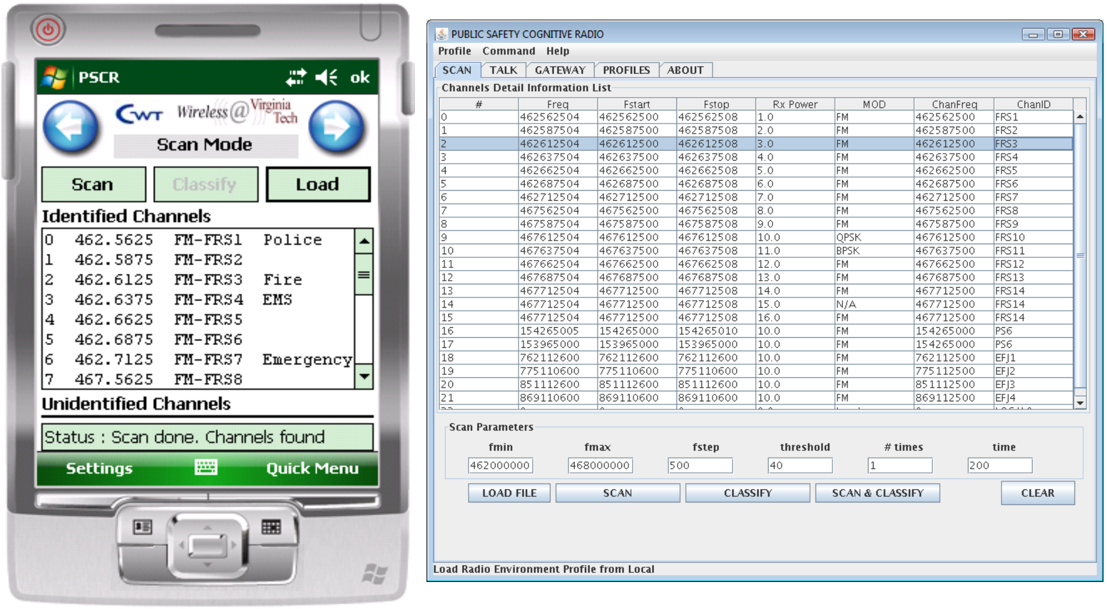
The PSCR supports the modes of operation explained in the earlier section. The remote access allows these modes to be accessed by remotely connected devices and to use the functionality of the PSCR. The graphical user interface (GUI) is ergonomically designed for minimum clicks and is similar is operation to a conventional push-to-talk radio. The easy to use mode scroll buttons provide quick access to the three modes. The operation of the PDA is described is further detail in Appendix A.

The layout of the GUI has been designed in collaboration with Gyu Hyun Kwon, graduate student, Human Factors Engineering and Ergonomics, Department of Industrial and Systems Engineering, Virginia Tech. Software verification and testing of the code was done by Nannan He and Xueqi Cheng, graduate students, Department of Electrical and Computer Engineering, Virginia Tech.

2.4.8.1 Scan Mode

This mode provides the “Scan” and “Classify” functionality for remotely detecting active public safety waveforms and classifying the modulation. The “Load” functionality allows the user to remotely load previously used waveform configurations. A waveform configuration consists of parameters defined in XML files waveform_*.xml and config_*.xml and is automatically generated by the framework. To the remote user, this entire operation is masked with a simple interface consisting of three buttons – Scan, Classify, and Load. The display for the returned channels splits the screen into an unidentified channel section and an identified channel section. The identified channels include those channels whose classification was successful and modulation type was obtained. The Scan button will scan for active waveforms and by default will return the frequency and other parameters such as transmit power, bandwidth and channel identifier. By default, the scan does not automatically classify the modulation since classification is more time consuming and thus all channel data returned by Scan is unidentified. The user can select an unidentified channel row and click ‘classify’. The frequency and channel information is now returned to the VPSCR for classification. If classification is successful, then the returned channel will be identified by a frequency as well as modulation type and listed as an identified channel. The Load button is used when communication with a known waveform such as FRS/FM is desired. The default channels loaded can be modified at the VPSCR database but not from the PDA.

For the convenience of the user, each channel can be assigned a channel name such as Police, Fire, Emergency etc. Channel names can be set independently on each PDA.



Identified Channels

0	462.5625	FM-FRS1	Police
1	462.5875	FM-FRS2	
2	462.6125	FM-FRS3	Fire
3	462.6375	FM-FRS4	EMS
4	462.6625	FM-FRS5	
5	462.6875	FM-FRS6	
6	462.7125	FM-FRS7	Emergency
7	467.5625	FM-FRS8	

Unidentified Channels

Status : Scan done. Channels found

Channels Detail Information List

#	Freq	Fstart	Fstop	Rx Power	MOD	ChanFreq	ChanID
0	462562504	462562500	462562508	1.0	FM	462562500	FRS1
1	462587504	462587500	462587508	2.0	FM	462587500	FRS2
2	462612504	462612500	462612508	3.0	FM	462612500	FRS3
3	462637504	462637500	462637508	4.0	FM	462637500	FRS4
4	462662504	462662500	462662508	5.0	FM	462662500	FRS5
5	462687504	462687500	462687508	6.0	FM	462687500	FRS6
6	462712504	462712500	462712508	7.0	FM	462712500	FRS7
7	467562504	467562500	467562508	8.0	FM	467562500	FRS8
8	467587504	467587500	467587508	9.0	FM	467587500	FRS9
9	467612504	467612500	467612508	10.0	QPSK	467612500	FRS10
10	467637504	467637500	467637508	11.0	BPSK	467637500	FRS11
11	467662504	467662500	467662508	12.0	FM	467662500	FRS12
12	467687504	467687500	467687508	13.0	FM	467687500	FRS13
13	467712504	467712500	467712508	14.0	FM	467712500	FRS14
14	467712504	467712500	467712508	15.0	N/A	467712500	FRS14
15	467712504	467712500	467712508	16.0	FM	467712500	FRS14
16	154265005	154265000	154265010	10.0	FM	154265000	PS6
17	153965000	153965000	153965000	10.0	FM	154265000	PS6
18	762112600	762112600	762112600	10.0	FM	762112500	EFJ1
19	775110600	775110600	775110600	10.0	FM	775112500	EFJ2
20	851112600	851112600	851112600	10.0	FM	851112500	EFJ3
21	869110600	869110600	869110600	10.0	FM	869112500	EFJ4

Scan Parameters

fmin	fmax	fstep	threshold	# times	time
462000000	468000000	500	40	1	200

Load Radio Environment Profile from Local

Figure 17. Scan Mode on PDA (left) and existing GUI on PSCR (right)

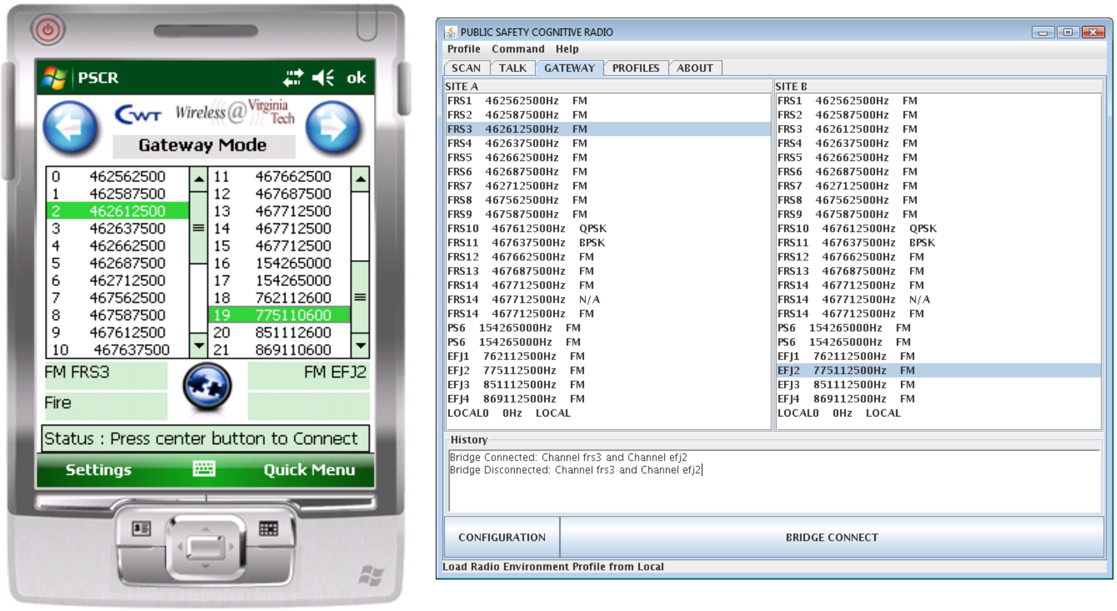
2.4.8.2 Gateway Mode

This mode provides bridging functionality for the PDA to allow a remote user to bridge two public safety waveforms together. Before two waveforms can be bridged, their channel modulations must be identified. In other words, Scan and Classify must be performed before bridging waveforms together. Alternatively, the Load functionality can be used to load a list of identified and saved channels.

Similar in operation to the VPSCR, the PDA interface provides an elegant two-listbox view and a single button for gateway operation. On the VPSCR, there were two buttons used for bridging waveforms together. One of these buttons initially set up the gateway framework (Configuration) and initialized the audio transmit and receive flowgraphs. Another button (Connect/Disconnect) was used to establish and later tear down the bridge. This functionality has been integrated into a single button interface which performs configuration and bridge formation together and later serves to tear down the bridge.

The two listboxes shown display an identical channel list and the user may select two different channels from either. The prototype VPSCR code is designed to use the lower frequency channel (RFX400 board) on Side A and the higher frequency channel (RFX900 board) on Side B. This configuration can be easily changed at the VPSCR. However, the user may select the channels on the PDA screen in any order. It should be noted that a daughterboard supporting the selected waveform must be present on at least one side of the USRP and the second waveform selected must conform to the frequency requirement of the other side of the USRP.

The PDA status bar provides the current state of the gateway, i.e. Connected or Disconnected. Errors in establishing the gateway are displayed in the same status bar.



0	462562500	11	467662500
1	462587500	12	467687500
2	462612500	13	467712500
3	462637500	14	467712500
4	462662500	15	467712500
5	462687500	16	154265000
6	462712500	17	154265000
7	467562500	18	762112600
8	467587500	19	775110600
9	467612500	20	851112600
10	467637500	21	869110600

FM FRS3 FM EFJ2

Fire

Status : Press center button to Connect

PUBLIC SAFETY COGNITIVE RADIO
Profile Command Help
SCAN TALK GATEWAY PROFILES ABOUT

SITE A

FRS1	462562500Hz	FM
FRS2	462587500Hz	FM
FRS3	462612500Hz	FM
FRS4	462637500Hz	FM
FRS5	462662500Hz	FM
FRS6	462687500Hz	FM
FRS7	462712500Hz	FM
FRS8	467562500Hz	FM
FRS9	467587500Hz	FM
FRS10	467612500Hz	QPSK
FRS11	467637500Hz	BPSK
FRS12	467662500Hz	FM
FRS13	467687500Hz	FM
FRS14	467712500Hz	FM
FRS14	467712500Hz	N/A
FRS14	467712500Hz	FM
PS6	154265000Hz	FM
PS6	154265000Hz	FM
EFJ1	762112500Hz	FM
EFJ2	775112500Hz	FM
EFJ3	851112500Hz	FM
EFJ4	869112500Hz	FM
LOCAL0	0Hz	LOCAL

SITE B

FRS1	462562500Hz	FM
FRS2	462587500Hz	FM
FRS3	462612500Hz	FM
FRS4	462637500Hz	FM
FRS5	462662500Hz	FM
FRS6	462687500Hz	FM
FRS7	462712500Hz	FM
FRS8	467562500Hz	FM
FRS9	467587500Hz	FM
FRS10	467612500Hz	QPSK
FRS11	467637500Hz	BPSK
FRS12	467662500Hz	FM
FRS13	467687500Hz	FM
FRS14	467712500Hz	FM
FRS14	467712500Hz	N/A
FRS14	467712500Hz	FM
PS6	154265000Hz	FM
PS6	154265000Hz	FM
EFJ1	762112500Hz	FM
EFJ2	775112500Hz	FM
EFJ3	851112500Hz	FM
EFJ4	869112500Hz	FM
LOCAL0	0Hz	LOCAL

History

```
Bridge Connected: Channel frs3 and Channel efj2
Bridge Disconnected: Channel frs3 and Channel efj2
```

Load Radio Environment Profile from Local

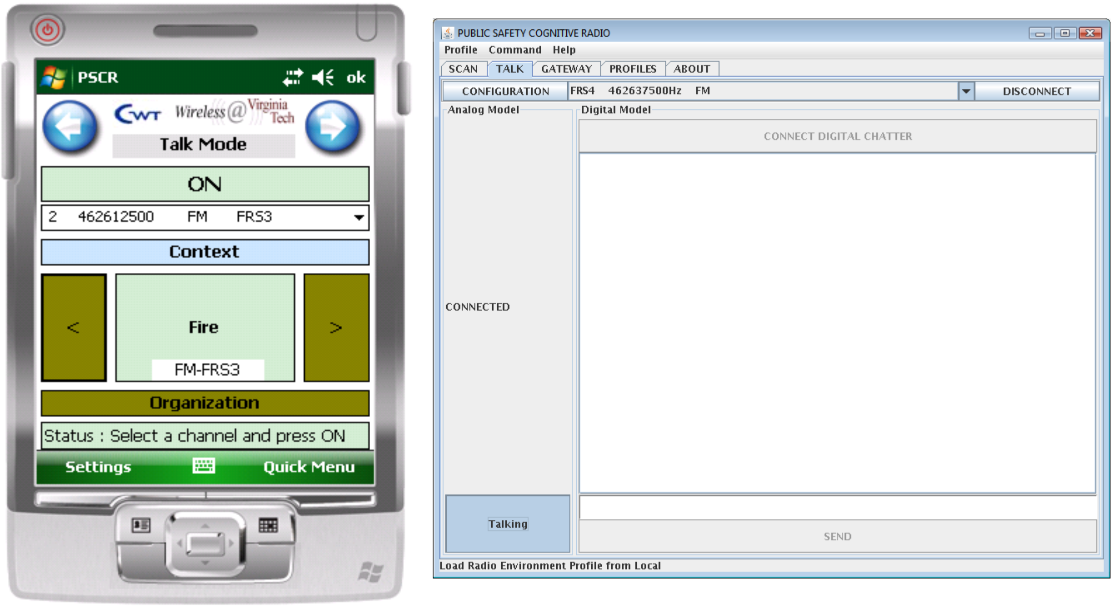
Figure 18. Gateway Mode on PDA (left) and existing GUI on PSCR (right)

2.4.8.3 Talk Mode

This mode provides the PDA with the Talk ability to establish a remote audio link between the PDA and a public safety waveform through the VPSCR. Before Talk mode can be utilized, the channel modulation must be identified. Currently, the PDA supports analog audio talk to the narrowband FM waveform, the operation of which was described in Section 2.4.7.1. Since digital audio is being sent to the VPSCR, the VPSCR can be modified to support digital audio transmission over digital modulation schemes which it currently supports – BPSK, QPSK and MPSK.

The VPSCR interface uses three buttons to provide talk functions. The Configuration button is used to initialize the audio framework, the Connect button is used for loading the audio framework with the selected modulation, frequency and audio parameters. This places the VPSCR in a default 'listen' state for audio from a public safety waveform. The Talk button initializes the audio-talk flowgraph to enable the VPSCR to talk to the public safety waveform through the microphone on the laptop.

The PDA sports a redesigned interface from the original GUI, for accessing the audio framework and flowgraph for analog talk between the PDA and the public safety waveform. The interface is designed to be similar in appearance to a push to talk (PTT) radio with a large central PTT button. The ON button will perform all the initialization and will place the PDA in 'listen' state. The central PTT button allows the PDA user to



2 462612500 FM FRS3

FM-FRS3

Status : Select a channel and press ON

FRS4 462637500Hz FM

CONNECTED

Load Radio Environment Profile from Local

Figure 19. Talk Mode on PDA (left) and existing GUI on PSCR (right)

talk to the public safety waveform. The PTT button turns blue to indicate the talk state from the PDA and returns to the original color when the state is terminated. The channel dropdown list is for easy viewing of the channel details as well as selecting the desired waveform. Channels can also be selected using the left and right channel scroll buttons. The channel identifier and modulation is displayed on the PTT button. The status bar provides continuous updates of the buffer time remaining and the current operating state as well as any error messages.

2.4.9 Summary

This chapter discussed the remote access architecture for the VPSCR and the implementation details using a personal digital assistant. This work is unique in the sense that it enabled the powerful features of a public safety cognitive radio to be used in a portable handheld device. This method addresses the immediate requirement to use cognitive radio features in a portable device by balancing the constraints of power requirements of a cognitive radio and lack of processing power of current generation handheld systems. In the future, more powerful mobile processing environments will ensure that further processing can be carried out on the PDA. Until such time, the remote access scheme will be an effective system.

This chapter initially discussed the motivation behind the remote access scheme, detailing the lack of portability of the PSCR. The VPSCR which enabled PSCR functionality in vehicles was introduced. Next, the remote access architecture was discussed with specific emphasis on audio processing both at the VPSCR and at the PDA side. The network operation where UDP commands are used for controlling the VPSCR was presented. Finally, some usability screenshots and comparison with the existing interface were shown.

This concludes the hardware implementation of this research work. In the next section, a service discovery protocol is designed and simulated to extend this VPSCR-PDA scenario in a wide area network providing quick and efficient access to remote services.

Chapter 3 : Service Discovery in Public Safety Networks

3.1 Introduction and Background

The PDA-VPSCR remote access architecture equips the first responder with the features of a cognitive radio in a portable device. Consider the scenario where the responder needs backup from headquarters based across the city. The current system does not support such a feature; however, a wide area wireless public safety network containing such devices can provide this capability. The responder will be able to discover other public safety personnel in proximity and to communicate with them. They can also share information about available spectrum and can establish networks in infrastructure damaged areas by utilizing unused spectrum. The users will be able to locate other VPSCRs, obtain their services, scan for active waveforms from across the city, and seamlessly establish communication with legacy radios.

However, beneath this ideal situation lies the need to discover users and their capabilities efficiently without overwhelming the underlying network with control traffic. The need for users to quickly detect the loss or failure of a neighboring user is of importance in a highly dynamic and mobile environment and the ability to locate users and devices in close proximity should be a priority. The following research work attempts to address this goal using service discovery in public safety networks.

This research work will introduce a new type of public safety network and present an optimized service discovery protocol to find and locate devices. The application of this section would be to allow PDAs to locate and access features from other VPSCRs and establish communication with legacy radios in the wide area network. This consequently also overcomes the problem of low transmit power of devices as long as a wide area wireless backbone is present to route data and audio. The infrastructure for a network, such as in Figure 20, exists; however the method of efficient feature determination does not. This is the focus of this research.

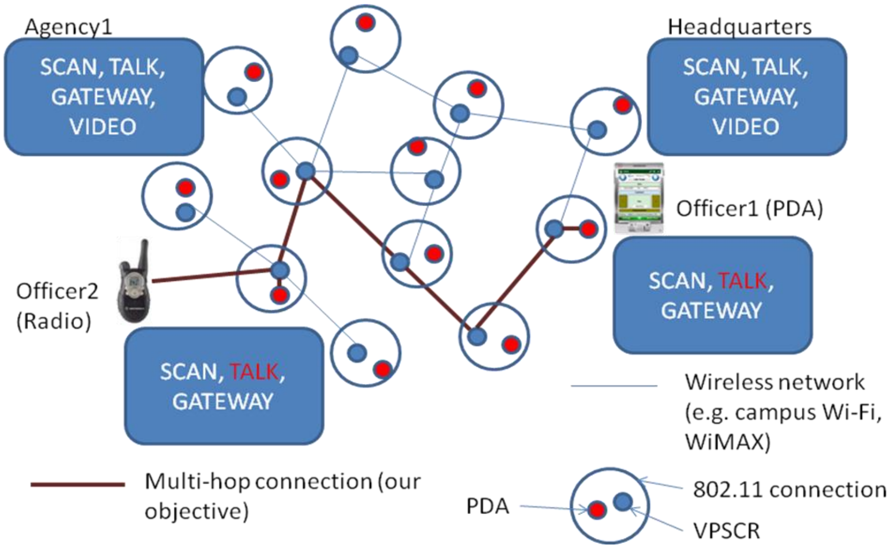

Agency1
SCAN, TALK, GATEWAY, VIDEO
Headquarters
SCAN, TALK, GATEWAY, VIDEO
Officer1 (PDA)
SCAN, TALK, GATEWAY
Officer2 (Radio)
SCAN, TALK, GATEWAY
Wireless network (e.g. campus Wi-Fi, WiMAX)
Multi-hop connection (our objective)
PDA
802.11 connection
VPSCR

Figure 20. Finding Services in a Wide Area Network

Definition of a **Service**: A service can be defined as a feature of a particular node/device that may be utilized by other connected nodes or devices. Connecting to a service involves initiating a transfer over one of the many schemes such as Hyper Text Transfer Protocol (HTTP), File Transfer Protocol (FTP), Session Initiation Protocol (SIP) etc., or a user defined proprietary protocol. Each scheme defines its own specification for initialization, handshake, packet structure, timing sequence and tear-down sequence. Commonly used services in a local area network (LAN) include network file storage, DNS, directory access (LDAP), authentication etc.

