

# The SpectrumWare Approach to Wireless Signal Processing

David L. Tennenhouse and Vanu G. Bose\*  
Telemedia Networks and Systems Group  
Laboratory for Computer Science, MIT

## Abstract

*The SpectrumWare project is applying a software oriented approach to wireless communication and distributed signal processing. Advances in processor and analog-to-digital conversion technology have made it possible to implement virtual radios that directly sample wide bands of the RF spectrum, and process these samples in application software. The elimination of dedicated hardware introduces tremendous flexibility into a wireless communications system. However, our approach goes further than the software implementation of traditional signal processing functions. We use processor and network memory to temporally decouple the sample streams from the software modules so that the bulk of the processing can be realized within virtual time programming environments. Decoupling relaxes the temporal constraints on the processing algorithms and their execution. This paper describes the SpectrumWare concept, our experimental approach, and the implications that this approach has for wireless signal processing algorithms.*

## 1. Introduction

The combination of processors, memories and analog-to-digital conversion technology have made it possible to extend the role of the computer to software-based processing of sampled signals. Speech, video, ultrasound, radar and baseband radio are examples of signals that are presently processed with digital technology. However, present day systems have two characteristics that limit the computational accessibility of the sampled data:

- The raw signal is subject to substantial analog pre-processing in advance of A/D conversion.
- Much of the digital processing is performed using customized, rather than general purpose, digital technology.

The SpectrumWare project is using wide bandwidth A/D technology [1] to overcome these limitations and vastly extend the “reach” of software-based systems. Instead

of tuning to a single channel, a SpectrumWare node is able to sample an entire radio frequency (RF) band and make all of its channels accessible to software processing. The software can then demodulate and decode any number of channels within the band, perhaps by assigning threads to different channels so as to give the user the appearance of processing many channels simultaneously. Transmission is accomplished in a similar manner, with channel-specific threads contributing to an outgoing sample stream that is used to synthesize the transmitted signal.

In the case of mobile networking, SpectrumWare pushes the analog components to the edge of the system and obviates the need for specialized digital hardware. More importantly, our approach permits *virtual* radios that have tremendous flexibility and scope for innovation – there is no fixed limitation as to the number of channels, the means by which they share the band, or the use of common modulation and coding schemes. We envisage a heterogeneous environment in which libraries of SpectrumWare software modules support a wide range of modulation and coding schemes

Figure 1 illustrates two SpectrumWare node configurations. Mobile nodes have directly attached RF transceivers. Stationary nodes may have attached transceivers but can also exchange sample streams with remote RF transceivers located elsewhere and accessible through a traditional network.<sup>1</sup> The stationary nodes and remote transceivers function as distributed “base stations” in which the antenna and A/D conversion functions are decoupled from the signal processing functions. A single stationary node can leverage multiple antenna sites and antenna sites can be shared by these processing nodes.

Recent efforts in software radios have been re-implementing traditional designs using software to emulate functions previously implemented in dedicated hardware [2]. In contrast, our virtual radio approach will involve the re-examination of the overall system from a software designer's perspective, with specific emphasis on the “virtualization” of RF resources.

For SpectrumWare to succeed it is not sufficient to push the digital and software domains to the edges of the system – it is also necessary to escape the real-time limitations of present day signal processing algorithms and the programming environments in which they are implemented. Accordingly, we are extending the reach of *virtual time* programming environments so that they can encompass all aspects of SpectrumWare processing. We are dedicating significant amounts of processor and network memory to the *temporal decoupling* of the sampled signals and the software processing modules.

---

<sup>1</sup>The Network ‘cloud’ provides a seamless connection between base transceivers and stationary processing sites. Peer transceivers could also be configured to directly exchange their sampled RF data, thereby implementing a distributed repeater which Steinbrecher has termed as a spectral *wormhole*.

---

\* A previous version of this work was presented at Mobicom'95, Berkeley, CA.

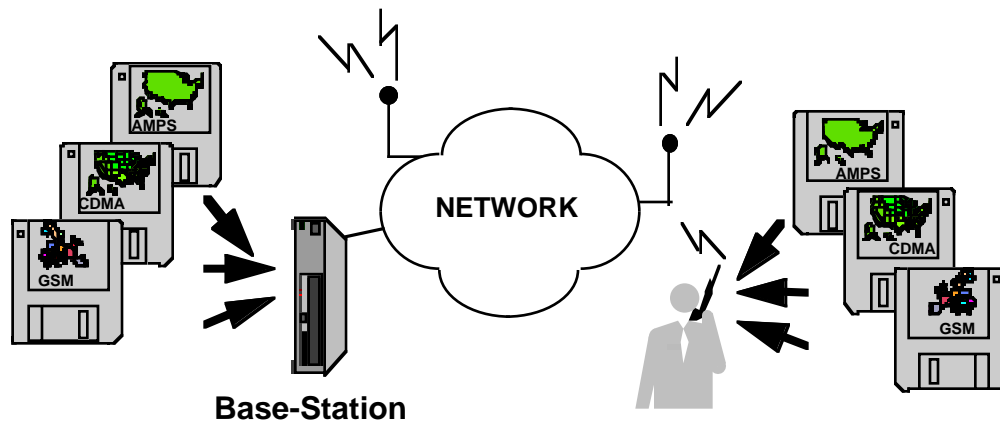


Figure 1: SpectrumWare Nodes

Temporal decoupling relaxes the fine grained temporal constraints on the signal processing algorithms and their execution environments, while maintaining an average data processing rate that meets the real-time requirement. As processors become faster, and the absolute timing constraints of a particular application remain fixed, the number of per sample instructions available to the algorithms will climb sharply.

