The Development of an SDH Simulator using a Software Defined Radio Platform

Angus Brandt, G-J Van Rooyen
Dept. of Electrical Engineering, University of Stellenbosch, Stellenbosch 7600, South Africa
{adbrandt, gvrooyen}@sun.ac.za

Abstract— **A *Synchronous Digital Hierarchy* (SDH) point-to-point link was implemented at a base *Synchronous Transfer Mode level 1* (STM 1) signal rate. The implementation was realized using a *Software Defined Radio* (SDR) architecture, which managed and linked the SDH atomic units into a STM-1 software radio. A *Common Object Request Broker Architecture* (CORBA) interface was used to implement a *Telecommunications Management Network* (TMN) to perform *the Operations, Administration and Maintenance* (OAM) duties via a graphical user interface.**

I. INTRODUCTION

The S*ynchronous Digital Hierarchy* (SDH) forms a critical part of the telecommunications backbone, and effective design, reliability analysis and training is essential to managing SDH networks effectively. This paper details the design of an SDH implementation using a *Software-Defined Radio* (SDR) platform. The main purpose of such an implementation is investigative; additionally, a software implementation may be used for operator training, fault simulation and even to assist hardware synthesis. Although a software-defined radio platform is used to realise the SDH functionality, it should be noted that this is motivated by efficiency and flexibility demands, although the system is not a "radio". A final objective of the research is to demonstrate the use of SDR techniques for a wider range of data-streaming applications.

This software SDH system would be useful as a training tool if it were built using the SDH *International Telecommunications Union* (ITU) standards. It should also be used to simulate certain network conditions to aid network designers, and should be capable of simulating fault conditions for statistical analysis. The SDH system will make use of the SDR architecture described in [5].

The following sections will mainly describe SDH and briefly give an overview of the SDR system used. Finally, a description will be given of how these two technologies have been combined to form a SDH system implemented on the Stellenbosch University SDR architecture.

II. THE SYNCHRONOUS DIGITAL HIERARCHY

The original goal of the Standards Committee was to establish an international specification for a common hierarchy among the European, North American and Japanese standards.

After five years of debate, the ITU adopted STM-1 at a rate of 155Mbits/s as the base granularity in the SDH network. In North America, the *Synchronous Optical Network* (SONET) specifications were based on a 51Mbits/s *Synchronous Transport Signal level 1* (STS-1) signal.

The most important SDH specification is the ITU-T

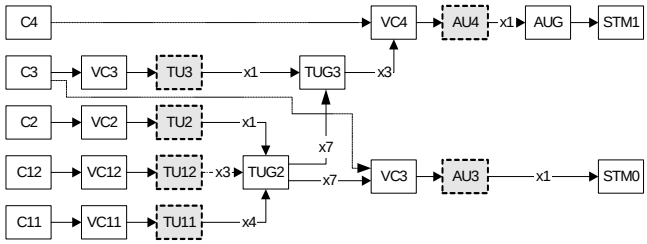

Figure 1: SDH Multiplexing Structure

G.707, "Network Node Interface for the Synchronous Digital Hierarchy (SDH)". It took the ITU four years to update the specifications. The new draft was approved in May 1996 and released in March 1997. Following this major G.707 revision, the ITU updated the G.783 specification, "Characteristics of Synchronous Digital Hierarchy (SDH) Equipment Functional Block". The main goal of these new specifications is to define the SDH interfaces and equipment that will ease the use of SDH services in the access network.

