IMPLEMENTATION OF A SHARED RESOURCE MODEL IN A TACTICAL RADIO SYSTEM

Lee Pucker
Spectrum Signal Processing, Inc.
Burnaby, B.C, Canada

ABSTRACT

Traditional tactical radio architectures are limited in their ability to scale up to support wideband waveforms operating beyond 2 GHz and to scale down to make more efficient use of processing resources when supporting narrowband channels. A move to a shared resource model in a tactical radio system addresses these issues by facilitating processor capacity assignment based on the specific needs of each instantiated waveform. A key enabling technology supporting this paradigm is the use of a packet switched fabric, such as RapidIO, that is capable of supporting the interprocessor connectivity required in a shared resource model while maintaining the performance necessary to support operation on a frequency hopped network. This paper examines the requirements for a tactical radio architecture employing a shared resource model, and explores some of the challenges inherent in the use of a switched fabric for this type of system.

INTRODUCTION

The JTRS Airborne, Maritime, and Fixed (AMF) Cluster’s pre-SDD request for proposal (RFP) identifies a number of key challenges that must be explored in defining a next generation software defined radio (SDR) architecture [1]. Among these challenges is the extension of the radio architecture to support wideband waveforms beyond 2 GHz, including both Advanced MILSATCOM waveforms and common data link (CDL) backbones, and the utilization of shared resources vs. dedicated resources, allowing multiple narrowband channels to operate on a common set of processing devices. This paper explores the requirements for a tactical SDR system that addresses these challenges by first reviewing the requirements that define a traditional tactical radio architecture and then exploring the differences between these requirements and those of a wideband radio system operating above 2 GHz. An architecture is then proposed that extends the traditional tactical radio model to support a mix of radio applications above and below 2 GHz on a common platform. This includes a brief exploration of SCA support for the proposed architecture as well as a discussion on some of the key challenges that must be addressed in adopting this type of architecture in a tactical radio system.

REQUIREMENTS FOR THE BLACK SIDE ARCHITECTURE IN A TRADITIONAL TACTICAL RADIO SYSTEM

Tactical radio networks are typically limited to peer-to-peer or broadcast communications with no centralized gateway acting as a base station or hub [2]. Payload data rates in a these type of networks are relatively constrained, usually 1 megabit per second or less [3], with wireless networking often supported by allowing each radio in the network to act as a repeater or bridge for other radio nodes. This type of ad-hoc structure allows the creation of dynamic, “infrastructureless” networks supporting potentially hundreds of mobile units while ensuring survivability of each network in the event that one or more nodes fail.

This paradigm implies that only a single user channel is supported on each active carrier. Thus, although a number of the waveforms supported in a tactical radio environment, such as Link-16 [4], may incorporate a multiple access scheme to allow multiple user channels to occupy the same carrier frequency, each tactical radio terminal associated with a given multiple access channel will only utilize a single logical access channel at any given time. Tactical radio systems can thus be architected with processing resources statically dedicated on a per channel basis. In this model, the failure of any one channel will not impact the operation of any other channel in the overall radio architecture, optimizing the availability of communications to the war fighter.

The architecture of a tactical radio system is further constrained by the synchronization requirements of the various supported networks. For example, waveforms such as SINCGARS [5] and LINK-16 employ frequency hopped spread spectrum techniques for interference suppression and transmission security. The windows for transmission and reception of signals in these types of waveforms are tightly controlled, and as such synchronization with the network requires precise knowledge of the temporal parameters of the waveform at the A/D and D/A converters within the SDR platform. This in turn requires very tight coupling between the RF front end and the first modem processing stage to ensure proper temporal alignment with the hopped network.

To illustrate this, consider the block diagram shown in Figure 1 representing the physical layer implementation of a sample frequency hopped waveform [6]. On the receive side, this architecture digitizes the IF signal from the RF front end in the converter assembly, and then "basebands" the signal using a narrowband digital down converter. Hop "bursts" are then acquired from the baseband signal and forwarded onto channel processing for demodulation and decoding. This process is reversed on the transmit side, with payload data being encoded and modulated in the channel processor and then staged as a hop "burst" which is forwarded to the converter assembly for transmission at the appropriate time.