In summary, our software-based approach to the processing of RF signals has many benefits, including:

- Increased flexibility:

Traditionally, analog hardware implemented in transmitters and receivers constrains the system to a particular modulation scheme and set of channel assignments. A software based system, utilizing direct RF sampling, can dynamically alter modulation and channel assignments.

- Rapid incorporation of advances:

Incorporating new modulation and coding methods into a SpectrumWare system is simply a software upgrade, requiring no new hardware.

- New channel coding algorithms:

Temporal decoupling relaxes the temporal constraints on the signal processing algorithms, thus creating opportunities for new approaches to signal processing.

We realize that our approach is somewhat aggressive and it will take some number of years before the underlying technologies and software mature. However, given the rates at which the sampling and processing technologies are improving, we are convinced that the bandwidth and variety of computationally accessible signals, as well as the complexity of their processing, will expand. In the not too distant future we will reach a point where SpectrumWare-like processing will be both feasible and desirable.

## 2. The Case for SpectrumWare

Opportunities to effect radical change in systems design are often rooted in a fundamental change in the technology trade-offs that have dictated the status quo.

In the case of SpectrumWare, the technologies in the balance are A/D conversion vs. general purpose digital processing, storage and transmission. The balance of power, which has historically favored A/D conversion, is undergoing a significant and historic shift. The SpectrumWare approach is not just becoming feasible – it may soon become the only way to achieve substantial improvements to signal processing applications.

### 2.1 A/D Conversion Technology

For the purposes of this discussion we distinguish three levels of commercially available A/D conversion devices:

- High resolution devices perform relatively precise conversions yielding 16 or 18 bit samples. These *narrowband* devices have a limited frequency range that is typically on the order of 100 KHz. They produce synchronous sample streams of up to 200 KSPS or 3.6 Mbps.

- Medium resolution devices perform less precise conversions yielding 10-14 bit samples. Some of the 10-bit *wideband* devices have a frequency range of up to 10 MHz. These devices produce sample streams at up to 200 Mbps.

- Low resolution devices yield 8 bit samples. These *broadband* devices have frequency ranges of up to 250 MHz and produce sample streams at 4 Gbps.

DARPA has announced a research goal of an 8 GHz, 8-bit convertor [3]. Table 1 compares the current research efforts [4] with what is commercially available. For this discussion, we restrict ourselves to the commercially available convertors, as we will be evaluating them with respect to commercially available processors and networking technology.

Resolution	Commercial Sampling Rate	Research Sampling Rate
16 bits	200 KSPS (Analog Devices)	5 MSPS (Hewlett-Packard)
10-14 bits	20 MSPS (Analog Devices)	50 MSPS (Hughes Aircraft)
8 bits	500 MSPS (Harris)	1 GSPS (Hewlett-Packard)

Table 1: Comparison of commercial and research grade Analog to digital convertors.

The wideband devices are of particular interest for two reasons: their resolution is sufficient to enable many interesting applications, especially in areas such as RF, where dynamic range is of prime importance; and mainstream digital technologies are crossing the threshold at which they can process their sample streams. Although the sample rates cited for these devices represent a present day snapshot, [1] suggests that the bandwidth converted at these resolutions is increasing relatively slowly. In contrast, the bandwidth that can be converted with lower fidelity is an order of magnitude wider and the absolute rate of their sampled data streams is on a steeper growth curve.

## 2.2 Processor and Memory Technology

At some fundamental level the clock rate of digital processors can at best asymptotically approach the limits of single bit A/D technology. However, processors operate in more controlled environments than sensory transducers with respect to dynamic range, noise, etc. and so their clock rates may someday exceed those of RF sampling technology. Furthermore, the digital abstraction allows us to construct much more complex systems than have been realized in the analog domain. These systems exploit many forms of parallelism – such as 64 and 128 bit wide buses, pipelining and multiple instruction units – to increase the computational power that can be brought to bear during each clock cycle.

Consider the case of a DEC Alpha-based workstation attached to an ATM-based network. Commercially available Alpha chips issue instructions 200 million times per second and can often issue two instructions in a cycle. At wideband sample rates approaching 20 mega-samples per second (MSPS) this corresponds to 10 instruction cycles per sample. A recently announced Alpha implementation will execute over 15 cycles per sample and will be able to issue up to 3 instructions per cycle. PC-like configurations incorporating 2-4 such processors will have the capacity to perform fairly extensive processing of wideband signal streams.

Recent improvements in memory, I/O and network bandwidth make the SpectrumWare approach feasible. Although memory has been an area of concern the

critical threshold has been crossed. The memory subsystem of an Alpha-based workstation can support a single memory-to-memory copy at 360-400 Mbps, which exceeds the rate of the wideband signal stream. The I/O system of such a workstation has been shown to support a single network/host transfer rate of 516 Mbps. Finally, a 200 Mbps sample data rate is only slightly more than the rate supported by commercially available ATM-based networks (155 Mbps) – and is less than one third the rate supported by research prototypes, such as the MIT VuNet [5].

## 2.3 Implications for Signal Processing Algorithms

The relative rates of progress in A/D and processor/memory technologies lead to two observations that have had considerable influence on our thinking concerning SpectrumWare algorithms:

- Any signal that can be sampled with resolution of 12 bits or better is now within reach of all of the major subsystems of conventional workstations and networks. The disparity in the rate of improvement between processor and wideband A/D technology is feeding an upwards spiral in the processing budget. The average number of instructions and memory copies per sample that can be made available to the application programmer is increasing and will continue to do so.

- In the long term the slow evolution of high and medium resolution A/D technology will limit the amount of sample data that can be fed to processors. Significantly increasing the flow of sample data through the processor will involve the development of broadband modulation and coding schemes that are compatible with low resolution (high-bandwidth) sampling.<sup>2</sup>

Software processing of raw RF data introduces many opportunities for novel signal processing algorithms. In particular, the requirement for sequential computation is relaxed; the degree and type of processing can be varied on a sample by sample basis; and multiple signals of different types can be processed simultaneously. We are pursuing this agenda at three levels:

- Software implementation of conventional modulation algorithms.
- New algorithms for traditional modulation schemes.
- Novel coding and modulation schemes that are software-friendly, and compatible with low resolution sampling.

