A PROTOTYPE ALL-SOFTWARE PUBLIC SAFETY INTEROPERABILITY SYSTEM

Alok Shah (Vanu, Inc., Cambridge, MA, USA; abshah@vanu.com);
Jeremy Nimmer (Vanu, Inc., Cambridge, MA, USA; jnimmer@vanu.com);
David Franklin (Ucentric Systems, Maynard, MA, USA; dfranklin@ucentric.com)

ABSTRACT

A prototype waveform interoperability system has been built by Vanu, Inc. for the US Army U.S. Army Communications-Electronics Command (CECOM). The system enables law enforcement and public safety agencies with incompatible radio devices to communicate with each other.

The interoperability system is designed to scale to the needs of a particular user. It does this by performing all of the signal processing in software. The system hardware is chosen to be generic and waveform-independent. The RF subsystem covers wide swaths of spectrum, the digital I/O hardware is software programmable, and the computing platform is based on a general-purpose processor.

Challenges encountered during design and development of this interoperability system included full-duplex operation in a half-duplex system, sample delays in the software and hardware, and fundamental differences in the waveforms.

1. INTRODUCTION

> "If there is a single item that we could do [it] is to make sure that police, fire, emergency responders can communicate with one another. Oftentimes I go into a community and there are all types of bands and frequencies used and folks, literally, who are responding to an incident can't talk to one another. So that is one single item I can put my finger on that we need to address immediately."
>
> *-Testimony of Joe M. Allbaugh, Director of FEMA, to the United States Senate, October 16, 2001[1].*

The public safety community in the United States and elsewhere faces a serious interoperability problem. Different agencies operate radio systems with incompatible frequencies and/or waveform protocols. Within a particular agency, newly acquired advanced digital radios may not be able to communicate with legacy analog radios. Communications interoperability problems hindered the rescue efforts following the terrorist attacks of 9/11, as noted in the McKinsey report on the disaster [2].

The top half of Figure 1 describes the fundamental problem. The user on the left has a two-way radio that is either at a different frequency or transmits a different waveform than the radio on the right. There are a number of ways to deal with this problem, but all of them come with some limitations. The approach shown in the figure, in which someone listens to the audio on one radio and repeats it into the second one, is a commonly used solution.

Vanu, Inc., under a contract with the US Army, has developed a software radio prototype to address this interoperability problem. The prototype demonstrates the technical viability of a solution to the interoperability issue faced by first responders, among others, around the world. An intuitive interface allows the operator to dynamically establish a two-way patch connection between any two supported waveforms, allowing users to communicate across agencies with their own radios.

2. SOLUTION OVERVIEW

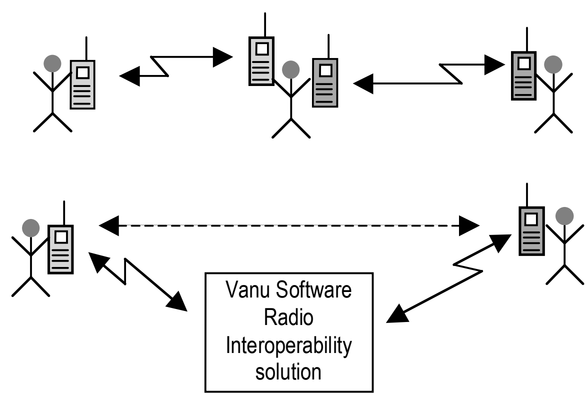

Figure 1: Interoperability: problem and solution

Public safety agencies currently have a handful of methods for providing communications between disparate radio systems. The most common ones are the following:

- Hand out new radios. An agency often keeps a backup supply of radios that can be loaned to other responders during an emergency. The advantage of this approach is that it guarantees interoperability. The disadvantage is that it requires personnel to use a radio that they may not be comfortable with in a high-pressure situation.
- Manual rebroadcasting of audio. An operator listens to the communications on one radio and repeats relevant commands into the second radio.

The advantage of this approach is that it allows responders to continue using their own radios. The disadvantage is that it is difficult for the operator to effectively repeat everything that he or she is hearing in real time.

- Use an automated interoperability system such as the JPS ACU-1000. These systems allow users to create patch connections by plugging radios into a chassis. This approach has a number of advantages, including that responders can continue to use their own radios and that the patch connections can be modified just by swapping radios. Disadvantages include system cost and the fact that audio cables must be available for each of the radios.

After examining these approaches and speaking with public safety personnel at various levels, Vanu, Inc. established a number of requirements for a public safety interoperability system. They include:

- Users should be able to continue using their own radios.
- The system should be intuitive to set up and operate.
- The system should be scalable both in terms of capacity and waveform support.
- The operator should be able to dynamically enable, disable, and modify patch connections.