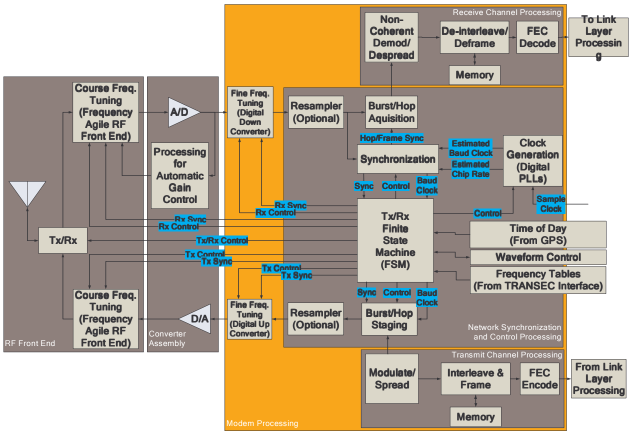


Figure 1: Block diagram illustrating the physical layer of an sample frequency hopped signal

Absolute time in this example is based on the sample clock of the A/D and D/A converters, with relative time tracked by tagging the signal data with an associated sample number. Thus network synchronization can occur through an acquisition and tracking algorithm specific to the waveform, with synchronization timing tracked by the waveform finite state machine (FSM), based on sample number, to identify the start and stop of each transmit and receive burst. This synchronization method requires access to the sample clock by the waveform FSM, and precise knowledge of the latency from the RF input to the synchronization block, measured in samples, to correctly define the processing delay, again in samples, associated with the synchronization timing. This delay can then be used in conjunction with similar knowledge of the latency from the burst-staging block to the RF output to ensure that a burst is correctly transmitted within the assigned window.

Support for frequency hopped waveforms such as this also requires special consideration be given to controlling the transmit and receive frequencies of the RF front end. Frequency agile waveforms generally break down into two distinct categories: slow hopping waveforms, where multiple data symbols are transmitted during each hop period, and fast hopping waveforms, where only a single baud is transmitted per hop [7]. Each hop can be characterized by a duty cycle that is split between the active transmission time and the setup time for the next hop. For many waveforms support for frequency agile operation can be achieved by creating and maintaining a local hop table in the RF front end, and then sending a hop sync whenever necessary to step to the next entry in the table.

However, for many waveforms the information necessary to calculate the transmit or receive carrier frequency for the next hop or set of hops may be sent in the header information of the current hop. For these types of waveforms the hop frequency can be updated as often as once per hop, requiring the control information to be sent and the new frequency to be set up during the tuning or guard period specified for the waveform. The header information associated with each hop is demodulated and decoded independent of the payload data, with a high-speed control path established between the waveform FSM and the RF front end.

The turn around time for processing the header data in these types of waveforms to generate the next hop frequency illustrates an overall need to control the latency within the modem processing of the tactical radio. This is especially true for connection-oriented waveforms, where an acknowledgement or response must be transmitted within a specified time period for each received data packet. This requires that the latency between the waveform components operating on the various processing devices within the modem architecture be deterministic, with the time to move data between components allowed to vary so long as the transport delay is guaranteed within a time period specific to that waveform.

TRADITIONAL BLACK SIDE ARCHITECTURE SUPPORTING THESE REQUIREMENTS

Software defined radio architectures supporting these requirements are well documented [8, 9, 10, 11, 12], with the black side architecture typically following a dedicated resource model. In this model, illustrated in Figure 2, the RF and modem resources are dedicated on a per channel basis, with multiple channels supported through the use of duplicate RF/Modem processing subsystems. The converter assembly for each channel may be implemented as a part of the modem processing or a part of the RF transceiver unit.