### 2.3.1 Conventional modulation algorithms.

The first step is to develop SpectrumWare software that emulates existing analog hardware. This provides an opportunity to demonstrate the approach and benchmark the software environment with known algorithms. We have designed algorithms to receive conventional radio

<sup>2</sup>[6] describes the hardware implementation of a CDMA scheme that uses low resolution samples.

broadcast signals (AM & FM), and will extend this work to the demodulation of AMPS cellular telephony signals.

The next step is to emulate digital RF systems in the SpectrumWare environment. Using a PCMCIA host adapter we plan to implement a SpectrumWare wireless LAN adapter, to demonstrate the flexibility and utility of the approach. We note that it should be possible to make other RF transmissions, such as GPS, accessible via SpectrumWare software.

### 2.3.2 Identification of new algorithms.

Temporal decoupling enables new algorithms that take advantage of relaxed constraints on computation. These *virtual time* algorithms may leverage large amounts of memory to process samples out of order, make multiple passes on the sample stream<sup>3</sup>, and/or leverage statistical behavior. Conventional coding schemes will be re-examined in this light, to identify specific areas where we can obtain significant performance.

The trade-off between memory access and add-multiply instruction execution is an area to be investigated. Add-multiply instructions have been the traditional cost metric, and so most algorithms have been designed to minimize the number of these operations. However, memory accesses, in particular cache misses, are costly operations for today's processors. By designing algorithms that balance instructions and cache misses, it may be possible to gain additional computational savings.

For example, the FM receiver outlined in section 4, has a target processing budget of about 4 instructions per sample. However, looking more closely at the design of the processor reveals that there is significantly more capacity available. On the DEC 21064, words are 64 bits long, so we could pack four 16-bit samples into each word. This opens a potential new class of signal processing algorithms where multiple samples are treated as single words.

The wideband sampling of the RF spectrum also provides opportunities to improve the robustness of symbol decoding. For example "soft-decoding" [7] uses information about a signal to determine a confidence factor for each decoded symbol. We can extend this approach to utilize information from the wideband signal that is not available to conventional receivers. Most soft decoding systems make decisions based on information obtained from the demodulated baseband signal. By directly sampling the RF spectrum, we are able to incorporate information about the received wideband signal, such as power levels on adjacent channels, into the soft decision process.

---

<sup>3</sup>Conventional DSP software does this to a limited degree, e.g., in the implementation of tap filters. However, these buffers are orders of magnitude smaller than those we envisage.

The move to general purpose processors and multi-tasking operating systems opens up many new areas for research into signal processing algorithms. In many cases, it would make sense to process bursts of samples as fast as possible, so that the excess processor capacity could be applied to other tasks during the intervals between bursts. Similarly, algorithms that perform scanning functions, e.g., monitoring channel occupancy, can grab a large burst of samples and process them on a background basis. We expect the development of virtual time algorithms to be a long term effort that will gain momentum as technology improvements make the approach feasible for a wider range of applications.

### 2.3.3 Novel modulation schemes.

Wireless communication presents a particularly interesting opportunity to develop software friendly modulation schemes. Unlike signals that occur in nature, the signals that are being sent and received over the wireless spectrum are artificial, so we can shape them to suit our purposes (subject to channel impairment). Additionally, we are free to modify the entire receiver structure, allowing for coding schemes that optimize resource usage [8].

We are especially interested in new approaches that leverage low resolution broadband sampling, especially those that simplify use of multiple access channels. Rather than requiring each transmitter to be time synchronized, we prefer schemes that support asynchronous transmission and reception. We note that CDMA has some of the desired characteristics but hypothesize that relaxing the temporal constraints might lead to alternative approaches that are easier to implement on general purpose platforms. Multiple passes through the received sample stream could allow a receiver to decode a varying number of simultaneous transmissions.

### 2.4 Escape from the synchronous strait-jacket

A near universal property of digital signal processing (DSP) environments is the synchronization of I/O and computation. Traditionally the DSP programmer carefully manages the computation on a fine grained basis to assure that all necessary work is completed in accordance with a strict schedule – in effect, the program runs lock step with the I/O. Although DSPs have evolved under different constraints from general purpose processors, there has been some convergence. For example, general purpose processors have recently incorporated features, such as single-cycle multiply-add capability, that were previously only found on DSP processors. The most significant remaining differences are the low latency of DSP interrupt mechanisms and DSP's avoidance of data caching. Not surprisingly these are the features that touch on the cornerstone of the DSP approach, the relatively synchronous processing of the sample stream.

As general purpose processors improve, their the absolute interrupt latency will decrease<sup>4</sup>, even if they continue to require more clock cycles per interrupt than DSPs. Since the requirements of individual latency-sensitive applications are fixed, the range of applications that can be supported on conventional processors will continue to expand. Furthermore, temporal decoupling of sampling and processing relaxes the latency constraints of many applications.

Caching may be a thornier issue, as many signal processing algorithms do not display the locality of data reference found in data processing. In some cases, such as the FFT of a very long sequence, the presence of caching can actually hurt performance. Instead of caches, newer DSPs include up to 4 Mbits of fast memory on chip that allows the programmer to manually “prefetch” samples into the high speed memory. None the less, we note that newer general purpose processors support prefetch instructions that allow the cache to provide the equivalent functionality. More importantly, we believe that there is an opportunity to develop new signal processing algorithms that exhibit locality, making the use of caches desirable.

