A Software-Defined Radio Receiver for APCO Project 25 Signals

S. Glass
Queensland Research Lab
NICTA
Brisbane, Australia
stephen.glass@nicta.com

V. Muthukkumarasamy
School of ICT
Griffith University
Gold Coast, Australia
v.muthu@griffith.edu.au

M. Portmann
School of ITEE
University of Queensland
Brisbane, Australia
m.portmann@itee.uq.edu.au

ABSTRACT

APCO Project 25 (P25) is the digital communications standard that has widespread deployment amongst emergency first-responders in several different countries. This paper describes the implementation of a low-cost software-defined radio receiver for APCO Project 25 signals. The OP25 Receiver has been developed as part of an investigation into the security of the P25 protocol suite and provides low-level access to the actual message traffic using the WireShark packet sniffer. The proposed OP25 Receiver is a useful diagnostic and security analysis tool. Our initial experience suggests that the flexibility of the software-defined radio approach is well-suited to meeting the varying needs of public-safety radio communications.

General Terms

Software-Defined Radio (SDR)

1. INTRODUCTION

In this paper we discuss a Software-Defined Radio (SDR) receiver which we are using to analyze traffic from public-safety communications using the the widely deployed APCO Project 25 (P25) standard. P25-based systems are used by first-responder emergency services across the US, Canada and Australia. Compared to the analog radio systems that preceded P25 the digital system is claimed to offer improved radio spectrum use, increased geographical coverage, centralized channel management (trunking) and the provision for both voice and data traffic.

The motivation for constructing an SDR receiver for P25 traffic is to meet the needs of an investigation into the security of public-safety radio communications. This investigation requires that the message traffic be captured and analyzed in detail. Using an SDR approach allows for the flexibility of complete low-level access to the message traffic without the expense associated with specialized P25 protocol analysis devices.

1.1 APCO Project 25

A P25 radio system consists of both fixed and mobile equipment. Fixed stations fulfil the roles of base station, trunking controller and repeater. A fixed station may provide data services and gateways to the public switched telephone network, private automatic branch exchanges and to other radio systems. Mobile radios may be either hand-held or vehicle-mounted and paired with Mobile Data Terminals (MDTs) for accessing data services. Whether fixed or mobile all P25 radios can operate in either analog frequency-modulation (FM) or digital modes. The digital mode can use an operator-chosen cryptographic cipher to protect message confidentiality. The P25 standard does not mandate the provision of encryption capabilities and so this feature is available on some, but not all, P25-compliant equipment.

The P25 standards are jointly administered by the Telecommunications Industry Association (TIA) and the American National Standards Institute (ANSI). To ensure the interoperability of P25 equipment the P25 standards defines a Common Air Interface (CAI) [2]. This is the core specification document and defines the modulation techniques, the frame types, their meanings and the physical layer representations that must be implemented by all P25-compliant radios.

P25 systems encode all voice traffic using the IMBE vocoder. This is a multi-band excitation vocoder which delivers relatively high quality speech from a low-bandwidth channel [5, 6]. Unfortunately, the standard makes no provision for the use of any alternative vocoder technologies which can be a problem when IMBE is used in noisy environments and performance is degraded [8]. Possibly the least satisfactory aspect of the IMBE vocoder is that it is a proprietary and patent-encumbered technology. The use of IMBE for voice coding may, therefore, require licensing fees be paid to the patent holder. The P25 standard requires, however, that such licenses are available on non-discriminatory terms.

1.2 Benefits of SDR approaches

Communications equipment making use of SDR approaches have a considerable edge in flexibility when compared to traditional radio design techniques. The increased flexibility must be weighed against additional computation, increased power usage and increased latencies. In many situations such costs can be offset by the additional utility afforded by an SDR platform.

In a disaster, infrastructure may be damaged requiring cooperation between various emergency agencies and civilian volunteers such as the Amateur Radio Emergency Service (ARES) and the Military Affiliate Radio System (MARS). Hurricane Katrina, for example, saw amateur operators assume the role of 911 dispatchers in Hancock County, Mississippi following the collapse of the conventional communications infrastructure. P25 explicitly allows for this because it supports operation using the analog FM mode used by civilian volunteers. Using an SDR approach extends this capability to enable a single station to simultaneously process many different types of analog and digital signals.