The modem architecture itself typically incorporates a FPGA for network synchronization and control processing, and either a GPP or DSP for the transmit and received channel processing, depending on the size, weight, and power limitations imposed upon the modem subsystem. When a GPP is incorporated into the modem architecture, waveform link layer processing may be supported directly by the modem. Conversely, when a DSP is employed in lieu of a GPP, a black side "host" may be utilized to provide a common GPP outside of the modem architecture that is shared across all modem channels. The tight coupling required between the modem FPGA and the RF front end to maintain network synchronization is supported through direct, independent connections for data and control. Similarly, the deterministic latency required between the waveform components is facilitated by direct connections between the processing elements of each modem channel.

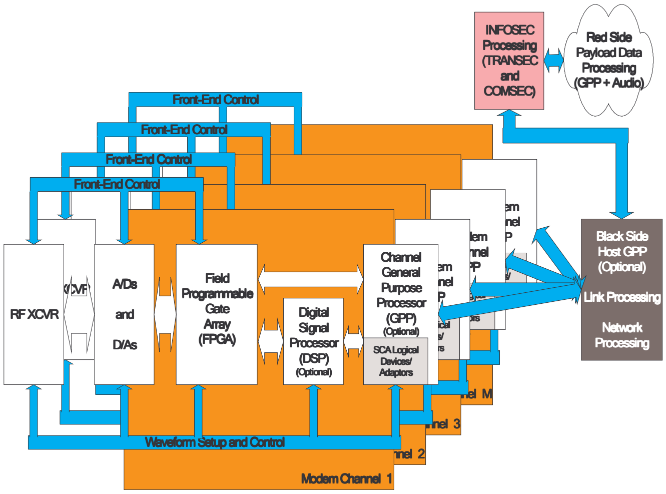


Figure 2: Typical black-side architecture for a multi-channel tactical radio system

LIMITATIONS WITH TRADITIONAL TACTICAL RADIO ARCHITECTURE

Inherent in this architectural model is a limitation in the ability to appropriately scale to support a waveform application's *specific* needs. Each channel in a dedicated resource architecture must be "over-designed" to support the most complex waveform that will operate on that channel. As a result, a modem architecture designed to support a single wideband waveform, such as the new Wideband Networking Waveform (WNW), may have the I/O and processing capacity to support a number of simultaneous channels of a less complex waveform, such as VHF FM. However, because 100% of the resources of a modem channel are allocated when any waveform is instantiated on that modem in this model, the processing resources may be significantly underutilized when running the less complex waveform.

This inability to scale is especially troubling when attempting to expand the role of the tactical radio to support advanced waveforms operating above 2 GHz. These waveforms differ from the more traditional tactical waveforms in two key areas:

- A communications terminal supporting Advanced MILSATCOM or high data rate CDL applications will often support multiple simultaneous user channels per carrier [13]. For example, the SMART-T terminal provides 16 simultaneous MDR capable channels [14]. As such, the channel based architecture typically employed in a traditional tactical radio system may not be suitable for applications operating above 2 GHz since the front end processing within the terminal's modem structure must support multiple backend processing "channels". Further, these waveforms often operate in a full duplex mode, requiring the architecture to support simultaneous transmit and receive processing.

- Waveforms operating above 2 GHz often sustain much higher data rates, and subsequently much higher bandwidths, than are supported by a traditional tactical radio architecture. For example, the MILSATCOM Advanced EHF Waveform supports payload data rates of 8 Mbps per user channel, with multiple user channels supported on each RF carrier [15]. Advanced communications systems of this type often involve composite payload data rates of 100's of megabits per second, requiring terminal architectures supporting instantaneous RF channel bandwidths of well over 200 MHz [16,17]. The front-end processing of these types of systems often exceed the capacity of any single signal-processing device, requiring the front-end waveform processing device, requiring the front-end waveform algorithms to be distributed across multiple processors operating in parallel to achieve the required throughput. This differs from the more traditional tactical radio architectures where lower data rates allow for more channelized processing.