The problem does not stop at the processor. Artificial restrictions are placed on the range of processing algorithms that are even considered by present day designers. It is widely held that to perform real time digital signal processing, programs must be executed within near synchronous environments, i.e., they must directly control the processor or be supported by real-time operating systems. While such a coupling appears to be a logical consequence of synchronous sampling, we believe that this linkage is unnecessary. In particular, our experience and those reported in [10] bear evidence that general purpose computer workstations running time sliced operating systems, such as UNIX, are increasingly capable of performing traditional DSP computations.

In recent years, the signal processing community has taken some steps towards a more relaxed temporal perspective, in both the hardware [11] [12] and software compression domains. However, we are advocating a wholesale change in temporal perspective that will bring a more complete suite of CS tools (virtual memory, multi-threading, etc.) to bear on signal processing problems.

### 2.5 Towards a computer science perspective on information theory

Information theory focuses on the relationship of channel bandwidth, signal level, noise and capacity. It also provides a bridge between various modulation and coding domains including frequency division, time division and code division. Many applications of information theory deal with trade-offs between the time

---

<sup>4</sup>Albeit at a rate that lags improvement in the technology as a whole [9].

and frequency domain. Other applications, especially in the realm of coding deal with the mapping of bits onto symbol alphabets that are suited to the available channel(s).

Ultimately, we are searching for a computer science perspective on information theory that provides additional insight as to how memory and processing resources can be leveraged.<sup>5</sup> It should also provide computer scientists with a more intuitive way to think about modulation and channel coding than present models that have their roots in analog signal processing. For example, computer science theorists are quite used to dealing with time/space trade-offs and we would like to extend that familiarity to time/frequency/code domains.

## 3. The SpectrumWare Approach

Present day approaches to signal processing are held back by their reliance on customized hardware and software components that emulate the operation of their analog predecessors. These designs are complicated by the temporal sensitivity of the sampled and synthesized information. The accepted norm is a tightly coupled approach that attempts to preserve the timing of the sampled data as it traverses the system.

The SpectrumWare project will explore a temporally decoupled approach. By pouring samples of the wideband RF signals into host memory we are able to take advantage of conventional operating systems and networks to process the signal, as opposed to special purpose DSP's with customized interconnection technology. [3] There are three basic technologies that enable the SpectrumWare approach:

- Wideband A/D converters that can sample large bands of the RF spectrum.
- Temporal decoupling mechanisms that bridge the gap between the transceivers and the pioneering software.
- A software environment that is capable of handling temporally sensitive data in a conventional multi-tasking environment.

### 3.1 Wideband RF Transceivers

Wide band A/D conversion allows the construction of simple RF transceivers, consisting of only: an antenna, pre-amplifier and a wideband A/D converter in the receive path; and a D/A converter and an amplifier in the transmit path. Directly attached transceivers transfer the sampled signal data directly into the memory of their local host. Similarly, software synthesized samples are transferred from memory to drive the D/A, thereby generating the transmitted waveform. Remote transceivers are similar in construction, but incorporate

---

<sup>5</sup>Since the advent of digital computers, the trade-off between processing power, memory and effective channel capacity has been recognized [13].

simple network interfaces that sink and source their sample streams.

Wideband conversion places some serious demands on the RF amplifier. The amplifier must be linear over the entire band that is to be digitized, and automatic gain control (AGC) is difficult to implement over a wideband. Advances in these areas will be critical to the realization of the full software radio paradigm.

SpectrumWare nodes can receive a wide variety of signals due to the flexibility of the software processing and the wide bandwidth of the analog section. The signal processing functions that are typically performed in customized software and hardware are realized in SpectrumWare software modules – portable application packages that execute on conventional operating systems and processors. This serves the dual purpose of using software to maximize the flexibility of the system and keeping the transceivers as simple as possible.

### 3.2 Temporal Decoupling

Although transceivers are subject to various constraints imposed by the real world, applications are typically cocooned within *virtual* execution environments supported by operating systems and processors. A key observation is that the different components of a SpectrumWare system will operate at substantially different time scales with pronounced differences in the granularities and predictability with which they process information:

- Transceivers operate on a fine grained basis, dealing with small amounts of data (8-16 bits) at relatively precise and periodic intervals.
- Application programs operate on a coarse grained basis, manipulating data in large units, such as video frames, that take varying amounts of time to process.
- Operating systems and packet networks occupy an intermediate position that bridges the gap between appliances and applications.

In the development of the SpectrumWare approach it will be necessary to address the discontinuities that arise at the junctions between timing regimes. A distinctive aspect of our architecture is the identification of the points at which different temporal domains abut and the careful design of *temporal decoupling* mechanisms that smooth the transitions at these boundaries.

Instead of extending the real-time paradigm across the whole system, the SpectrumWare approach extends the boundaries of the virtual time environment, by: (i) time-stamping and temporally decoupling sampled information at the edge of the system; and (ii) providing a virtual time programming environment in which it is possible to implement applications that process temporally sensitive information. An advantage of this approach is that it accommodates system components that achieve their long term performance targets through dynamically scheduled bursts of high speed activity. Human users, time sliced processors and network links

are examples of such dynamically multiplexed resources.

Figure 2 illustrates the various control loops that might co-exist amongst the mutually asynchronous components supporting transceiver-application communication. Our architecture explicitly recognizes the role of memory as the elastic glue that binds together the pipeline components. Memory buffers support temporal decoupling by allowing information to flow between adjacent components operating at different instantaneous rates. They play an important role as *burst absorbers* – the SpectrumWare analog of automotive shock absorbers.

