DESIGN AND IMPLEMENTATION OF A STANAG 5066 DATA RATE CHANGE ALGORITHM FOR HIGH DATA RATE AUTOBAUD WAVEFORMS

by

Stephan Schulze

Submitted in partial fulfilment of the requirements for the degree

Master of Engineering (Computer Engineering)

in the Faculty of Engineering, the Built Environment and Information Technology

UNIVERSITY OF PRETORIA

AUGUST 2005

ABSTRACT

DESIGN AND IMPLEMENTATION OF A STANAG 5066 DATA RATE CHANGE ALGORITHM FOR HIGH DATA RATE AUTOBAUD WAVEFORMS

by

Stephan Schulze

Supervisor: Prof. G.P. Hancke

Department: Electrical, Electronic and Computer Engineering

Degree: Masters in Engineering

KEYWORDS:

STANAG (North Atlantic Treaty Organisation Standardization Agreement) 5066, DRC (Data rate Change) algorithm, BER (Bit Error Rate), FER (Frame Error Rate)

SUMMARY

HF communication has been used for more than a century and to this day still fulfils an important function in communications networks. In order to interface with modern communications protocols, solutions have to be designed to facilitate data communication over HF (High Frequency). STANAG 5066 is one such solution which provides an application independent ARQ (Automatic Repeat Request) bearer service for client applications.

A need exists within the STANAG 5066 specification for a DRC algorithm. The objective of such an algorithm is to select the optimum data rate and interleaver size, based upon current HF channel conditions, to maximise the data throughput over the HF link.

In this dissertation previous implementations of DRC algorithms were studied and evaluated. In literature it was found that algorithm implementations used the FER and no channel information to make a data rate choice. This resulted in algorithms that tended to oscillate between data rate choices, and was very slow to react to changes in the HF channel.

A new DRC algorithm was designed and simulated that uses the SNR (Signal-to-Noise Ratio) and the BER estimate to make a data rate choice.

The DRC algorithm was implemented in a commercial STANAG 5066 system and tested using HF data modems and a simulated HF channel.

The results of the implementation and testing show that the designed DRC algorithm gives a better performance, is quicker to adapt and is more robust than previous DRC algorithms. This is also the first DRC algorithm that has been designed to use channel information, such as the SNR and BER, to make a data rate choice.

OPSOMMING

DIE ONTWERP EN IMPLEMENTERING VAN 'N STANAG 5066 DATA-TEMPO-VERANDERINGS-ALGORITME VIR HOË SPOED "*AUTOBAUD*" GOLFVORMS

deur

Stephan Schulze

Toesighouer: Prof. G.P. Hancke
Departement: Elektriese, Elektronies en Rekenaar Ingenieurswese
Graad: Meesters in Ingenieurswese

SLEUTELWOORDE:
STANAG (Noord Atlantiese Vredes Organisasie Standaardisasie Ooreenkoms) 5066, Data-tempo-veranderings-algoritme, BER (Bis Fout Tempo), FER (Raam Fout Tempo)

OPSOMMING

HF kommunikasie word alreeds vir langer as 'n eeu gebruik en vervul tot vandag toe 'n belangrike funksie in kommunikasienetwerke. 'n Nuwe koppelvlak moes gevind word om met moderne kommunikasieprotokolle te werk. STANAG 5066 is een so oplossing wat 'n applikasie-onafhanklike ARQ draerdiens bied vir applikasieprogramme.

Daar bestaan binne die STANAG 5066 spesifikasie 'n behoefte aan 'n data-tempo-veranderings-algoritme. Die oogmerk met so 'n algoritme is om die optimum datatempo en interflegger grootte te kies, gebaseer op die huidige HF-kanaal omstandighede, om sodoende die data deurvoer te maksimaliseer.

In hierdie dissertasie is vorige ontwerpe van data-tempo-veranderings-algoritmes bestudeer en geëvalueer. In literatuur is gevind dat ander data-tempo-veranderings-algoritmes gebruik alleenlik die FER sonder enige kanaalinligting om datatempo keuses te maak. Die uitwerking hiervan is dat datatempokeuses geossilleer het en dat die datatempokeuses baie stadig was om te reageer op HF-kanaal veranderinge.