ADDRESSING THESE LIMITATIONS THROUGH SHARED RESOURCE PROCESSING

Addressing these limitations in an efficient manner requires a fundamental change in the way in which a tactical radio is architected. One paradigm which allows the processing resources associated with each modem channel to scale to a waveform's specific needs is to dynamically assign processor resource capacities from a "processor pool" (see Figure 3). Thus, if a waveform only requires a portion of an FPGA and a portion of a

GPP for the modem processing functions, the remaining capacity of these devices, as well as the capacities of any other devices associated with a given hardware implementation, need not be allocated, making them available for other waveform applications. Conversely, if a waveform requires multiple FPGAs operating as a "processing array" to support advanced waveforms above 2 GHz, these can also be allocated from the pool, potentially decreasing the total number of channels that can operate on the radio at any given time, but providing support for extremely wideband applications. This model can be further extended to allow each processor to support waveform components for multiple RF channels maximizing the efficiency of the overall processing.

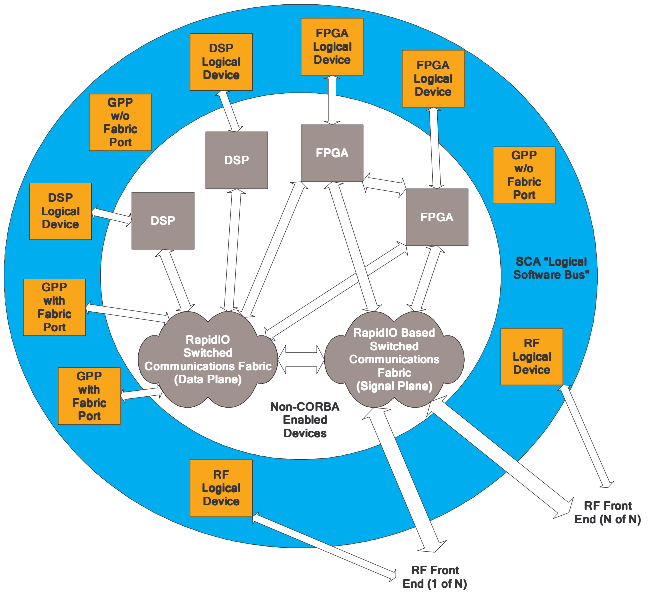


Figure 3: Creation of a processor pool through the use of a RapidIO switched fabric in an SCA compliant radio

A number of architectural elements are required in the tactical radio system to support this type of shared resource processing. First, a minimum unit of scalability must be established within the overall modem architecture. This would typically take the form of a "modem processing engine" capable of supporting some level of base waveform requirement. Next, a communications infrastructure must be established to support the required hard real-time connectivity between the modem processing engines (the data plane), and between the modem processing engines and the various RF front ends (the signal plane) [18]. These elements are explored separately in the following sections.

THE MODEM-PROCESSING ENGINE

One approach to creating a processor pool would be to create a processing engine for each type of processor required in the radio platform. Thus a modem architecture requiring the use of FPGAs, DSPs, and GPPs would have independent FPGA based processing engines, DSP based processing engines, and GPP based processing engines. While this model allows the number and type of processors to be optimized for a radio platform, it is limited in its ability to scale down in that the minimum deployable system would require at least three processing engines in the overall modem architecture. This may be unacceptable depending on the allocated space for the modem processing, and as such, a modem-processing engine would more generically incorporate the number and types of processing resources required to provide a single modem channel consistent with the traditional tactical radio model. This allows the radio architecture to be scaled down to a single channel, if necessary, with all of the radio resources provided.

The processing resources incorporated on the modem-processing engine for this model must each support "partitioned operation". This allows the waveform components from multiple independent waveforms to operate concurrently within the processing device, with each of these waveform components capable of being set up and torn down dynamically without impacting the components associated with other waveform channels. While this capability is fairly easy to achieve for a GPP through the use of a partitioned real-time operating system, it is more difficult to achieve for devices such as FPGAs due to the way in which these devices are architected and the lack of a supporting infrastructure.