By temporally decoupling the sampling from the computation, we relax the constraint that the system handle a new sample during every sample interval,  $t$ , to a looser constraint that the operating system provide the application with sufficient resources to process  $N$  samples every  $Nt$  seconds. On a coarse grained basis, the application is keeping pace with the sample stream. However, finer grained analysis would show that the application is sometimes executing faster than the stream while at other times it is slower. While this use of buffering is not new, we can take advantage of this temporal relaxation in ways that are typically shunned by present day signal processing systems:

- The processor can use techniques, such as data caching, which lead to variability in the time taken to execute a sequence of instructions.
- Variations in the time taken to process individual samples can be leveraged. Time *saved* during the processing of one subset of the samples can be applied to the processing of neighboring samples.
- In multitasking environments each application runs a fraction of the time, executing bursts of instructions at a super-rated pace that exceeds the rate of its symbol stream. The resources *saved* by one application can be re-assigned and used by others.

We are not advocating systems that exhibit completely unpredictable behavior. Instead, we are leveraging differences in the granularity at which deterministic behavior need be achieved. Consider an application whose output is only observed at human perceptual granularities, i.e., KHz bandwidths. It is perfectly reasonable for the application to exhibit statistical behavior at finer granularities, i.e., MHz sampling rates, provided its coarse grained behavior is fairly predictable. There are three orders of magnitude difference between these two granularities. We believe that the granularity required by a large number of applications can be realized within virtual time execution environments.<sup>6</sup>

---

<sup>6</sup>Network transmission latencies virtually ensure that wide area networked applications can be realized in this manner. The applications that are excluded are those that have a feedback path of short duration, i.e., the latency between sampling analog information and generating an analog response must be finer than can be achieved by virtual time systems. In present

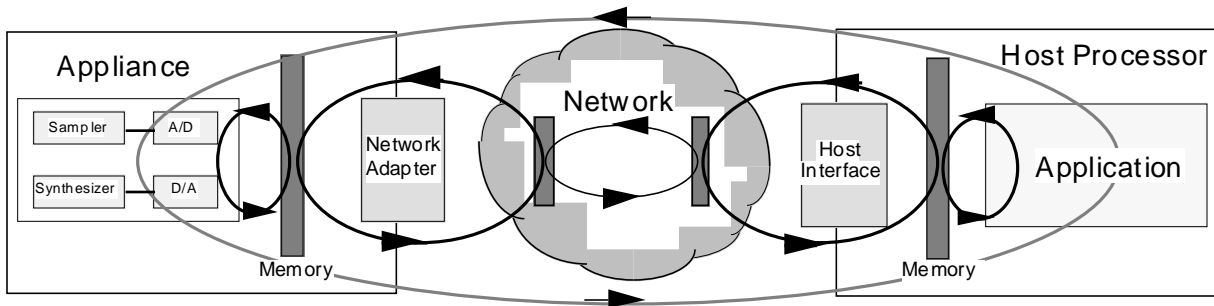


Figure 2: Temporal decoupling and closed loop control.

Furthermore, the granularity that can be realized by these environments continues to improve with the technology.

Although each process can operate at its own instantaneous rate, it is clear that the long term average flow through all stages of the pipeline must be balanced. SpectrumWare relies on transport level protocols to provide closed loop feedback on an end-to-end basis.

The ViewStation project [5] has demonstrated the temporal decoupling of ATM-based video cameras. The temporal decoupling achieved by this system allows the application software to execute at a pace that dynamically varies in accordance with the data and processing resources made available to it. Our experience with the ViewStation has shown that this approach is able to satisfy the requirements of many temporally sensitive applications. The decoupling of the application execution environment is crucial if software is to operate on platforms with dynamically varying loads (e.g., multi-tasking environments) and be portable across platforms executing at different speeds.

SpectrumWare is one or more orders of magnitude more aggressive than the ViewStation, i.e., it will extend the virtual time software domain to embrace a much larger fraction of the total signal processing task. In the ViewStation environment, the applications operate on audio and video frames which have been extracted from the channel. Dedicated hardware implements a number of channel processing functions, including RF filtering and demodulation, A/D conversion, NTSC synchronization and color separation, etc. In the SpectrumWare environment these functions are re-ordered and all except for the simplest of RF filtering and A/D conversion are performed in temporally decoupled software.

### 3.3 The SpectrumWare Software Environment

SpectrumWare software supports the integration of a range of sensory and RF signals within a consistent programming framework. This framework must achieve a balance between the desire for medium-independent technologies and the need to accommodate

environments the excluded applications are those whose feedback paths are less than about 10 msec.

heterogeneity, i.e., the inherent differences amongst the media and the manner in which they are processed. Furthermore, it must be efficient and responsive enough to support the software-based implementation of temporally sensitive operations, such as channel and source coding. The ultimate objective is a virtual time environment that is competitive with present day DSP environments but executes on commercial workstations and operating systems.

Our prototype software is based on the VuSystem [14] which has demonstrated the feasibility of software-based approaches to conventional media, including audio, video, text and graphics. The VuSystem implements a two level strategy, as illustrated in figure 3, in which programs are partitioned along an *in-band* axis, that supports the flow of temporally-sensitive information, and an *out-of-band* axis that supports the event driven program components, including the user interface and the configuration and control of the in-band processing *pipeline*. This approach localizes the portions of the software over which programmers must observe a limited degree of temporal discipline.

SpectrumWare's in-band processing components are arranged into processing modules that pass dynamically-typed and time-stamped *payloads* through input and output ports. The time-stamps, which may be generated by the remote appliances, allow for the eventual re-synchronization of the virtual time stream with the real world.

