Satellite links and software defined radio for remote communications and sensor data gathering

Linda M. Davis  David Haley  André Pollok  Ying Chen

Institute for Telecommunications Research
University of South Australia
www.itr.unisa.edu.au
{linda.davis, david.haley, andre.pollok, ying.chen}@unisa.edu.au

Abstract—
Remote communications, remote sensing and sensor data gathering are critical in supporting activities across Australia’s environment, resource, defence and security sectors. Satellite links are an obvious choice for serving our vast land and marine territories, as well as international deployments.

This paper gives an overview of state-of-the-art technology in the form of software defined radio (SDR) technology for both the space and ground terminals of a satellite link. Like terrestrial applications, SDR in space offers possibility for cognitive and adaptive operation, multi-mode operation, radio reconfiguration, remote upgrade as well as the potential to accommodate new applications and services without hardware changes.

Index Terms—satellite, reconfigurable radio, software-defined radio, SDR, remote communications, situational awareness, remote sensing, data gathering

I. INTRODUCTION & BACKGROUND

In 2009, Australia’s defence white paper endorsed space-based platforms as key elements of our future defence force, providing capability for remote communications, command and control, as well as situational awareness via remote sensing and data gathering [1]. This recognises the suitability of satellite platforms, not just for coverage of Australia’s vast land and marine territories, but also in support of international deployments.

As allies, Australia and the United States share access to ground-based and space-based joint facilities. For example, Australia is a partner to the United States in the Wideband Global Satellite (WGS) system, which includes six (and potentially more) geostationary (GEO) wideband satellites. Operated by the United States, WGS provides broadband communications for fixed and large mobile platforms [1]. The first three satellites (Block-1) have been launched to date. The target date for launch of the second three satellites (Block-2) is late 2011 to early 2012. The WGS satellite payload can filter and route 4.875 GHz of instantaneous bandwidth, supporting communication links in the Ka-band and X-band, with data rates ranging from 2.1 Gbps to more than 3.6 Gbps [2]. The WGS is complemented by narrowband UHF low earth orbit (LEO) satellite capabilities primarily for mobile land forces. Australia will share capacity on the communications payload aboard Intelsat IS-22, scheduled for launch in 2012. These systems will enhance satellite capabilities currently provided by the Optus-C1 satellite [3].

Traditionally satellite payloads have implemented bent-pipe or transparent routing. That is, the satellite-based modem acts as a relay, receiving the analog signal from the earth-station, amplifying, then retransmitting to the ground-based terminal, or vice versa. Such simple physical layer implementation is limiting in terms of capacity, geographic coverage, configurability, policy-based security, and support for rapidly evolving applications [4]. With increasing digital processing capability being made available onboard the satellite, more sophisticated functionality is possible, including regenerative relaying, dynamic bandwidth resource allocation, packet switching, traffic aggregation / disaggregation and routing, satellite multicasting and broadcasting via spot-beams, and cooperative relaying [4, 5].

In terms of hardware capability and modem technology, space-based systems lag terrestrial technology due to need for space-hardening and flight-testing along with far-stricter size, weight and power (SWaP) constraints. Furthermore, in order to obtain return on investment in build, launch and mission, the operational service lifetime is long compared with terrestrial system deployments. Geostationary satellites are expensive (typically > US$ 200m to build) and have long mission design life (typically 15+ years). Smaller platforms such as LEO microsatellites can be deployed much faster (12-24 months) and cheaper (typically < US$ 20m) [6].

A recent technical innovation in satellite systems is the application of software-defined radio (SDR) payloads [6]. With large parts of the radio waveform defined in software, the radio waveform can be changed in operation through software control, and additional waveforms and modifications can be implemented with software changes, all without modification of the SDR platform (the combination of hardware and operating environment where the waveform application is running) [7]. SDR in space, like its terrestrial counterpart, offers the possibility of adaptive and cognitive operation, multi-band and multi-mode operation, radio reconfiguration, remote upgrade and the potential to accommodate new applications and services without hardware changes. This latter feature is particularly attractive in realising the return on investment in design, build and launch of satellite payloads. An additional benefit of the move
to SDR is portability of the software waveform applications. An application can be ported to different underlying SDR platforms, allowing for interoperability and staggered hardware upgrades. However, the flexibility and portability offered by SDR results in trade-off in implementation efficiency including size, power, and perhaps even performance [8].