Figure 1 shows the generally accepted SDH multiplexing scheme. A plesiochronous 2Mbit/s tributary signal at 32 bytes per 125µSec frame, is mapped into a 34 byte synchronous *Contain*er (C-12). The two extra bytes are for justification and frequency adaptation. After addition of *Path Overhead* (POH) information for path management the *Virtual Container* (VC-12) is obtained. When the VC-12 is aligned in the *Tributary Unit* (TU-12), a pointer is added which indicates the phase of the particular VC-12, which changes during transmission. After addition of this pointer the TU-12 is obtained. Phase variation can be due to Jitter from regeneration and multiplexing equipment and Wander due to temperature differences within the transmission media. In the first multiplexing stage three TU-12s are inserted into one *Tributary Unit Group* (TUG-2). The second multiplexing stage combines seven TUG-2s and VC-3 *Path Overhead* (POH) information in the VC-3. After addition of pointer information for phase alignment the *Administrative Unit* (AU-3) is obtained. The Third Multiplexing stage multiplexes three AU-3s to the *Administrative Unit Group* (AUG). Finally in the fourth multiplexing stage N AUGs are byte interleaved. STM-N is obtained after addition of *Regenerator Section Overhead* (RSOH) for the management of repeater sections and *Multiplex Section Overhead* (MSOH) for the management of multiplex sections.

Thus a STM-1 can transmit 3(TU-12)×7(TUG-2)×3(TUG-3) = 63×2Mbit/s tributaries or 3×34Mbit/s tributaries or one-140Mbit/s tributaries.

III. NETWORK SEGMENTATION: SDH LAYERS

SDH segments the network into a *Regenerator Section* (RS), a *Multiplex Section* (MS), a *High Order Path* (HP) and a *Low Order Path* (LP). This allows for the probable location of an error or defect and provides a comprehensive view of the network's performance.

- The RS begins and ends with every element in the network, except for purely passive components like amplifiers.
- A MS begins and ends with any element, such as an add-drop multiplexer or digital cross connect, that is capable of integrating multiple traffic sources.
- The HP begins at the origin of the data stream and ends at its destination; LP applies to lower-rate virtual tributaries, such as 2Mbits/s circuits carried over SDH.

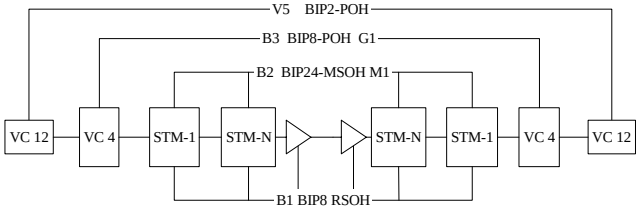

Figure 2: Segmentation of an STM1 SDH link

The SDH overhead is likewise divided into a Regenerator Section, a Multiplex Section, a High Order Path, and Low Order Path as shown in figure 2. Terminating equipment manipulates the overhead associated with each layer. Every network element is considered section-terminating equipment, so they all affect the section overhead such as framing (A1, A2), regenerator section trace (J0), and parity (B1). Pointer bytes (H1-H3) are transmitted on row 4, and are considered part of the administrative unit. Multiplex Section terminating equipment has access to the regenerator and multiplex section overhead, including parity (B2), protection switching (K1/K2), and synchronous status messages byte (S1). Path Terminating Equipment (PTE) marks the endpoints of the connection and is where the parity (B3), High Order Path Trace (J1), Signal Label (C2), and other path overhead are originated. Low order Virtual Containers, primarily used for 2.048Mbits/s and 1.544Mbits/s signals, also have their own overhead, including signal label V5 byte, bits 5-7, parity (BIP-2, part of the V5 byte), and low order path trace byte (J2) that are controlled by low order path terminating equipment.

The following is a general overview of SDR with some of the key ideas that went into designing the SDR architecture.

IV. A GENERAL OVERVIEW OF SOFTWARE DEFINED RADIO

The SDR Forum defines SDR as follows: [2]

> *Software Defined Radio (SDR) is a collection of hardware and software technologies that enable reconfigurable system architectures for wireless networks and user terminals.*

The SDR forum was founded to promote this technology. To ensure a common view the term "software radio" was proposed to be a generic term by using a set of tiers to describe facilities present in each system:[3]

- Tier 0. Hardware Radio (HR)
- Tier 1. Software Controlled Radio (SCR)
- Tier 2. Software Defined Radio (SDR)
- Tier 3. Ideal Software Radio (ISR)
- Tier 4. Ultimate Software Radio (USR)

A software radio fits into the tiered structure depending on the degree or means by which the device parameters can be changed by software. SDR is therefore a Tier 2 software radio.