The Vanu Software Radio virtual patch prototype uses a software framework on top of off-the-shelf hardware components to satisfy all of the above requirements. A java-based graphical user interface (GUI) allows the operator to choose two radios, combine them into a patch connection, and press “CONNECT” to enable communications. The operator can just as easily disable a patch connection or modify the radios within a connection. The prototype system currently supports one connection at a time using any two of the FM, APCO Project 25, and GSM waveforms. It can be scaled to support more simultaneous connections and additional waveforms.

Once the operator has established a patch connection between two radios at a given frequency pair, the software application tunes the hardware and creates the necessary signal processing chains in software. The system then listens for any transmission of either waveform type on the defined frequencies. Once a valid signal is received, the system processes it down to the source voice, re-modulates that source in the format of the second radio, and transmits it at the appropriate frequency. In this manner, users of incompatible radios are able to communicate directly.

3. SOFTWARE ARCHITECTURE

The software architecture of the interoperability system is based on modular components. By dynamically connecting modules based on operator requests, the system can support a variety of virtual patch connections.

There are three types of modules in the system: hardware channel modules, waveform modules, and audio arbitration modules. Each virtual patch has exactly one audio arbitration module, and one or more hardware and waveform modules, as shown in Figure 2.

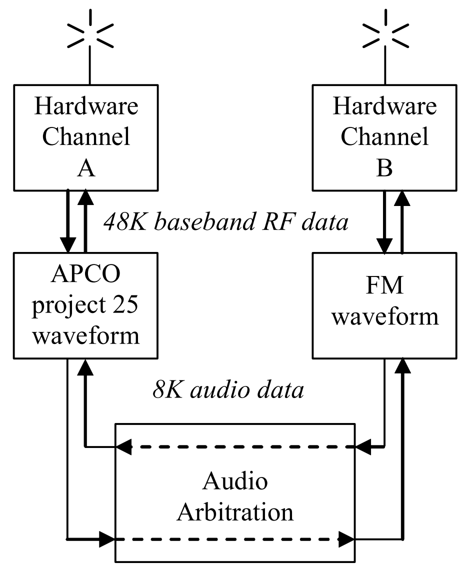

Figure 2: Software Block Diagram

The system transfers audio data and baseband RF data between different modules using Sprockit™ ports. Sprockit is a programming framework developed by Vanu, Inc. to support the implementation of data intensive real-time software radio applications [3]. As part of its interface, each module (set of objects) provides a set of Sprockit ports as follows.

The hardware modules provide a digital channel for baseband RF data. Each module has one input and one output port, each typically for interleaved I-Q data at the rate appropriate for the channel type.

The waveform modules encapsulate all waveform-specific properties, such as modulation and vocoding. Waveform modules have four ports: one audio input and one output port (each for 8k linear audio), and one data input and output for baseband RF data. A waveform module also has the responsibility of deciding when its RF input reflects a valid incoming transmission. For an FM waveform, this is done with energy threshold detection on the carrier; for GSM, this is done by observing DTMF signals for push-to-switch behavior (see Technical Challenges section below).

The audio arbitration module mediates the communication by selectively enabling connections between its audio ports. The arbitration module has one pair of audio ports (input and output) for each waveform in a patch. If the arbitration module receives valid audio from a given waveform, that audio is passed along to the opposite waveform(s) and all other audio inputs are muted.

Given this modular architecture, various substitutions are possible. For instance, if only one transmitter is available for a patch, hardware modules that share the transmitter and mediate access to it may be used. When the system is idle, nothing is transmitted. When one waveform receives a signal, the other waveform will produce RF data to be transmitted, and the hardware will retune to the proper frequency before transmitting.

As will be described in the Technical Challenges section, the original 48k sampling rate for RF data led to overly-high latencies when mixing in GSM waveforms. To address this, we introduced a hardware module that used a higher sampling rate to the hardware, but then resampled the data before passing it along to its external Sprockit port.

Finally, we note that the system is designed to permit the separation of the control unit, with its java-based GUI, and processing unit, consisting of signal-processing software and hardware. The GUI application and processing application communicate over a TCP connection and can therefore run on different machines if desired. One possible use case involves the processing system in the back of a van and the GUI running on a laptop in the front seat, or even on a handheld PDA.

4. HARDWARE ARCHITECTURE AND PLATFORM

The prototype hardware consists entirely of off-the-shelf components and provides essential flexibility and scalability to the system. Large public safety communities have different needs than smaller ones. The virtual patch is designed to scale to the needs of a particular user by running software on general-purpose processors. The RF and digital I/O hardware was chosen to be wideband and waveform-independent. That way patch capacity can be added to existing frequency bands simply by adding more processing power, and new frequency bands can be supported simply by adding more RF cards.