The P25 standard is changing in response to legislative and technical changes. The increasing competition for bandwidth has resulted in new FCC requirements. The existing Phase I transmissions use a 12.5kHz channel and 4FSK modulation and is due to be superceded. The Phase II transmissions use a 6.25kHz channel and $\frac{\pi}{4}$ DQPSK modulation. An SDR approach can quickly adapt to such changes because processing is partitioned into blocks with well-defined interfaces. The demodulation stage can be easily replaced to allow for reception of different modulation schemes whilst sharing the common code for packet assembly and decoding.

Vendors often implement proprietary extensions to the P25 standard into their equipment. This can inhibit interoperability and lock emergency responders into using equipment from a single manufacturer. The technical obstacles to handling proprietary extensions are relatively minor since it involves nothing more than modifying the SDR programs. The commercial and legal obstacles may prove more substantial.

Experience of disasters shows how effective communications can make the difference between life and death. The tragedy of Hurricane Katrina was compounded by major failures of the emergency communications infrastructure — resulting in the loss of human lives that might otherwise have been prevented [3]. This experience highlights the need for emergency first-responder organisations to have robust, effective and interoperable communication capabilities. The US has launched the SAFECOM program which defines the requirements for public-safety voice, data and video communications [12]. The latter requirement for video communication imposes much higher bandwidth requirements than can be met by existing P25 communication systems. Cognitive radio approaches can satisfy the requirement for increased bandwidth by opportunistic use of bands that are underutilized by their primary users [7]. Cognitive radio uses SDR technology to rapidly reconfigure the physical layer transmissions. SDRs approaches can, therefore, ensure interoperability and backward-compatibility with existing equipment and enable cognitive radio techniques to provide video and other high-bandwidth services.

An area where SDR has significant potential benefit is in the use of free and open-source software. Standardized SDR hardware and free software SDR frameworks allow for the rapid development of new and improved services which can be distributed under free and open source licenses. An alliance of manufacturers and other interested parties has initiated a free software project which aims to create a reference implementation for a P25 trunking controller[10].

2. IMPLEMENTATION

In this section we describe the hardware and software components that comprise the SDR implementation of a P25 receiver. This program is known as OP25[1].

2.1 Hardware

The equipment used for this investigation is a Universal Software Radio Peripheral (USRP) as shown in figure 2. This is a low-cost (≈ 750 USD) SDR designed to work with the GNU Radio framework. The USRP itself is responsible for sampling the input signals and the samples are sent to a GNU/Linux laptop computer for processing by the OP25 Receiver program. Daughter boards provide for frequency translation, amplification and filtering to enable receive and transmit access to the VHF and UHF bands used for public-safety communications.

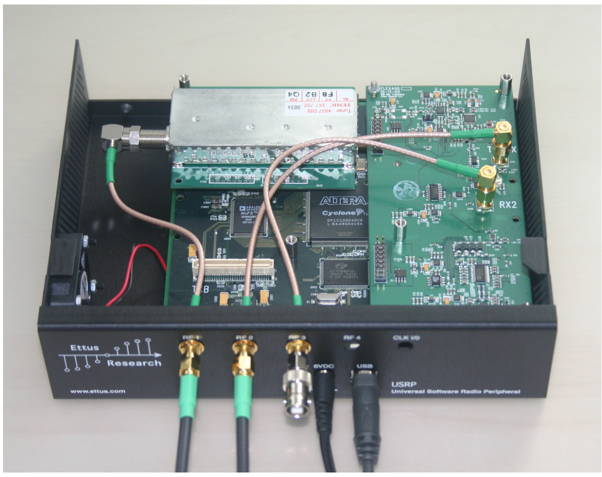


Figure 2: The USRP Software-Defined Radio with 80–870MHz VHF/UHF receiver (top left) and 400–500MHz UHF transceiver (right) daughter boards.