‘n Nuwe data-tempo-veranderings-algoritme is, ontwerp en gesimuleer, wat die SNR en die BER skatting gebruik om ‘n datatempokeuse te maak.

Die data-tempo-veranderings-algoritme is geïmplementeer in ‘n kommersiële STANAG 5066 stelsel en is getoets met HF data modems in ‘n gesimuleerde HF kanaal.

Die resultate van die implementering en toetsing wys dat die ontwerpte data-tempo-veranderings-algoritme beter werkverrigting toon, vinniger reageer en meer robuus is as vorige algoritmes. Hierdie is ook die eerste data-tempo-veranderings-algoritme wat ontwerp is om kanaalinligting soos die SNR en BER te gebruik om datatempokeuses te maak.

LIST OF ABBREVIATIONS

ACK	Acknowledgement
AGC	Automatic Gain Control
ARQ	Automatic Repeat Request
ARP	Address Resolution Protocol
AWGN	Additive White Gaussian Noise
ALE	Automatic Link Establishment
BER	Bit Error Rate
BERT	Bit Error Rate Test
bps	Bits per second
CAS	Channel Access Sublayer
CFTP	Compresses File Transfer Protocol
CCIR	Consultative Committee for International Radio
COTS	Common off the shelf
CRC	Cyclic Redundancy Check
CSMA	Carrier Sense Multiple Access
DiffServ	Differential Services
DSCP	Differential Services Code Points
DRC	Data Rate Change
DRM	Digital Radio Mondial
DTS	Data Transfer Sublayer
EOM	End of Message
EOT	End of Transmission
EOW	Engineering Order Wire
FER	Frame Error Rate
FRAP	File Receipt Acknowledgement Protocol
HDLC	High-level Data Link Control
HF	High Frequency
HFCS	High Frequency Channel Simulator
HFTP	High Frequency Token Protocol
HMTP	HF Mail Transfer Protocol
Hz	Hertz (Unit of Frequency)

ICMP	Internet Control Message Protocol
ID	Identification Number
ISI	Intersymbol Interference
LAN	Local Area Network
LQA	Link Quality Analysis
LSB	Least Significant Bit
MAC	Media Access Control
MAN	Management Sublayer
MHz	Mega-Hertz
MSB	Most Significant Bit
MTU	Maximum Transmission Unit
NATO	North Atlantic Treaty Organisation
OTA	Over the Air
POP	Post Office Protocol
PPP	Point-to-Point Protocol
PDU	Protocol Data Unit
QoS	Quality of Service
RC-66	RapidM Rapid Communicator-66
RCOP	Reliable Connection Orientated Protocol
RM6	RapidM HF Data Modem & ALE Controller
RX	Receive
SAP	Subnet Access Point
SIS	Subnetwork Interface Sublayer
SMTP	Simple Mail Transfer Protocol
SNMP	Simple Network Management Protocol
SNR	Signal-to-Noise Ratio
STANAG	NATO Standardisation Agreement
SOM	Start of Message
TCP	Transport Control Protocol
TDMA	Time Division Multiple Access
TLC	Transmit Level Control
TOS	Type of service
TX	Transmit
UML	Unified Modelling Language

TABLE OF CONTENTS

CHAPTER 1 INTRODUCTION

1.1 HF COMMUNICATIONS

Radio communication has been an integral part of worldwide communication since the implementation of the radio in the earliest part of the previous century. Radio communication that utilizes the bandwidth between 3 and 30 MHz is called HF (High Frequency) or *“shortwave”* communication and is especially used for communication over long distances. Even though HF radio communication technology has been in operation for a long time it is still popular today because of the recovery and redundancy advantages the technology offers over satellite and more modern terrestrial implementations.

The reason that HF communication can take place over such long distances is related to the properties of the earth’s ionosphere. If a HF signal hits the ionosphere and is sufficiently bent the signal will hit the earth again at a distant location. The signal can then rebound to the ionosphere and from the ionosphere back to earth at an even greater distance away from the source. One hop, or from earth to ionosphere to earth, allows for communication over great distances, for 1 hop < 4000 km (kilometres), for 2 hops between 4000 and 7000 km and for 3 hops 7000 to 12000 km from the source. The propagation properties of HF radio signals are based upon the frequency used as well as environmental conditions. For example, a signal that propagates well during the day may, not do so during the night. This can be due to changes in temperature, to name but one reason.