The United States Federal Communications Commission (US FCC) adopted a more focused definition for SDR:

> *A radio that includes a transmitter in which the operating parameters of frequency range, modulation type or maximum output power (either radiated or conducted) can be altered by making a change in software without making any changes to hardware components that affect the radio frequency emissions.*

Although the definition is more specific than the broader definition provided by the SDR Forum, the FCC definition can be considered as a subset of the broader definition [4].

An SDR does not process the *Radio Frequency* (RF) signal directly and makes use of a hardware stage to down convert the RF signal to an *Intermediate Frequency* (IF) or directly to *Baseband* (BB) for processing by the software that defines the SDR. Hence the software running on the radio defines the functions of the radio. The software defines processes such as modulation and all aspects of signal processing.

V. SOFTWARE DEFINED RADIO ARCHITECTURE

In this paper, specific use is made of the SDR architecture developed by J.J.Cronjé in his thesis "Software Architecture Design of a Software Defined Radio System" [5]. Here J.J.Cronjé sets about designing and implementing a general-purpose SDR system. The SDH design discussed in this paper makes specific use of this SDR platform, developed and implemented at *Stellenbosch University* (SUN).

The following paragraphs should serve as an overview of this architecture.

In searching for a SDR hardware platform, and using the work of Bose [6] and Pucker [7], where a comparative study is made between *applications specific integration circuits* (ASIC), *digital signal processors* (DSPs), *general purpose processors* (GPPs), and *field programmable gate arrays* (FPGAs), Cronjé [5] makes the following conclusion:

> *No single platform is ideally suited to SDR systems… Until a platform is developed that caters specifically for SDR systems, the optimal solution is a hybrid solution that contains FPGAs and DSPs or GPPs.*

This conclusion is important since the realisation of a real system depends strongly on the availability of hardware that would effectively and closely mimic the performance of hardware radios while at the same time moving closer to the conceptual universal software radio.

The SDR system architecture in Figure 3 was designed with a layered approach to aid control and communications. Specifically defined interfaces provide upper and lower layer interconnects. Each layer is independent of the others,

and the complete system is built up by interconnecting these layers [5].

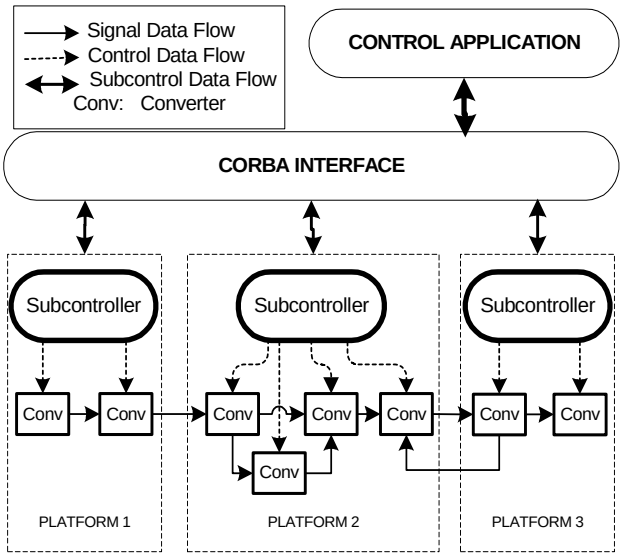

Figure 3: SDR system Architecture

The layers are defined as follows:

Converter layer

In the context of the design shown in Figure 3, a converter is an atomic unit that performs a well-defined signal processing function [5]. Each such signal-processing block is abstracted as a unit that converts incoming sample streams into output sample streams – hence the name "converter". In the converter layer, such blocks (e.g. amplifiers, demodulators, modulators) are defined and linked via specified ports, to ultimately realise a software–defined radio system. The converter layer therefore has to be optimised for high-speed signal processing to maximize throughput and minimize latency. It is this optimisation that makes it suitable for the simulation of SDH networks.