2.2 GNU Radio

The OP25 Receiver has been developed as a free software project and is based on the successful GNU Radio and WireShark free software packages. The OP25 Receiver program is built using the GNU Radio framework which allows SDR programs to be written in C++ and Python. GNU Radio provides a large collection of signal-processing blocks which transform their input signal(s) into their output signal(s) in some well-defined way[4]. A software radio connects such signal processing blocks together to perform the necessary signal processing. Using the GNU Radio framework allows for rapid prototyping of SDR programs and ensures hardware independence because the framework can use other hardware than the USRP. GNU Radio itself is free software for which the source code is freely available and which the user is entitled to modify and re-distribute. If a radio needs a signal-processing block that isn't present then it can be written (often using an existing block as a starting point) and added to the framework.

[1] The complete source code for the OP25 Receiver (including the necessary patches for WireShark) can be obtained from the principal author.

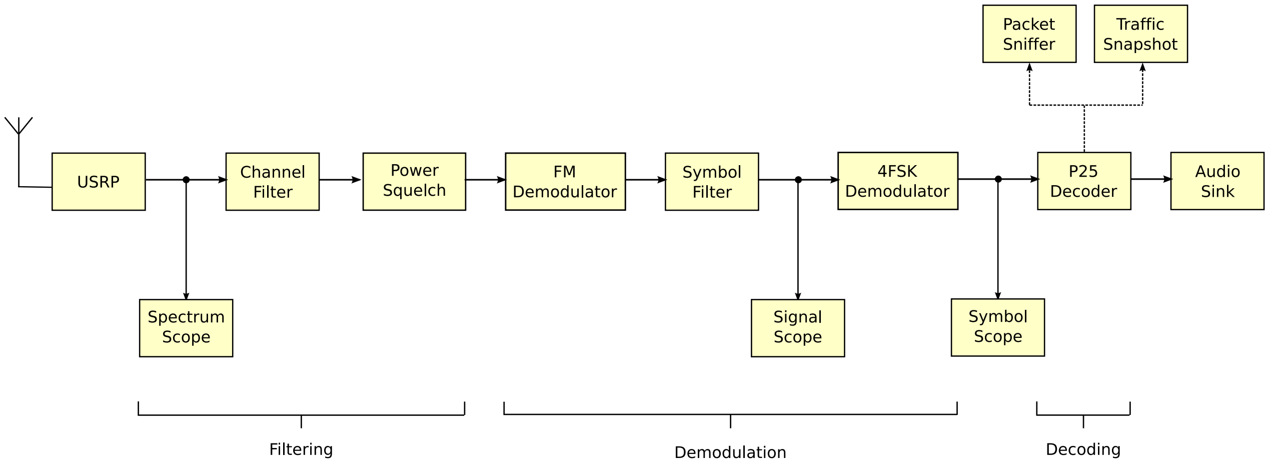


Figure 1: Block diagram for OP25 Receiver

2.3 P25 Receiver

The OP25 Receiver is a Python program. This program connects together the signal-processing blocks responsible for sampling the relevant part of the radio spectrum, extracting the signal of interest and decoding it for the user. The receiver produces digital audio as its output and sends the decoded P25 frames to the WireShark network protocol analyser where they can be analysed in detail. A block diagram for the OP25 Receiver is shown in figure 1 showing how the signal-processing blocks are connected together. The OP25 Receiver can be thought of as comprising three main stages: filtering, demodulation and decoding.

2.3.1 Filtering

P25 signals occupy channels that are each 12.5kHz wide. The USRP can process approximately 6MHz of the radio spectrum at one time, allowing hundreds of P25 signals to be received simultaneously. In the OP25 Receiver only one channel is selected by the filtering stage for subsequent processing. It is possible to process several in parallel limited only by the available processor resources.

2.3.2 Demodulation

The demodulation stage transforms the baseband signal into a stream of symbols. P25 makes use of a modulation scheme known as C4FM (continuous 4-level FM) in which four different frequencies are used to encode a two bit symbol at a rate of 4800 symbols/s. To demodulate the baseband signal the OP25 Receiver first uses a narrow-band FM demodulator to recover the baseband signal and then a 4-level Frequency Shift Keying (4FSK) demodulator to translate that signal into a stream of two bit symbols. The 4FSK demodulator is not provided by the GNU Radio library. Instead a non-standard block was made available by a GNU Radio user which meets this requirement.