The HF radio medium has a number of key challenges that must be taken into account when transmitting data packets and email. These include very low SNR (Signal-to-Noise Ratio) radio signals, multipath fading channels, signal propagation variation based upon hour, season and sunspot cycle and a limited channel capacity. Receiver noise and interference can affect the received signal and can come from both internal and external sources. However, the external receiver noise greatly exceeds the internal receiver noise. Received signal quality is measured in SNR, the higher the SNR the better the received signal.

Although most HF research is currently being carried out for military applications, a new standard has emerged in the commercial arena called Digital Radio Modial (DRM). DRM is a new standard for short, medium and long wave digital radio transmission and also the only non-proprietary digital AM radio system.

1.2 HISTORY OF DATA COMMUNICATION OVER HF

HF Data communication started with Morse code and later more improved techniques evolved that increased the data transmission speed. These techniques take the variability of the HF medium into account.

1.2.1 HF modems

Because most conventional radios cannot transmit data directly into the HF medium, a HF modem is required. The function of the HF modem is to convert digital data to an audio signal using a modulator for transmission by the radio and to convert the received audio signal back to digital data using a demodulator at the receiver. This allows the HF radio to TX (transmit) and RX (receive) both data and voice signals.

1.2.2 HF waveforms

Waveforms were developed that added FEC (Forward Error Correction) to the data stream. This allowed data errors to be detected and corrected at the HF modem demodulator. The BER (Bit Error Rate) gives a measure of the number of one-bit errors over a certain number of bits as detected by the HF modem demodulator and is used to measure the performance of the waveform.

1.2.2.1 Interleaver

Interleaving is the process of randomizing the data stream in order to reduce the BER. Most errors occurring in the HF medium are burst errors. When a data steam is transmitted serially over the interface an entire section of adjacent data will be lost when such a burst error occurs.

It is very difficult for the FEC algorithm to detect and correct the corruption of adjacent data blocks. When the data bits are interleaver, randomly distributed over the entire message, and again de-interleaved at the receiver, the FEC has a greater possibility of correcting burst errors. The interleaver size refers to the size of an interleaver block in bits and this relates to the interleaver length which is measured in seconds and is determined by calculating the time taken to transmit an interleaver block at a certain data rate.

1.2.2.2 Autobaud and non-autobaud waveforms

An autobaud waveform is a type of waveform where the receiver does not need to have exact knowledge of the transmitter data rate and interleaver size in order to correctly receive the signal. An autobaud waveform includes the data rate and interleaver size in the header information of the waveform. STANAG (North Atlantic Treaty Organisation Standardization Agreement) 4539 [1], MIL-STD (U.S. Military Standard) 188-110A [2] and MIL-STD 188-110B [3] are examples of autobaud waveforms. A non-autobaud waveform does not have this header information and as such, the sender and receiver data rates and interleaver size have to be synchronized before communication can take place.

1.2.3 HF data communication requirements

The demand placed upon hardware and communication infrastructure has grown because of an increase in the size and complexity of military, business and civilian applications, as well as their communication needs and requirements. The modern communication development trend is one of integration, where all types of communication systems can inter-operate with one another in a large WAN (Wide Area Network) type system. This WAN model, for military use, includes the local command and control station, the soldier in the field, vehicles, planes and ships. All these components need to be able to communicate with one another using different communications hardware over different mediums and using changing technologies.

For WAN integration the existing HF radio communications infrastructure will need to interact with modern communications protocols. The need has therefore arisen for the HF network to support TCP/IP (Transport Control Protocol/Internet Protocol) services in order to inter-network with other types of network applications. The benefits of using TCP/IP services are:

- the application independence of the TCP/IP protocol;
- the large number of existing applications and services that use TCP/IP as bearer medium;
- that with TCP/IP there is no need for a large number of gateways in the system (using TCP/IP services decreases the number of protocol conversions required at network end-points).