Service Discovery (SD) can be simply explained as a system or protocol either to advertise or to find services on a network. In other words, it allows users and devices to locate each other’s features in a way which minimizes the need for underlying network knowledge. Since services operate at the application layer, traditional service discovery has been viewed as an application process.

3.1.1 Service Discovery Characteristics

1. SD allows a node to distribute a Uniform Resource Identifier (URI) which may contain information such as a web address, server address and port, uniquely

identifiable index or a variety of other parameters depending on the type of architecture.

2. SD is optimized for reliability, low overhead and low discovery time as well as scalability or a suitable compromise of these factors depending upon the targeted network.
3. SD is traditionally an application layer protocol which has also been integrated with other layers of the network stack towards cross-layer architecture.
4. SD allows users to obtain a list of services and provides suggestions for selecting the most suitable service.

3.1.2 Service Discovery Does Not

1. SD is not concerned with the actual data transfer involved in the service utilization process.
2. SD does not specify the protocols used after the services have been obtained. This will be specific to the services being utilized.
3. The SD scheme should not be specific to a particular network – although actual implementations are usually network type specific.

3.1.3 SD in Wired Networks

SD is an essential part of modern networks where finding printers, routers, fax machines, network shares, streaming music servers and portable devices is made possible. Some of the well known SD protocols are Jini[36], UPnP [37], Salutation [38] and SLP [39], [40]. In such cases the network is usually a Local Area Network (LAN) or Wi-Fi infrastructure with considerable network capacity where devices do not leave or change their location often. In these relatively reliable networks, SD protocols are generally optimized for scalability and discovery time.

3.1.4 MANETs

A mobile ad-hoc network (MANET) is a constantly mobile network composed of wireless devices which are continuously moving, changing position or losing connectivity. Each node may be connected to any number of single hop neighbors, only constrained by hardware limitations, and is capable of independent motion. Challenges in MANETs are in the form of dynamically changing routes and having to perpetually evaluate node presence and neighbor states.

3.1.5 SD in MANETs

Service Discovery is of special importance in MANETs where constantly changing topologies and routing paths create uncertainties in locating a previously known service. MANETs are comprised of devices scattered in an unsystematic manner with each node acting as a router for traffic. The factors affecting service discovery are the low capacity wireless networks, unstable routing table entries, random node and path losses, multiple redundant paths, lack of a central information agency, lack of a caching mechanism, node mobility and battery or resource limitations. Examples of research in this area are Konark [41], Bluetooth SDP [42], SSDP for MANET [43] and Splendor [44] among others. An excellent feature comparison of SD protocols is given in [45].

3.2 Classification of Service Discovery Protocols

SD Protocols come in a variety of flavors, and consequentially, there are various methods of classification; for instance – based on location of service cache (directory based, directory less), based on storage mechanism (structured, distributed), based on virtual topologies (clustered, backbone based, DHT based) and based on flooding mechanism (global, controlled) among many other types [46].

This thesis is, however, concerned with a type of service discovery optimized for a special type of MANET where the search mechanism is of primary importance and the research conducted is for optimization of this mechanism. Therefore the search mechanism classification is explained in further detail below.

3.2.1 Search Mechanisms

Service search or advertisement consists of the technique used for distribution of service information. Search techniques are of key importance in the availability of services across a network and can influence the selection of the best possible service. It is widely regarded that distributed information involves less service location effort while a centralized storage information causes difficulty in scalability [45]. Therefore the search mechanism needs to be suited for the type of service storage used in the network. Search can be broadly classified into two categories depending on the type of location and storage scheme that is utilized by the node advertising the service:

1. **Proactive** or Push based SD (Advertisement): Services or agents on behalf of services will periodically advertise their presence. Nodes maintain services in their individual caches for a while until after the advertisement is no longer heard.
2. **Reactive** or Pull based SD (Discovery request): Services or agents are passively connected devices and other nodes which require these services initiate a request or discovery message. The reply contains service information such as state, availability and location. Services are updated by an explicit update message.

Protocols, in general, depend on both mechanisms although the extent by which each scheme is used decides its classification. Many service discovery protocols such as SLP and Jini [36] use a centralized agent that performs service request related functions while others such as Konark [41], UPnP [37] and DEAPSpace [47] use a distributed approach and each node maintains its own service cache. From this discussion, it becomes clear that there is an abundance of service discovery protocols with numerous classifications, applications, and operating mechanisms. Although we use the term ‘Advertisement’ for (1) and ‘Discovery’ for (2); in this thesis they are used interchangeably to indicate the hybrid nature of most SD protocols.

3.2.2 Routing Protocols for Ad-hoc networks

Routing is the process in which packets traverse along the network from the source to the intended destination along a determined path. Routing is optimized through a combination of metrics, depending on the application, which include – efficient routes (end-to-end path, end-to-end average delay), load distribution, and packet loss. In this thesis we consider routing for MANETs in combination with Public Safety Networks (PSNs) since routing for wide-area and fixed networks are optimized for completely different metrics. Classical routing can be classified by the search mechanism as distance vector (DV) and link state (LS).

Distance Vector – Each node maintains destination and next hop information in its routing table along with the cost metric in the routing table. Each node exchanges state knowledge with only its immediate neighbors and is constantly monitoring their condition. Distance Vector protocols use route metric calculation along with outgoing interface to choose the best path using least-cost algorithms such as Bellman-Ford [48].

Link State – Each node periodically floods the network announcing its presence and each node receiving the advertisement will use information in the message to form a map of network connectivity with itself as the map center. Link State protocols determine shortest path individually, using algorithms such as Dijkstra's [49].

3.2.3 Proactive and Reactive Protocols

DV and LS routing protocols are a subject of much research and some well known protocols implemented include ad-hoc on demand distance vector (AODV), optimized link state routing (OLSR), destination sequenced distance vector (DSDV), dynamic source routing (DSR) among many others. A further classification under which both types of traditional protocols fall is the route retrieval which can be both proactive and reactive.

Proactive protocols actively determine the routes during network convergence and each node continuously maintains a table of routing information accurate. When a connection is has to be established, the node refers to the routing table to determine the path, usually next hop, to the destination. Proactive protocols need to maintain a routing table with entries for every node in the network. Common proactive protocols are OLSR and global state routing (GSR).

Reactive protocols determine the routes based on when the route is needed. This initially introduces a set up delay in establishing a link. Reactive protocols drastically reduce the overhead involved in route determination since they do not maintain routes to any node unless required. Common reactive protocols are AODV and DSR.

In the following section, the suitability of OLSR for service discovery is described followed by current research work in this area.

3.2.4 OLSR

Optimized Link State Routing (OLSR) [50] is a proactive link state routing protocol that has gained much interest in the wireless and MANET community due to the optimized nature of its flooding mechanism in disseminating network broadcasts and link state routing information. OLSR uses the concept of multi point relay (MPR) nodes which are specially designated to broadcast messages during the route convergence process. This technique significantly reduces the number of redundant messages and hence the link overhead of routing compared to simple flooding as in Figure 21.

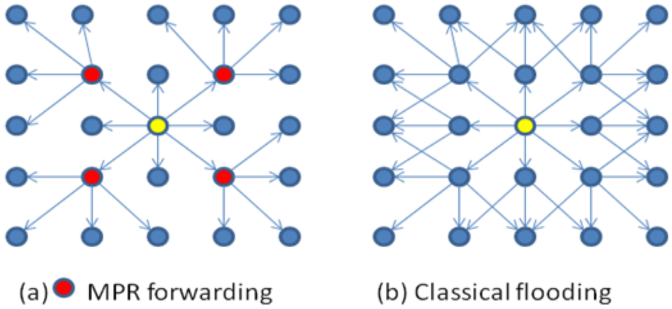


Figure 21. OLSR Forwarding vs Classical Flooding

OLSR generates link state information only at the MPRs. These MPRs distribute first, link state information using a path to other MPRs ensuring an optimized delivery mechanism with each node receiving the information. Second, an MPR node reports links between itself and those nodes that maintain it as an MPR. This technique avoids distribution of the link state information of every link and hence is further optimized. Thus non-optimized local paths are avoided and are not visible to the network as a whole.

3.3 Integrated Routing and Service Discovery

Traditional Service Discovery is an application level procedure that operates independently of other network functions like link/neighbor discovery and routing. In such cases service discovery must always occur after neighbor discovery and routing [51] since it uses application layer protocols, e.g., MARE [52] and Konark [41]. Hybrid application discovery involves the use of both a directory agent as well as nodes advertising their own services and is the approach that many MANET discovery protocols use [53] and [54]. In [55] the authors explore the effectiveness of reactive and proactive routing protocols in response time for service discovery and conclude that in most cases a hybrid approach is a suitable compromise.

Application layer service discovery has been shown to be ineffective in MANETs due to various reasons which include - low capacity wireless links which are unsuitable for increased control traffic overhead, constant status updates and the lack of extensive caching mechanisms.

There have been significant research efforts in MANET cross-layer service discovery protocols [56], [57], [58] with each protocol intended for a particular function or a particular routing protocol. Many such efforts range from service discovery over AODV [59], and over OLSR [60], [61], [62], as well as separate routing protocols like GSR [63]. The benefits of the cross-layer approach in DSR and DSDV protocols outperforming traditional service discovery has been outlined in [64]. Evaluation criteria for such comparisons often include the effective overhead, end-to-end service discovery time and network capacity utilization for routing and service function.

3.3.1 Existing OLSR based Service Discovery

In this section existing service discovery protocols for OLSR are discussed. This is an area generating a huge interest in the research community as can be seen by other published work. The interest stems from the fact that OLSR is a very extensible routing protocol in terms of the requirements and the flexibility of its packet structure.

The standard OLSR packet is defined by the RFC 3626 [50] uses a standardized packet encapsulation format. Each of the three message types (HELLO, TC and MID) are defined as their own message and encapsulated in the standard structure. Multiple messages can be added to a single OLSR packet. This allows protocol architects to develop additional message formats that could be sent along with standard OLSR messages or independently. Second, the concept of using MPRs for optimized flooding allows additional application layer information to be efficiently distributed along the network with minimized overhead.

The research on SD in the OSLR protocol unanimously utilizes MPRs for distribution of service discovery information. In [65] the authors compare the advantages of OLSR over WCPD for SD and they conclude that OLSR outperforms WCPD in terms of coverage, though at the cost of bandwidth consumption. There has been significant research in utilizing OLSR for SD as evident in [60], [61], [62] and [66].

In [60] the authors introduce a new message format called service discovery message (SDM) for distributing service discovery information along the MPR path. They conclude that the overhead in introducing a new message type is insignificant in comparison to the overall OLSR message volume. They compare the performance of their scheme in terms of total number of services discovered and message overhead. Since the service discovery messages are transmitted over the MPR set independently of the standard OLSR messages, it implies that routing and neighbor discovery must occur before services can be advertised. Therefore it can be stated that service discovery is not truly taking place in parallel with routing, instead the routing protocol is emulating the application layer service discovery by using the MPR nodes.

In [61] the authors propose a cross-layer service discovery for OLSR based on Bloom filters and service caching for bandwidth constrained environments. They compare the performance of their Mercury protocol to others such as SLPManet [67] and PDP [68] and conclude that it is superior in terms of delay and overheard by using Bloom filter compression [69]. They also analyze the false-positives generated in service discovery by using Bloom filters. The Mercury project [70], is a plugin for the OLSRD [71] implementation.

In [62] the authors have introduced a multicast DNS (mDNS) based encapsulated message for OLSR which uses distributed DNS caches in a backbone extension to the wireless mesh network. They conclude that caching and message hops are definitive factors in determining message overhead. They compare the performance of their protocol in terms of total number of services found, delay and overhead.

In [66] the authors propose a service discovery protocol to support real-time multimedia applications over OLSR using the session initiation protocol (SIP) and they discuss the reduced overhead and call setup time for their protocol.

In [72] the author discusses a priority based service discovery over OLSR for ad-hoc networks for using quality of service guarantees for time sensitive implementations. It also discusses a MAC layer modification for cross-layer performance enhancement. The work analyses the round trip delay and message overhead metrics.

The discussion above clearly shows the abundance of research efforts in service discovery for OLSR. Many of the implementations are designed with a particular goal in mind; for instance, service guarantees [72], multimedia performance [66], or low capacity networks [61]. In the following section, the need for a specialized service discovery protocol for public safety is analyzed.

3.4 Motivation

"Need for an optimized Service Discovery protocol for a Public Safety Network"

The above OLSR based service discovery protocols are not suitable for Public Safety Networks (PSNs) due to a variety of reasons. More specifically:

1. Public Safety Networks require **service prioritization** depending on the **proximity of services** and not the application type. This is an important difference since PSNs do not need services from all parts of the network. Such a scheme will only introduce unnecessary overhead, instead PSNs must use nearby resources if available. Thus the metric 'number of services discovered' is only suitable for nearby services and 'discovery delay' must be compared for separate scenarios with regard to distance from service.
2. PSNs differ from MANETs in their **network infrastructure**. PSNs consist of a stable fixed **backbone** which is not bandwidth or processing power limited. Public Safety networks use specialized devices in the network and are not as bandwidth constrained as MANETs.
3. PSNs usually operate with a **base station** that every officer uses to communicate with the dispatcher and other officers. Peer-to-peer communication is possible but infrequent.
4. PSNs are delay and **overhead sensitive** and therefore must completely exploit the proactive nature of the protocol (for nearby services) as well as the **lower overhead** component of a reactive mechanism (for services further away). SD in PSNs needs a dual proactive-reactive scheme to provide a balance in the constraints of responsiveness and overhead.

3.5 Proposed Public Safety Network

In this section, a new public safety network scheme utilizing the PDA and the previously introduced PSCR is proposed. The public safety network will support multiple such heterogeneously connected devices in the wide-area wireless network. Scenarios for such a network are plentiful and include, for example, a city-wide WiMAX network or a campus Wi-Fi network (that provides near-complete coverage in the designated area) providing quick response to public safety responders. The objectives of this PSN are:

1. A public safety network scheme utilizing a partial network backbone consisting of a small number of fixed infrastructure points. This differs from the architecture of a traditional PSN with a single base station.
2. The few fixed infrastructure points, are identical to regular nodes in hardware, and are connected to (mostly) self contained ad-hoc networks called Public Safety Zones (PSZ).
3. The backbone is designed to be used in case services or features located in a particular PSZ are not sufficient. Hence the use of the term mostly in (2).
4. Service Discovery advertisements (or flooding mechanism) are optimized to reduce overhead and network utilization for non-essential traffic. Advertisements are zone specific for immediate access to nearby services.
5. A network should support both proactive and reactive forms of service discovery for exploiting the features of a dual proactive-reactive network consisting of ad-hoc and infrastructure components.

The need for service discovery in public safety stems from the realization that devices utilizing the network are not identical in all respects and may support different features. In other cases, a particular feature may be too expensive to be included in all public safety devices (video conferencing, for example) and only a select number of devices may need to access such features at a given time. It is logical to introduce only a limited number of unique devices in the network and allow nodes to find and utilize these scarce services. In these respects, PSNs and MANETs differ in topology, although it may be stated that a PSN consists of MANETs connected through a backbone network.

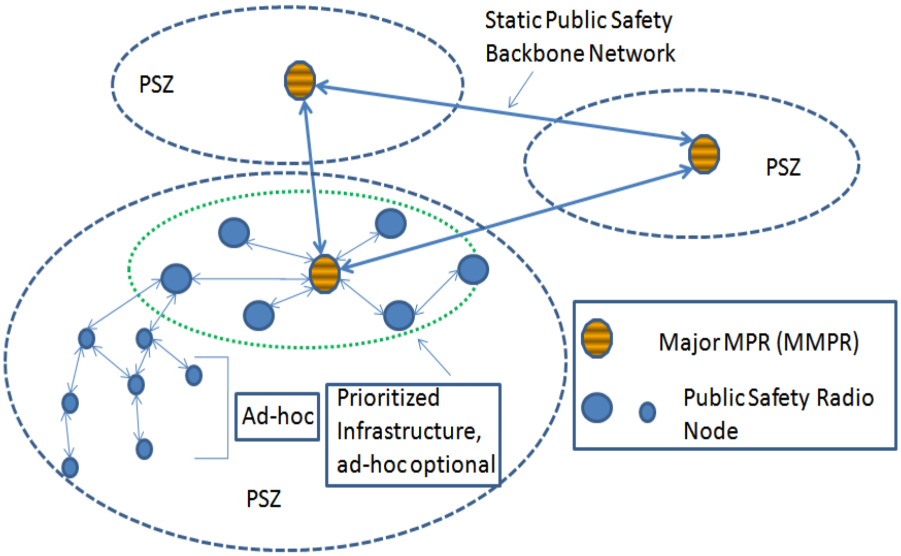


Figure 22. Public Safety Network (PSN)

3.5.1 Definition of a Public Safety Service (PSS)

A public safety service can be defined as a feature specific to a public safety node that may be advertised to other nodes and utilized remotely. A node can choose to not advertise some of its services and in this case they are hidden and not accessible. A public safety service is identified by PSS localization field and the source identifier as well as a globally unique identification number.

3.5.2 PSS Localization Classification

Public safety features can be classified according to their intended operation as well as their propagation mechanism. There are three types of proposed localizations for PSSs:

1. **Neighborhood services (NS)**
2. **Single-PSZ services (SPSZ)**
3. **Multi-PSZ services (MPSZ)**

Neighborhood services are intended for immediately connected neighbors or 2-hop neighbors. The motivation behind this classification is quick, near-immediate responses from a limited number of nearly nodes. For example, a first responder at the scene of a disaster can quickly find nearby responders and request assistance. The first responder can quickly search nearby nodes for utilized spectrum and initiate audio connections with other emergency response services. This is the quickest response service and is developed as a proactive service discovery mechanism. In this protocol, the backbone network is not utilized.

Single-PSZ services are those service announcements which are distributed within an individual PSZ. They are intended for moderate speed response services that can cover a much wider region than neighborhood services and a much larger number of nodes. For example, the first responder can request extra manpower or backup from headquarters which may be located a number of hops away. In this protocol, backbone node(s) in individual PSZs act as regular nodes. The backbone network is not utilized since services propagate in a single PSZ. The propagation method is implemented as a hybrid service discovery protocol optimized for lower overhead and reduced network load, without employing the backbone network.

Multi-PSZ services are intended for a very wide area of coverage encompassing a number of PSZs and effectively the entire public safety network which for example, may be distributed over an entire city. The first responder can request multiple response mechanisms for the scene of the disaster for different agencies. This is designed as a hybrid service discovery protocol that utilizes cache mechanisms in the backbone network. MPSZ services are not considered for simulation in this thesis.

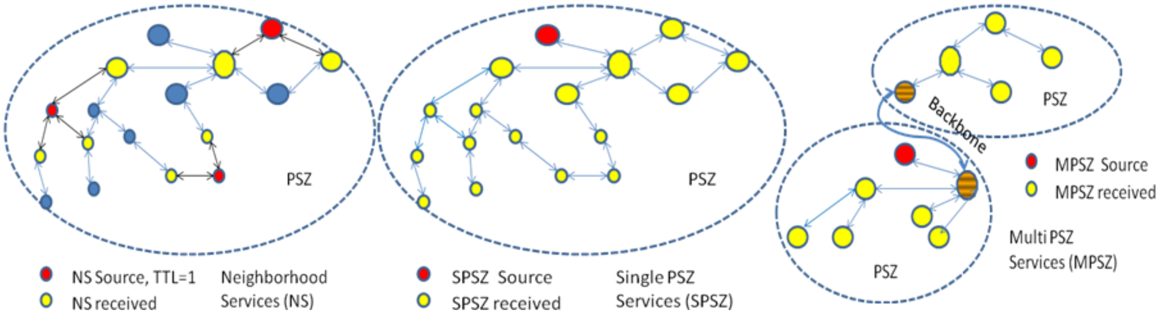
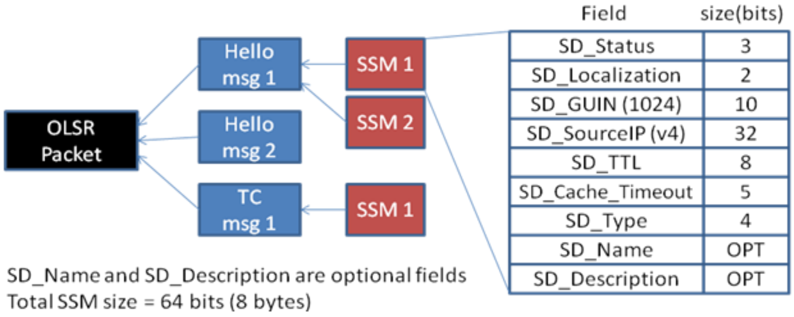


Figure 23. (L to R) a. NS b. SPSZ c. MPSZ Service Propagation Threshold

3.5.3 Service Sub Message

Service component in an OLSR message is injected as a Service Sub Message (SSM) which is a sub-sub-message defined under the hierarchy of a HELLO or TC message. The SSM is not another OLSR message type; instead the SSM can be inserted into HELLO or TC messages independently without affecting the functionality of the underlying OLSR protocol. Multiple SSMs can be injected into a single HELLO or TC message similar to the way multiple HELLO messages can be inserted into on OLSR packet. The limit on this number decides how many services a node can advertise in a single packet. The Figure 24 shows the structure of the SSM, and the associated fields.