2.3.3 Decoding

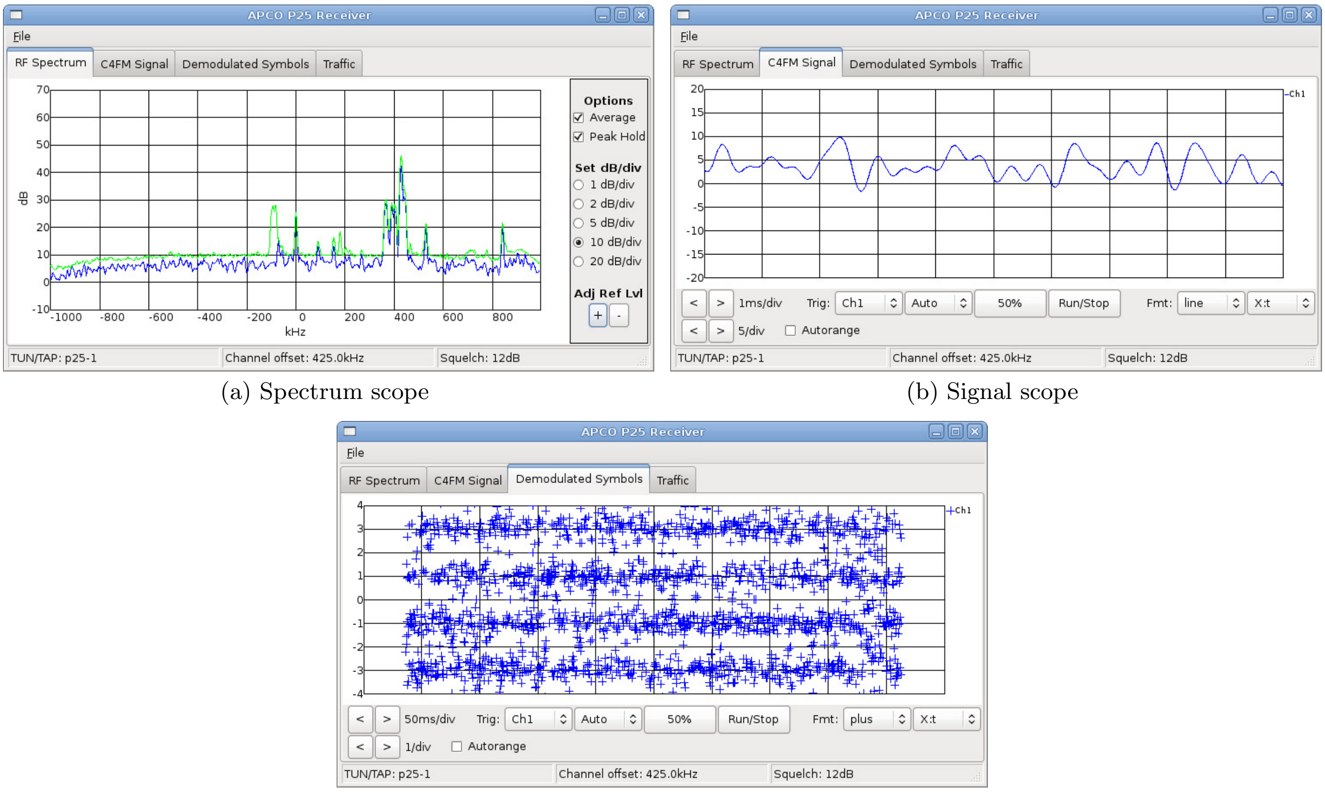
The core of the OP25 Receiver is implemented using a custom-built C++ signal-processing block. This custom-built C++ signal-processing block reconstructs the frames from the symbol stream and comprises of:

- Correlation — each frame is identified by a fixed Framing Sequence (FS) which, when detected, causes subsequent symbols to be aggregated into a frame.
- Error correction — To protect frame contents from interference P25 uses the BCH, Golay, Hamming and Reed-Solomon forward error correction codes (in full and shortened forms). These codes need to be applied as appropriate to the frame type to compensate for errors in reception.
- De-interleaving — To protect against fading the symbols are interleaved throughout the frame to obtain maximum benefit from the forward error correction. The frame body is recovered by de-interleaving the symbols.
- Voice decoding — voice frames in P25 are encoded using the IMBE vocoder and so the voice signal needs to be reconstructed from its encoded form.