The challenge of using TCP/IP over HF is the following:

- HF channel capacity is very limited and will not support bandwidth intensive TCP/IP applications;
- the large protocol overhead of TCP/IP packets will waste HF bandwidth;
- the HF medium is very variable and this will result in a number of retransmissions over the HF channel that will consume even more channel capacity.

1.3 ARQ (AUTOMATIC REPEAT REQUEST) PROTOCOL DEVELOPMENT

In order to support HF data communication and perhaps even TCP/IP over HF, a need has arisen for a general, open and interoperable protocol for data applications. This is especially true for ship-to-shore communication that supports email and other PC (Personal Computer) based internet applications. A number of propriety data link and ARQ transport protocols have been proposed to support data communication between nodes over HF. However, most implementations were application specific; this means that there exists no interoperability between different vendor implementations. The new protocol has to be compatible and inter-operate with the current NATO (North Atlantic Treaty Organization) WAN network. The new protocol requires a common air interface, ARQ functionality and a common terrestrial interface. For previous work in the area of an HF ARQ protocol the reader is referred to [4], [5] and [6] as well as the maritime gateway project [7].

In 1996 the Allied Naval Communications Agency commissioned the design and implementation of a new protocol for ship-shore communication. The protocol described the interfaces, data formats, functions and procedures for HF data communication.

For any protocol to meet NATO requirements the two external interfaces have to be adequately defined. The HF (air) interface and the non-HF (client) interface (Figure 1.1). The advantage of defining open interfaces is to allow interoperability between different vendor implementations.

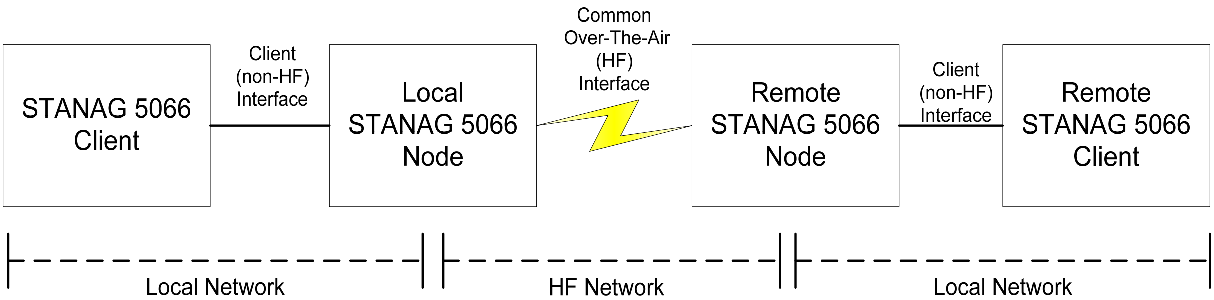

Figure 1.1: Interfaces of the STANAG 5066 protocol.

The end-result of the protocol design was the STANAG 5066 [8] NATO standardization agreement. For future reference, a STANAG 5066 node is the implementation of the standardization agreement as an application server that delivers STANAG 5066 services to client applications. STANAG 5066 [8] defines an international standard for data communication in HF networks. One important advantage of the STANAG 5066 architecture is the fact that it is application-independent and delivers a generic bearer service for client applications.

STANAG 5066 is an ARQ (Automatic Repeat Request) type protocol that controls the transmission of packets sent OTA (over the air) interface and was developed using a layered design approach, much like TCP/IP (Transport Control Protocol/Internet Protocol). An example of a complete STANAG 5066 system can be seen in Figure 1.2. There exists a client-server relationship between the application specific clients and the STANAG 5066 node. The clients will use the node to facilitate communication to a peer client application over a HF link. The STANAG 5066 node itself does not need any client specific knowledge in order to deliver the ARQ service. STANAG 5066 was initially developed to support ship-to-shore communication using STANAG 4285 [9] and STANAG 4529 [10] waveforms. STANAG 4285 and 4529 are non-autobaud waveforms for data communication on 3 kHz radio channels. STANAG 4285 supports data rates up to 2400 bps and includes FEC and equalization.