Partial reconfiguration of an FPGA device in this manner is generally specific to the device. For example, Xilinx's Virtex II and Virtex II Pro devices organize their internal resources in columns [19]. Each column or portion of a column can be independently reconfigured using the devices' configuration port. The regular structure of these columns allows waveform resources to be created that are relocatable within the device. Thus, capacity allocation within the device can be managed through an SCA enabled logical device with waveform resources loaded and unloaded, as necessary, based on the availability of the required column structure.

I/O support within the device for this paradigm is provided through a channelized I/O structure incorporating a fixed interface to the outside world [20]. Logical channels are established between the various waveform resources and the I/O structure dynamically based on the specific needs of each instantiated waveform component. This structure also supports intercomponent connectivity within the device to maximize the modularity of the waveform architecture.

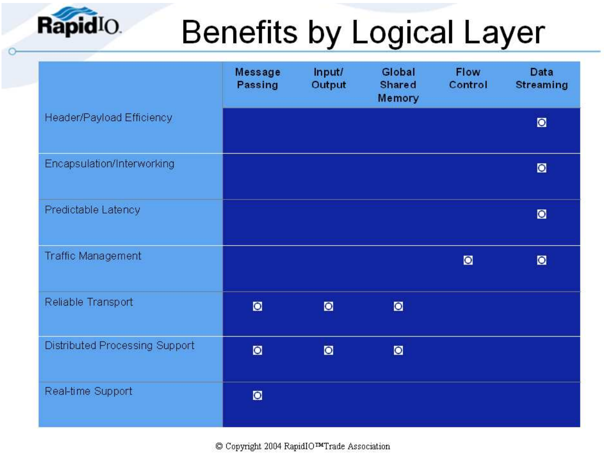
DATA PLANE SUPPORT THROUGH A SWITCHED FABRIC INTERCONNECT

Once the modem-processing engine is defined, a communications infrastructure must be created to support the required connectivity within the overall processing architecture. Within the processing pool, this implies a need for a data plane providing "any to any" connectivity between the processing devices of the modem-processing engines, allowing the number and types of processing devices associated with any given waveform channel to be dynamically assigned during waveform instantiation. This requirement is best addressed through the creation of a data plane based on a switched fabric interconnect such as RapidIO. In this type of architecture, data is routed between processing devices based on a destination address embedded in each transmitted packet, with the switched fabric providing a transport layer capable of end-to-end routing and multiple links. As such, switched fabric technologies provide for efficient support of the logical channels necessary to interconnect the waveform components associated with each instantiated waveform application across multiple disparate processing elements throughout the overall radio architecture.

Support for the deterministic latency through the switched fabric interconnect required for data plane communications implies a need to allocate fabric capacity on a per channel basis as a part of the overall setup of each waveform. Properties that must be allocated include both sustained bandwidth and end-to-end transmission latency. A number of switched fabric architectures provide support for these features. For example, RapidIO offers extensions to its base protocol stack to include flow control and data streaming to provide for traffic management and predictable latency [21]. Features associated with these extensions to the RapidIO protocol are summarized in Figure 4.

SIGNAL PLANE SUPPORT THROUGH A SWITCHED FABRIC INTERCONNECT

The use of a switched fabric can also extend to the signal plane to provide for an interconnect architecture switching between the various RF front ends and modem processing engines. This architecture must support not only the one RF to one modem paradigm utilized in a traditional tactical radio model, but also the one-to-many connectivity required to support distribution of IF or baseband signal data from a single RF front end to multiple back end processing engines for extremely wideband applications, and the many-to-one connectivity required to allow multiple RF channels to be processed on a single modem-processing engine. A packet switched fabric such as was identified in the previous section lends itself well to these topologies, supporting both reconfigurable data flows and high speed low latency connections.