Figure 4 is a mock-up of the visual programming interface [15] to a software based receiver. The in-band software pipeline mimics the block diagram of a

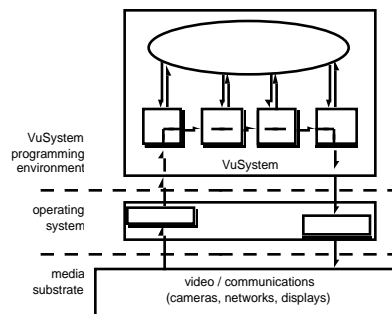


Figure 3: The ViewStation / VuSystem architecture.

conventional hardware-based receiver. However, by changing the function of the RecFilterblock, while the receiver is running, the type of demodulation or channel assignment can be changed. Similarly, the decoding algorithm and decision thresholds can be altered.

Sampling widebands of the RF spectrum allows multiple channels to be processed simultaneously. Thus several communications channels can be opened simultaneously along with applications that monitor other channels for certain information. In order to fully exploit these opportunities we plan to incorporate multi-threading capability into the programming framework.

### 3.4 Summary

The SpectrumWare approach utilizes wideband sampling, and temporal decoupling to enable the software processing of RF signals on conventional platforms. Temporal decoupling allows the samples to be processed in large blocks, and a whole new range of algorithms that leverage statistical behavior may be brought to bear on the problem space.

In section 4 we describe our experimental testbed, provide details of our approach and describe the application of the SpectrumWare concept to a simple RF application – the tuning of AM and FM radio stations. Although we expect to use SpectrumWare in more challenging environments this familiar application conveys the *flavor* of our approach.

## 4. A SpectrumWare Testbed

The SpectrumWare testbed will have three primary components: the network backbone, the RF transceivers and the software environment. The testbed leverages networking components that have been developed within the ViewStation project. The backbone of the testbed is an ATM-based network linking 6 VuNet switches, whose ports can source and sink traffic at up to 700 Mbps.

### 4.1 Transducer Appliances

We consider RF transceivers to be instances of a broader class which we refer to as *transducer appliances*, illustrated in figure 5. These devices, either locally attached or remote and network-accessible, support the wideband digital sampling of analog

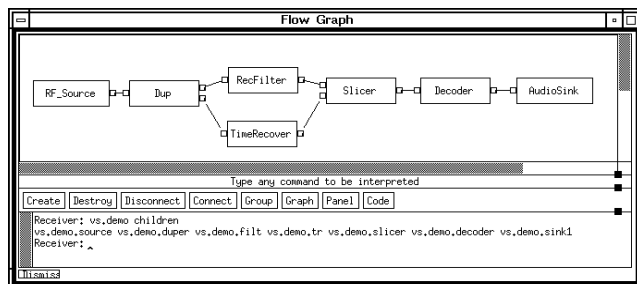


Figure 4: SpectrumWare Representation of a Digital RF receiver

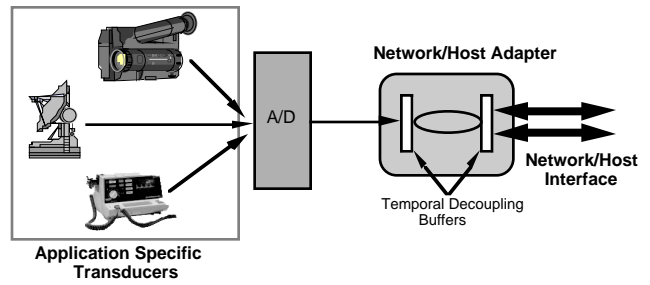


Figure 5: Generic Transducer Appliance

information and/or the synthesis of analog waveforms. The goal is to make the appliances as simple as possible consisting only of a transducer with minimal filtering and amplification, a wideband digital converter and either a network or host interface.

Directly attached appliances have two levels of temporal decoupling. The word/cache level FIFO provides fine grained temporal decoupling by allowing a sequence of samples arriving from the synchronous data stream to be collected into moderately sized units. These units are sized so as to correspond to the word or cache line length of the memory subsystem of the processor to which the device is attached. The second level of decoupling is provided by the memory subsystem itself. The incoming sample stream is *poured* into the memory so that it can be examined at the convenience of the software. In practice, the software is likely to process the samples in relatively large blocks which may be many multiples of the memory page size. Furthermore, this sample processing may be performed in bursts as dictated by the behavior of the application program and the time-slicing of the processor among applications.

The memory subsystem provides a similar decoupling function for synthesized waveforms. The applications can generate outgoing samples in bursts which are absorbed by the memory and played out to the appliance in word/cache line units which are then played out on precise boundaries.

For network appliances, the fine grained decoupler allows the synchronous data stream to be collected into packets or ATM cells. The second level of decoupling memory is located at the appliance/network interface where larger FIFO's buffer the packets (cells) to be transmitted. Similarly, the outgoing data path has relatively large buffers at the network/appliance interface and a smaller playout FIFO adjacent to the D/A converter.

### 4.2 Prototype Appliances

Several networked information appliances are being prototyped and incorporated into the testbed. Figure 6 shows a block diagram of the appliance adapter that has been built. This appliance is generic; by changing the transducer and filter blocks, the appliance can be customized for various applications. The section of the diagram within the dotted line is application specific

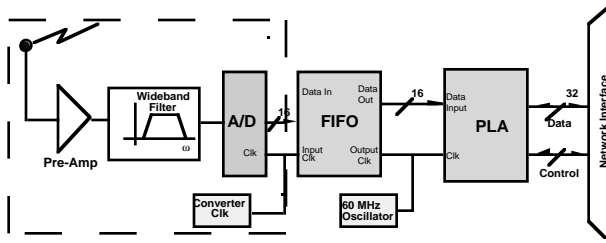


Figure 6: RF Transducer Network Appliance

and is implemented on a separate daughter card. Three different types of appliances have been designed using this approach:

- RF Receiver Appliance

An appliance capable of digitizing the AM broadcast band has been designed. This frequency band was chosen to provide a simple example which would allow us to test the system and demonstrate some of the signal processing approaches. This appliance is described in detail in the next section.