The hardware architecture of the virtual patch is shown in Figure 3. The RF and processing subsystems sit in a standard VME chassis, while the digital I/O cards plug into the processing cards. The VME interface was chosen for the backplane because it is supported by the chosen RF transceiver vendor, DRS Signal Solutions.

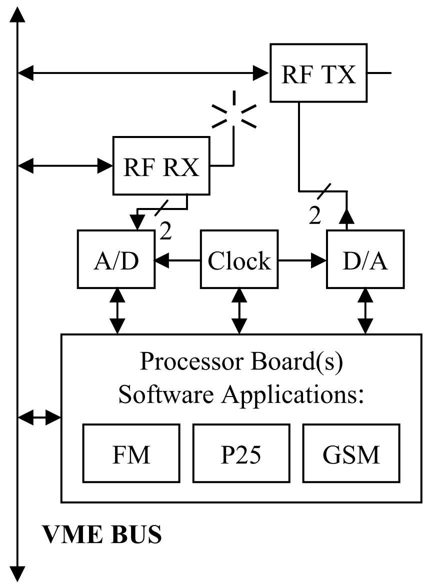

Figure 3: Hardware Architecture

Each hardware subsystem can be described in more detail:

- *RF TX, RF RX*: The DRS SI-7002 dual-channel transmitter and SI-9136 dual-channel receiver card both provide bandwidths of up to 25 MHz over a frequency range of 20 MHz to 3 GHz. This range covers all relevant public safety and military communications bands. RF parameters such as frequency, gain, and bandwidth can be set from the processor card over the VME bus. The RF cards interface with the digital I/O cards as an analog signal at an intermediate frequency of either 70 MHz (receive) or 16.25 MHz (transmit).
- *Processor boards*: The prototype system can support any VME processor card that runs the Linux operating system. The system currently includes the Concurrent Technologies VP110, which has a 1.2-GHz Intel Pentium III processor installed. All signal processing takes place in software running on the general-purpose processor; no DSPs or FPGAs are used for waveform-specific signal processing. The system requires three PCI Mezzanine Card (PMC) slots for the digital I/O cards. In the current prototype, the VP110 has one PMC slot onboard and uses an additional PMC carrier VME card to get the other two.
- *A/D, D/A, Clock*: The digital I/O subsystem consists of three PMC cards from Echotek Corp. The ECDR-GC314 and ECTR-GC314 are digital receiver and transmitter cards, respectively. The receiver card contains both a set of A/D converters, for sampling an analog signal into a digital one, and a set of digital downconverters, for filtering and resampling the digital samples. The transmitter

card has an equivalent set of digital upconverters and D/A converters. Each card supports multiple analog IF signals and up to 12 narrowband signals for parallel signal processing. A third Echotek PMC card, the ECSG-1R3ADC-PMC, is required to provide time-aligned sample clocks to the A/D and D/A cards. The clock card programmably generates up to three low phase noise clock signals between 1 MHz and 100 MHz. It also has a trigger to provide time alignment to the other cards. Using the clock card, the GC314 cards generate timestamps that can be used to calculated relative time offsets, which is important for cellular waveforms such as GSM.

The RF and digital I/O components chosen for this prototype provide exceptional system flexibility. The wide frequency range and bandwidth offer coverage for all relevant bands and waveforms, the multiple channels enable two simultaneous receive signal processing chains, and the digital up- and downconverters allow computationally intensive channel selection and sample rate conversion to occur without using processor cycles.

The cards described above all fit within a seven-slot VME chassis and enclosure in the prototype virtual patch. Additional capacity could be achieved by adding RF, digital I/O, and processor cards and moving to a larger chassis.

5. TECHNICAL CHALLENGES

A number of interesting challenges arose during development and testing of the virtual patch. A few of these issues are discussed below.

Two-way radios are inherently half-duplex devices, while cellular handsets are full-duplex. The virtual patch prototype includes support for patch connections between half-duplex radios (FM and APCO Project 25) and full-duplex radios (GSM), under the use case that a civilian official (for example, the governor) might want to communicate directly with public safety personnel. A user of a public safety radio presses a push-to-talk button to alternate between transmitting and receiving, but no such button exists on a standard GSM handset. As a full-duplex device, the GSM phone is always transmitting, which poses a problem for the system. The standard assumption is that only one device in a patch connection will be transmitting at any given time, and so the virtual patch operates by listening for a valid signal at the two patched frequencies. If the GSM phone is always transmitting a valid signal, then the patch will always be re-transmitting it, and the second device in the patch connection will never be able to take over the channel for its own transmission. The solution to this issue was to turn the GSM phone into the equivalent of a half-duplex device. A number of methods of enabling this were attempted by the development team:

- A voice activity detector based on the one in the GSM specification was implemented to determine whether or not the GSM user was speaking. When the user was not speaking, the half-duplex radio was allowed to transmit. The problem with this approach is that the microphones in today's GSM handsets are extremely sensitive. Even if the user of the phone was not speaking, nearby voices were detected by the voice activity detector, and the virtual patch rarely gave the channel to the half-duplex radio.
- A push-to-talk mode was implemented in which the GSM phone user would hold down a key while speaking. The virtual patch recognized the DTMF tone and only gave the channel to the GSM user when it received DTMF. The problem with this approach is that the GSM handsets that were tested muted their microphone when the user pressed a button.
- A "push-to-switch" mode was implemented in which the GSM phone user would press a button to toggle between speaking and not speaking. The virtual patch recognized the DTMF tone and alternated which radio owned the transmit channel. This option worked quite well in practice and was easy to understand and use.

Another issue that arose during system testing was feedback caused by sample delay in the system. In the normal case, the virtual patch transmits a signal at frequency A and then goes back to listening on both frequencies. However, the delay through the transmit path causes the end of the frequency A transmission to be received by the virtual patch, which then re-transmits the signal on frequency B. Again, the end of the frequency B transmission is received by the virtual patch and re-transmitted on frequency A. This pattern continues for some time. The solution is to introduce hysteresis into the system. By quantifying the delay through its transmit path, the virtual patch prototype was modified to not start listening on the two channels until a fixed amount of time after the transmission is completed.

A third interesting challenge was that cellular waveforms, unlike most two-way radios, generally have highly constrained real-time requirements. For example, layer 2 of the GSM full-rate traffic channel requires a turnaround time between receiving a timeslot and transmitting a response of approximately 42 ms. The implication of this fact is that a virtual patch system that supports GSM must achieve sufficiently low sample latency through the software and hardware. The system is currently set up such that both Echotek digital I/O channels are aligned in time and must be advanced at least one page, corresponding to 1024 samples, at a time. The issue is that one page of a low-data-rate waveform like APCO Project 25 corresponds to a substantial amount of time (1024 samples at 48 ksamples/second = 21.3 ms), and advancing the GSM samples through the hardware only once every 21.3 ms in each direction (receive and transmit) would make it very difficult to support the required GSM roundtrip time. The chosen method of compensating for this was to increase the sample rates of the FM and Project 25 hardware channel modules to rates similar to that of GSM, thereby making one page of FM or Project 25 samples a much smaller amount of time. The hardware channel modules then decimate the sample streams down to the 48 ksamples/second that the waveforms support. Rate matching mechanisms of this type seem likely to be required in patch systems that support future sophisticated waveforms, even if GSM is not one of them.

6. CONCLUSION

The prototype waveform interoperability system described in this paper takes a step toward enabling law enforcement and public safety agencies with incompatible radio devices to communicate with each other.

Several challenges were encountered during design and development of this interoperability system, such as full-duplex operation in a half-duplex system, sample delays in the software and hardware, and fundamental differences in the waveforms.

Despite these challenges, the prototype demonstrates the technical viability of a solution to the interoperability issue faced by first responders. The system also addresses operational requirements. Care was taken to account for the facts that users needed to continue using their own radios, that the system needed to be intuitive to use, that the system had to be scalable, and finally that the system operator required the ability to dynamically enable, disable, and modify patch connections.

7. BACKGROUND ON VANU, INC.

Vanu, Inc. evolved out of the SpectrumWare software radio research project at MIT in the mid 1990s. Since its inception in 1998, the company has focused on waveform software development for software radios. The extent to which Vanu Software Radio systems implement signal processing in software distinguishes them from other software radio design techniques. Vanu systems perform as much signal processing as possible in application-level software running on top of standard processors and operating systems. The advantage of moving signal processing into software is increased flexibility: a Vanu, Inc. waveform (air interface) implementation can run on a range of devices, from handhelds to scalable infrastructure.

Vanu, Inc. licenses software radio technologies and waveforms and provides design-consulting services to wireless device manufacturers, system integrators, and service providers. The company’s core competencies include strong software engineering, use of innovative signal processing algorithms, development of portable code, and the creation of commercial-off-the-shelf-based system design.

REFERENCES

[1] September 11, 2001: Attack on America. Hearings before the United States Senate Committee on Environment and Public Works. October 16, 2001. Statement of Joe M. Allbaugh, Director,FEMA.

[2] “Increasing FDNY’s Preparedness.” Prepared by McKinsey & Co.

[3] J. Chapin, V. Bose, "The Vanu Software Radio System", 2002 Software Defined Radio Technical Conference, San Diego, November 2002. http://www.vanu.com/publications/vanuinc-sdrforum-sd-submitted.pdf