Subcontroller layer

The main function of the subcontroller layer is to manage the converters under its control. Using a round-robin scheduling method, all converters are given the opportunity to process their samples, and subsequent output data are then passed to the next converter based on a "push" principle. This method stores the computed output of a converter at the input buffer of the next converter instead of its own output buffer.

CORBA interface

The *Common Object Request Broker Architecture* (CORBA) standard is defined by the members of the *Object Management Group* (OMG). CORBA is a platform and programming language independent *Remote Procedure Call* (RPC) architecture that offers an elegant way to overcome cross-platform RPC difficulties.

For the system design shown in Figure 3, the CORBA interface only channels control information between the main application and the subcontroller layer. The interlayer communication requires very little bandwidth, as any communications are limited to system parameter modifications, the addition of new components and the initiation and termination of signal processing.

Control application (Main application)

The control application provides the user interface to the system. A *Graphical User Interface* (GUI) is typically implemented here to allow the user to interact with the components of the system. For example, a general-purpose GUI may allow the user to add, select and remove subcontrollers, and to query and adjust their signal processing parameters.

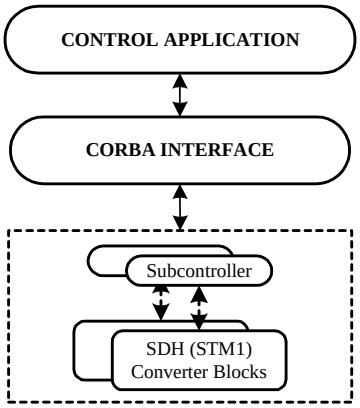

Figure 4: Block diagram of an SDH system built on SDR

VI. BUILDING THE SDH NETWORK USING SDR

Figure 4 illustrates a high-level overview of the SDH STM1 transmission system, using the SUN SDR architecture for component management.

A number of converter blocks are used to represent various SDH stages and path levels. These converter blocks are then combined to form a STM 1 structure. Finally two subcontrollers are used to allow for a distributed system, spanning various processors to support parallel processing of the SDH converter blocks. These subcontrollers ultimately