Field	size(bits)
SD_Status	3
SD_Localization	2
SD_GUIN (1024)	10
SD_SourceIP (v4)	32
SD_TTL	8
SD_Cache_Timeout	5
SD_Type	4
SD_Name	OPT
SD_Description	OPT

SD_Name and SD_Description are optional fields
Total SSM size = 64 bits (8 bytes)

Figure 24. Packet Format and Hierarchy of SSM

These fields are explained in the Table 4, and it should be noted that the functionality or purpose of individual fields may vary depending on the type of localization (SD_Localization) used. Some fields such as the readable name and description have been omitted from the implementation. The total size of the SSM sub-sub-message is 8-bytes.

Each service is associated with an SSM containing these fields and is propagated along the network. All fields of the SSM are stored in the cache.

Table 4. SSM Field Contents and Details

Functionality	Field	Description	Contents
State maintenance	SD_Status	This field is used to maintain the status of the proactive or reactive services across the network. The functionality varies depending on the type of dissemination used	0 = Ignored 1 = Add/update service 2 = Optimized 3 = Delete service
Propagation	SD_Localization	This field defines the localization of the service and thereby deciding the dissemination method	0 = Ignored 1 = NS 2 = SPSZ 3 = MPSZ
Identification	SD_GUIN	The service Globally Unique Identification Number. This is a sufficiently large randomly generated number providing a unique identifier to the service	10-bit random sequence
Identification	SD_SourceIP	This stores the source IP address (or potentially any uniquely identifiable parameter) which will associate a service with its origin	32-bit IPv4 address
Propagation, state maintenance	SD_TTL	Decremented by 1 after reaching a node. 1) To determine the maximum extent of service reach (at TTL=0) 2) To prevent messages from cycling indefinitely in the network 3) To discard redundant and duplicate paths	8-bit integer (0 to 255)
Cache maintenance	SD_Cache_Timeout	1) how long a proactive NS will remain in the cache without an update query 2) a S/MPSZ reactive service threshold for withdrawing a service from circulation	5-bit integer (0 to 31)
Description, classification	SD_Type	This integer is used for contextual classification and hierarchy of services. Services can be uniquely categorized using SD_Type and SD_GUIN	4-bit integer (0 to 15)
Description, classification	SD_Name	Optional field that identifies the service by a readable name	Optional. Not implemented.
Description, classification	SD_Description	Optional field that identifies the service by a shot description	Optional. Not implemented.

3.5.4 Cache Repository

Each node maintains three cache data structures intended for services. First, the Local Cache (LCache) repository is initialized with a list of services that a node possesses and intended for distribution to other nodes. The LCache will contain both active services which are intended for circulation and inactive services which are only for use by the individual nodes themselves. Second, the Received Cache (RCache) repository maintains services received by a node and those which are in active circulation. Contents of the RCache may be forwarded when propagating a service if a number of factors are satisfied. Finally, the Deleted Cache (DCache) repository contains a list of services which have been removed from circulation. The DCache is used for identifying deleted services and preventing further spread of the service as well as for proactively deleting services during propagation. Contents of the DCache are purged after suitable intervals to ensure that a deleted service has been completely removed from the network. The cache structures maintain all classifications of service with all types of localizations.

Each element in the three cache repositories is comprised of a data structure entitled 'SD_Element' show in Figure 25. The fields in the data structure are composed of all fields contained in an SSM and additional fields for forwarding and cache status.

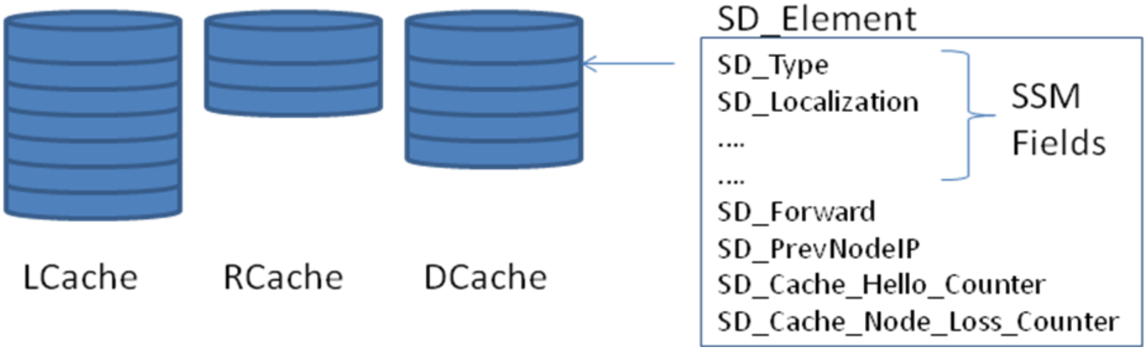


Figure 25. Cache Structure and Contents

3.5.5 Information dissemination mechanism

The key feature of this service discovery mechanism is the location based service dispersal technique. In this regard, the protocol offers three different mechanisms for each of the three localizations defined previously. In the figure, the primary motivator for location based services is shown in Figure 26. It is observed that the primary influencing factor in the service discovery protocol varies from quick response to low overhead and finally scalability as distance from the service origin changes.

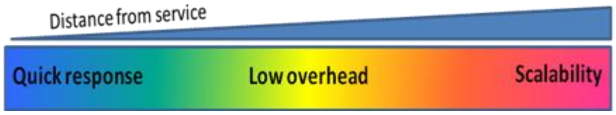


Figure 26. Service Objective versus Distance from Source

The need for a proactive protocol stems from the fact that nearby services require immediate updates when services are being advertised or updated. The proactive nature of the protocol provides the continuous update mechanism piggybacked on the OLSR HELLO messages which are broadcast regularly. Services further away require moderate speed response to updates however the deciding factor here is the overhead associated in maintaining the state across the network. The lack of a well defined network hierarchy can cause a network to perform poorly in terms of scalability. To address scalability in a wide area network, the proposed network consists of autonomous zones (PSZs) which provide routing and location of services from within. It is expected that the need for a user within such a PSZ to access services outside the network is rather unlikely. However to provide this ability, a minimal backbone network provides a scalable framework.

Neighborhood services are piggybacked purely on the OLSR HELLO messages and are added to every HELLO message sent by the node advertising a service. This interval may be changed, for example, to allow SSMs to be added only after a particular multiple number of HELLOs. The NS model is used to distribute services to nearby nodes ideally for 1-hop or 2-hop neighbors. However, this protocol is extensible to support discovery over the entire PSZ as well, at the expense of increased overhead and redundancy. This is a proactive protocol which depends on updates to maintain services in the RCache. If a service update has not been received for an integral multiple of a fixed number of HELLOs sent by a node (Cache_Timeout), then the service is deleted from the RCache. The NS method is ideally suited to fast changing services and links.

SPSZ services are initially transmitted by HELLO messages to the nearest MPR node. From here, they are propagated as an SSM in TC messages. The TC messages traverse the MPR set and reach all nodes in the PSZ using OLSR. This hybrid protocol is optimized for preventing redundant SSMs from being transmitted when the network topology or service does not change - this is ideal for a dynamically changing topology with stationary phases. The second optimization is in the form of cache updates to newly joined nodes. The network automatically detects node loss and additions and will advertise services to the node without querying the service source. The third optimization is in the form of service deletions when a node along the advertisement path fails. All nodes further along the advertisement path will remove the service from the cache after a Cache_Timeout while status quo is maintained for services in the connected region. This can be termed as an internally self-maintaining network.

MPSZ services are distributed in the same way as SPSZ services and are fully cached at the backbone node. Then using an underlying layer, the services among the backbone nodes are constantly updated. At the other end, the backbone node which is part of another PSZ will distribute the service using regular OLSR TC messages. The MPSZs are introduced as a way to distribute services in a wide area network, and are not considered for the performance tests.

3.6 Service Discovery Models

In this section, the details of the various models for service discovery in the proposed public safety network are presented along with the implementation code and design flowchart. This will provide a thorough understanding of the algorithm which can further be deployed in any existing OLSR package.

3.6.1 Overview of the Simulator

The discrete event simulator used is OMNeT++ 4.0 [73] with the INET framework [74]. OMNeT++ provides a base for performing simulation of discrete events in networks using C++. It is further extended using the INET framework which is a collection of models for 802.11 MAC and PHY, other PHY layers, routing protocols, app layer models, mobility models and more specifically MANET protocols like AODV, DYMO and OLSR. The OLSR model is an OMNeT++ implementation adapted from the widely popular UM-OLSR package [75]. This service discovery protocol and all related code is an extension to the OLSR model and all the code is embedded within the methods of the OLSR model.

3.6.2 Node level model

The node is a standard mobile ad-hoc node consisting of the blocks shown in Figure 27. The PHY and the MAC layer contained in the WLAN block use the default 802.11 implementation. The network layer is a generic IPv4 provider which enables access to IGMP, ICMP and ARP along with routing table management features. The application layer consists of a TCP or UDP middle layer over a user defined application. The application layer is not modified for this simulation. When number of nodes in the network is increased the node density is kept constant.

This thesis is concerned with the "manetrouting" block which contains the OSLR protocol and the service discovery additions. All changes are directly implemented in the OLSR message handlers as service discovery components for cross-layer functionality.

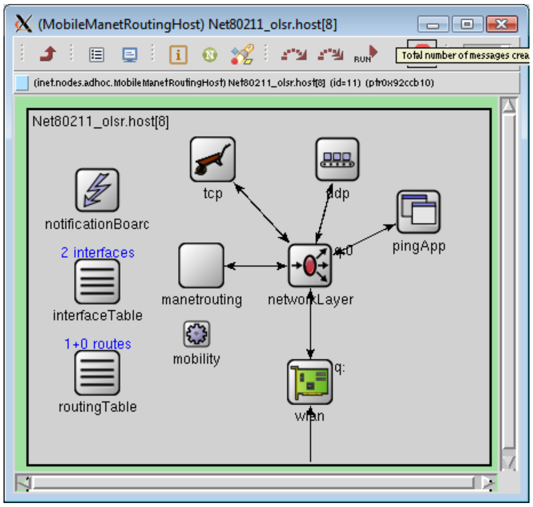

(MobileManetRoutingHost) Net80211_olsr.host[8]
Total number of messages crea
(inet.nodes.adhoc.MobileManetRoutingHost) Net80211_olsr.host[8] (id=11) (ptr0x92ccb10)
Net80211_olsr.host[8]
notificationBoarc
2 interfaces
interfaceTable
1+0 routes
routingTable
tcp
udp
pingApp
manetrouting
networkLayer
mobility
wlan
q:
q0

Table 5. Radio Parameters for the Simulation

Parameter	Value
Transmitter power	0.75mW
802.11 Radio channel #	0
Receive sensitivity	-90dBm
SNIR threshold	4dB
Thermal noise	-110dBm
Path loss alpha coefficient	2
Bit rate	54Mbps
RTS threshold	3000 bytes

Figure 27. Simulation node model and the 'manetrouting' block which implements service discovery

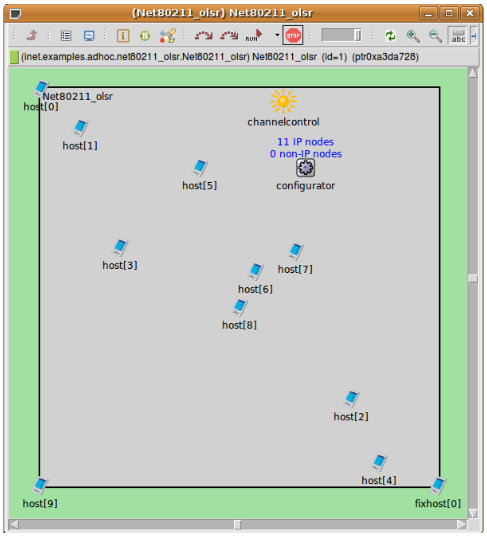


Figure 28. 10(+1 fixed) Node Example Network

3.6.3 Algorithm – service handler

An OLSR message is processed to extract the OLSR link and neighbor state information, when a node receives any OLSR packet. After this process, service extraction takes place to determine if the packet contains any services as part of the sub messages. This procedure is common for all types of OLSR messages, specifically for TC and HELLO which contain the SSMs for service discovery.

An OLSR packet may contain multiple SSMs and the following process is repeated until all SSMs are processed individually. First, the service GUIN and SourceIP is checked with the LCache to determine if the service message is originally from the receiving node. If so, the SSM is discarded since a node should not update its service cache using its own service message. Next, if the service received is present in the DCache, then it is discarded as the service message is stale and no longer valid.

The service details are compared to entries in the RCache using the GUIN and SourceIP and other updated fields. If the service exists in the RCache, and the TTL of the service received is greater or equal to the stored value, then the stored values are updated with the SSM from the message. This ensures that only the shortest path is selected for updating the cache since other paths may contain redundant or stale information. If the service does not exist in the RCache, a new entry is created.

Once all SSMs from the packet are processed, the cache deletion operation occurs. For each service in the RCache, the service Status is checked for messages which are to be deleted (3). Such entries in the RCache are then moved to the DCache and service Forward is set to true, to allow the delete message to be generated in the next OLSR packet. The delete message is used to explicitly remove the service entries from the RCache of other nodes.

After the service discovery processing occurs, the OLSR message handler code continues its operation normally. The entire process in shown in Figure 29 and this is implemented as part of the OLSR message receive handler.

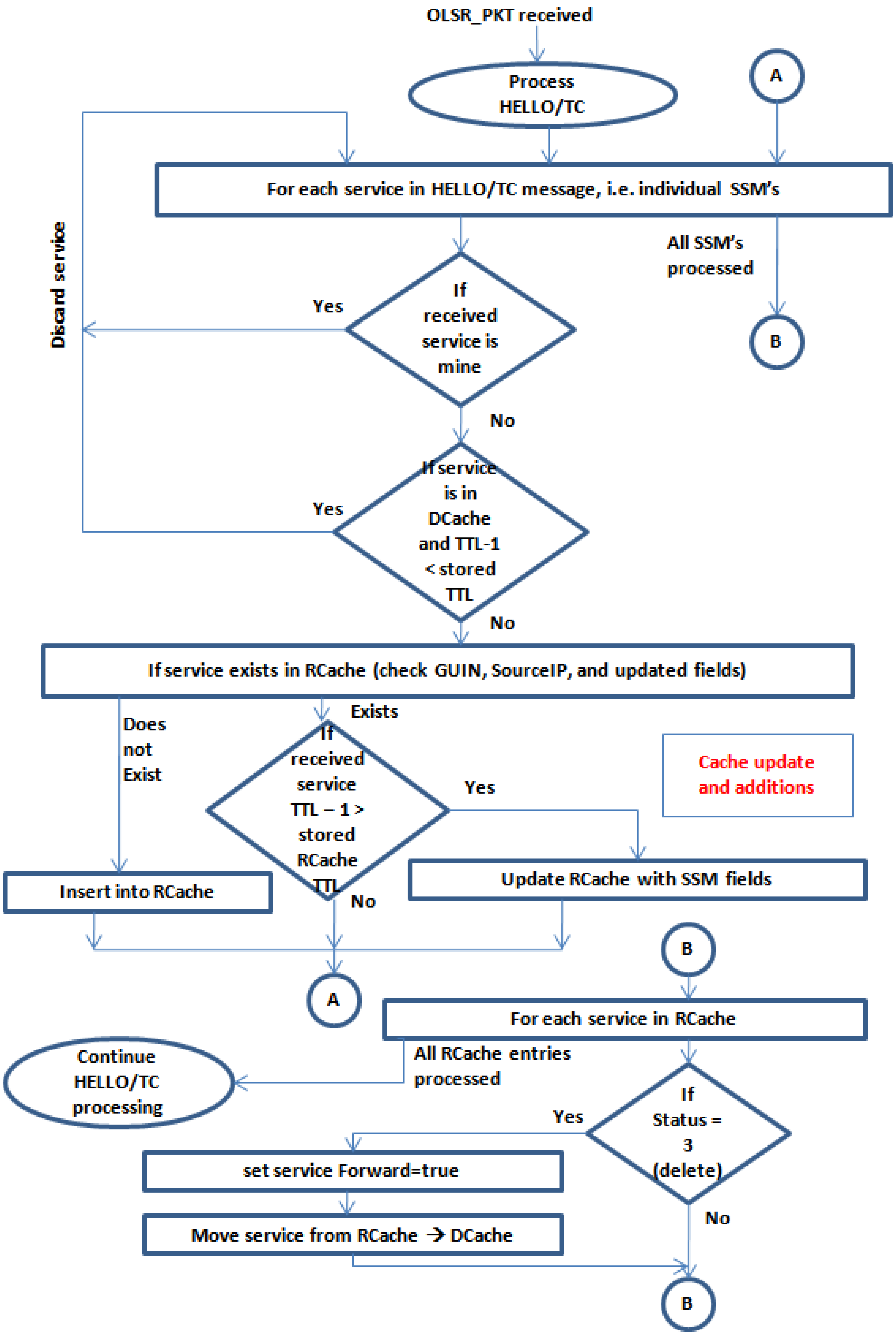

OLSR_PKT received
Process HELLO/TC
A
For each service in HELLO/TC message, i.e. individual SSM's
All SSM's processed
B
Discard service
Yes
If received service is mine
No
Yes
If service is in DCache and TTL-1 < stored TTL
No
If service exists in RCache (check GUIN, SourceIP, and updated fields)
Does not Exist
Exists
If received service TTL – 1 > stored RCache TTL
Yes
No
Cache update and additions
Insert into RCache
Update RCache with SSM fields
A
B
For each service in RCache
All RCache entries processed
Continue HELLO/TC processing
Yes
If Status = 3 (delete)
No
set service Forward=true
Move service from RCache → DCache
B

Figure 29. Service Discovery Message Handler Algorithm

3.6.4 Algorithm – service dissemination

The propagation scheme uses HELLO and TC messages for service dissemination. The procedure is discussed in this section and depicted in the flowchart in Figure 30. Some parts of the algorithm are specific to HELLO generation (orange) while others are for TC generation (blue) and some is commonly processed code (white).

Each node independently processes a timer expired event for the HELLO and TC messages. The timer expired event causes the Send_hello or Send_tc methods to be invoked which generate and send the OLSR messages. The default timer duration is 2sec for HELLO and 5sec for TC.

Initially, an extra stage of processing occurs if it is a HELLO timer – for all NS (LOC=1) services existing in the RCache, the Cache_Hello_Counter is incremented by 1 and the RCache is updated. This field maintains the number of HELLO messages sent between service queries. If the Cache_Hello_Counter exceeds the Cache_Timeout, the NS service is considered stale and is discarded from the RCache. This is part of the cache maintenance procedure for the proactive NS protocol. The field is reset to the original value when an update message is received.

The first and second optimization is for messages with Status=2 i.e. the Optimized scheme and is used with TC messages only. Each node maintains a snapshot of the neighborhood condition each time a TC is sent. While sending a TC message, the current neighborhood condition, using the OLSR Neighbor List, is compared to the previous snapshot. If the current condition indicates that a new node has joined or disconnected, then the service Forward flag is set which allows for insertion of the SSM into the packet later during this algorithm. If there are not updates in the neighborhood, the Forward is set to false which does not insert the SSM into the TC message thereby lowering overhead for this hybrid SPSZ protocol.

Next, for each service in the LCache, conditions are checked and the SSM is created if independent conditions for HELLO and TC are satisfied depending on the type of packet. Similarly, the RCache is processed for elements that are eligible for insertion into the packet. The DCache insertion procedure is only performed for TC messages. Initially services in the DCache are scanned and Staus = 3 is assigned as well as the Cache_Timeout is decremented for all deleted services. If the Forward condition and

Cache_Timeout is valid then a deleted service SSM is generated and inserted into the TC. This is used for propagating service deletion requests in the network. The HELLO and TC processing requires separate conditions as shown in the dual condition block with orange and blue colors for HELLO and TC respectively.

The second optimization technique is used for automatically removing services from the RCache, as well as informing other nodes that the service no longer exists. Using the updated OLSR Neighbor Set each node checks availability of the previous node (using the PrevnodeIP field) from where the service was received. If this node has failed, the Cache_Node_Loss counter is decremented in the RCache. After a particular threshold, if the service is not updated from another source, the service is removed from the RCache and moved to the DCache. This effectively will propagate a service deletion message and the service will be removed from other nodes receiving the deletion message. This is applicable for both SPSZ and MPSZ services. Along with the previous optimizations, this procedure provides the service discovery protocol with the ability to maintain services automatically in the network.

