The VITA Radio Transport (VRT) protocol is an emerging standard for Software Definable Radio (SDR) applications. It was developed to provide interoperability between a diversity of SDR components by defining a transport protocol to convey digitized signal data and receiver settings. As such it provides an infrastructure to maintain sample-accurate alignment of signal data and discrete events between multiple receivers that are either collocated or separated by large distances.

This paper provides an overview of the standard and portrays the benefits of VRT in an example RF receiver and DSP architecture.

1. MOTIVATION FOR THE VRT STANDARD

Interoperability of communication radios has been a focal point for many SDR architectures. For example, the Joint Tactical Radio Service (JTRS) defines the next generation receiver architecture to provide both voice and data interoperability between military services and also emergency civilian services. The interoperability of hardware and software is based upon an open architecture framework for the radio referred to as the Software-Compliant Architecture (SCA). A common core framework must be loaded on every JTRS-compliant radio which makes it interoperable with portable waveform definitions that can operate on any JTRS-compliant radio.

The VRT protocol addresses these requirements by defining a transport packet with unique signal data and signal context information. The signal data packet provides a broad range of data formats to support most digitizers and signal processing formats. The context packets convey sensors internal settings such as frequency, bandwidth, gain and delay and also convey spatial information. Both packet types support time stamping so that signal data from multiple receivers can be time-aligned to enable coherent and synchronous processing. With these features the VRT standard makes it possible to correlate information from a diversity of radio providers to enhance signal detection and geo-location capabilities. It thus eliminates dependency upon a single source for receiver and DSP equipment.

1.1 VRT Packet Structure Provides Interoperability

Interoperability of communication radios has been a focal point for many SDR architectures. For example, the Joint Tactical Radio Service (JTRS) defines the next generation receiver architecture to provide both voice and data interoperability between military services and also emergency civilian services. The interoperability of hardware and software is based upon an open architecture framework for the radio referred to as the Software-Compliant Architecture (SCA). A common core framework must be loaded on every JTRS-compliant radio which makes it interoperable with portable waveform definitions that can operate on any JTRS-compliant radio.

The VRT standard significantly differs from JTRS since it addresses the interoperability of receivers based upon a common packet protocol that is independent of the signal type or waveform type. It does not define a software framework, but rather defines a packet framework to convey signal data and receiver settings independent of the type of signals and/or waveforms being observed. Unlike the SCA standard, the architecture of the signal processing devices are not defined in the VRT standard and thus the equipment provider is free to define their own architecture based on a variety of technologies including application-specific integrated circuits (ASICs), field-programmable
gate arrays (FPGAs), digital signal processors (DSPs), and general-purpose personal computers. It can be used in conjunction with JTRS to enhance its capabilities or it can be deployed for many other radio applications such as signal surveillance, radar, Electronic Warfare (EW) and communication applications that do not use JTRS waveforms. In the past these applications used custom proprietary architectures where a specific radio architectures and components were developed for each unique application and implementation. This made it very difficult and costly to upgrade, to provide scalable architectures and to add new features. With the performance capabilities of today’s technology there is no need to have custom hardware for most applications. The capabilities of modern radios and signal processors make it possible to develop generic products to meet the requirements of all these applications. They can either be configured at the factory for the specific application or dynamically changed in the field as the mission requirements dictate. The commonality between these types of SDR radios include:

- One or more high performance analog tuners;
- Integrated digital receiver functions including digital down conversion (DDC), channelization, digital spectrum processing, and signal detection;
- Dynamic routing of signals within a receiver to the digital receiver resources;
- Sample-accurate time stamping of data;
- Ability to send one or more of the digital receiver channels out over an industry standard physical link, most often serialized clock and data.

The following section provides an overview of how the VRT transport protocol handles the diversity of receiver applications and architectures previously described.

2. OVERVIEW OF VRT PACKET FEATURES

The VRT standard resolves the dilemma of interoperability among SDR applications by providing a rich set of features for signal data packets and context data packets that can be used for a wide range of applications. Some of the key features of VRT include:

- A transport layer definition that can be transparently layered upon standard link interfaces such as Gigabit Ethernet, S-FPDP, RapidIO, USB, Aurora and most any link interface [2][3];
- Separate packets for signal data and context information to optimize throughput;
- Sample-accurate timestamping of signal data instrumental for direction finding (DF), time difference of arrival (TDOA), beamforming, and other emitter localization techniques;
- IF data packets supporting a wide range of digitized sample types: 1 to 32 bits, real, complex, and floating point;
- Context packets to convey a comprehensive set of receiver attributes such as frequency, bandwidth, gain, delays, sample rates, geolocation, and inertial navigation parameters;
- Sample-accurate timestamping of context events such as changes to receiver settings/status;
- Class codes to encapsulate all the options of a VRT packet into a single 32-bit field with class code relaying how to decode the packets to the device receiving the packets;
- Stream Identifiers (SID) to associate packets from the same signal source providing a multiplexing capability across a link, to associate signal data packets with context packets, and to identify parent-child relationships between components in a receiver