SDR is the software ascendant evolution of reconfigurable radio. An early adopter of reconfigurable technology for space applications was Australia’s own FedSat microsatellite communications payload. FedSat was launched 2002 by the Australian Cooperative Research Centre for Satellite Systems, becoming the first Australian-built satellite to be launched since WRESAT in 1967. The FedSat communications payload utilised field programmable gate array (FPGA) components for baseband digital signal processing and included a code upload mode allowing it to be reprogrammed while in orbit [9]. FedSat was operational for over four years until its on-board batteries reached the end of their operational lifetime. From the same era, another example of FPGA-based reconfigurable satellite payload design is given in [10], where the reconfigurable computing platform was designed for mitigation of radiation effects rather than reconfigurable application.

The evolution from reconfigurable and reprogrammable devices to software reconfigurable (software defined) radio has been underpinned by advances in the enabling technologies, foremost analog-to-digital and digital-to-analog converters (ADCs, DACs), but also general purpose processors (GPPs), digital signal processors (DSPs) and FPGAs. For the most part, research and development of SDR has been driven by the demand for flexible and reconfigurable radio communications in support of military and public safety operations [7], including the Joint Tactical Radio System (JTRS) [11, 12].

The combination of SDR and satellite technologies is highly suited to defence applications in remote communications, remote sensing and sensor data gathering (i.e. collection of data from distributed remote ground terminals). Even in the case that the satellite communications payload is not SDR based (such as in bent-pipe links) the benefits of SDR may be realised to some extent through its use for ground-terminals and earth stations.

II. Software-Defined Radio

Despite the wide use of the term software-defined radio (SDR), there is no single agreed definition, primarily due to slightly varying perspectives of key players in the field (radio implementers, network operators, service providers etc). For example, the Wireless Innovation Forum [13], formerly known as the SDR Forum, defines SDR succinctly and generically as a “radio in which some or all of the physical layer functions are software defined” [14].

In an ideal SDR world, the analog front-end of the radio would be restricted to functionality that cannot be implemented digitally such as the antenna, radio-frequency (RF) filtering, receive and transmit amplification [15]. In other words, the aim is to perform the conversion between the analog and digital domains as close to the antenna as possible in order to maximise flexibility and reconfigurability. With current technology, such an ideal SDR is not yet practical or cost effective. Major challenges include high-resolution wideband sampling in a large range of carrier frequencies [16, Ch. 4], transceiver filtering and linearity [16, Ch. 2].

Traditionally space-based communications were dominated by hardware-based radios with no or only little reconfigurability. Due to the high cost associated with the design, deployment and operation of a satellite, the life cycle of a satellite payload is typically considerably longer than that of terrestrial transceivers [6]. As a result, the waveform implemented on the payload is often outdated within the lifetime of the satellite [17]. The adoption of SDR technology in satellite communications is highly attractive as it provides the flexibility to remotely reprogram the SATCOM equipment once bug fixes, improved algorithms or new/evolving communication standards are available [7]. Furthermore, the possibility to change waveforms enables multi-band or multi-mode operation of the SDR payload.

In the following sections, we will discuss the hardware and software architecture of an SDR.

A. Hardware Platform

As mentioned earlier, an SDR has a digital and an analog subsystem. The digital processing units available on most SDR platforms offered today are general purpose processor (GPP), digital signal processor (DSP) and field programmable gate array (FPGA) as illustrated in Fig. 1. The reconfigurability of these processing units provides the flexibility of the SDR. The GPP is typically responsible for management and control of the radio [18]. The analog subsystem typically comprises the RF front end, including transmit power amplifier and receive low-noise amplifier, transmit and receive filters, up/down converters to intermediate frequency (IF) or baseband and analog-to-digital converters (ADCs, and DACs).

Traditional hardware-based radios are typically implemented as application-specific integrated circuits (ASIC), which are highly optimised in terms of their computational efficiency, power consumption and circuit area [7]. It is obvious that such application-specific optimisation of the circuitry is precluded in SDRs, where the focus is on flexibility. For example, compared to an ASIC, the corresponding FPGA implementation is typically characterised by higher power consumption and larger silicon area [7].