Once a complete frame has been received the error-corrected and de-interleaved frame contents are encapsulated in an Ethernet frame and sent to the WireShark network protocol analyser using the TUN/TAP device. This is a virtual network interface that enables other programs to receive and process traffic decoded by the OP25 Receiver. Voice frames are subject to an extra step in which the compressed voice codewords are extracted for post-processing by the IMBE voice decoder.

2.4 P25 Receiver User Interface

The graphical user interface for the OP25 Receiver program is shown in figure 3. Four panels display the signal at various stages of processing and are used for both control and diagnostic purposes. The “spectrum scope” shown in figure 3(a) displays a dynamic visualisation of the frequency domain. This display is the receiver’s principal control and allows signals of interest to be identified and the receiver tuned approprirately. The user can click at any point within the frequency/power graph to set the frequency and RF squelch threshold.



☑ Average
☑ Peak Hold
○ 1 dB/div
○ 2 dB/div
○ 5 dB/div
● 10 dB/div
○ 20 dB/div

TUN/TAP: p25-1 | Channel offset: 425.0kHz | Squelch: 12dB

(a) Spectrum scope

1ms/div | Trig: Ch1 | Auto | 50% | Fmt: line | X:t
5/div | □ Autorange

TUN/TAP: p25-1 | Channel offset: 425.0kHz | Squelch: 12dB

(b) Signal scope

50ms/div | Trig: Ch1 | Auto | 50% | Fmt: plus | X:t
1/div | □ Autorange

TUN/TAP: p25-1 | Channel offset: 425.0kHz | Squelch: 12dB

(c) Symbol scope

Figure 3: The OP25 Receiver graphical controls

The remaining displays provide diagnostic information. The "signal scope" panel is shown in figure 3(b) and is used in conjunction with the "symbol scope" of figure 3(c) to assess the quality of the received signal to visualize the time domain for the selected channel. The latter plots the distribution of symbols which should form four clearly-separated streams for a strong input signal.

2.5 WireShark Network Protocol Analyser

The WireShark network protocol analyser is used to recognize, filter and dissect P25 network traffic. WireShark is a free software network protocol analyser which is in widespread use and provides comprehensive facilities for inspecting and analysing network traffic. The standard WireShark distribution does not support the analysis of P25 message traffic but, because it is free software, we have been able to extend it to meet this requirement.

Figure 4 shows the modified version of WireShark being used to inspect P25 message traffic. The top pane shows a stream of frames from which a voice frame has been selected. The middle portion of the window shows the logical contents of the frame and allows the user to traverse its content. The bottom part of the display shows the physical representation of octets that correspond to the part of the frame of interest.