Finally, the HELLO or TC message is created and encapsulated in the OLSR headers and sent to the network layer.

The simulation does not implement the service dissemination mechanism for the MPSZ scheme and only single zone of coverage is considered for evaluation.

Summary of optimizations:

1. Automatic cache maintenance: New node additions (joins) will receive all SPSZ services in the next received TC message without query or notification to the service source.
2. Automatically removing service from unavailable routes: If a node along the path fails, a service deletion message is propagated further along the dissemination path and the connected part of the network is unaffected
3. Hybrid scheme with overhead optimization: The SPSZ protocol does not broadcast update messages during stationary non-changing scenarios. Service update is propagated only when either routes or service change.

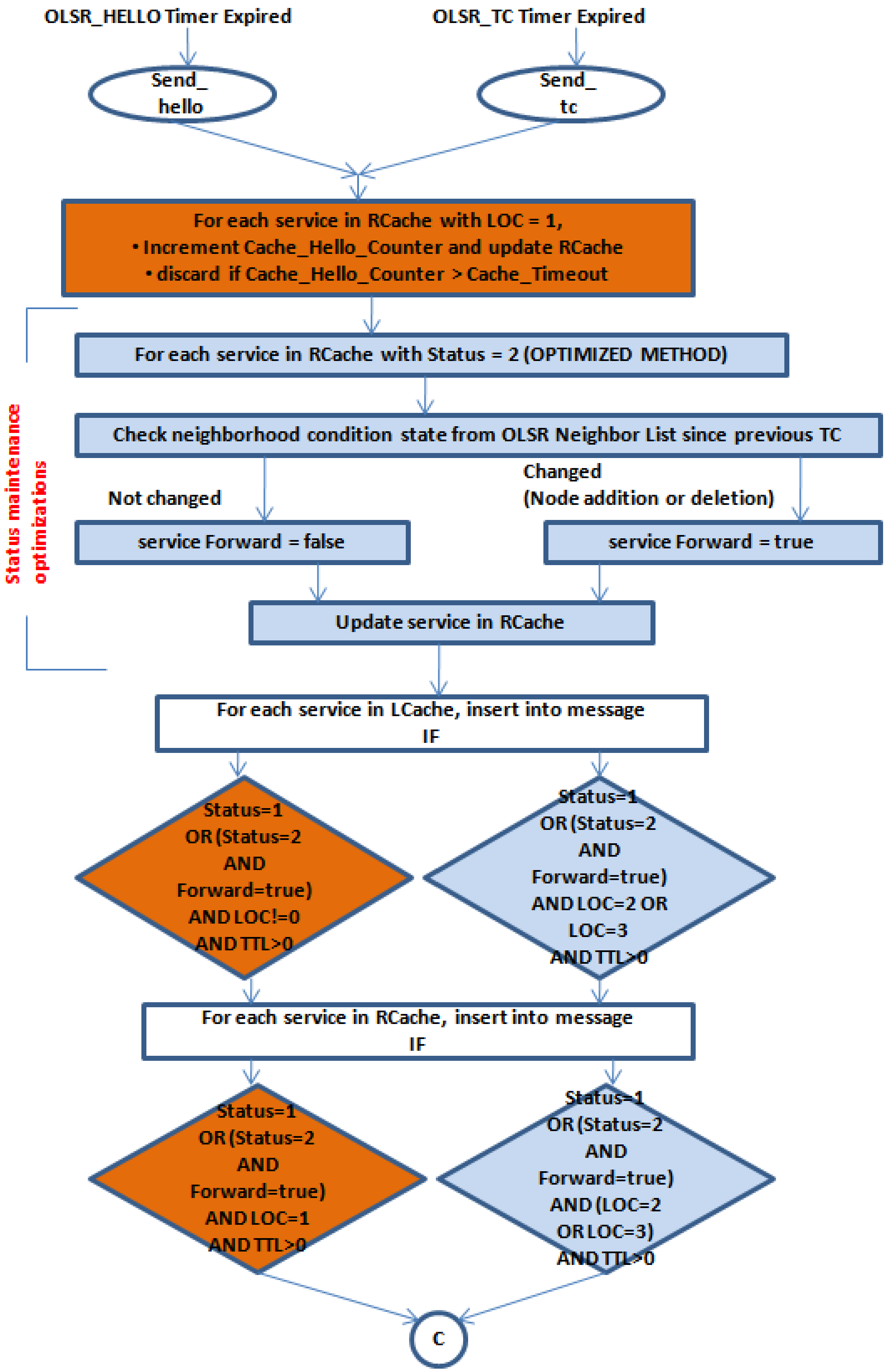

OLSR_HELLO Timer Expired
OLSR_TC Timer Expired
Send_ hello
Send_ tc
For each service in RCache with LOC = 1,
• Increment Cache_Hello_Counter and update RCache
• discard if Cache_Hello_Counter > Cache_Timeout
Status maintenance optimizations
For each service in RCache with Status = 2 (OPTIMIZED METHOD)
Check neighborhood condition state from OLSR Neighbor List since previous TC
Not changed
Changed (Node addition or deletion)
service Forward = false
service Forward = true
Update service in RCache
For each service in LCache, insert into message IF
Status=1 OR (Status=2 AND Forward=true) AND LOC!=0 AND TTL>0
Status=1 OR (Status=2 AND Forward=true) AND LOC=2 OR LOC=3 AND TTL>0
For each service in RCache, insert into message IF
Status=1 OR (Status=2 AND Forward=true) AND LOC=1 AND TTL>0
Status=1 OR (Status=2 AND Forward=true) AND (LOC=2 OR LOC=3) AND TTL>0
C

Figure 30. Service Dissemination Algorithm (continued on next page)

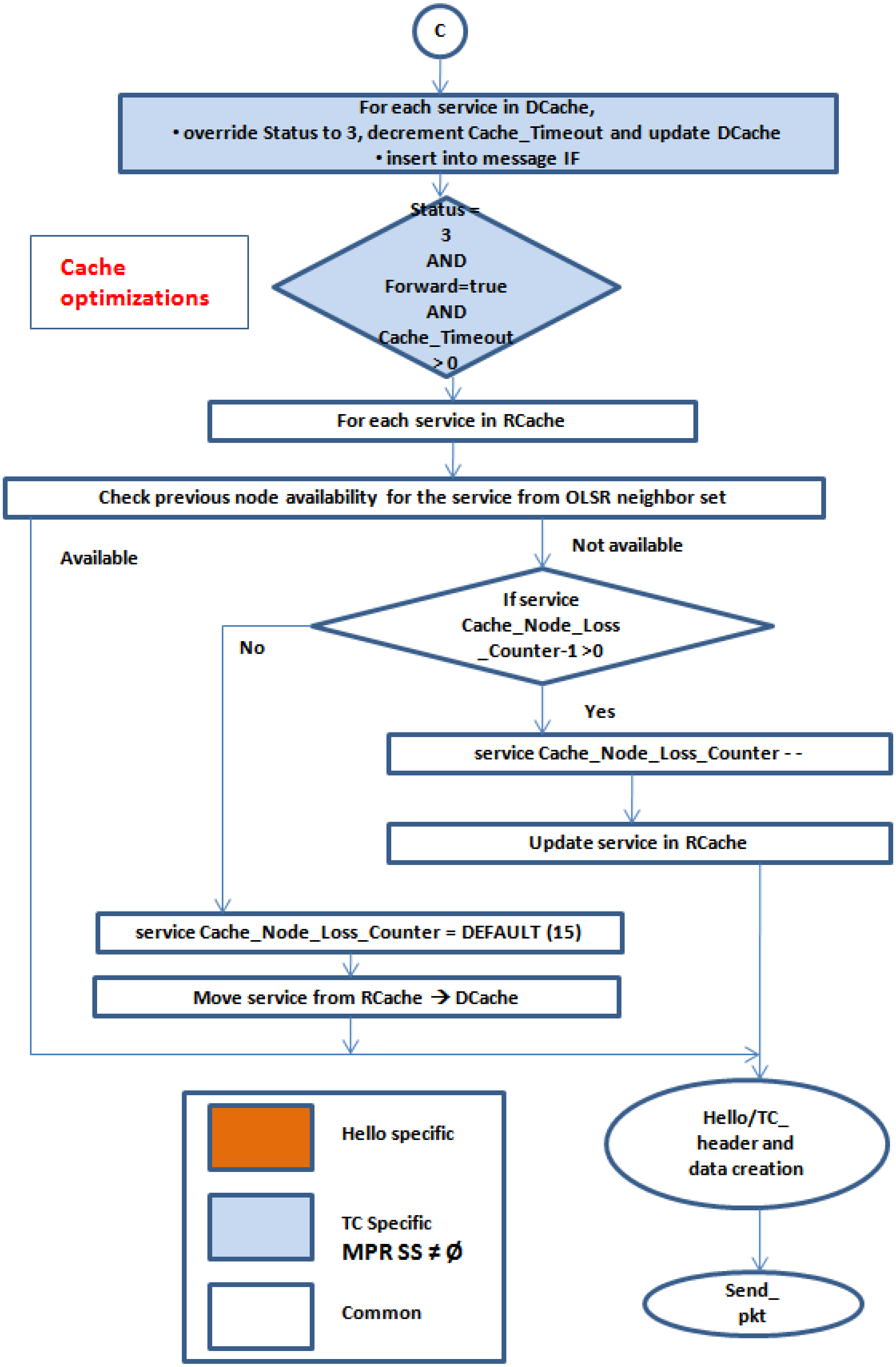

C
For each service in DCache,
• override Status to 3, decrement Cache_Timeout and update DCache
• insert into message IF
Cache optimizations
Status = 3 AND Forward=true AND Cache_Timeout > 0
For each service in RCache
Check previous node availability for the service from OLSR neighbor set
Available
Not available
If service Cache_Node_Loss _Counter-1 >0
No
Yes
service Cache_Node_Loss_Counter - -
Update service in RCache
service Cache_Node_Loss_Counter = DEFAULT (15)
Move service from RCache → DCache
Hello specific
TC Specific
MPR SS ≠ Ø
Common
Hello/TC_ header and data creation
Send_ pkt

Figure 31. Service Dissemination Algorithm (continued from previous page)

3.7 Test Protocols

The simulation is used to evaluate the performance of five test cases and protocols in various scenarios. These test cases compare the implementation of the optimized scheme with baseline service discovery standards as well as service discovery without the optimizations.

1. **LO1_FLOOD:** This baseline test involves OLSR HELLO message based cross-layer flooding to distribute the service to the entire network. The SSMs are added to the HELLO message and propagated in a similar way. This scheme uses the proactive cache maintenance method. SD_Localization = 1, SD_TTL = 255.
2. **LO1_TTL1:** This test involved using service discovery for distributing Neighborhood Services (NS) to 1 hop neighbors. This is a special case of LO1_FLOOD and uses the proactive scheme for cache maintenance. SD_Localization = 1, SD_TTL = 1
3. **LO1_TTL2:** This test involved using service discovery for distributing Neighborhood Services (NS) to 1 hop and 2 hop neighbors. This is a special case of LO1_FLOOD and again uses the proactive scheme for cache maintenance. The purpose of this test is to evaluate the scalability of the NS scheme in multi-hop environments. SD_Localization = 1, SD_TTL = 2
4. **LO2_OPT:** This is the **new protocol of chief interest** which needs evaluation against the other schemes. This protocol uses the **optimized** status and cache maintenance mechanism for distributing SPSZ messages over a zone (MPSZ method is not simulated although the procedure is similar). The SSMs are propagated over a combination of HELLO and TC messages. SD_Localization = 2, SD_TTL = 255, SD_Status = 2
5. **LO2_REG:** This is the baseline protocol for comparison using the regular OLSR MPR distribution mechanism without any optimizations for distributing SPSZ over a zone. This protocol is similar in operation to the existing protocols described in Section 3.4.1. The SSMs are propagated over a combination of HELLO and TC messages. SD_Localization = 2, SD_TTL = 255, SD_Status = 1

It should be noted that FLOOD, OPT and REG are global (SPSZ) protocols which will be evaluated while the TTL1, TTL2 protocols are for local distribution only.

3.7.1 Evaluation Metrics

The metrics used in the OMNeT++ simulation are standard parameters used in research to evaluate the performance of service discovery protocols, as well as particular routing protocols. Each metric is compared against test variables which help evaluate the efficiency under different scenarios. The metric calculation is performed for each of the 5 test protocols (propagation methods) and performance is graphed.

1. **Message overhead:** The total overhead involved in the service discovery is graphed in terms of number of excess SSMs as well as excess bytes generated for service discovery. For each node, statistics were collected for number of HELLO and TC messages sent and number of SSMs sent for each type as well as the bytes generated. This metric is useful in determining the excess network load of the service discovery protocol. A lower overhead for routing and service discovery is desirable.
2. **Number of updates:** This is a count of the number of unique update messages sent or forwarded for a particular service by each node. The update messages include redundant advertisements as well as legitimate updates. This metric is useful in determining the service discovery overhead of the messages in terms of processing and network load. A lower number of updates is desirable in a hybrid scheme and simultaneously this does not have an effect on the responsiveness.
3. **Dissemination time:** The time required for a service to reach every node in the zone (or system) is known as dissemination time. The time when a service was received by each node is recorded and the senders timestamp is subtracted from the maximum values of the received times. This metric is useful for determining the speed at which services are distributed in a network.
    a. **Convergence Time:** This is the time required for the cross-layer service protocol to disseminate to all nodes during the network discovery process (convergence) at the start. This is useful in evaluating the effectiveness of a cross-layer approach.
    b. **Stasis Time:** This is the dissemination time after the network has formed and is stable. At this time, each node has a complete list of routing table

entries. This is useful in evaluating the real dissemination time when routes do not change after the network and routes have stabilized.

4. **Cumulative SSMs generated:** The cumulative SSMs generated as a function of time are helpful in analyzing the service discovery trend in a dynamically changing network to evaluate scalability. A scalable protocol is characterized by a slow rising trend in the cumulate SSM curve over time, which indicates lower network utilization for service discovery.

3.7.2 Test variables

Test variables are user controlled parameters which evaluate the performance of the scheme with each variation. The test variables chosen are:

1. **TC interval (seconds):** The TC interval is amount of time between TC broadcasts for an MPR node. The TC interval determines the rate at which topology changes are distributed to nodes in the network. A low TC interval implies faster routing updates at the expense of increased overhead. **(3sec, 5sec,7sec,10sec,12sec,15sec)**
2. **Number of service providers (%):** This variable is the number of nodes which possess an active service for distribution, as a percentage of the total. This is useful for determining the impact of increasing the number of services. **(30%, 50%, 70%, 100%)**
3. **Number of nodes:** The number of nodes is varied and the metrics are recorded. The node density is kept constant by adjusting the square grid area accordingly. The network scalability is evaluated with this vest vector. **(10nodes, 25nodes, 50nodes, 100nodes)** in (500m^2, 790m^2, 1118m^2, 1581m^2 grid)
4. **Individual node statistic:** This is a graph showing a particular statistic for all nodes. As an example, the number of updates is plotted against node number.
5. **Time trend (seconds):** This evaluates the performance of a particular metric as a function of time. For example, cumulative SSMs generated over time.
6. **Node velocity (meters/second):** This metric is obtained for randomly changing node velocities. The mobility model is the Random Waypoint model which is standard for all network mobility tests. **(0m/s, 5m/s, 15m/s, 25m/s)**

3.8 Results and discussion

3.8.1 Message overhead and TC interval

A network consisting of 11 nodes is simulated with varying TC intervals. All nodes are stationary, each node records statistics individually and, finally, the sum of all bytes and number of messages is obtained for each type of protocol. Each node has 1 service of the specific protocol for each test, which makes it a total of 11 services in the network at a given time. The tests were run for 1000sec.

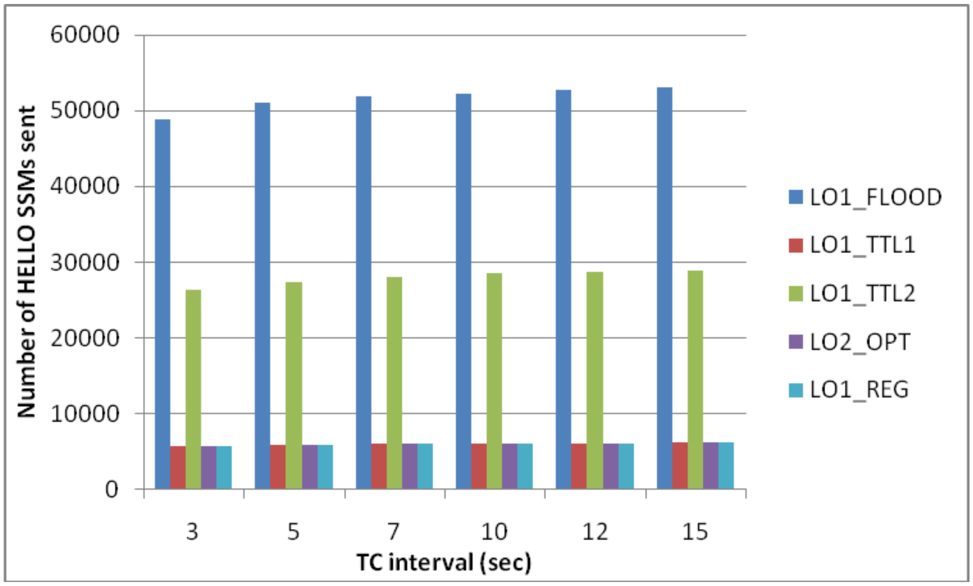


Figure 32. Number of HELLO SSMs vs TC interval

The sum of HELLO SSMs transmitted by nodes in all five tests is shown in Figure 32. It can be seen that in the LOC1_FLOOD method, and with LOC1_TTL2, the metric far exceeds the SSMs transmitted by the other methods. Also as expected, the number of HELLO SSMs in each method is not noticeably affected by the TC interval in the LO2_OPT and LO2_REG schemes. It can be concluded that LOC1_FLOOD and to a certain extent LOC1_TTL2, demand a significant overhead.

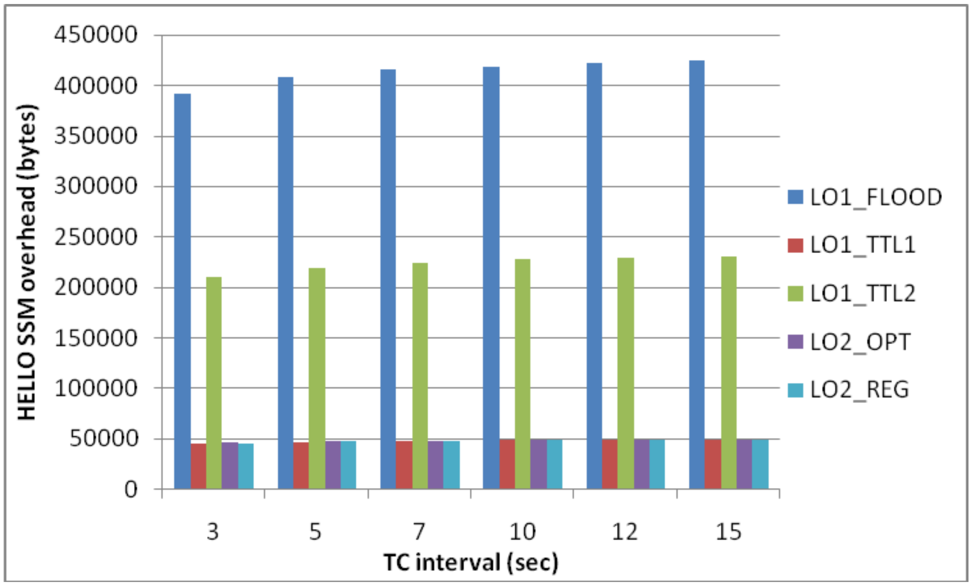


Figure 33. HELLO SSM Overhead in bytes vs TC interval

The Figure 33 shows the SSM overhead in bytes for all nodes in the five tests as a function of TC interval. The FLOOD and TTL2 overhead in bytes is significantly higher than the other methods which limits the scalability of the NS protocol for scenarios larger than 1 hop. And for larger networks, the other protocols are more suitable.