Factors that the system designer needs to take into account when choosing an SDR platform are not limited to the computational requirements of the waveform application(s), but should also include SWaP constraints, cost and reconfiguration time.

For space applications, an additional challenge is that electronic equipment requires radiation hardening, which reduces the processing performance to such a degree that space-hard processors lag one to two generations behind their terrestrial equivalents [19]. These additional resource constraints place strict requirements on waveform design and SDR software architecture.

B. Software Architecture

In an ideal SDR, waveform applications can be ported to another SDR platform (potentially from a different vendor) without any modifications of its software implementation. It is the
The objective of the SDR software architecture is to introduce a transparency layer that decouples the waveform application from the underlying hardware as shown in Fig. 1. This layer is also known as middleware [15, Sec. 1.4]. Other aspects of the software architecture are to achieve modularity, adaptability and reconfigurability. Current SDR software architectures have not yet reached the maturity that they provide perfect abstraction away from the hardware and many challenges remain. Some of these challenges will be discussed below.

Adopting the concept of object orientation, the abstraction of hardware components is achieved by representing them as objects within the middleware layer. The middleware also provides services that allow different objects to communicate with each other, typically via some standard interface [15, Sec. 1.4]. The middleware layer is made up of all those software components that are not specific to the waveform application, e.g. the operating system, hardware drivers and resource management [15, Sec. 1.4].

In this paper, we will focus on the Software Communications Architecture (SCA), which is considered a de facto standard [7]. However, we will also briefly discuss another software architecture, namely the Space Telecommunication Radio System (STRS).

**Software Communications Architecture (SCA)**

This software architecture specification has been developed for the Joint Tactical Radio System (JTRS), a US defence program (see Sec. II-C). Despite being originally intended for the military domain, SCA has also found its way into industry and research. Starting with terrestrial communications, where SCA now has widespread acceptance, SCA has more recently also found its way into industry and research. Starting with terrestrial communications, where SCA has also found its way into industry and research. Starting with terrestrial communications, where SCA has also found its way into industry and research. Starting with terrestrial communications, where SCA has also found its way into industry and research. Starting with terrestrial communications, where SCA has also found its way into industry and research. Starting with terrestrial communications, where SCA has also found its way into industry and research. Starting with terrestrial communications, where SCA has also found its way into industry and research. Starting with terrestrial communications, where SCA has also found its way into industry and research. Starting with terrestrial communications, where SCA has also found its way into industry and research. Starting with terrestrial communications, where SCA has also found its way into industry and research. Starting with terrestrial communications, where SCA has also found its way into industry and research.

SCA is a CORBA-based (Common Object Request Broker Architecture) software architecture that aims to standardise high-level interfaces for deploying, configuring, controlling and monitoring the hardware and waveforms within an SDR [18]. SCA is a distributed system architecture, i.e. applications can be partitioned across different target processors, thereby allowing parallel processing. Hardware and application modules are abstracted as logical devices and components, both of which are CORBA containers that define standard interfaces and module properties.

Even though SCA contributes greatly to the portability of applications, SCA-compliance is not sufficient for all aspects of portability. For example, SCA supports portability by defining a subset of OS functions that can be used by the application [18]. On the other hand, SCA is still more or less limited to the GPP on current SDR platforms [7] and thus, waveform components that are executed on the DSP or FPGA are not CORBA-compliant and hence not easily portable. Another implication is that interprocessor communications need to be implemented using some non-CORBA mechanism.

The reasons for SCA being limited to the GPP are twofold: (i) implementations of CORBA object request brokers (ORB) used to be available for GPPs only and (ii) CORBA may not achieve the required throughput for interprocessor communications. In recent years, the first issue has been addressed and ORB implementations are now also available for DSPs and FPGAs [7, refs in]. The second point, however, remains challenging. SDR implementations of many modern high-throughput waveforms require efficient data transfers between GPP, DSP and FPGA. Using CORBA in conjunction with the transport mechanism TCP/IP not only limits the throughput, but also faces challenges with latency. Alternatives are available, but usually reduce portability.