- Cellular Telephony Appliance

A Cellular Wideband receiver, on loan from Steinbrecher Corp., digitizes a 10 MHz swath of the cellular spectrum with 12 bit resolution at a sample rate of 25.6 MHz. The appliance produces data at a rate of 307.2 Mbits/sec, which is within the capacity of our ATM network. We are presently developing a simple digital interface that will bridge the receiver to the generic appliance adapter.

- Generic Medical Appliance

A multi-input, low-band width (90 KHz sampling rate) appliance (a generic platform for medical instrumentation) has been built. By attaching the appropriate transducer, this platform will be used to monitor signals such as EKG, EEG, ultrasound, temperature and blood oxygen simultaneously.

Both the local and remote (network-based) appliances pour their sample streams into host memory. For example, to decouple the cellular band data stream into segments of 100 Hz granularity (i.e. roughly ten milliseconds of data) would require multiple buffers each 384 K-bytes in size. It is quite reasonable to have this amount of memory available to the processor.

In addition to these networked appliances, we are currently developing a directly connected PCI-based appliance. This will provide much higher throughput from appliance to memory by utilizing the 132 MByte/sec PCI bus. Using the experience we gain from this design we plan to implement a PCMCIA appliance. Coupled with high performance laptop computers, this will enable mobility experiments as well as rapid prototyping and analysis of wireless communication systems.

### 4.3 Prototype Application

The software processing of AM and FM radio is the first stepping stone that will allow us to build up our tools

and benchmark our performance against existing (hardware-based implementations). As we gain experience, we will move on to more novel algorithms and complex modulation schemes, such as CDMA.

Directly digitizing the RF data alters some of the basic considerations in receiver design. For example, there is no need for an IF stage. IF was introduced into receivers to eliminate the need to construct costly, high frequency, variable filters and amplifiers. However, emulation of such filters is neither costly or difficult in software. The AM receiver consists of: an A/D converter that directly samples the low end of the spectrum up to and including the entire AM band, at a sample rate of 3.5 MSPS with 12-bit resolution; a variable band pass filter implemented in software that selects the channel of interest directly from the RF data; and an envelope detector which outputs the modulating signal.

The envelope detector uses a look-ahead scheme which allows it to reduce the computational overhead. Once the first maximum is found, the detector can predict approximately where the next maximum will occur. Thus the next maximum can be found by searching over a small window in the neighborhood of the next expected maximum. The size of the window is determined by the noise present in the signal, and can even be dynamically varied depending upon a running estimate of the noise. We intend to implement such an algorithm and calibrate its performance on the testbed.

FM demodulation presents a slightly different problem, since the carrier frequency is in the range of 88- 108 MHz. While the direct sampling of these signals is within the range of current 8-bit A/D converters, higher resolution converters are not yet fast enough. By using bandpass sampling [16], we can capture the entire FM band by sampling at 43 MHz, instead of sampling the entire band at upwards of 200 MHz, which would be a challenge not only for the state of A/D's but also for processing the signal on today's RISC processors. We will first bandpass filter the received signal between 87.1 and 108.9 MHz, then sample the filtered waveform at 43.55 MHz, which brings the entire FM band down to baseband. The bandpass filter removes the out of band energy, thus avoiding aliasing although the sampling rate is much less than the Nyquist rate. Bandpass sampling is a generally useful technique for wireless communication, since most types of broadcast or communication devices are assigned to a specific frequency range. The issues involved with processing data sampled at this rate were discussed in section 3.3.

## 5. Application Domains

The following subsections discuss two SpectrumWare application domains: virtual radios, which have been the subject of this paper; and virtual instruments. As previously mentioned wireless communication is of particular interest because, subject to channel impairment, we can shape the sample streams to meet our purposes. However, SpectrumWare also enables

virtual instruments, which can be easily configured for a wide range of applications by simply changing the transducer and a minimal number of components implemented on the daughter cards. Ultimately we hope to gain an understanding of the rules that apply to any sampled medium. Not just the relationship between sample rate, bits per sample, but also statistical properties that relate average instructions per sample, distribution of instructions per sample and memory requirement.

### 5.1 Virtual Radios

A traditional wireless RF system (e.g. a cellular telephone) is realized entirely in dedicated hardware. The receiver is capable of listening to only one of a pre-determined set of channels using a specific stack of modulation, channel coding and source compression standards. The SpectrumWare approach pushes the digital conversion to the stage before the channel processing, allowing the channel assignment and coding to be flexible. Additionally it is possible to scan channels simultaneously, since a broad section of the spectrum is captured.

Cellular telephony service providers are facing an interesting problem: channel and source coding are advancing so rapidly that the cellular community is presented with a multitude of standards (e.g. AMPS, GSM, CDMA). Upgrading the transmission standard however, requires service providers to install new hardware at the base stations and users to upgrade their handsets. Furthermore, a cellular phone user can only use his/her handset on compatible networks. SpectrumWare eliminates both of these problems - upgrading standards is simply a software upgrade on both the handsets and base stations. Furthermore, a single handset can use different transmission standards by simply activating the appropriate software.

Clearly, the SpectrumWare approach is ideally suited for mobile applications, where the user can seamlessly migrate between communications systems using different frequencies, modulations and coding. However, the power constraints in the mobile environment demand attention. Part of the problem is being addressed by emerging technology, for example lower power A/D converters and low power RISC processors (e.g. ARM processor). SpectrumWare can also reduce the power utilization in software. For example, by implementing a standby mode in which the processor and convertor only wakeup for short intervals to determine if anyone is trying to communicate with the mobile host. This is done by digitizing and storing a time-slice of the spectrum, after which the convertor can shut down again, and then processing the samples to look for communications addressed to the mobile host.