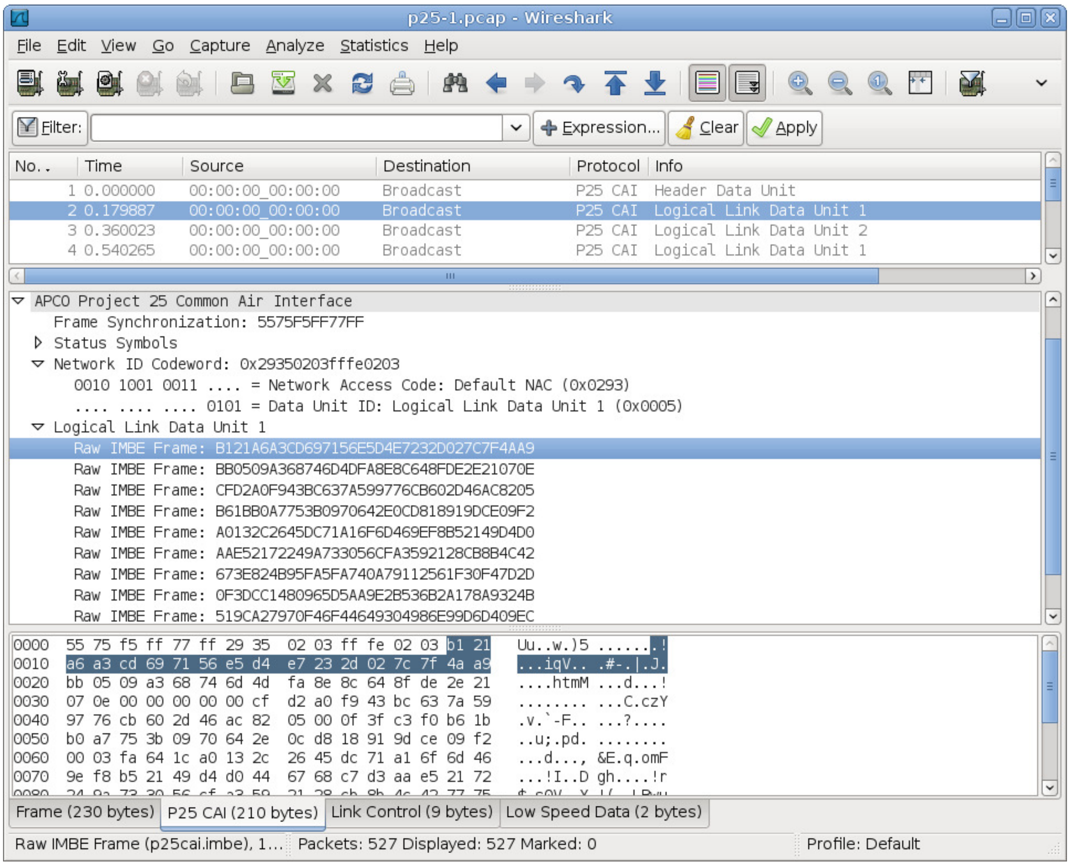
The P25 specifications mandate only the physical layer representation of the frame. They do not specify how such frames should be represented after de-interleaving and error correction. To retain as much information as possible frames are passed from the receiver to WireShark using the physical frame representation. This allows for network traces to be exchanged with other tools which make use of the P25 physical frame layout, but each such tool must be able to de-interleave the received frames.

3. DISCUSSION

This section considers the issues arising from the implementation of the OP25 Receiver and examines the potential impact of SDR technologies in public-safety communications.