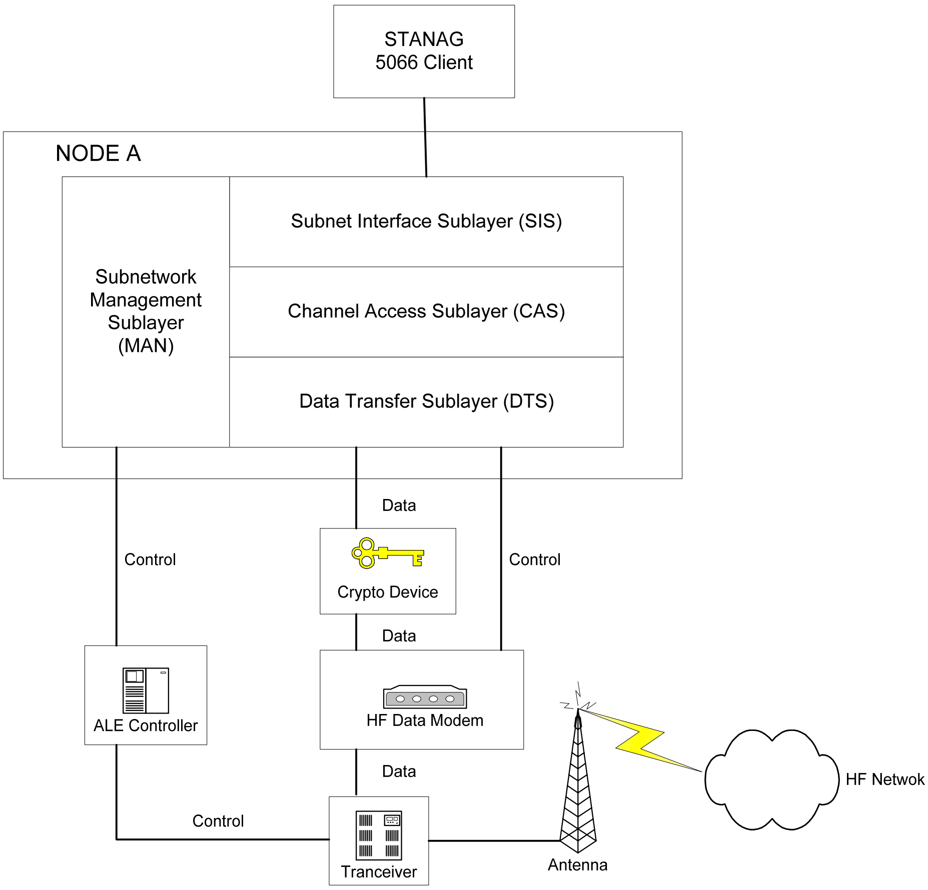

Figure 1.2: Complete STANAG 5066 HF Data Communications System

The STANAG 5066 protocol sublayers, as can be seen in Figure 1.4, are:

- **Subnet Interface Sublayer (SIS):** The SIS provides a common interface for all subnetwork clients that use the services provided by the STANAG 5066 node. All primitives or data packets and PDUs (Protocol Data Units) exchanged between sublayers and external clients are well defined. This results in network clients that are vendor interoperable. Each client connecting to the STANAG 5066 node uses a SAP

(Subnet Access Point) number. The SAP number is unique to each client connecting client and functions much like a port number in TCP/IP.

- **Channel Access Sublayer (CAS)**: The CAS defines functions needed for accessing the physical channel, i.e. the radio spectrum, using a HF radio and antenna. The CAS assumes that the frequency or "channel" selection function is handled by an external process. This external frequency selection function can either be handled by ALE (Automatic Link Establishment) [11], a human operator or an automated process.
- **Data Transfer Sublayer (DTS):** The DTS handles data transmission to a remote node and provides a reliable data link service for connected clients. The layer will ensure that D_PDUs (DTS Protocol Data Units) or *"Frames"* are delivered, based upon data exchange rules, to a remote node. The automatic change of data rate is also handled in this layer through the modem remote control interface. The remote control interface allows the node to send control information to and receive status information from the HF modem. This includes setting the current data rate and waveform type as well as receiving current channel condition information from the modem, such as the current RX data SNR and BER values. The interface between the STANAG 5066 node and the HF modem can be seen in Figure 1.3.