The combination of these features in a standardized transport language is unique to the VRT standard and yields new capabilities for SDR architectures. For instance, a device implementing this standard can effectively convey signal data and convey receiver settings for a broad range of applications including communications, radar, EW, and others. This enables an SDR to become a multi-functional receiver that can simultaneously output unique data streams for the different functions it supports. This is of special value to DoD architectures where each unique functional requirement of a receiver is typically implemented in a custom radio for that application [5]. In many instances, all of these functions can be implemented in a single multi-functional SDR device reducing the size, weight, and power of the combined capability. Having an infrastructure based upon a common standard such as VRT also reduces development risk, schedule, recurring cost and life-cycle cost of a system architecture.

2.1 Packet Approach to System Synchronization

VRT provides the interconnection of radio system components via data structures instead of direct wiring. This allows system components to be synchronized in time using modern gigabit data distribution techniques rather than a directly wired connection. Hence spatially distant systems can cooperate with signal processing to facilitate such isochronous applications as TDOA, synthetic aperture radar, DF or various beamforming applications. For example, traditional SDR system configurations typically included ribbon cable interconnection methods to allow digitized data, sample clock and synchronization signals to be distributed among system components. Many of these systems used proprietary schemes for connection, data format and synchronization logic. VRT standardizes the
methods used for data transfer and system synchronization. It eliminates the need for parallel interfaces for sample-accurate synchronization by providing sample-accurate time-stamping of signal data, sensor settings and external events.

2.2 IF Data Packets Convey Signal Information

Both IF Data packets and Context packets are integral to VRT, helping provide the standard’s interoperability benefits.

Figure 1 shows the form for the IF data packet. The first word is a header that is common to both the IF data packet and the context packet. (See Table 1 for a summary of IF data packets and context packets.) The header contains the packet type, option bits for the extended header and trailer, a rolling counter to ensure proper reception of all packets, and the packet size. The header is followed by optional extended header words in both packet types, which include the stream identifier, the class identifier, the integer time-of-day timestamp and a 64-bit fractional-seconds timestamp. In the IF data packet, the header and optional extended header are followed by the signal data payload and a 32-bit trailer word as shown in the Figure 1. In the context packet, the timestamp field location is followed by a 32-bit context indicator field and the selected context fields.

2.3 Data Items Fields Support Complex Signal Sample Types

In an effort to reduce the ambiguity that arises from disparate data formats from different vendors, VRT defines a rich set of data format standards ranging from one-bit fixed-point to 64-point complex and several floating point formats. In addition, VRT defines a signal data format construct called the item packing field, which provides channel tags and event flags with sample accurate alignment. Figure 2 depicts the item packing field; the data sub-field in the diagram is effectively the same as a signal sample.

Example of the range and accuracy of several fields are shown in Table 1. Even though these fields provide a large range and fine accuracy, their use will not significantly impact the link’s bandwidth since the context packets are sent only when a change in the context information is available. Thus the context packets will be sent at much lower rate than the signal data packets.
Table 1. Sample of VRT Context Fields Range and Accuracy

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Maximum Range</th>
<th>Minimum Range</th>
<th>Resolution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time Stamp</td>
<td>136 years</td>
<td>Present Time</td>
<td>1 picosecond or 1 sample</td>
</tr>
<tr>
<td>Frequency and Bandwidth</td>
<td>+8.79 teraHz</td>
<td>-8.79 teraHz</td>
<td>0.95 microHz</td>
</tr>
<tr>
<td>Gain or Power</td>
<td>+256 dB (or dBm)</td>
<td>-256 dB (or dBm)</td>
<td>1/128 dB</td>
</tr>
<tr>
<td>Sample Rate</td>
<td>+8.79 teraHz</td>
<td>0 teraHz</td>
<td>0.95 microHz</td>
</tr>
</tbody>
</table>

2.4 Information Classes and Class Documentation Provide Interoperability

A powerful aspect of VRT is the introduction of a standard documentation mechanism to allow the proper interpretation of system functionality from vendor to vendor. This is facilitated by the introduction of the concept of information classes.

An information class refers to a set of constituent data and context packet streams related to an associated application. The class documentation requirement allows standard specification of all aspects of a system or system components so that VRT system integrators can collect specifications from VRT component vendors and understand the precise operation of the aggregate system.

The information class allows a complex set of system interdependencies to be specified so that a system block diagram can be ascertained simply by interpreting the information class documentation.