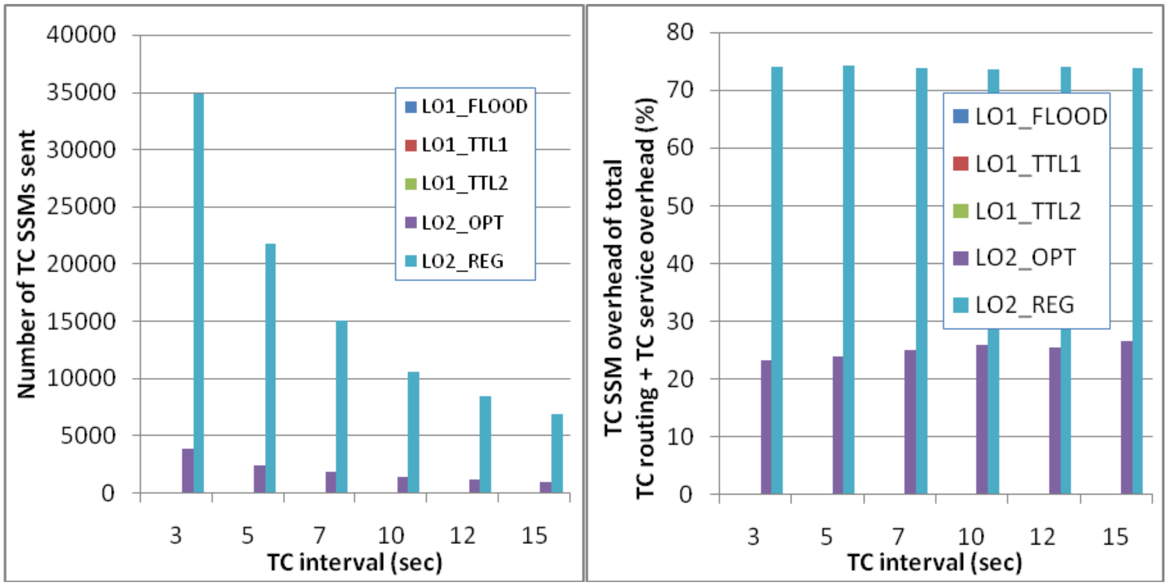


Figure 34. Number of TC SSMs sent (left) and % TC overhead (right) vs TC interval

The Figure 34 (left) shows the overhead statistics for the TC SSMs as a function of TC interval. It can be observed that as the TC interval increases, the number of TC SSMs reduce however as a percentage of the total overhead Figure 34 (right), it is nearly constant for both methods. This constant characteristic is a result of queuing OLSR TC information until the next TC message is sent and this indicates that service discovery overhead over TC is significant compared to the routing and HELLO message overhead, because the routing overhead in terms of bytes is nearly independent of the TC interval. The LO1 techniques do not use TC as a dissemination scheme and hence their values are zero. Finally, the overhead in the LOC2_OPT method is considerably lower than the overhead in the LOC2_REG method in both graphs as a result of the optimizations.

3.8.2 Message overhead and service providers

These tests explore the relationship between the message overhead and the number of services in a network. A stationary network is consisting of randomly selected nodes are initialized with 1 service at startup. Of the 11 nodes, 10 nodes are simulated with 30%, 50%, 70% and 100% of the total services (i.e. for example; in the 50% scenario, 5 of the 10 nodes distribute 1 service each.). The 11th node is an observer node which also participates in service discovery but without possession of a service. The test is run for 1000 seconds. The TC interval is the 5 sec default.

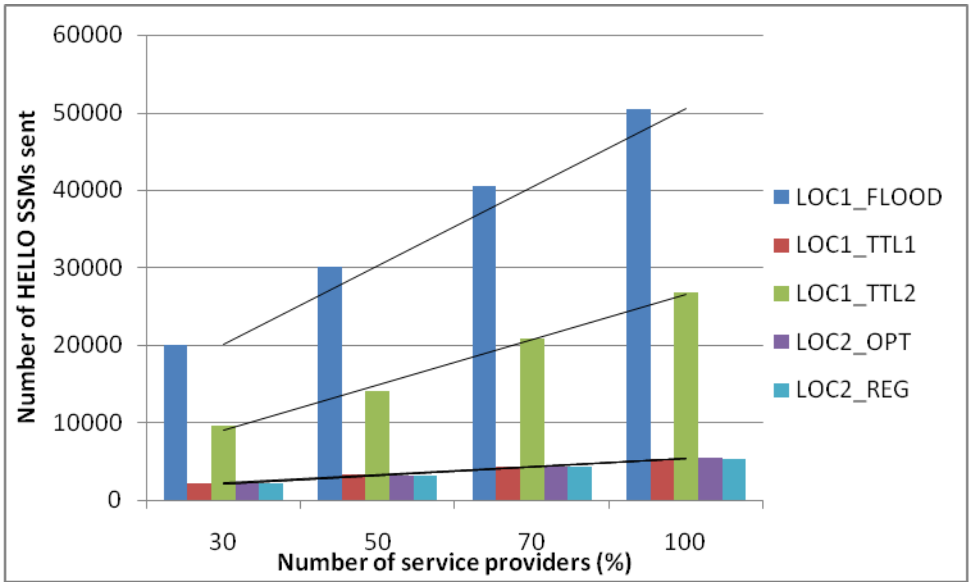


Figure 35. Number of HELLO SSMs vs Number of service providers

In the figure, as expected, the LOC1_FLOOD technique and LOC1 in general, does not scale well as the number of services in the network is increased. This is indicated by the relatively high angle of the slope trend. Second, the OPT and REG methods provide relatively better scalability when the number of services is increased.

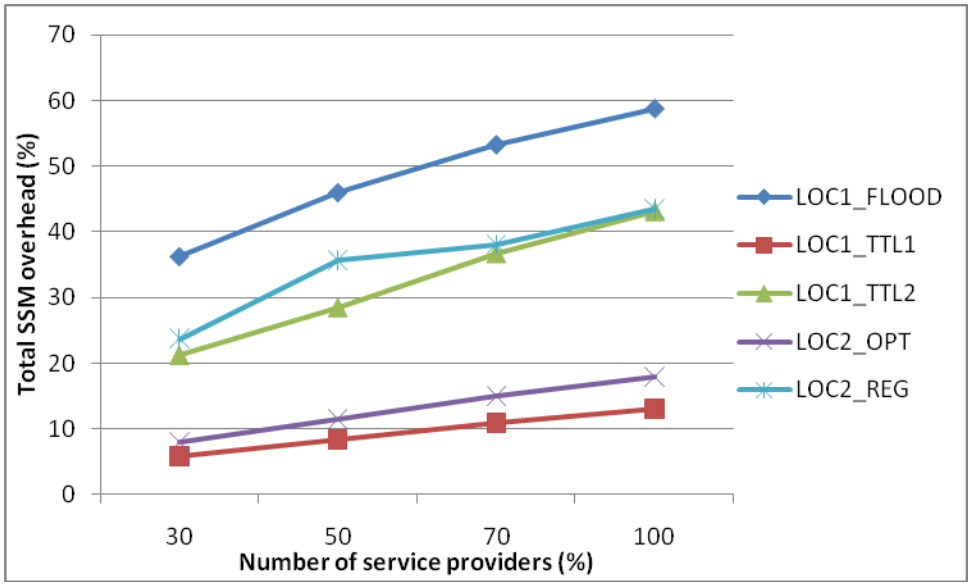


Figure 36. Total SSM overhead vs Number of service providers

The total SSM overhead when compared to the number of service providers indicates that LOC1_FLOOD SSM messages consume between 35-58% of the service and routing bytes in all cases. Conversely, the LOC2_OPT method provides a modest rise and consumes less than 20% of the overhead even for 100% of service providers. The REG method has also a high degree of total overhead approaching 45% with 100% of services. The LOC1_TTL1 and LOC1_TTL2 methods do not provide entire zone coverage, and TTL2 has a high overhead.

3.8.3 Number of updates and number of nodes

The relationship between the update messages forwarded or sent by each node is recorded for a static and stationary network. The TC interval is the default 5 seconds. Since routes and services do not change in this test, updates are redundant and serve no purpose which contributes to excessive processing and network overhead. This metric is evaluated for 350 seconds.

The graph of average number of updates for the three methods and different number of nodes indicates that the LOC2_REG method requires a large number of updates in the TC SSMs, and to a lesser degree, the LOC1_FLOOD method. The lack of an optimized cache maintenance scheme in the LOC2_REG and the LOC1 methods leads to a high number of messages being transmitted even when the network is stationary over long periods of time. This is because the underlying service discovery is proactive in nature. The OPT method performs well and it can be observed that with increasing number of nodes, the update number actually decreases. This can be attributed to the optimized cache maintenance scheme which is hybrid in nature.

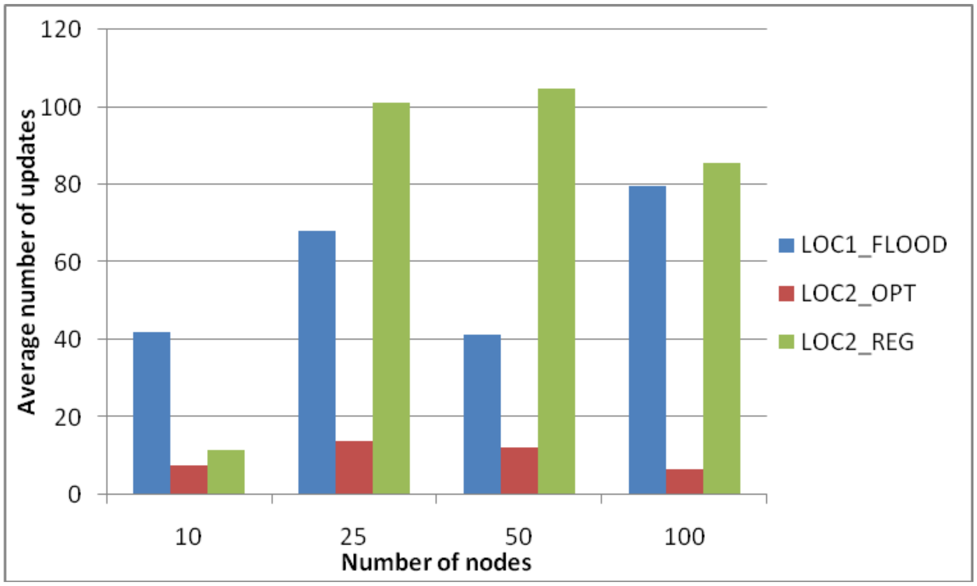


Figure 37. Average number of updates vs Number of nodes

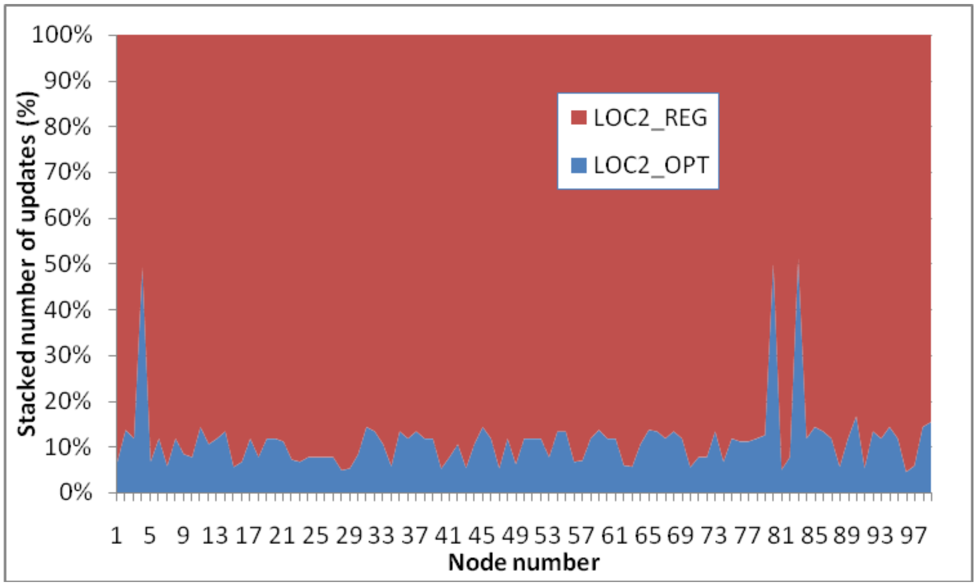


Figure 38. Stacked number of updates vs Node number (100 nodes)

The number of updates as a stacked percentage is shown for the two methods for the 100 node topology. It is interesting to note the differences in the REG and OPT method by presence of the optimized neighbor status maintenance mechanism that leads to the significant variation in their performance. A majority of these updates are redundant since neither the network topology nor the service parameters are changing.

3.8.4 Dissemination time and number of nodes

The convergence dissemination time is of importance because in this cross-layer protocol, the service discovery occurs simultaneously with the routing. It is expected that the convergence dissemination time would be greater than the stasis (after convergence) time. The convergence time is measured from t=0sec until every node has received the service. The TC interval is the default 5 sec. Stasis time is measured after 30 sec and from the instant the source node inserts the service. The number of nodes is varied between 10, 25, 50 and 100 and each case has one observer node.

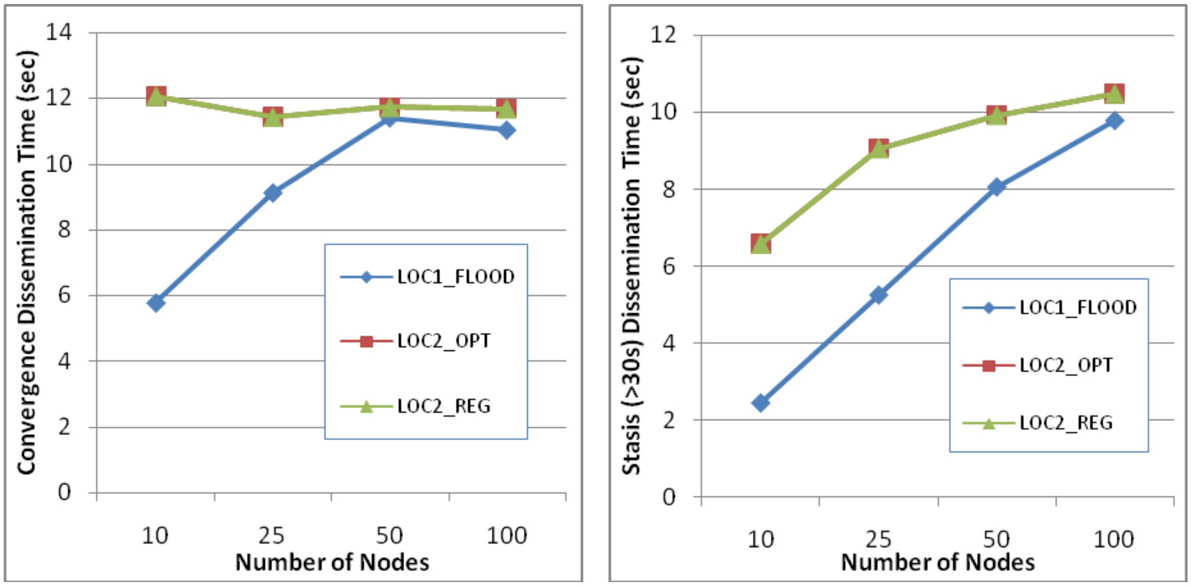


Figure 39. Convergence time (left) and Stasis time (right) vs Number of Nodes

In both the convergence time and stasis time, the LOC1_FLOOD technique outperforms the others in rapidness. This difference becomes relatively insignificant when the number of nodes increases due to the increased routing and discovery overhead. The dissemination times for the LOC2_OPT and LOC2_REG method, are identical and this is attributed to the fact that there is no difference in the two methods with regard to message handling and broadcast procedures. The other important detail is the difference between the stasis and convergence curves which indicates that service discovery occurs faster if the network is already formed, but this is not significant as the number of nodes increases. It can be concluded that the FLOOD technique has a better dissemination time for a small number of nodes and this advantage diminishes for larger networks.

3.8.5 Dissemination time and node velocity

The nodes follow the random waypoint mobility model with speeds up to 25 meters/sec in discrete steps. The dissemination time is observed for the nodes with these velocities and is also compared to the static (0 m/s) and stasis (>30sec) scenario. The TC interval is the default 5 sec and number of nodes is 10 + 1 observer.

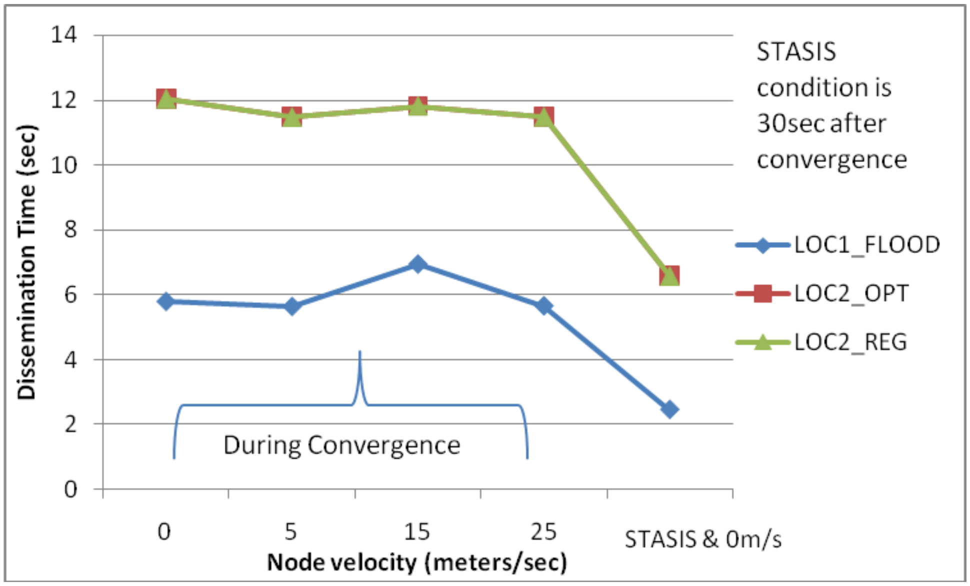


Figure 40. Dissemination time vs Node velocity

The convergence dissemination time, as discussed earlier, is found to be greater than the stasis dissemination time since it occurs during the route building phase. The node velocities do not significantly affect the performance of the LOC2_OPT and LOC2_REG methods. This is attributed to the OLSR protocol which proactively maintains states and accounts for link failures by changing the routing table. As the routes are broken and new routes formed, the service discovery occurs simultaneously with routing and services are available immediately when the new route is determined.

3.8.6 Dissemination time and TC interval

The TC interval is important for propagating services using the LOC2_OPT and LOC2_REG methods and bears a direct relationship on the dissemination time. A larger TC interval consequently will increase the dissemination time, and the extent of the impact is determined in this test scenario. It is expected that dissemination time will be a linear function of the TC interval and a propagation delay factor. The figure shows the comparison between the convergence and stasis dissemination times for various TC intervals and 10 nodes.

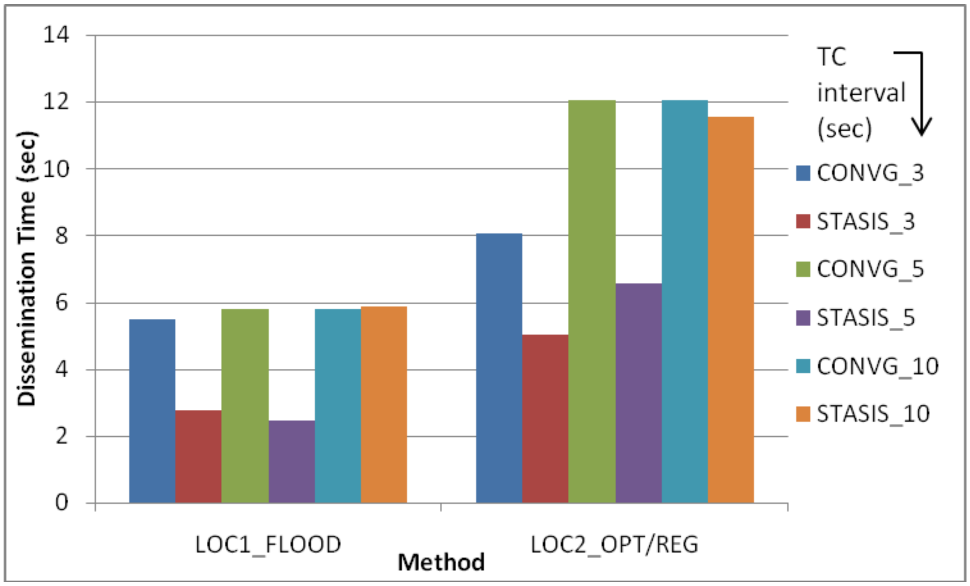


Figure 41. Dissemination time vs method for varying TC intervals

The time required for dissemination using the LOC2_OPT/REG methods is always more than the FLOOD technique in each scenario. This can be explained by the variation in the HELLO and TC intervals. The HELLO interval is much smaller, in this case 2 sec, compared to the TC interval. The propagation delay involves distribution of the SSM through HELLO flooding and since the HELLO timers expire nearly simultaneously, the SSM is available for insertion. FLOOD can begin distribution much earlier while the OPT/REG methods must necessarily wait for the first TC message to be sent. The first TC message is forwarded to the entire network using the MPR paths and this is the propagation delay in LOC2 methods. Again the difference between convergence and stasis conditions can be observed. Finally, unlike the OPT/REG methods, the FLOOD distribution is not affected by the TC interval.