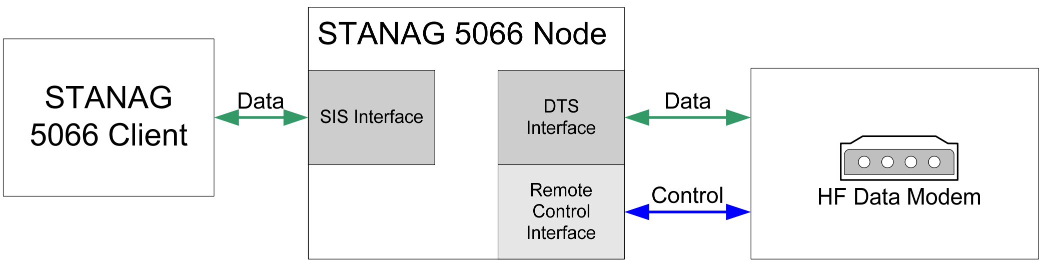

Figure 1.3: Interface between the STANAG 5066 node and the HF modem.

- **COMSEC (Communications Security) Sublayer:** This layer provides data security and confidentiality using external hardware crypto equipment.
- **Modem Sublayer:** The function of the modem sublayer is to transmit a digital signal over an analogue channel by modulating the digital signal to an audio signal at the sender and demodulating the received audio signal to a digital signal at the receiver.

- **ALE Sublayer:** The ALE sublayer will be used to *make* and *break* physical links defined by an operating frequency as well as to provide a frequency selection function to the STANAG 5066 node. The ALE Sublayer searches through a set of predefined frequencies for a remote node. When the remote node is found a link is made to this node and as soon as data transfer to the node is complete, the link is broken.
- **Radio Equipment Sublayer:** This sublayer will be responsible for the tuning of the HF radio to the correct operating frequency, i.e. set the current RX and TX frequencies as well as interfacing with an antenna.
- **Subnet Management Sublayer:** The subnet management interface is the layer that is capable of interfacing with all other layers of the STANAG 5066 protocol stack and provides management and configuration support.

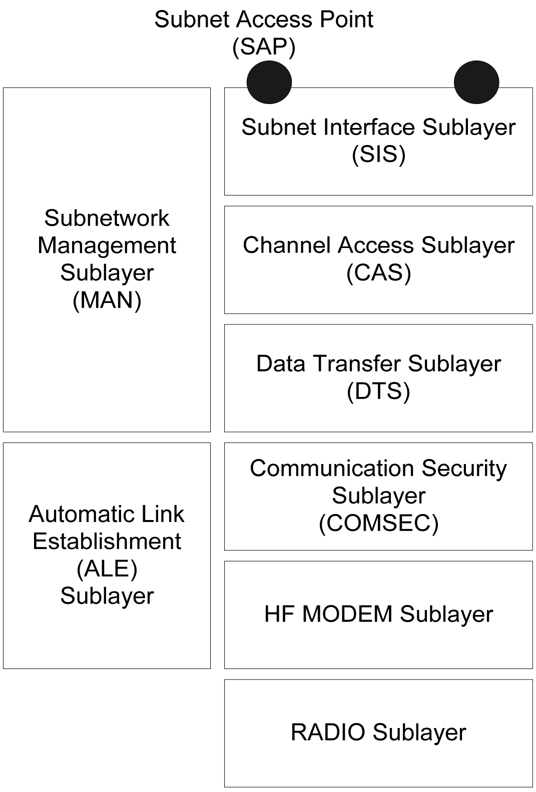

Figure 1.4: The STANAG 5066 protocol sublayers.

Communication between adjacent sublayers functions through the passing of *primitives* or data packets between sublayers. Communication to a peer sublayer at a remote node functions through the passing of PDUs. Figure 1.5 shows the inter-node and intra-node communication.