3.1 IMBE Decoding

The use of the patent-encumbered IMBE vocoder was discussed in the introduction. To decode IMBE the most common approach is to make use of hardware devices or "dongles" which embody licensed copies of the vocoder algorithms. The OP25 Receiver can use the DVSI VC55-PR hardware dongle. This accepts compressed voice bits via an RS232 serial line and produces an analog audio output directly. This decompressed audio is not made available to the OP25 Receiver and so cannot be processed by subsequent blocks. An alternative software implementation of the IMBE decoder can also be used. This is a non-optimized implementation of the IMBE decoder as described by the vocoder specification [1]. At present, this is a stand-alone C++ program which is not integrated with the OP25 Receiver. The OP25 receiver writes the compressed voice codewords to file for post-processing by the software IMBE decoder.



```
No. .  Time        Source               Destination        Protocol   Info
    1 0.000000   00:00:00_00:00:00    Broadcast          P25 CAI    Header Data Unit
    2 0.179887   00:00:00_00:00:00    Broadcast          P25 CAI    Logical Link Data Unit 1
    3 0.360023   00:00:00_00:00:00    Broadcast          P25 CAI    Logical Link Data Unit 2
    4 0.540265   00:00:00_00:00:00    Broadcast          P25 CAI    Logical Link Data Unit 1
```

```
▽ APCO Project 25 Common Air Interface
    Frame Synchronization: 5575F5FF77FF
  ▷ Status Symbols
  ▽ Network ID Codeword: 0x29350203fffe0203
      0010 1001 0011 .... = Network Access Code: Default NAC (0x0293)
      .... .... .... 0101 = Data Unit ID: Logical Link Data Unit 1 (0x0005)
  ▽ Logical Link Data Unit 1
      Raw IMBE Frame: B121A6A3CD697156E5D4E7232D027C7F4AA9
      Raw IMBE Frame: BB0509A368746D4DFA8E8C648FDE2E21070E
      Raw IMBE Frame: CFD2A0F943BC637A599776CB602D46AC8205
      Raw IMBE Frame: B61BB0A7753B0970642E0CD818919DCE09F2
      Raw IMBE Frame: A0132C2645DC71A16F6D469EF8B52149D4D0
      Raw IMBE Frame: AAE52172249A733056CFA3592128CB8B4C42
      Raw IMBE Frame: 673E824B95FA5FA740A79112561F30F47D2D
      Raw IMBE Frame: 0F3DCC1480965D5AA9E2B536B2A178A9324B
      Raw IMBE Frame: 519CA27970F46F44649304986E99D6D409EC
```

```
0000  55 75 f5 ff 77 ff 29 35  02 03 ff fe 02 03 b1 21   Uu..w.)5 .......!
0010  a6 a3 cd 69 71 56 e5 d4  e7 23 2d 02 7c 7f 4a a9   ...iqV.. .#-.|.J.
0020  bb 05 09 a3 68 74 6d 4d  fa 8e 8c 64 8f de 2e 21   ....htmM ...d...!
0030  07 0e 00 00 00 00 00 cf  d2 a0 f9 43 bc 63 7a 59   ........ ...C.czY
0040  97 76 cb 60 2d 46 ac 82  05 00 0f 3f c3 f0 b6 1b   .v.`-F.. ...?....
0050  b0 a7 75 3b 09 70 64 2e  0c d8 18 91 9d ce 09 f2   ..u;.pd. ........
0060  00 03 fa 64 1c a0 13 2c  26 45 dc 71 a1 6f 6d 46   ...d..., &E.q.omF
0070  9e f8 b5 21 49 d4 d0 44  67 68 c7 d3 aa e5 21 72   ...!I..D gh....!r
```

Figure 4: Wireshark packet sniffer being used to inspect P25 traffic

On a modern dual-core CPU the performance of this vocoder is sufficiently fast that real-time speech decoding is possible (for example, decoding an 8 second speech fragment takes approximately 60 milliseconds of elapsed time).

3.2 Equipment Issues

Computing equipment is increasingly common in emergency service vehicles but requires careful interface design and a focus on safety. One example using P25 communications together with sophisticated computing capabilities is Project 54 at the University of New Hampshire [9]. This uses a streamlined interface to provide access to a number of different services in police patrol cars. Safety is the primary concern but there are physical and cost issues that must also be considered.

3.2.1 Power, size and processing constraints

The use of a computer for processing and the power requirements of the radio itself imply that the type of software radio described here is not yet a suitable replacement for hand-held mobile radios. For vehicle-mounted mobile radios and for fixed stations these are much less significant issues. The processing requirements of an SDR program are quite substantial — hundreds of thousands of samples must be processed every second. Modern computers can easily cope with this processing load but older equipment often does not have the necessary CPU resources and I/O bandwidth.

3.2.2 Cost

P25 equipment can prove quite costly. Several manufacturers offer radios with two otherwise identical models, one of which supports P25 and the other analog-only operation. The difference in price between the radios is usually substantial; one P25 radio is often as expensive as two analog-only radios. In contrast, the expense of a software radio is comparable with the cost of a vehicle-mounted P25 mobile radio. The SDR approach is considerably more flexible than a hardware radio but requires a suitable computer to perform the necessary signal processing.

The major cost in a P25 system is often incurred in expensive fixed station equipment. P25 systems allow for voice and data but for small operators the costs of adding data support to the fixed stations can prove to be prohibitive. Ramsey *et al.* have implemented a simple P25 data transceiver using a computer's sound card as a 4FSK modem [11]. This system is known as Project 54. The computer is interfaced with a conventional analog FM transceiver. This approach allows for data operation to be provided at a small fraction of the capital costs that would otherwise be required. The OP25 Receiver enjoys the same low cost advantages as Project 54 but can operate on multiple channels and integrate voice and data operation into a single unit. Project 54 enjoys the ability to transmit at high power that makes use of radio hardware purpose-designed for public-safety applications.