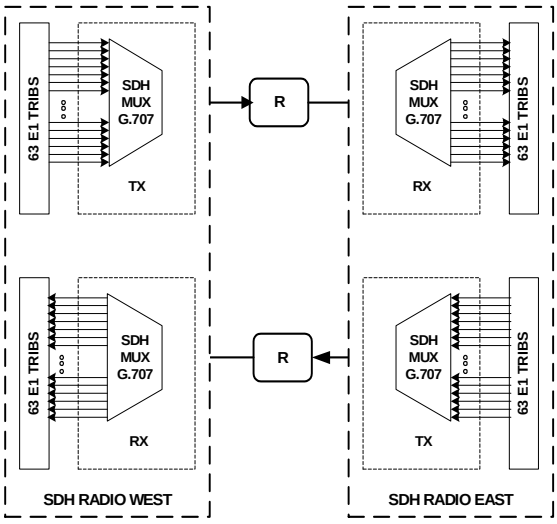

Figure 5: STM terminal -direction EAST and WEST

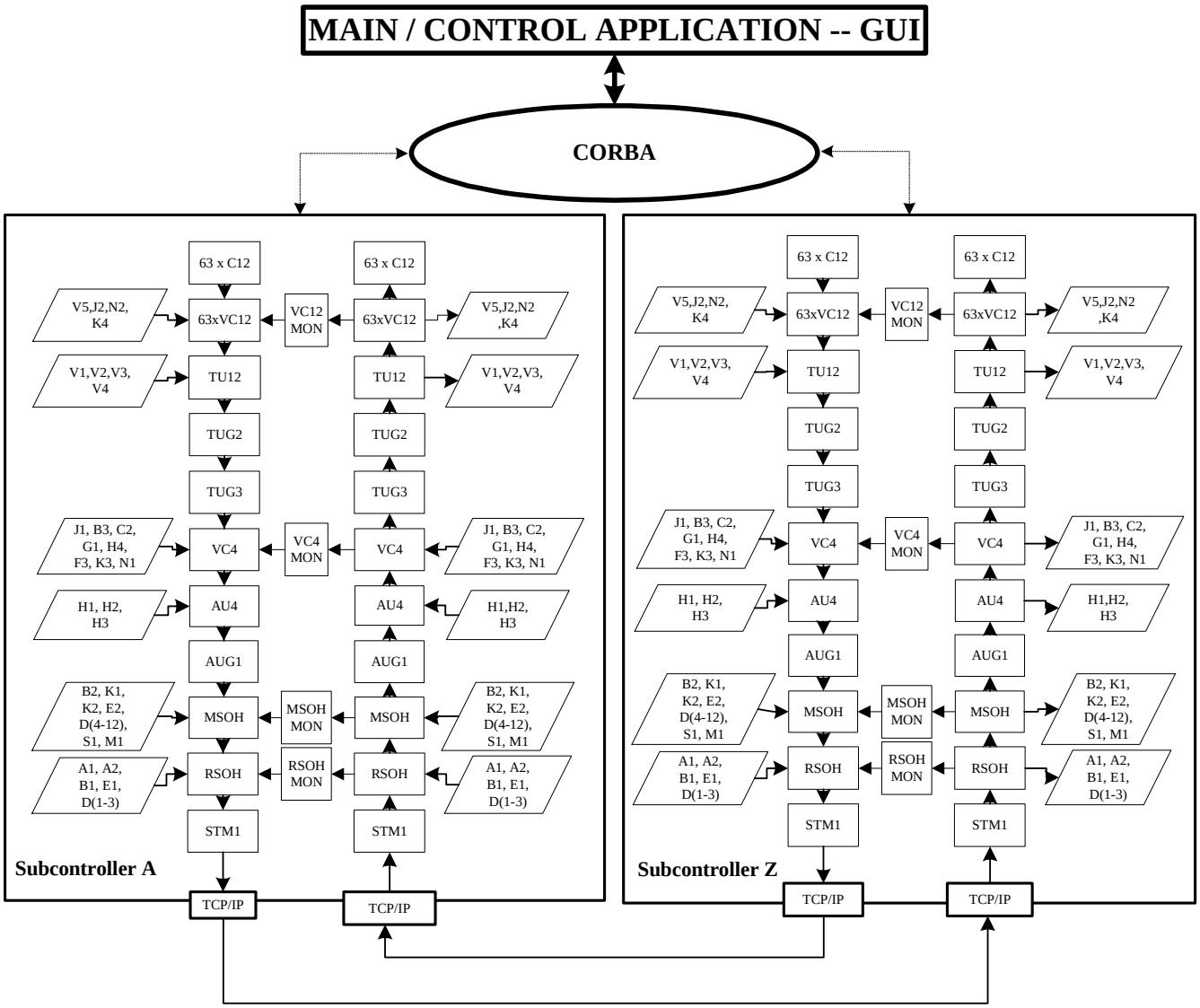

Figure 6: SDH radio using SDR

represent two SDH radios at an STM-1 level as shown in figure 5. Each subcontroller is responsible for the functional execution of the converters under its control and in turn communicates to the main application via the CORBA interface.

Extensive use was made of recommendation G.783 [8] in defining the components and methodologies for the SDH system. The recommendation illustrates specification methods based on functional decomposition of equipment into atomic and compound functions. The equipment is then described by its *Equipment Function Specification* (EFS), which lists the constituent atomic and compound functions, interconnections and overall performance objectives: [8]

> *The internal structure of the implementation of this functionality (equipment design) need not be identical to the structure of the functional model, as long as all the details of the externally observable behaviour comply with the EFS.*

Clearly the specifics of each component's internal operation are left to the designer's discretion, as long as externally observable behaviour complies with the EFS. Figure 5 illustrates an STM1 point-to-point SDH radio. The architecture of the multiplexer includes 63 tributaries, each having a capacity of 2,048 Mb/s. The regenerator section layer begins and ends with every element in the network, except for purely passive components like amplifiers.