RapidIO. Benefits by Logical Layer

	Message Passing	Input/ Output	Global Shared Memory	Flow Control	Data Streaming
Header/Payload Efficiency					●
Encapsulation/Interworking					●
Predictable Latency					●
Traffic Management				●	●
Reliable Transport	●	●	●		
Distributed Processing Support	●	●	●		
Real-time Support	●				

© Copyright 2004 RapidIO™Trade Association

Figure 4: Benefits of extensions to RapidIO specification currently under development [22]

Use of a switched fabric, however, in support of the signal plane requires additional consideration be given to network synchronization. Switched fabric interconnects generally operate asynchronous to the sample clock, and as such have no inherent mechanism for accommodating the fixed latency necessary to maintain synchronization with a frequency hopped network. This issue can be addressed by passing the sample clock from each RF front end to the modem processing engine, and then tagging the transported samples with a "sample count" that is maintained on both sides of the asynchronous fabric. This count can be used by the waveform FSM to precisely determine the latency through the signal plane by comparing the sample number embedded in the received data stream with a locally maintained sample number operating on that same clock. This technique allows for support of multichannel synchronization as well, allowing packets from different channels to be re-aligned for MIMO or beamforming applications based on the embedded sample count.

Use of a switched fabric for the signal plane transport also allows for control of the RF transmit and receive frequencies through the insertion of hard real time control packets into the reverse path between waveform FSM and the RF front end. As with the signal data, this must have a dedicated capacity allocation to ensure that the hop rate for the specific waveform can be maintained. The tight coupling required to maintain the hop synchronization can be achieved through a direct wire line connection from the waveform FSM to the RF front end. This model can be extended to provide support for interfacing to multiple RF front ends through a circuit switched architecture, which maintains the “zero latency” point- to-point connectivity required for this type of interconnect while allowing any FSM to connect to any RF front end.

PROPOSED SHARED RESOURCE TACTICAL RADIO ARCHITECTURE

A tactical radio architecture supporting shared resource processing is illustrated in Figure 5. In this architecture, 1 of N RF front ends connect to 1 of M modem processing engines, with M typically smaller than N for a given deployment model. The RF front ends include a combination of both traditional narrowband tactical channels and wideband channels supporting advanced MILSATCOM and common data link waveforms. An FPGA is placed in the RF front-end of this architecture to facilitate zero latency IF processing, such as the processing associated with the front-end automatic gain control functions and I/Q balancing.

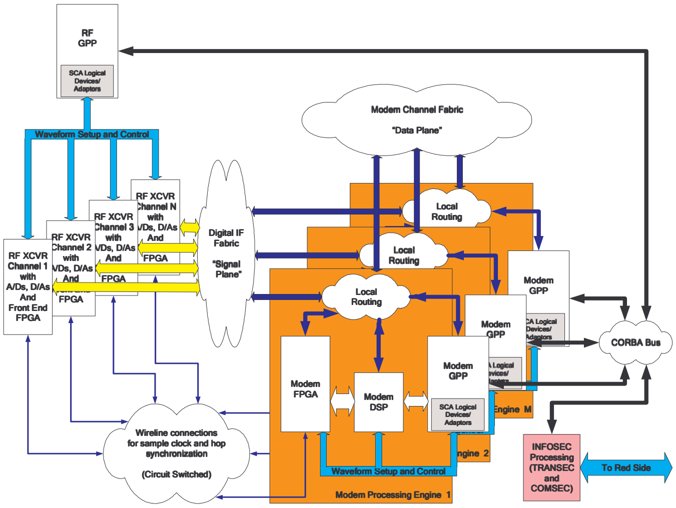


Figure 5: High level black side architecture of a shared resource tactical radio incorporating independent signal, data, and control planes.

It should be noted that the use of a switched fabric between the RF and modem-processing has the added benefit of allowing these two subsystems to be hosted in geographically disparate locations. For example, in a shipborn environment, the RF subsystem could be placed on a communications tower with the modem processing, interconnected via the switched fabric, located in the radio shack.

SCA CORE FRAMEWORK SUPPORT FOR A SHARED RESOURCE MODEL

SCA support for a shared resource architecture requires that the packet switched fabric be provided for through the logical devices associated with each processing element, as illustrated in the simple application shown in Figure 6. Each logical device defines one or more connections between the processing device it represents and the switched fabric through which it wishes to communicate. Connections between application components are then made via the logical devices on which they are loaded. These concepts are explored in detail in [12] and the reader is referred there for additional information.