3.8.7 Cumulative SSMs and time

The cumulative SSM curves for a particular node provide an insight into the process of service propagation for each of the test cases. The results are repeated for node mobility and its effects on the OPT and REG methods is discussed. Each test is run for 350 seconds and with 10 + 1 nodes. The Random WP mobility model is used for motion and the statistics are collected for every node. This test case below presents the values for a single node (number 8) which is selected since - 1) it is an MPR node and 2) it lies approximately in the topology center of the network. TC interval is 5 seconds.

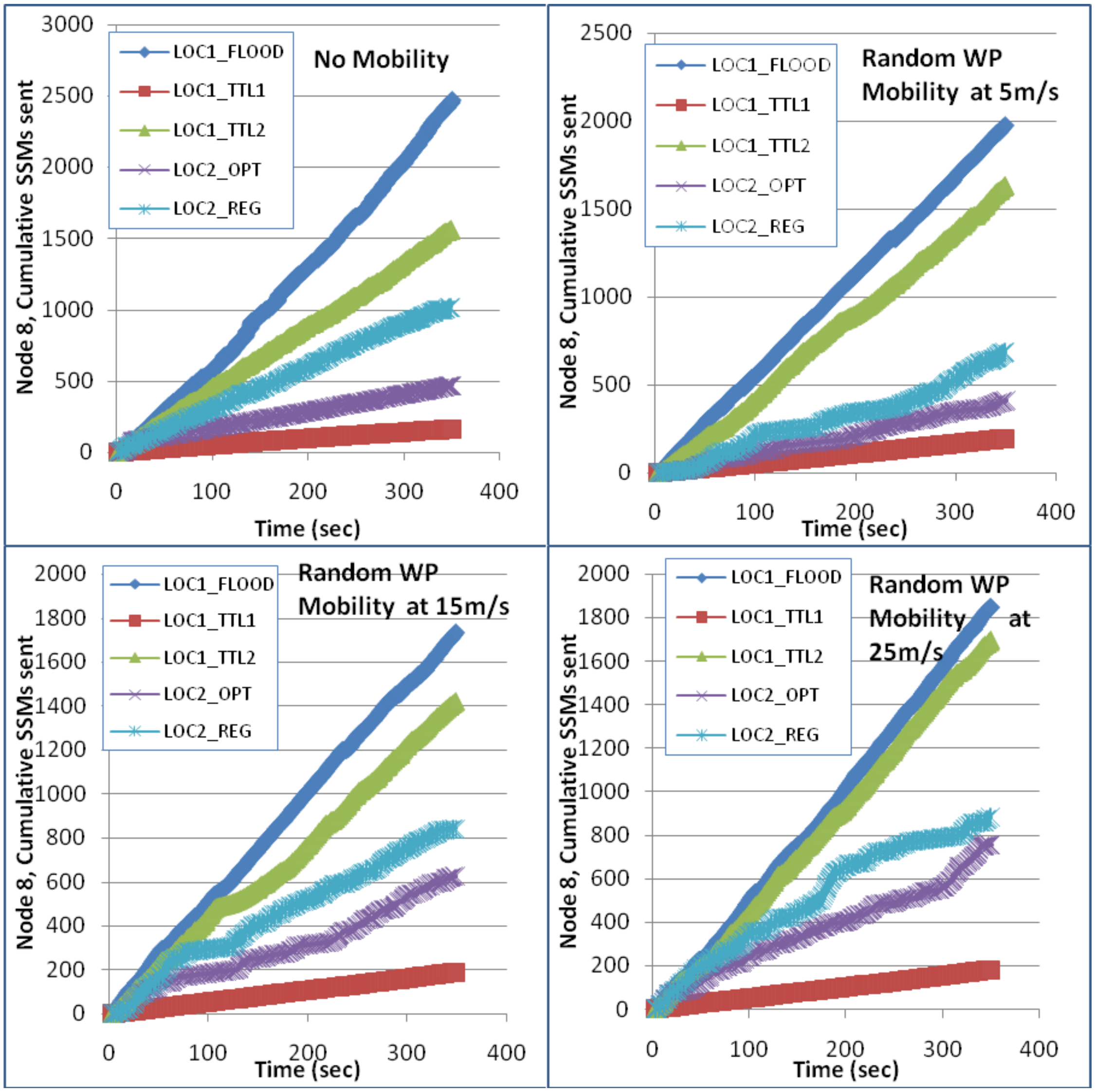


Figure 42. Cumulative SSM vs Time

The FLOOD and LOC1 techniques have a high rate at which SSMs are transmitted. The rate at which SSMs are sent using the LOC2_OPT method modestly continue to increase with increasing mobility, the exception being 5 m/s, which is the effect of the neighbor state maintenance mechanism. As the mobility rate, and hence the rate at which new connections are formed and destroyed increases, the characteristics of the OPT curve approaches that of the REG curve (ref the 25m/s case). The OPT method outperforms the REG and FLOOD methods since the curve is comparatively slow rising in all test cases.

3.9 Table 6. Summary of Results from Simulation

Metric	Test vector	Conclusion and Significance
Message Overhead	**TC interval**	LOC2_OPT generates far lower number of SSMs and the overhead remains nearly flat with changing TC interval compared to the other methods. This provides scalability when TC update rate is increased.
	Number of service providers	As number of service provider's increase, the SSM overhead of LOC2_OPT outperforms other protocols by a comparatively modest increase allowing it to accommodate more service providers without affecting network load.
Number of updates	**Number of nodes**	The LOC2_OPT outperforms the other protocols in terms of number of updates, mostly redundant, for a static network promoting lower processing and network overhead in the nodes.
Dissemination Time	**Number of nodes**	LOC2_OPT is slower compared to LOC1_FLOOD for lower number of nodes and the disadvantage diminishes as number of nodes is increased. LOC2_OPT therefore performs well for higher number of nodes.
	Node velocity	Node velocity does not affect the performance of any protocol significantly. Stasis case needs a lower time for dissemination compared to convergence. LOC2_OPT is slower than LOC1_FLOOD (and consequently all LOC1 cases) in all cases.
	TC interval	LOC2_OPT dissemination time is directly affected by the TC interval while LOC1_FLOOD is not. LOC2_OPT is slower than LOC1_FLOOD in all cases.
Cumulative SSMs	**Time (with Mobility)**	The LOC2_OPT outperforms all protocols in the tests which validates the effectiveness of the optimizations.

Chapter 4 : Conclusions

The chapter highlights the salient features of this research work and provides a brief summary of the contributions along with the potential for future research.

4.1 Summary

In Chapter 2, I discussed one of the applications of cognitive radios for providing public safety interoperability using an existing reconfigurable framework. I have discussed the need for portability in the public safety cognitive radio system and highlighted some of reasons why complete portability is not currently feasible. An immediate solution to this problem is by providing remote access to cognitive radio so that a first responder can remotely utilize its features. This chapter discusses the implementation details of the VPSCR remote access architecture as well as the PDA front end. The control sequence, network access protocol and audio services are some of the features which provide seamless integration of the PDA with the PSCR framework.

In Chapter 3, I introduced the need for extending the PDA and VPSCR scenario to a wide area network. To summarize, many devices provide unique capabilities and it may be expensive or prohibitive to equip every device with all these features. By using a service discovery protocol in a wide area network, it would be possible to efficiently locate and utilize these services. Existing service discovery protocols are described and their inability to be used in public safety networks is discussed. A public safety network is proposed for discovery of devices and provides a hierarchical structure for service localization depending on the proximity. An algorithm is presented for a new hybrid service discovery protocol which is optimized for cache and state maintenance. Finally, the protocol is analyzed with baseline service discovery schemes in simulation and the results corroborate the effectiveness of the three optimizations

4.2 Contributions

1. A remote control architecture and framework is developed for the VPSCR which enables access from any remote device supporting the extensible protocol.
2. The three VPSCR modes of operation – Scan, Gateway, and Talk have been fully ported and are now remotely accessible.
3. Authentication and state maintenance control logic is developed on the VPSCR which also remotely identifies the status of the hardware.
4. Digital and analog audio interoperation between the PDA and analog waveform is developed and implemented. Audio processing and buffer structure for the PDA has been designed after audio rate analysis of the stream. Digital and analog audio processing has been developed on the VPSCR.
5. A public safety network is proposed utilizing a reduced number of backbone nodes to provide wide area coverage of public safety features.
6. A novel service discovery protocol is designed to optimize the service distribution based on proximity of the service source.
7. Three service discovery optimizations have been introduced to reduce overhead and improve scalability. They provide automatic cache maintenance mechanisms in the network.
8. A full protocol model has been developed using the OMNeT++ simulator.
9. The optimized protocol was compared with other service discovery methods and the protocol outperforms the competition in every test with the exception of dissemination time. Since this protocol is optimized for overhead reduction and scalability, dissemination time is a secondary consideration.

4.3 Potential future work

The possible next stage would be to extend the remote access functionality with a web based interface to the VPSCR enabling remote access from any networked machine. Real-time video and data services can be provided to the remote devices to extend the feature list of the system. A more secure authentication scheme is desirable without relying on the inherent 802.11 encryption and which can be used over any radio link. A digital audio communication scheme between multiple PDAs, utilizing the VPSCR as a base station, would also be useful for the public safety community when communicating with a variety of radios. Since digital audio is being streamed to the VPSCR, a next step would be to allow digital modulation of the audio data over supported waveforms such as BPSK, QPSK and MPSK.

The immediately obvious next step would be to integrate the service discovery protocol into an OLSR implementation. This can be achieved without significant effort since the protocol simulation is C++ based and majority of the code resides in the message handler and sender methods of OLSR. Future efforts could also extend the service discovery simulation and implementation to the proposed MPSZ backbone, which has been discussed but not simulated. Future applications of the service discovery protocol may include wireless sensor networks consisting of large number of nodes which exploit the optimizations for lower overhead and scalability.

Appendix A – PDA operation guide

a. Authentication and Association

Authentication involves exchange of identity information to allow the VPSCR to recognize an authorized PDA and create a connection. The current prototype does not use username and password authentication, however this feature could be incorporated in future releases. PDA's access and control of the VPSCR can be accepted or denied by the VPSCR.

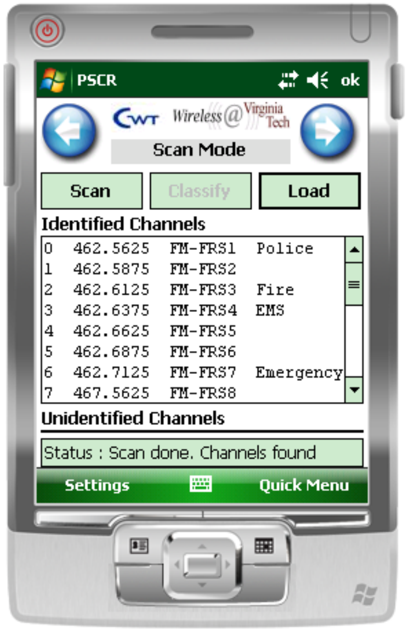
Association involves the exchange of connection information and parameters such as IP addresses, port numbers and additionally, supported codecs. Both the VPSCR and PDA exchange this information. Association occurs after successful authentication and can be initiated either by a PDA or the VPSCR.

Association can be automatic or manual. Manual association occurs when the user presses the "***Associate***" menu button. Automatic association occurs when the user directly uses any of the features of the VPSCR. On successful association the PDA will display the ***Associated status*** message.

On association, the PDA can access a number of VPSCR services which include (but are not limited to):

1. ***Scan Mode***: The VPSCR can scan the radio environment, in a specified spectrum range, for active public safety waveforms and provide this list to the PDA. The scan mode also includes classification of unknown modulation schemes. Commonly used public safety waveforms and frequencies can also be loaded from a database.
2. ***Gateway Mode***: The PDA can issue a command to establish a bridge between two incompatible public safety waveforms using different frequencies and modulation schemes.
3. ***Talk Mode***: The PDA can establish an audio connection with any of the public safety waveforms found or returned from the database.

The mode scroll buttons, Figure 2, are used to switch between Scan, Gateway and Talk mode.



Identified Channels

```
0  462.5625  FM-FRS1  Police
1  462.5875  FM-FRS2
2  462.6125  FM-FRS3  Fire
3  462.6375  FM-FRS4  EMS
4  462.6625  FM-FRS5
5  462.6875  FM-FRS6
6  462.7125  FM-FRS7  Emergency
7  467.5625  FM-FRS8
```

Unidentified Channels

Status : Scan done. Channels found

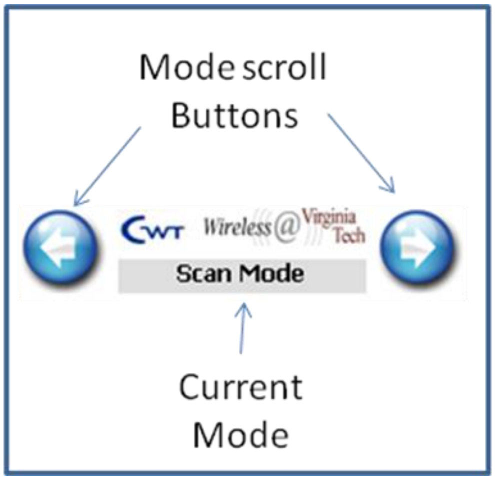
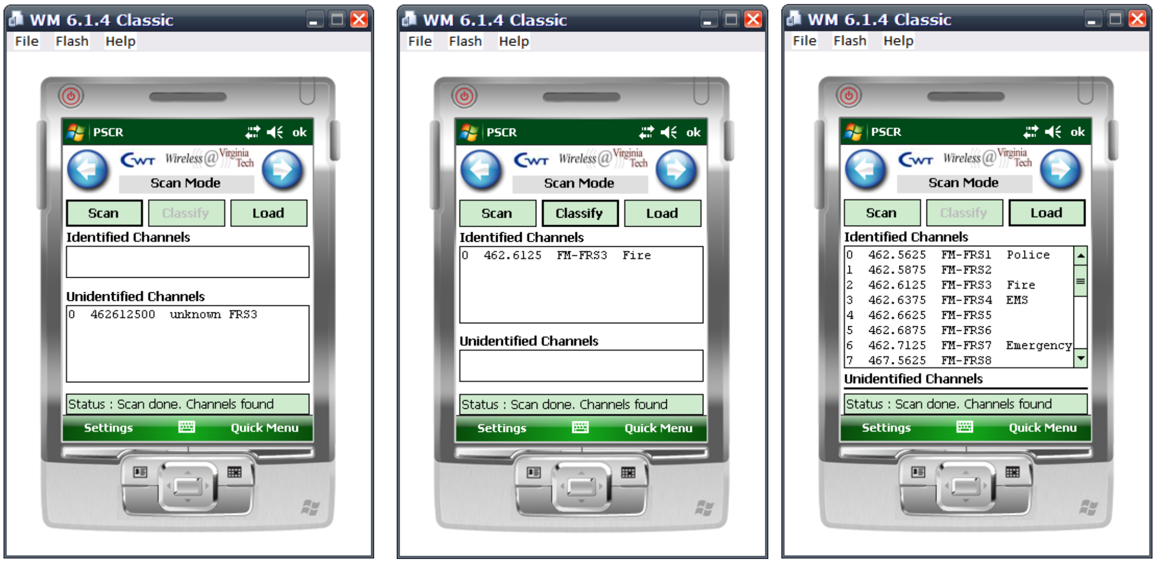


Figure 1 (left) - Sample PDA screenshot
Figure 2 (above) - Mode selection scroll buttons

b. Scan Mode

The VPSCR hardware is a sophisticated public safety cognitive radio capable of scanning and classifying active public safety waveforms. The results of this operation are relayed to the PDA and displayed on the screen.



```
0   462612500   unknown FRS3
Status : Scan done. Channels found
```

```
0   462.6125   FM-FRS3   Fire
Status : Scan done. Channels found
```

```
0   462.5625   FM-FRS1   Police
1   462.5875   FM-FRS2
2   462.6125   FM-FRS3   Fire
3   462.6375   FM-FRS4   EMS
4   462.6625   FM-FRS5
5   462.6875   FM-FRS6
6   462.7125   FM-FRS7   Emergency
7   467.5625   FM-FRS8
Status : Scan done. Channels found
```

Figure 3

a) Scan spectrum
b) Classify the found channel
c) Load common waveforms from database

Figure 3 (a) shows the results of a scan between 461 MHz and 468 MHz. A single channel was found and was displayed as an ‘unidentified channel’ with modulation type ‘unknown’. Setting the desired frequency range is explained in the section on Administrative settings.

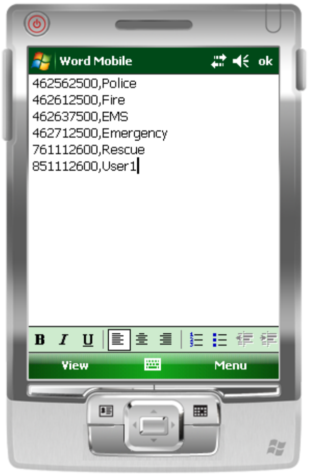
Figure 3 (b) shows the result of a classification for the ‘unidentified channel’. The channel was determined to be a FM-FRS3 signal that is assigned to “Fire” public safety waveforms. Channel Name assignment is explained below.

Figure 3 (c) shows the result of a load from database for commonly used public safety waveforms.

Figure 3 (d) (right), Channel Name Assignment:

The channel name database is a single file - ***\ChannelNames.txt*** located on the PDA. The file stores the names of public safety agencies associated with a particular channel frequency. If a Channel does not have a name associated with it, then it should not be entered here. This file can be manually updated by the law enforcement officer using the PDA.

An example ChannelNames.txt file is shown in Figure 3(d).

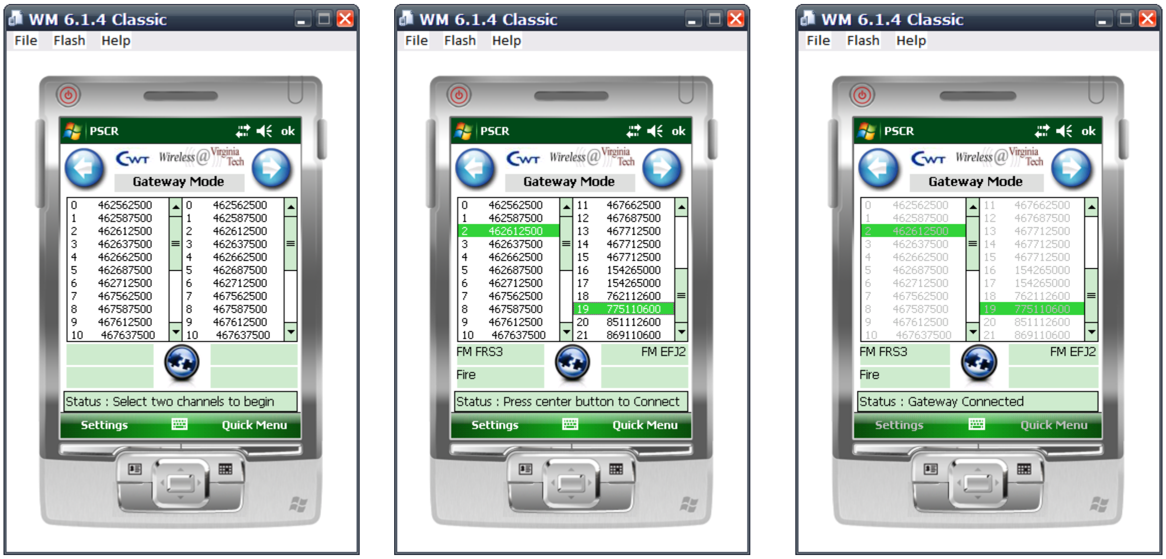


```
462562500,Police
462612500,Fire
462637500,EMS
462712500,Emergency
761112600,Rescue
851112600,User1
```

Note: The file must be saved as comma separated ***ASCII text format***, shown in the figure, and not in MS Word (.doc) or other unsupported formats.

c. Gateway Mode

The Gateway Mode allows the VPSCR to bridge two incompatible public safety waveforms so that public safety officials using incompatible radios can talk to each other. The PDA can issue a remote command to the VPSCR to initiate and terminate the bridge.



0	462562500	0	462562500
1	462587500	1	462587500
2	462612500	2	462612500
3	462637500	3	462637500
4	462662500	4	462662500
5	462687500	5	462687500
6	462712500	6	462712500
7	467562500	7	467562500
8	467587500	8	467587500
9	467612500	9	467612500
10	467637500	10	467637500

Status : Select two channels to begin

0	462562500	11	467662500
1	462587500	12	467687500
2	462612500	13	467712500
3	462637500	14	467712500
4	462662500	15	467712500
5	462687500	16	154265000
6	462712500	17	154265000
7	467562500	18	762112600
8	467587500	19	775110600
9	467612500	20	851112600
10	467637500	21	869110600

FM FRS3 | FM EFJ2
Fire

Status : Press center button to Connect

0	462562500	11	467662500
1	462587500	12	467687500
2	462612500	13	467712500
3	462637500	14	467712500
4	462662500	15	467712500
5	462687500	16	154265000
6	462712500	17	154265000
7	467562500	18	762112600
8	467587500	19	775110600
9	467612500	20	851112600
10	467637500	21	869110600

FM FRS3 | FM EFJ2
Fire

Status : Gateway Connected

Figure 4
(a) Gateway mode default screen
(b) Selecting two waveforms
(c) Gateway connected status message

Figure 4 (a) displays two lists where a channel from each list can be selected to bridge together.
Figure 4 (b) shows two channels being selected for a bridge. In the figure, FM-FRS3 (Fire) at 462.6125 MHz and FM-EFJ2 at 775.1106 MHz are selected. The details for a selected channel are individually displayed below the lists for each side.
Figure 4 (c) shows a connected gateway. The other options on the screen are grayed out.