A client/server relationship exists between any two adjacent network layers, where each transport network layer provides transport to the layer above and uses transport from the layers below. The layer providing transport is termed a server and the layer using transport is termed client. A detailed breakdown of Figure 5 is shown in Figure 6 and expands further on specifics of each of the subcontrollers' range of responsibilities and converter connectivity.

Network management functions are performed by three sets of SDH overhead bytes, namely path, multiplex section and regenerator section overheads.

VII. TRAIL TERMINATION FUNCTION

Each layer's signal integrity is monitored by the trail termination function. In the source direction it generates and adds error detection code and trail trace identification. It conveys back remote error indication signals containing the number of error detection code violations in the received signal and remote defect indicator signal.

In the sink direction it monitors for some or all of the following: bit errors, connection status, near-end performance, far-end performance, server signal fail and signal loss.

To enable single-ended maintenance, the defect status and number of error detection code violations detected at the sink trail termination are conveyed back to the source trail termination. The defect status is conveyed via the *Remote Defect Indicator* (RDI) signal and the number of error detection code violations is conveyed via the *Remote Error Indication* (REI) signal.

Degradation of the signal results in the detection of anomalies and defects. As a consequent action of the detection of certain near-end defects, the signal is replaced by the *all-ONEs* (AIS) signal and RDI is inserted in the return direction. The defects are reported to the fault management process.

VIII. LOWER ORDER PATH LAYER

The VC12 blocks in figure 6 forms the lower order path layer. At this stage the lower order path overhead is added in the transmit direction and VC12 MON performs the function of monitoring the VC12 overhead bytes. Four lower order path overhead bytes are added during a multiframe period. A multiframe period consists of four frames with frame duration of 125µSec per frame, thus the multiframe period is 500µSec. VC12 MON ensures that alarm conditions are tested for mainly via the V5 byte. Only the lower order path layer has access to the lower order path overhead bytes. This allows for easier fault location during fault conditions and individual paths can be monitored for transmission errors via the V5 byte. On the receive path, the VC12 lower order path overhead is removed, and only the payload is transmitted to the receiving C12 container.

Each converter has been coded to perform a defined task with the result of each block being passed on to the input of the next converter block. Figure 6 shows the arrangement for the layer defined as the lower order path layer made up of the C12 and VC12 converter blocks.

SDR allows for a distributed system, thus the subcontrollers can be implemented on various processors and interlinked using TCP/IP.

IX. HIGHER ORDER PATH LAYER

The VC4 blocks in figure 6 forms the higher order path layer. In this layer the J1, B3, C2, G1, G2, H4 F3, K3 and N1 bytes[1] forms the higher order path overhead. These higher order path overhead bytes are added in the transmit direction to perform the functions as outlined below:

- End-to-End communication bytes: J1, B3, C2 and G1.
- Payload-specific bytes: H4, F2 and F3.
- Byte reserved for future standards K3.
- Path Network Operator Byte (NOB): N1

VC4-MON performs a monitoring function and reports any anomalies on the receive side, back to the transmit side. Only the higher order path layer has access to the higher order path overhead. As is the case for VC12, each converter block is well defined with the inputs and outputs connected as defined in the ITU-T SDH multiplexing structure (figure 1). Three TUG-3 converter blocks are byte interleaving multiplexed to form the VC 4. Thus the higher order path layer is constructed through the interconnection of converter blocks TUG2, TUG3 and VC4.