### 5.2 Virtual Instruments

We have built a foundation that allows us to prototype a virtual medical instrument. The range of signals to be

supported include ultrasound, EKG, EEG, oxygen and temperature sensors. Ambulances and bed-side nodes could be equipped with a single instrument that can be dynamically configured by snapping appropriate transducers into place and activating instrument-specific software. Virtual instruments overcome a significant preparedness problem — in mobile environments the premium on space and weight may limit the number of physical instruments that can be carried. Our approach equips the para-medic with a generic instrument that can be customized in the field to meet the needs of each patient.

Another type of virtual instrument that will be prototyped is the *virtual oscilloscope*. The test instrument industry has already taken the first step towards this goal. Products such as LabView<sup>7</sup> provide software control of existing test instruments and the possibility of downloading captured data for off-line processing at a workstation. Our approach will extend this concept to the point where the test instrument is reduced to an A/D converter and workstation software.

## 6. Conclusion

The three main advantages of the SpectrumWare approach are: a reduction in the cost and complexity of the hardware; the ability of the system to incorporate technological advances; and the potential for new approaches to signal processing algorithms. These advantages are realized by performing as much of the computation as possible in software.

The SpectrumWare testbed provides a vehicle for studying the impact of software based processing on wireless communications. As the A/D and processor technology improves, it will be possible to implement more involved, and more flexible processing algorithms on even wider bands of sampled RF spectrum. We expect that significant effort in the future will be put towards developing software signal processing algorithms to leverage these technologies and provide more flexible, robust wireless communications.

The software approach provides tremendous flexibility for mobile communications systems. Changing modulation schemes or frequency assignments involves the loading and/or configuration of software modules. Mobile units moving between service providers, or continents, can be dynamically configured to inter-operate with local facilities. Similarly the base station software can be upgraded to support new modulation schemes without having to upgrade hardware, especially the RF components that may be located at remote antenna sites.

SpectrumWare is a long term approach that will become increasingly practical as technology advances. In the meantime it provides an experimental sandbox that allows us to investigate: the nature of the boundary between virtual time and real time; mechanisms that

---

<sup>7</sup>National Instruments, 604 Bridge Point Parkway, Austin TX.

leverage statistical behavior; and the granularity with which sequences of samples are processed. We believe that this approach will lead to a computer science perspective on the underlying channels supporting mobile communication systems and, more generally, on signal processing and information theory.

## Acknowledgments

The authors would like to thank the many individuals who have contributed to the SpectrumWare effort, especially Don Steinbrecher for the helpful discussions and insight, the reviewers for their helpful comments and Bill Stasior and David Wetherall for their contributions and editing of the manuscript. This research was supported by the Advanced Research Projects Agency under contract No. F30602-92-C-0019 (monitored by AFSC, Rome Laboratory) and contract No. DABT-6395-C-0060 (monitored by US Army, Fort Huachuca).

## References

- [1] Harris, F., Steinbrecher, D., *Wireless Interface Design for Digital Processors*. In ICUPC'93 Tutorial 5, October 1993.
- [2] Lackey, R., Upmal, D., *Speakeasy: The Military Software Radio*, IEEE Communications Magazine, pp.56-61, Vol. 33, No. 5, May 1995.
- [3] Mitola, J., *Software Radios Survey, Critical Evaluation and Future Directions*, In Proc. IEEE National Telesystems Conference, May 19-20, 1992, Wash. D.C.
- [4] Wepman, J., *Analog-to-digital Convertors and Their Applications in Radio Receivers*, IEEE Communications Magazine, pp.39-45, Vol. 33, No. 5, May 1995.
- [5] Tennenhouse, D.L., et al, *A Software Oriented Approach to the Design of Media Processing Environments*, IEEE Int. Conf. on Multimedia Comp. and Systems, Boston MA, May 1994.
- [6] Sheng, S., Allmon, R., Lynn, L., O'Donnell, I., Stone, K., *A Monolithic CMOS Radio System for Wideband Digital CDMA Communications*, Calgary Wireless Conference, 1994.
- [7] Lee, E. A., Messerschmidt, D. G., *Digital Communication*, Kluwer Academic Publishers, 1994
- [8] Yun, L.C., Messerschmitt, D.G. *Power Control for Variable QOS on a CDMA Channel*, In Proc. IEEE MILCOM, Fort Monmouth, N.J. pp. 178-182, Oct, 1994
- [9] Anderson, T., Levy, H., Bershada, B., Lazowska, E. *The Interaction of Architecture and Operating System Design*, 4th Int. Conf. on Arch. Support for Programming Languages and Operating Systems, Pages 108-120, Santa Clara CA, April 1991.
- [10] Stewart, L. C., Payne, A.C., Levergood, T.M., *Are DSP Chips Obsolete ?* Technical Report CRL 92/10, Cambridge Research Laboratory, Cambridge, MA, 1992.
- [11] Analog Devices, *SHARC Preliminary Data Sheet*, One Technology Way, P.O. Box 9106, Norwood, MA, ADSP-21060/62, November 1994
- [12] Grove, A., *Synopsis of Comdex 1994 Keynote Address*, <http://www.intel.com/intel/news/speeches/cdxkey.html>
- [13] Fano, R.M., *Digital Computers in Communications Engineering*, In Proc. Computer Applications Symposium, 1961.
- [14] Lindblad, C., Wetherall, D., Tennenhouse, D., *The VuSystem: A Programming System for Visual Processing of Digital Video*, ACM Multimedia '94, San Francisco, October 1994.
- [15] Wetherall, David, J., *An Interactive Programming System for Media Computation*, Technical Report MIT/LCS/TR-640, MIT, September 1994.
- [16] Brown, A., Wolt, B., *Digital L-band Receiver Architecture with Direct RF sampling*, In IEEE Position Location and Navigation Symposium, pp. 209-216, April 1994.