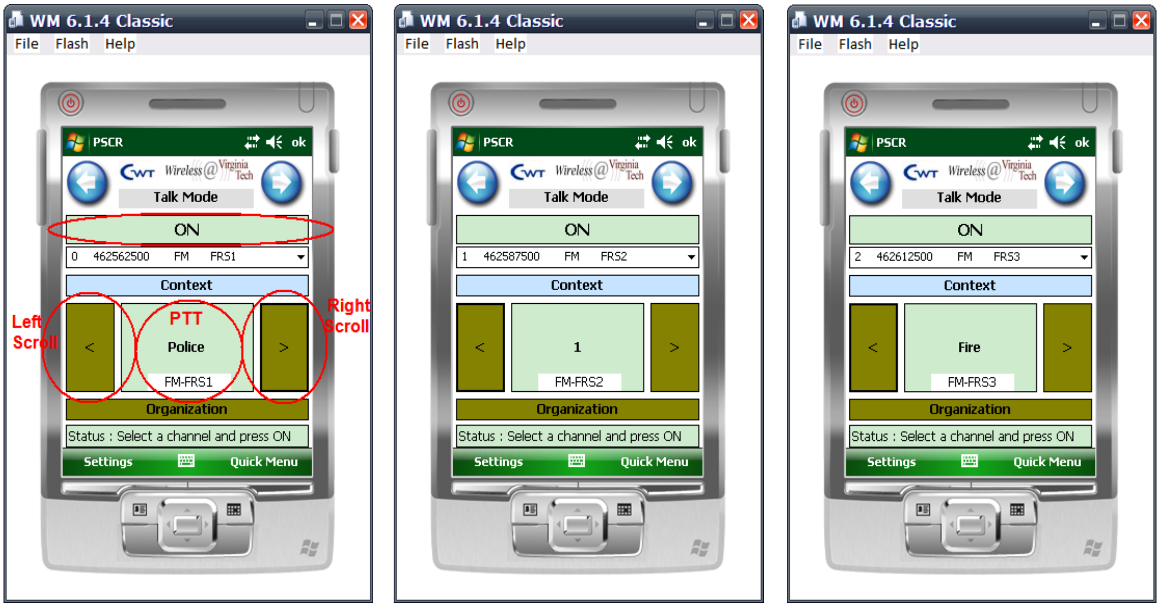
The ***center button***, located below the two lists, is used to create and destroy the bridge. The "***Gateway Connected***" message is an acknowledgement from the VPSCR that the gateway has been successfully established. All options and menu screens are disabled after the gateway is connected. They are re-enabled when the gateway is disconnected.

The "***Gateway Disconnected***" message is an acknowledgement from the VPSCR that the gateway has been torn down.

Note: Being able to bridge two waveforms is hardware dependent. The hardware used is a Universal Software Radio Peripheral (USRP) that uses two independent RF daughterboards for the bridge. The bridge will not be successful if the hardware platform or daughterboards do not support the selected waveforms' operating frequency.

d. Talk Mode

The Talk mode is used to connect to an active public safety waveform and establish an audio link between the PDA and the public safety waveform (through the VPSCR).



0 462562500 FM FRS1

1 462587500 FM FRS2

2 462612500 FM FRS3

Status : Select a channel and press ON

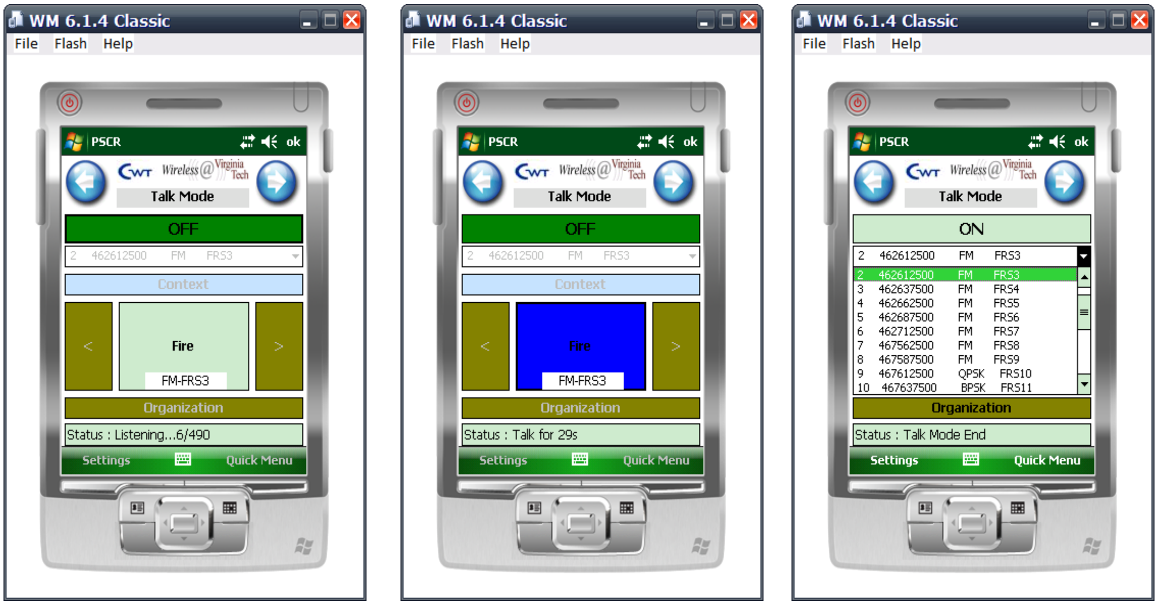
Figure 5
(a) Talk mode screen
(b) Left button pressed once
(c) Left button pressed again

The talk mode screen is shown in Figure 5 (a). The screen consists of six buttons of which two are of chief importance (circled in red). The “ON” button will initiate the Talk mode to the specified waveform or public safety agency. For example, in Figure 5 (a), if the ON button were pressed, it would create an audio link to the Police (FM-FRS1) waveform. Similarly for Figure 5 (b) and 5 (c) the ON button would create audio links to Channel 1 (FM-FRS2) and Fire (FM-FRS3) waveforms respectively. After the ON button is pressed, the PDA will automatically start ***listening for incoming audio.***

The Left and right scroll buttons will scroll the currently selected waveform. It can be observed from 5 (a) (b) (c) that after pressing the left scroll button the text on the buttons scrolls left and simultaneously changes the waveform displayed on the large button located under ‘ON’.

The large button immediately below the ON button, also circled in red, has a changing text depending on the selected waveform. This serves as a ***Push-To-Talk (PTT) button***. For example, in Figure 5(a), the PTT will allow the PDA to talk to the Police, in Figure 5 (b) to Channel 1 and in Figure 5 (c) to Fire waveforms. The ON button needs to be pressed to initiate the Talk mode before PTT can be operated. The PTT will begin streaming audio from the PDA to the selected public safety waveform.

The context and organization buttons are currently unused and will be used in the future to assign logical categories to the public safety waveforms. For example, waveforms that belong to the same group would be indicated by the background color of the organization button.



Status : Listening...6/490

Status : Talk for 29s

2	462612500	FM	FRS3
2	462612500	FM	FRS3
3	462637500	FM	FRS4
4	462662500	FM	FRS5
5	462687500	FM	FRS6
6	462712500	FM	FRS7
7	467562500	FM	FRS8
8	467587500	FM	FRS9
9	467612500	QPSK	FRS10
10	467637500	BPSK	FRS11

Status : Talk Mode End

Figure 6

a) Talk mode (listening)

b) Talk mode (talking)

c) Talk mode easy dropdown list

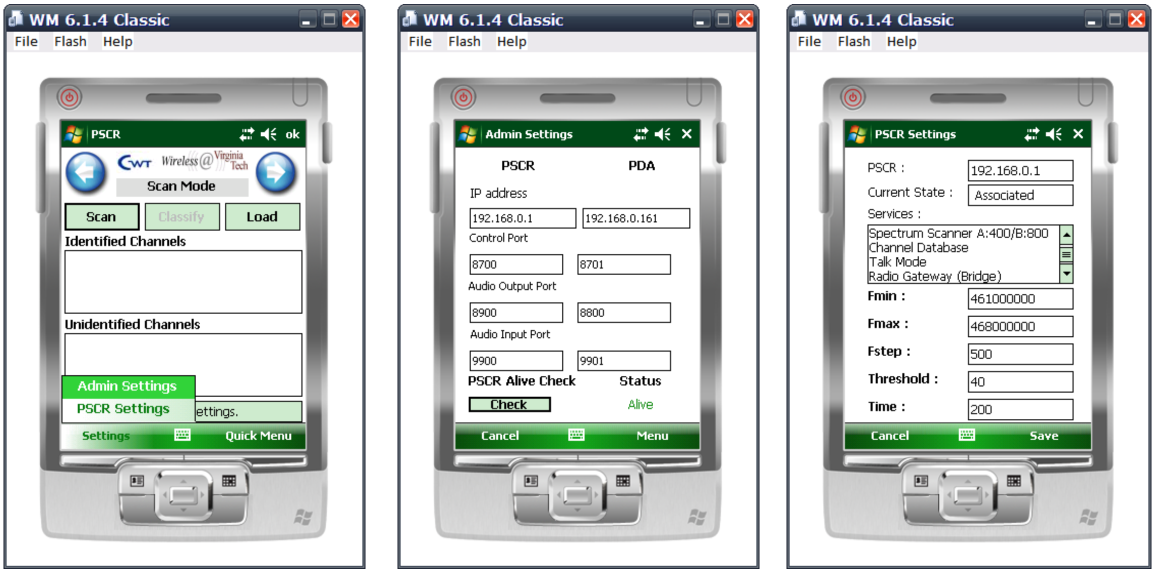
Figure 6 (a) displays the screen after the "ON" button has been pressed. This initiates an audio connection between EMS waveform (462.6375 MHz FM-FRS4) and the PDA. After pressing the ON button, the text changes to "OFF" and background color turns green to indicate that listening is in progress. If "OFF" is pressed then Talk mode will end. **The status bar indicates the length, approximately 4 minutes, for which the listen can run without re-initiating the connection. This is due to limitations of this device with low buffer memory**. Future work will eliminate this minor inconvenience. The connection can be manually re-initiated by pressing the "OFF" button and subsequently pressing "ON".

Figure 6 (b) shows the screen after the PTT button has been pressed. The PTT button turns Blue to indicate that an audio stream from PDA has been initiated. The user can talk into the microphone and audio will be transmitted from the PDA to the public safety waveform. To begin listening (and stop talk) the user should press the PTT button again. **The talk from PDA can be a maximum of 30 seconds in a single PTT session, due to limitations of the buffer size in the mobile device**. Future work will eliminate this limitation. The button can be reactivated immediately after 30 seconds are complete. The PTT button changes to the original color when listening begins (talk stops).

Figure 6 (c) shows the easy dropdown menu for waveforms and frequencies. A user can select a waveform from this list. This selection reflects the changes to the scroll buttons as well. The procedure for initiating listen and PTT (talk) is the same; however this menu is a quick alternative to using the left/right scroll buttons. All menu screens, dropdowns and buttons, apart from ON/OFF and PTT, are disabled during talk mode. They are re-enabled when talk mode is turned off.

e. Administration and Settings Menu

The administration and settings menu is located on the bottom left corner of the screen. This screen allows access to radio parameters and VPSCR connection parameters.



	PSCR	PDA
IP address	192.168.0.1	192.168.0.161
Control Port	8700	8701
Audio Output Port	8900	8800
Audio Input Port	9900	9901

PSCR Alive Check: Check
Status: Alive

PSCR : 192.168.0.1
Current State : Associated
Services :
Spectrum Scanner A:400/B:800
Channel Database
Talk Mode
Radio Gateway (Bridge)
Fmin : 461000000
Fmax : 468000000
Fstep : 500
Threshold : 40
Time : 200

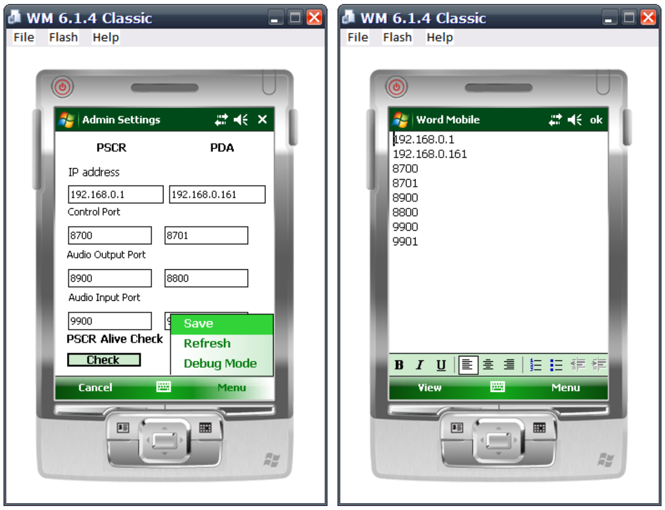
Figure 7
(a) Left Settings menu popup
(b) Admin Settings screen
(c) PSCR Settings screen

Figure 7 (a) shows the two options when the Settings button is selected.

The **Admin settings** screen is shown in Figure 7 (b). The “Association” parameters are displayed here. These parameters are either exchanged after Association or from the \Settings.txt file shown in Figure 8(b). The PSCR Alive Status check is used to determine if the VPSCR is active for the specified settings (IP and ports).

The **PSCR Settings** screen, Figure 7 (a) displays the IP address of the VPSCR and its association status. The services display the list of services that the PSCR supports. The automatic exchange of services has not yet been fully implemented. This field is populated locally. The Save button (bottom right) is used to save the settings for the current run.

Scanner/Classifier parameters:
Fmin, Fmax = Minimum and maximum frequency range for the spectrum scanner.
Fstep = Frequency step threshold for the scanner.
Threshold = Noise floor threshold for the scanner and classifier.
Time = Time for scanning, determines the accuracy and length of the scan.
Note: The Scan parameters are hardware dependent. Please consult the PSCR hardware documentation on selecting the correct parameters for the specific USRP.



	PSCR	PDA
IP address	192.168.0.1	192.168.0.161
Control Port	8700	8701
Audio Output Port	8900	8800
Audio Input Port	9900	9
PSCR Alive Check		

```
192.168.0.1
192.168.0.161
8700
8701
8900
8800
9900
9901
```

Figure 8 (right)
a) Admin Settings Right menu
b) \Settings.txt file

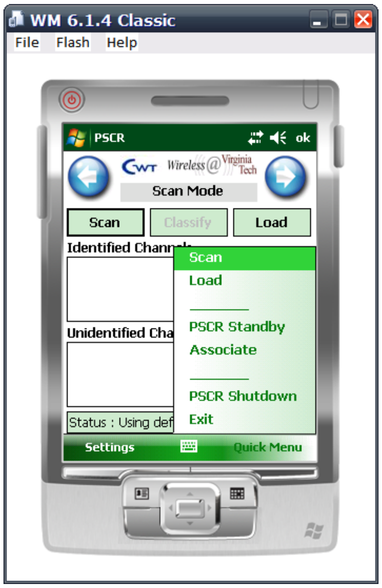
The right menu shown in Figure 8 (a) is for Save, Refresh and Debug Mode. The Debug Mode is used only for troubleshooting the Talk mode - this should only be used by developers. The Debug mode will change the appearance of the Talk mode to display a log of behind the scenes operation.

The \Settings.txt file contents are shown in Figure 8 (b). The file is in ASCII text format. The explanation of the parameters is given below:

192.168.0.1	= VPSCR IP address
192.168.0.161	= PDA IP Address
8700	= VPSCR Control Port
8701	= PDA Control Port
8900	= VPSCR Audio Output Port
8800	= PDA Audio Output Port
9900	= PSCR Audio Input Port
9901	= PDA Audio Input Port

These are default association parameters which can be overridden by the \Settings.txt file.
Note: Changing these parameters will require subsequent changes to the VPSCR settings file located on the VPCSR machine which follows the same format.

f. Quick Access Menu



Status : Using def

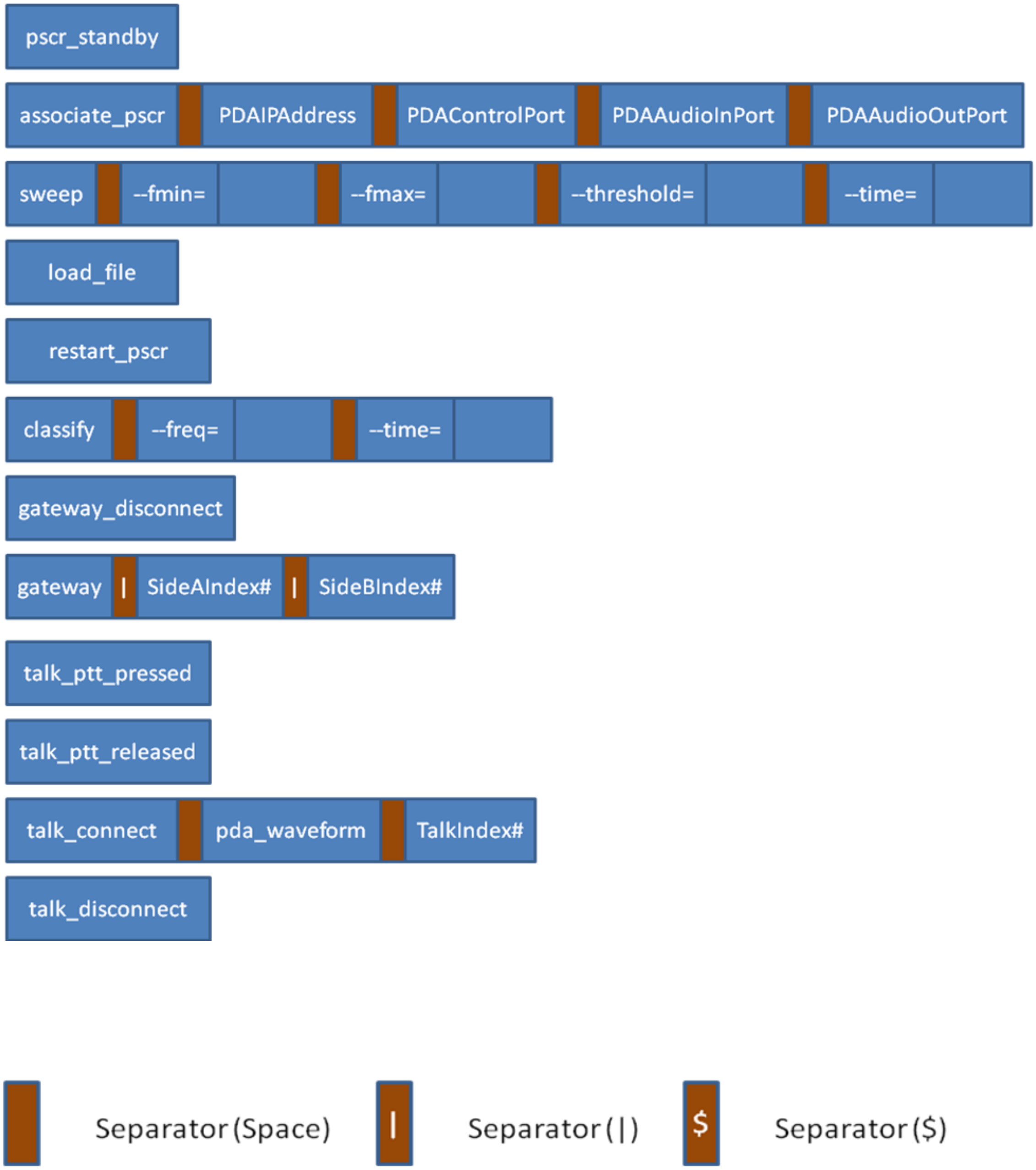
Figure 9 (right), Quick Access Menu

The Quick Menu appears on all mode screens for quick access to these functions:

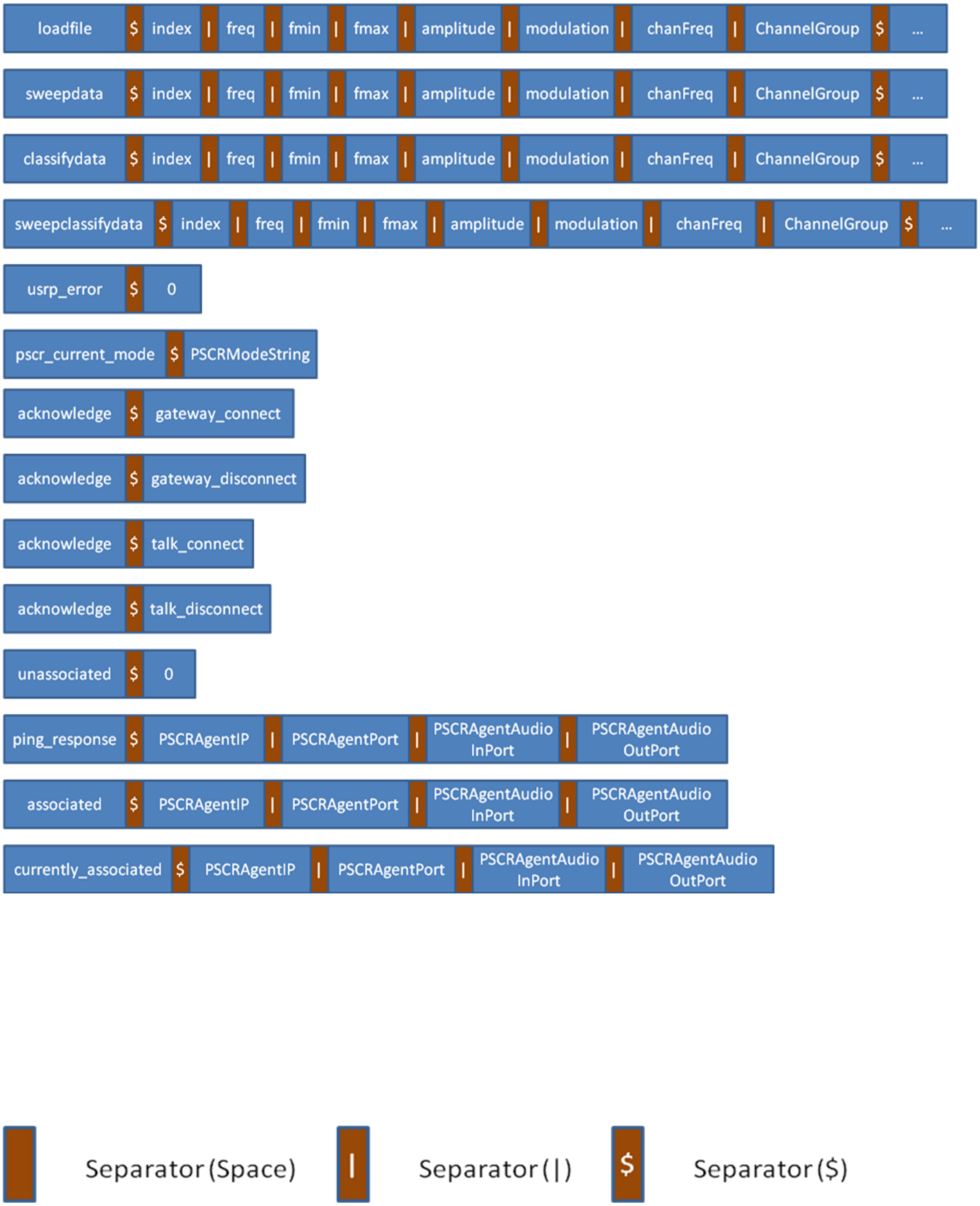
1. **Scan**
2. **Load**
3. **PSCR Standby** – PSCR Standby is used if the PDA application terminates abnormally during a mode. If the PSCR is currently active, the status message will display the current mode. PSCR Standby is used to return the PSCR to its original standby condition and wait for new commands.
4. **Associate** – Manual Association button.
5. **PSCR Shutdown** – To exit the PSCR application on the PSCR laptop. This does not power off the hardware.
6. **Exit** – To exit the PDA application. This will cause the PSCR to stop all Talk and Gateway connections.

Appendix B – VPSCR Command Format

a. PDA to VPSCR command formats


pscr_standby
associate_pscr ▮ PDAIPAddress ▮ PDAControlPort ▮ PDAAudioInPort ▮ PDAAudioOutPort
sweep ▮ --fmin= ▮ --fmax= ▮ --threshold= ▮ --time=
load_file
restart_pscr
classify ▮ --freq= ▮ --time=
gateway_disconnect
gateway | SideAIndex# | SideBIndex#
talk_ptt_pressed
talk_ptt_released
talk_connect ▮ pda_waveform ▮ TalkIndex#
talk_disconnect
▮ Separator (Space)
| Separator (|)
$ Separator ($)

b. VPSCR to PDA command formats



```
loadfile $ index | freq | fmin | fmax | amplitude | modulation | chanFreq | ChannelGroup $ ...
sweepdata $ index | freq | fmin | fmax | amplitude | modulation | chanFreq | ChannelGroup $ ...
classifydata $ index | freq | fmin | fmax | amplitude | modulation | chanFreq | ChannelGroup $ ...
sweepclassifydata $ index | freq | fmin | fmax | amplitude | modulation | chanFreq | ChannelGroup $ ...
usrp_error $ 0
pscr_current_mode $ PSCRModeString
acknowledge $ gateway_connect
acknowledge $ gateway_disconnect
acknowledge $ talk_connect
acknowledge $ talk_disconnect
unassociated $ 0
ping_response $ PSCRAgentIP | PSCRAgentPort | PSCRAgentAudio InPort | PSCRAgentAudio OutPort
associated $ PSCRAgentIP | PSCRAgentPort | PSCRAgentAudio InPort | PSCRAgentAudio OutPort
currently_associated $ PSCRAgentIP | PSCRAgentPort | PSCRAgentAudio InPort | PSCRAgentAudio OutPort
```

▨ Separator (Space) | Separator (|) $ Separator ($)

Rohit D. Rangnekar

Appendix C – List of Abbreviations

CE	Cognitive Engine
CR	Cognitive Radio
CWT	Virginia Tech Center for Wireless Telecommunications
DCache	Deleted Cache
FRS	Family Radio Service
GPP	General Purpose Processor
GUI	Graphical User Interface
LCache	Local Cache
LOC	Localization
LOC1_FLOOD	Localization=1, Protocol=NS Flood Global
LOC2_OPT	Localization=2, Protocol=SPSZ Optimized
LOC2_REG	Localization=2, Protocol=SPSZ Regular
MANET	Mobile Ad-hoc Network
MPR	Multi Point Relay
MPSZ	Multi-PSZ Service
NAT	Network Address Translation
NBFM	Narrowband FM
NPT	Network Port Translation
NS	Neighborhood Service
OLSR	Optimized Link State Routing
PCM	Pulse Code Modulation
PDA	Personal Digital Assistant
PSCR	Public Safety Cognitive Radio
PSD	Power Spectral Density
PSN	Public Safety Network
PSS	Public Safety Service
PSZ	Public Safety Zone
PTT	Push-To-Talk
RAS	Remote Agent and Service Discovery Block
RCache	Received Cache
SD/P	Service Discovery/Protocol
SDR	Software Defined Radio
SPSZ	Single-PSZ Service
SSM	Service Sub Message
TC	Topology Control message
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
USB	Universal Serial Bus
USRP	Universal Software Radio Peripheral
VPSCR	Vehicular Public Safety Cognitive Radio
WM	Windows Mobile
WPA	Wi-Fi Protected Access
WSGA	Wireless System Genetic Algorithm
XML	Extended Markup Language

Bibliography

[1] J. Mitola, "The Software Radio," *IEEE National Telesystems Conference*, 1992.

[2] "History of SDR," http://www.ecnmag.com/Software-Defined-Radios-Adapt-to-Change.aspx.

[3] J. Mitola, III and G. Q. Maguire, Jr., "Cognitive radio: making software radios more personal," *Personal Communications, IEEE*, vol. 6, pp. 13-18, 1999.

[4] G. Feng, C. Qinqin, W. Ying, C. W. Bostian, T. W. Rondeau, and L. Bin, "Cognitive Radio: From Spectrum Sharing to Adaptive Learning and Reconfiguration," in *Aerospace Conference, 2008 IEEE*, 2008, pp. 1-10.

[5] R. V. Prasad and P. Pawelczak, "Cognitive Functionality in Next Generation Wireless Networks: Standardization Efforts," in *IEEE Communications Magazine*. vol. 46, 2008.

[6] S. Haykin, "Cognitive Radio: Brain-Empowered Wireless Communications," *IEEE Journal on Selected Areas in Communications*, vol. 23, pp. 201-220, Feb 2005.

[7] T. W. Rondeau, "Application of Artificial Intelligence to Wireless Communications," in *Electrical and Computer Engineering*. vol. Doctor of Philosophy: Virginia Polytechnic Institute and State University, 2007, p. 198.

[8] N. Jesuale and B. C. Eydt, "A Policy Proposal to Enable Cognitive Radio for Public Safety and Industry in the Land Mobile Radio Bands," in *New Frontiers in Dynamic Spectrum Access Networks, 2007. DySPAN 2007. 2nd IEEE International Symposium on*, 2007, pp. 66-77.

[9] J. Donovan, "Cognitive Radios solve a host of problems," *Portable Design*, pp. 8-11, Jan 2007.

[10] "Summary of "Lessons Learned" from Events of September 11 and Implications for Business Continuity," http://www.sec.gov/divisions/marketreg/lessonslearned.htm, 2002.

[11] L. E. Miller, "Wireless Technologies and the SAFECOM SoR for Public Safety Communications ", 2005, http://www.antd.nist.gov/wctg/manet/docs/WirelessAndSoR060206.pdf.

[12] B. Le, "Building a Cognitive Radio - From Architecture Definition to Prototype Implementation, in Electrical and Computer Engineering," in *Electrical and*

Computer Engineering. vol. Doctor of Philosophy: Virginia Polytechnic Institute and State University, 2007.

[13] D. A. Scaperoth, "Configurable SDR Operation for Cognitive Radio Applications using GNU Radio and the Universal Software Radio Peripheral," in *Electrical and Computer Engineering*. vol. Master of Science: Virginia Polytechnic Institute and State University, 2007.

[14] "Ettus Research LLC.," http://www.ettus.com.

[15] "GNU Radio Homepage," http://www.gnuradio.org.

[16] B. Le, P. Garcia, Q. Chen, B. Li, F.Ge, M. ElNainay, T. W. Rondeau, and C. W. Bostian, "Public Safety Cognitive Radio Node System," in *Software Defined Radio Technical Conference*, Denver, Colorado, 2007.

[17] T. W. Rondeau, B. Le, C. J. Rieser, and C. W. Bostian, "Cognitive Radios with Generic Algorithms: Intelligent Control of Software Defined Radios," in *Software Defined Radio Forum Technical Conference*, Phoenix, Arizona, 2004.

[18] "Project25 (P25)," http://www.project25.org/.

[19] S. Cass, "Hardware for your software radio," in *IEEE Spectrum*, 2007.

[20] "EF Johnson 5100 radios," http://efjohnson.com/products/5100.asp.

[21] "Homeland Security and Technology Division," in *Strategies for States to Achieve Public Safety Wireless Interoperability* http://www.nga.org/cda/files/0903INTEROP.pdf.

[22] E. Reyes, "Basic public safety communications and interoperability," http://www.policeone.com/police-products/communications/articles/127284-Basic-public-safety-communications-and-interoperability/.

[23] F. Ge, Q. Chen, Y. Wang, T. W. Rondeau, B. Le, and C. W. Bostian, "Cognitive Radio: From Spectrum Sharing to Adaptive Learning and Reconfiguration," in *2008 IEEE Aerospace Conference*, Big Sky Montana, MT, 2008.

[24] J. D. Crockett, "DSP Architectures for Wireless Communications," in *International Symposium on. Advanced Radio Technologies*, 1998.

[25] Ultralife, "Battery Technology and Moore's Law," http://www.ultralifecorp.com/battery_trends.php?ID=6.

[26] R. Godwin-Jones, "Emerging Technologies Mobile-Computing Trends: Lighter, Faster, Smarter," *Language Learning & Technology*, vol. 12, pp. 3-9, 2008.

[27] F. Koushanfar, A. Prabhu, M. Potkonjak, and J. M. Rabaey, "Processors for Mobile Applications " in *International Conference on Computer Design*, 2000.

[28] ARM, "Cortex-A8 specifications," http://www.arm.com/products/CPUs/ARM_Cortex-A8.html.

[29] J. Brito, "Sending Out an S.O.S.: Public Safety Communications Interoperability as a Collective Action Problem," *FEDERAL COMMUNICATIONS LAW JOURNAL*, vol. 59, pp. 457-492, 2007.

[30] B. A. Myers, R. C. Miller, B. Bostwick, and C. Evankovich, "Extending the windows desktop interface with connected handheld computers," in *Proceedings of the 4th conference on USENIX Windows Systems Symposium - Volume 4* Seattle, Washington: USENIX Association, 2000.

[31] "Network Address Translation RFC," http://www.ietf.org/rfc/rfc4787.txt.

[32] M. Sweetgall, "G711 implementaion in C# with source," http://www.codeproject.com/KB/security/g711audio.aspx.

[33] "TELNET RFC," http://tools.ietf.org/html/rfc854.

[34] "Microsoft Waveform Audio," http://msdn.microsoft.com/en-us/library/aa910179.aspx.

[35] "Microsoft Platform/Invoke Library Samples," http://www.microsoft.com/downloads/details.aspx?FamilyId=B1F5CCAA-ADA2-42D4-8B70-95DC7D8F678C&displaylang=en.

[36] "Jini," http://www.jini.org/wiki/Main_Page.

[37] "UPnP Device Architecture v. 1.0.1 December 2003. ," http://wwvw.upnp.org/resourcesJdocuments.asp.

[38] "White Paper: Salutation Architecture Overview," in *Salutation Consortium* http://www.salutation.org/whitepaper/originalwp.pdf, 1998.

[39] E. Guttman, "Service Location Protocol: Automatic Discovery of IP Network Services," *IEEE Internet Computing,* July 1999.

[40] E. Guttman, C. E. Perkins, and M. Day, "Service Location Protocol, Version 2," in *Internet RFC 2608,* 1999.

[41] S. Helal, N. Desai, V. Verma, and L. Choonhwa, "Konark - a service discovery and delivery protocol for ad-hoc networks," in *Wireless Communications and Networking, 2003. WCNC 2003. 2003 IEEE*, 2003, pp. 2107-2113 vol.3.

[42] "Specification of the Bluetooth System Core, Version 1.0B volume 1,1999. Part E," Bluetooth SIG, http://www.bluetooth.com/link/spec/bluetooth_e.pdf.

[43] F. Sailhan and V. Issarny, "Scalable Service Discovery for MANET," in *Pervasive Computing and Communications, 2005. PerCom 2005. Third IEEE International Conference on*, 2005, pp. 235-244.

[44] Z. Feng, M. Mutka, and L. Ni, "Splendor: A secure, private, and location-aware service discovery protocol supporting mobile services," in *Pervasive Computing and Communications, 2003. (PerCom 2003). Proceedings of the First IEEE International Conference on*, 2003, pp. 235-242.

[45] R. S. Marin-Perianu, P. H. Hartel, and J. Scholten, "A Classification of Service Discovery Protocols," University of Twente, Enschede, Technical Report 1381-3625, June 2005.

[46] C. N. Ververidis and G. C. Polyzos, "Service discovery for mobile Ad Hoc networks: a survey of issues and techniques," *Communications Surveys & Tutorials, IEEE*, vol. 10, pp. 30-45, 2008.

[47] R. Hermann, D. Husemann, M. Moser, M. Nidd, C. Rohner, and A. Schade, "DEAPspace: transient ad hoc networking of prevasive devices," *Comput. Netw.*, vol. 35, pp. 411-428, 2001.

[48] R. Bellman, "On a Routing Problem," *Quarterly Applied Mathematics*, vol. 16, pp. 87-90, 1958.

[49] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numerische Mathematik*, vol. 1, pp. 269-271, 1959.

[50] "Optimized Link State Routing RFC3626," www.ietf.org/rfc/rfc3626.txt.

[51] V. Atanasovski and L. Gavrilovska, "Efficient service discovery schemes in wireless ad hoc networks implementing cross-layer system design," in *Information Technology Interfaces, 2005. 27th International Conference on*, 2005, pp. 496-501.

[52] M. Storey, G. Blair, and A. Friday, "MARE: resource discovery and configuration in ad hoc networks," *Mobile Networks and Applications*, vol. 7, pp. 377-387, 2002.

[53] N. Klimin, W. Enkelmann, H. Karl, A. Wolisz, and D. Ag, "A Hybrid Approach for Location-based Service Discovery in Vehicular Ad Hoc Networks," in *Proc. of 1st Intl. Workshop on Intelligent Transportation (WIT)*, 2004.

[54] A. Boukerche and K. Abrougui, "Performance evaluation of a hybrid adaptive service discovery protocol for next generation Heterogeneous Vehicular Networks (HVNs)," in *Proceedings of the 3nd ACM workshop on Performance monitoring and measurement of heterogeneous wireless and wired networks* Vancouver, British Columbia, Canada: ACM, 2008.

[55] J. Hoebeke, T. V. Leeuwen, S. Eeckhout, W. Puype, I. Moerman, BartDhoedt, and P. Demeester, "Comparison of Proactive and Reactive Resource and Service Discovery in Mobile Ad Hoc Networks," in *11th Symposium on Communications and Vehicular Technology in the Benelux (SCVT 2004)*, 2004.

[56] B. Raman, P. Bhagwat, and S. Seshan, "Arguments for Cross-Layer Optimizations in Bluetooth Scatternets," in *Proceedings of the 2001 Symposium on Applications and the Internet (SAINT 2001)*: IEEE Computer Society, 2001.

[57] R. Koodli and C. E. Perkins, "Service discovery in on-demand ad hoc networks," Internet draft, 2002.

[58] P. E. Engelstad and Y. Zheng, "Evaluation of Service Discovery Architectures for Mobile Ad Hoc Networks," in *Proceedings of the Second Annual Conference on Wireless On-demand Network Systems and Services*: IEEE Computer Society, 2005.

[59] D. Arias-Torres and J. A. García-Macías, "Performance Analysis of Two Approaches to Service Discovery in Mobile Ad Hoc Networks," in *Advanced Distributed Systems*, 2005, pp. 368-376.

[60] J. L. Jodra, M. Vara, J. M. Cabero, and J. Bagazgoitia, "Service discovery mechanism over OLSR for mobile ad-hoc networks," in *Advanced Information Networking and Applications, 2006. AINA 2006. 20th International Conference on*, 2006, p. 6 pp.

[61] J. Flathagen and K. K. Øvsthus, "Service Discovery using OLSR and Bloom Filters," in *The 4th OLSR Interop Workshop*, Ottawa, Canada, 2008.

[62] M. Krebs, K. H. Krempels, and M. Kucay, "Service Discovery in Wireless Mesh Networks," in *Wireless Communications and Networking Conference, 2008. WCNC 2008. IEEE*, 2008, pp. 3093-3098.

[63] D. Chakraborty, A. Joshi, and Y. Yesha, "Integrating service discovery with routing and session management for ad-hoc networks," 2004.

[64] A. Varshavsky, B. Reid, and E. d. Lara, "The need for cross-layer service discovery in MANETs," Report CSRG-492, 2004.

[65] T. Leclerc, L. Ciarletta, A. Andronache, and S. Rothkugel, "OLSR and WCPD as Basis for Service Discovery in MANETs," in *Proceedings of the 2008 The Second International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies*, 2008.

[66] L. Li and L. Lamont, "Service Discovery for Support of Real-time Multimedia SIP Applications Over OLSR MANETs," in *OLSR Interop & Workshop 2004*, San Diego, USA, 2004.

[67] M. A. E. Saoud, T. Kunz, and S. Mahmoud, "SLPManet: service location protocol for MANET," in *Proceedings of the 2006 international conference on Wireless communications and mobile computing* Vancouver, British Columbia, Canada: ACM, 2006.

[68] C. Campo, C. García-Rubio, A. M. López, and F. Almenárez, "PDP: A lightweight discovery protocol for local-scope interactions in wireless ad hoc networks," *Computer Networks,* vol. 50, pp. 3264-3283, 2006.

[69] B. H. Bloom, "Space/time trade-offs in hash coding with allowable errors," *Commun. ACM,* vol. 13, pp. 422-426, 1970.

[70] "Mercury project for OLSR," http://sourceforge.net/projects/olsr-mercury/.

[71] "OLSRd implementation," http://www.olsr.org/?q=download.

[72] K. Mehrjoo, "An Extensible Information Dissemination Scheme over the Optimized Link State Routing Protocol for Mobile Ad Hoc Networks," in *Electrical and Computer Engineering*. vol. Master of Science: Virginia Polytechnic and State University, 2007.

[73] "OMNeT++ Homepage," http://www.omnetpp.org/.

[74] "INET Framework for OMNeT++," http://inet.omnetpp.org/.

[75] "UM-OLSR Homepage," http://masimum.inf.um.es/?Software:UM-OLSR.