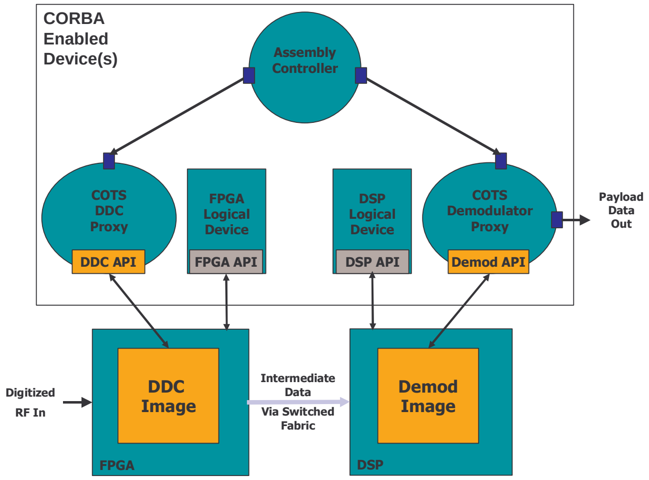


Figure 6: Sample application using packet switched fabric

KEY CHALLENGES ASSOCIATED WITH THE USE OF A SWITCHED FABRIC INTERCONNECT IN A TACTICAL RADIO ARCHITECTURE

While numerous advantages can be achieved in processing efficiency and waveform support by moving to a shared resource architecture there are a number of challenges that must be addressed as well before this type of solution is viable in a deployed environment. Chief among these is the power utilization and cost inherent in the architecture. The additional components necessary to support the switched fabric structure represent an increase in cost per unit and power that is dissipated within the radio. Over time it is expected that technology advances will obviate these issues. For now, however, this type of architecture is only appropriate for programs were the benefits accrued by the architecture outweigh these costs.

Availability is also an issue. Any packet switch utilized in this architecture potentially acts as a single point of failure in the system since data from multiple channels flows through that switch. Mechanisms must be employed, therefore, to maintain high availability in the system such as the use of a mesh fabric, eliminating the switch as a single point of failure, or the incorporation of a redundant fabric switch with failover capabilities.

This issue also extends to the processing devices in a shared resource environment. In a dedicated resource model, if a processor fails, it only effects the operation of a single communications channel. However, in a shared resource model, where a single processor may be partitioned to support multiple simultaneous waveforms, then the failure of any one device could impact multiple channels. This issue is addressed in the commercial telecommunications world by instantiating a redundant copy of each waveform operating on the platform to guarantee 99.999% uptime performance. This type of solution significantly increases the size, weight, power, and costs of the radio platform and as such it may be necessary to identify an alternative solution given the constraints in a specific deployed environment.

CONCLUSIONS

The adoption of a shared resource model compensates for the limitations inherent in a traditional tactical radio architecture. Implementation of this type of model requires that resources are pooled using RapidIO or a similar switched fabric interconnect. These enabling technologies support the "any-to-any" data flows required in a shared resource model while allowing, through extension of the basic protocol stack, the deterministic, low latency connectivity necessary to support synchronization and control on frequency hopped networks. Support for the SCA in this type of architecture can also be provided for within the confines of the SCA 2.2 specification, allowing a platform supporting a switched fabric interconnect to be fully compliant with the principles of the JTRS program. While a number of technical challenges must be addressed with respect to the power, cost, and availability requirements associated with a military tactical radio before a shared resource architecture can be fully deployed, the use of a shared resource architecture accrues significant advantages in both the number and type of waveforms that can be supported on the tactical radio platform.