3.3 Further Work

The OP25 Receiver is useful as a diagnostic tool but does not provide support for either trunking or for decryption of secure traffic. The use of the WireShark network protocol analyser to inspect the trunking control channel is possible but full support for trunking is not planned at present. Support for decryption, integrated software IMBE decoding and $\frac{\pi}{4}$ DQPSK modulation are planned for a future revision of the OP25 Receiver.

We have already indicated that transmission capabilities are of interest in our investigation and for practical public-safety communication devices transmission capability is essential. From the security analysis perspective will allow for the investigation of active attacks. This can be accomplished using the GNU Radio framework we have discussed here where the transmit process is the inverse of the receiver process and in which special care is taken to ensure interference is minimized.

4. CONCLUSIONS

This project demonstrates the use of SDR for receiving public-safety communications signals. The inherent flexibility of SDR enables interoperability with existing analog and existing digital systems and facilitates the transition to next-generation public-safety communications technologies. SDR platforms in public-safety communications can enable cognitive radio approaches which can meet the emerging requirements for high bandwidth and robust communications.

5. ACKNOWLEDGMENTS

The authors would like to acknowledge the support of both NICTA and the (ISC)^2 consortium. NICTA is funded by the Australian Government as represented by the Department of Broadband, Communications and the Digital Economy and the Australian Research Council through the ICT Centre of Excellence program; and the Queensland Government.

6. REFERENCES

[1] *Project 25 Vocoder Description.* Number ANSI/TIA/EIA-102.BABA-1998. Telecommunications Industry Association, 2500 Wilson Boulevard, Arlington, VA 22201, USA, May 1998.

[2] *Project 25 FDMA Common Air Interface Description.* Number TIA-102.BAAA-A. Telecommunications Industry Association, 2500 Wilson Boulevard, Arlington, VA 22201, USA, January 2003.

[3] *A Failure Of Initiative: Final Report of the Select Bipartisan Committee to Investigate the Preparation for and Response to Hurricane Katrina.* US Government Printing Office, Washington, DC 20402-0001, USA, February 2006. Available online at `http://www.gpoaccess.gov/katrinareport/mainreport.pdf`.

[4] GNU Radio. Project website. `http://www.gnuradio.org`.

[5] D. Griffin and J. Lim. Multiband excitation vocoder. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 36(8):1223–1235, August 1988.

[6] J. C. Hardwick and J. S. Lim. The application of the IMBE speech coder to mobile communications. In *International Conference on Acoustics, Speech, and Signal Processing (ICASSP-91)*, volume 1, pages 249–252, April 1991.

[7] S. Haykin. Cognitive radio: brain-empowered wireless communications. *IEEE Journal on Selected Areas in Communications*, 23(2):201–220, February 2005.

[8] Z. Jinjie and C. Zhigang. Investigation of IMBE parameters' sensitivity to noise. In *IEEE Global Telecommunications Conference (GLOBECOM 98)*, volume 6, pages 3734–3739, 1998.

[9] A. L. Kun, W. T. Miller III, and W. H. Lenharth. Computers in police cruisers. *IEEE Pervasive Computing*, 3(4):34–41, October–December 2004.

[10] OpenP25. Project website. `http://www.openp25.org`.

[11] E. R. Ramsey, W. T. Miller III, and A. L. Kun. A software-based implementation of an APCO Project 25 compliant packet data transmitter. In *2008 IEEE International Conference on Technologies for Homeland Security*, Boston, MA, 12–13 May 2008. Institution of Electrical and Electronics Engineers.

[12] The SAFECOM Program. *Public Safety Statement of Requirements for Communications and Interoperability.* Department of Homeland Security, P.O. Box 57243 Washington, D.C. 20073, October 2006. Available online at `http://www.safecomprogram.gov/safecom/library/technology/1258_statementof.httm`.