An in-depth treatment of SCA and the related challenges can be found in [7, 18].
Space Telecommunication Radio System (STRS)

While SCA is currently the predominant software architecture, STRS is an architecture that is specifically tailored towards resource-constrained systems and as such, is of particular interest for space applications. Unlike SCA, STRS is an open architecture standard, which is developed by NASA [19].

Similar to SCA, STRS aims to decouple waveform and platform hardware. However, STRS does not define a particular mechanism or ports via which components communicate with each other, i.e. CORBA is possible but not the only option. The application programming interface (API) defined by STRS provides interfaces to manage the waveform applications and the SDR platform they are running on. The API also provides device control and data transfer mechanisms. Furthermore, NASA limits the deployment of STRS to a specific SDR platform rather than requiring portability to different platforms.

C. Standardisation

Due to the need to support communications for multiple missions, often requiring fast deployment, as well as increased interoperability with allied forces, the United States Department of Defence (DoD) formed the Joint Tactical Radio System (JTRS) Joint Program Office (JPO) in 1997 to develop a family of SDRs. The driving force behind this effort was the need for cost reduction and enhanced flexibility of the military’s radio systems for field operations. To increase reconfigurability, interoperability, re-use and upgradability of waveforms, a common SDR software architecture was developed – the SCA [18].

III. Satellite Link SDR Challenges and Research

Software defined radio is an ideal candidate for payload development as it allows post launch upgrades. However, careful consideration must be given to the potential resource requirements of future applications in order to future proof the payload platform itself. The temptation to over-engineer must be tempered against typically tight size, weight and power (SWaP) constraints [6]. Furthermore, link availability and bandwidth limitations can present a challenge for the code upload process. Ground terminals are typically less expensive, may be replaceable and do not rely upon the use of space grade components which often lag their terrestrial counterparts in terms of performance. Hence there exists a risk for terminal development to out-run the capability of the satellite.

The Wireless Innovation Forum defines four tiers of software defined radio architecture [13]. Most existing SDR platforms are classed as Tier 2, that is, software defined radio with modulation and baseband processing in software allowing for multiple fixed-frequency fixed-function RF hardware. Tier 3 requires a programmable RF front-end with digital conversion at the antenna, and Tier 4, based on a similar architecture, is capable of fast transitions between communication protocols. Research is ongoing in the move to Tier 3 for both satellite and terrestrial applications. This is a particular challenge in the case of satellite applications due to the lack of processing power required to sample and generate RF at the antenna [6].

While the software communications architecture has become popular for terrestrial applications, its associated implementation requirements have limited adoption into the space segment. The SCA introduces memory footprint, computational and latency overheads to a system that is already resource-constrained [7]. The reliance upon FPGA and DSP to meet space application SWaP requirements, and the limited availability of space grade components, presents further challenges in terms of portability and multi-vendor interoperability. Architecture standardisation is further motivated by the need for platform design modification between missions due to part obsolescence and new mission requirements [20].

Maintaining security of the SDR system presents a further challenge, as its reconfigurability may present attackers with the opportunity to install malicious code [21]. The legitimate code upload process itself must also be protected to prevent modification and eavesdropping.

There is also a significant level of research ongoing into the development of technologies that are enabled by SDR, such as cognitive radio and dynamic spectrum access (DSA) [22, 23], and cross-layer techniques [24].

IV. Conclusion

Satellite technology is well-recognised as a significant asset to defence, both in terms of intelligence gathering and in support of deployments around the globe. The large investment required and the long operational lifetimes of satellites make SDR an attractive complementary technology. SDR is suited to the satellite payload as well as the ground-based terminals and earth station. Satellite SDR deployments highlight and amplify some of the challenges facing terrestrial SDR implementations and present challenging design, development and research problems. Nevertheless, the flexibility of SDR can deliver real benefit in terms of multi-mode utilisation of the space-based asset, upgradability of satellite and ground terminal applications and reconfigurable, adaptive and cognitive operation for applications including remote communications, remote sensing and sensor data gathering.

REFERENCES

Satellite links and software defined radio for remote communications and sensor data gathering