X. MULTIPLEXER SECTION

Multiplex Section Terminating Equipment (MSTE) is a network element that originates and or terminates STM-N signals. MSTE can access, modify or terminate the MSOH or perform any combination of these functions. MSOH MON in figure 6 performs a monitoring function, thus tracking the status of the MSOH bytes. The MSOH bytes are: B2 for the *Multiplex Section Bit Interleaved Parity code* (MS BIP-24), and determines if a BIP-24 error has occurred over the multiplex section. K1 and K2 is the automatic protection Switching (APS) bytes. The rest of the bytes are the D4-D12, S1 and E2 bytes.

XI. REGENERATOR SECTION LAYER

The regenerator section overhead contains only the information required for the elements located at both ends of a section. A *Regenerator Section Terminating Equipment* (RSTE) is a network element that generates an STM-N signal for long haul transport. RSTE can originate, access, modify or terminate the RSOH, or can perform a combination of these functions.

XII. CONCLUSIONS

The SDR-based SDH system was designed to mimic the behaviour of a SDH system. As more SDH functionality is added to the network the system can grow to become an effective SDH training tool, as well as perform statistical network analysis.

The system also demonstrates the use of SDR to effectively provide the communications handling of the SDH implementation.

A network segmentation approach was used to develop the SDH network, and once configured on the SDR architecture, each layer could be built and tested independently. Together SDH and SDR provide an effective development and training tool as each converter block provides insight into the workings of an SDH network. Furthermore, these converter blocks are written such that the code can be transported to a hardware platform in an attempt to realise such a system. The SDR architecture automatically provides an *Application Programming Interface* (API) that facilitates interfacing

to the system. This API could easily be used to implement a GUI for network management.

XIII. FUTURE WORK

SDH was optimised for voice transport only, which sets different requirements for the network than the growing data transport services. The technology enabling voice over data is called VoIP. Legacy SDH cannot support the growing data service demand. Next-generation SDH has a transport mechanism, which enables the concurrent existence of legacy and new services over the same network without disturbing each other. The technology is commonly known as the *Data over SDH* (DoS) concept. The DoS scheme consists of three technologies: *Generic Framing Procedure* (GFP), *Virtual Concatenation* (VC) and *Link Capacity Adjustment Scheme* (LCAS), all standardised by ITU-T.

Implementation of the DoS schemes will allow almost every data transport technology to be efficiently mapped over SDH.

Furthermore, statistical models can also be built by adding statistical data to the converter blocks, to aid network designers in resources management. Each converter block can then be assigned a mean time between failure value and the throughput vs. resources can then be optimised in line with the predefined network operators' service level agreement.

REFERENCES

[1] ITU-T Rec. G.707/Y.1322. "*Network node interface for the Synchronous Digital Hierarchy (SDH)*" December 2003.

[2] SDR Forum, "SDR Forum FAQs." http://www.sdrforum.org/faq.html. January 2004.

[3] SDR Forum."Definitions & Semantics" http://www.sdrforum.org.

[4] SDR Forum."Overview and Definitions of Software Download for RF Reconfiguration DL-DFN Document SDRF-02-A-0002-V0.00"6 August 2002.

[5] CRONJÉ, J., *Software Architecture Design of a Software Defined Radio System*. MSc thesis, Stellenbosch University, October 2004.

[6] BOSE, V., Design and Implementation of Software Radios Using a General Purpose Processor. PhD thesis, Massachusetts Institute of Technology, June 1999.

[7] PUCKER, L., "CommsDesign – Paving Paths to Software Radio Design." http://www.commsdesign.com/csdmag/sections/cover_story/OEG20010521S0118. June 2001

[8] ITU-T Rec. G.783. "*Characteristics of synchronous digital hierarchy (SDH) equipment functional blocks*" April 1997.

[9] Ming-Chwan Chow, Understanding SONET/SDH Standards and Applications, 1st edition.1995, Andan Publishers.

[10] GREEN, P., "Progress in Optical Networking" *IEEE Communications Magazine*, Jan. 2001, pp. 54-61.