3. EMERGING VRT-BASED SDR COMPONENTS

DRS Signal Solutions, Pentek and other SDR providers are beginning to release products based upon constructs of VRT. DRS offers a suite of receiver products in the HF and VHF/UHF frequency ranges that provide digitized VRT output packets as a standard feature. These products utilize a variety of digital interfaces including Gigabit Ethernet, Rapid-IO, S-FPDP, USB and Aurora as the link layer interface that VRT is layered upon. They come as both chassis mounted cards, such as VME or VXS [4], and man-portable modules.

To present an example implementation, the DRS SI-9147 dual-channel VHF/UHF VXS tuner is described, with respect to the types of VRT signal data packets it can source, along with the Pentek 4207 VXS DSP card. Integrating the VRT protocol into the product provides the key capabilities to manage the different types of signal data packets available in a product, to identify the DSP process from which the packets came from in the radio, to convey the signal routing through various DSP elements, and to relay the delay of the signal route. This information is critical for any geolocation algorithm such as DF, beamforming, radar or TDOA.

Without a VRT type of mechanism each vendor must provide a custom proprietary mechanism to identify and manage the different signal streams from a receiver, making it more difficult and costly to develop applications interoperable with different receivers.

The SI-9147s have individual synthesizers for each of its two RF tuner channels, enabling each to be individually tuned or multiple channels to be coherently tuned for DF and beamforming applications. Dual digitizers are built into the VXS single-slot module that provides 16 bits at an 80 MSPS sampling rate. The output of each ADC is fed into an array of FPGAs that route the data to delay memory or to one of the 36 ASIC-based DDCs as shown in Figure 6. The DDCs are individually controllably having 32,768 different decimation settings to provide output bandwidths from 17 MHz to 1 kHz. With this basic configuration, dozens of different signal data packet options can be selectively chosen. Loading the FPGA with demodulation and other advanced DSP capabilities can easily lead to hundreds of combinations of outputs available from a single receiver.

![Figure 6. DRS Dual-Channel VXS Receiver Enabled with VRT Signal Data Packets.](image)

The Pentek 4207 is a single or dual Freescale MPC8641 Altivec™ PowerPC processor board featuring a host of I/O support including a built-in dual optical Fibre Channel interface, and dual Gigabit Ethernet interface. It also includes two PMC module sites, both equipped to accept XMC (switched-fabric PMC) modules.

The Pentek 4207 also features up to 4 GB DDR2 SDRAM for program and data memory, and a Xilinx Virtex-4 FPGA to support gigabit serial fabrics or for custom user programming. The Pentek 4207 is optimized for embedded applications that require high-performance input/output (I/O) processing and processing such as wideband data acquisition and software radio.
The interoperability of the DRS SI-9147 and the Pentek 4207 has been developed, tested and demonstrated as shown in Figure 8. This simple architecture demonstrates the ability of the cards to use VRT as a common transport protocol to packetize and time stamp signal data from a receiver, to use Serial Rapid IO as a link layer interface, and to send it over multiple lanes of the VXS backplane at 3.125 Gbps per lane.

A typical architecture for these types of SDRs uses four SI-9147s to provide eight RF channels that send data to four or more Pentek 4207 DSP cards as shown in Figure 9. The architecture uses a Serial Rapid IO switch card to enable dynamic routing of signals from any of the receiver cards to any of the signal processing cards and/or between signal processing cards. The number of combinations of routing signals through a signal processing flow has now increased an order magnitude from the hundreds described for just a single SI-9147 to thousands of different combinations.

The system shown in Figure 9 can be used for multifunction SDR architectures that can simultaneously implement radar, communications, EW and surveillance functions. Utilizing a high-performance fabric, the receiver and DSP resources are dynamically allocated as needed to different functions, which will dynamically change based upon the mode of operation and the priority of each function. The typical modes of operation are search, direction finding, beamforming and set-on receiving. Table 2 shows these modes of operation with respect to SDR functional applications. Table 3 demonstrates the number of resources that may be dynamically switched in for each function.
The signal data packet was defined only from the perspective of a receiver sending data out. The framework of this packet is being used to define a packet to send data to an exciter for transmission. The context packet was also just defined to convey the state of a radio. Its framework is also being used to define a packet to control receivers and upconverters.

5. CONCLUSION

Dramatic changes are occurring in high-performance RF sensor and signal processing architectures that enable general purpose COTS components to now be used where in the past only custom stove-piped components were viable. The VRT standard provides a data transport infrastructure to develop products that support these revolutionary architectures and move the industry from custom application-specific components and designs to multifunction dynamically configurable architectures. The use of VRT constructs makes it possible to develop architectures that are not locked in to a single supplier.

The VRT standard defines a transport data protocol that provides interoperability for many SDR radio applications independent of physical link, application and the internal architecture of the radio. The use of the standard lowers the risk, the schedule, cost and life-cycle-cost of developing new applications.

6. REFERENCES