REFERENCES

1. Request for Proposal (RFP) for Successful Completion of Pre-System Development and Demonstration (pre-SDD) for the Airborne and MaritimelFixed Station Joint Tactical RadioSystem (JTRS) Program, RFP FA8709-04-R-OOO2, http://herbb.hanscom.af.mil/esc_opps.asp?rfp=R712
2. Zygmunt J. Haas and Siamak Tabrizi, "On some challenges and design choices in ad-hoc communications", *MILCOM 1998 - IEEE Military Communications Conference*, no. 1, October 1998, pp. 187 - 192
3. JTRS Joint Operational Requirements Document Version 3.2, http://herbb.hanscom.af.mil/esc_opps.asp?rfp=R712
4. Link-16 Summary, May 1997, http://prodevweb.prodev.usna.edu/SeaNav/NS40x/NS401_old/introduction/html/linksum.html
5. Janne J. Lehtomäki, "Maximum based detection of slow frequency hopping signals", *IEEE Communications Letters*, no. 5, May 2003 pp. 201-203
6. Reference Designs for Tactical Military Communications, Spectrum Signal Processing Inc.
7. Don J. Torrieri, Principles of Secure Communication Systems, Artech House, May 1992
8. Dave Dohse, Lynn Bush, Greg Osborn, and Eric Christensen, "Successfully Introducing CORBA Into the Signal Processing Chain of a Software Defined Radio", *COTS Journal*, Jan 2003, pp.32 - 37
9. Peter G. Cook and Wayne Bonser, "Architectural Overview of the SPEAKeasy System", *IEEE Journal on Selected Areas in Communications*, vol. 17, no. 4, April 1999, pp. 650 - 661
10. Arthur Akhurst, "Scalability Aspects of the SRA", Software Defined Radio Forum Contribution, 25 May 2001, http://www.sdrforum.org/2001_docs.html
11. David Murotake, "JTRS SCA Platform Hardware Scalability", Software Defined Radio Forum Contribution, 13 Nov 2001, http://www.sdrforum.org/2001_docs.html
12. Lee Pucker and Geoff Holt, " Extending the SCA Core Framework Inside the Modem Architecture of a Software Defined Radio", IEEE Communications Magazine, March 2004
13. Richard A. Williams and Heywood I. Paul, "Military X and KA - bands media mix analysis for the wideband gapfiller satellite system", MILCOM 2000 - IEEE Military Communications Conference, no. 1, October 2000, pp. 1214 - 1218
14. Wade A. Leonard, "MILSTAR and DOD teleport", MILCOM 2000 - IEEE Military Communications Conference, no. 1, October 2000, pp. 1181 - 1186
15. John Mogor and John Wilson, "Wideband Communication Options for the E6-B", http://www.mitre.org/support/papers/tech_papers99_00/wilson_wideband/wilson_ideband.pdf
16. Michael A. Rupar and David R. Beering, "A Ka-band high data rate shipboard satellite terminal", MILCOM 1999 - IEEE Military Communications Conference, no. 1, October 1999, pp. 1490 - 1494
17. Len P. Calabretta, Matthew Ryerse, and Lawrence Goeller, " X-Band, Ka-Band, and Future MILSATCOM Requirements: and Outing in Trade Space", MILCOM 1998 - IEEE Military Communications Conference, October 19
18. Sam Fuller, "RapidIO Update, Bus Board and Fabric 2004", http://www.busandboard.com/pdfs-20040119/1500-fuller.pdf
19. Xilinx, "Two Flows for Partial Reconfiguration: Module Based or Difference Based", http://www.xilinx.com/bvdocs/appnotes/xapp290.pdf
20. Spectrum Signal Processing, "Flexible Channel System", Patent Application (Patent Pending)
21. RapidIO Trade Association, "RapidIO Solves the Communications Fabric Conundrum with RapidFabric™ Extensions http://www.rapidio.org/about/RapidFabric/RapidFabric_Presentation/RapidFabric_White_Paper.pdf
22. Eran Strod, "RapidFabricTM: The RapidIO Open Communications Fabric Initiative", http://www.rapidio.org/about/RapidFabric/RapidFabric_Presentation/Strod_Electronica-ESC_March_2004_without_quotes.pps