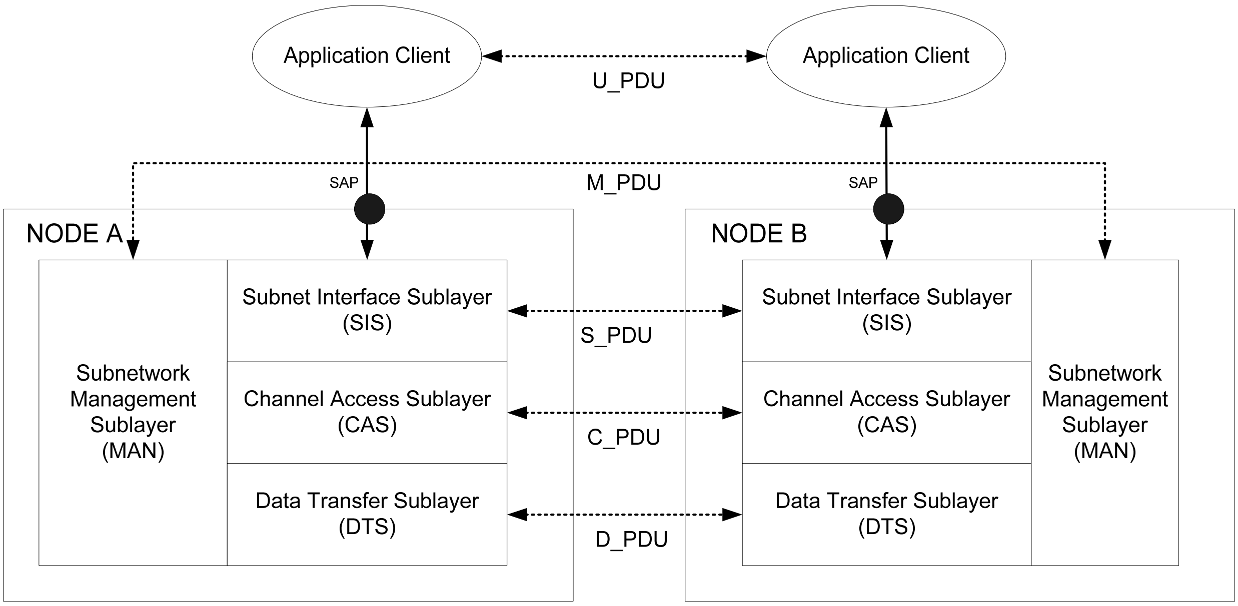

Figure 1.5: Peer layer communication between remote STANAG 5066 nodes.

1.4 PROBLEM STATEMENT

The HF communication medium is very diverse and challenging for an application that tries to use it as a bearer. This poses implementation problems for system designers trying to ensure the best amount of throughput on a link between two nodes. To achieve the largest throughput the following requirements have to be met based upon channel conditions:

- the best available channel must be used;
- the highest possible data rate must be used;
- the channel utilization should be high;
- the protocol overhead should be low;
- the system must adapt to changing channel conditions and avoid new link setup.

The purpose of a DRC (data rate change) algorithm is to select the highest possible data rate, measured in bps (bits per second), and interleaver size to use as well as to change that data rate and interleaver size based upon changing channel conditions, where a channel is defined by a certain RX and TX frequency in the 3 to 30 MHz spectrum. The best data rate and interleaver size is selected by the receiving node because the receiving node is in the best position to determine what the sending node settings should be when transmitting data to it.

RapidM (Pty) Ltd. (the company where the author is employed) is currently in the development stage of a HF email gateway solution. The RC-66 (Rapid Communicator 66) system can be seen in Figure 1.6.

The problem addressed in this dissertation is that a DRC algorithm has not yet been designed or implemented to manage data rate selection for the high data rate waveform STANAG 4539 for use in the RC-66 STANAG 5066 node.

1.5 RESEARCH OBJECTIVE

The objective of this research is to design, simulate and implement a DRC algorithm that uses current channel condition information to determine the optimum data rate and interleaver size to use in STANAG 5066 node to node communication. The DRC algorithm should recognize changes in channel conditions and then adapt the data rate and interleaver accordingly to produce the highest possible throughput, measured in bps, on the channel. Furthermore, the DRA algorithm designed and implemented should provide better performance than other DRC algorithms presented in literature.

The throughput is determined by the waveform used, interleaver size, D_PDU frame size and data rate. For the purposes of this research it is assumed that the waveform used will be STANAG 4539.

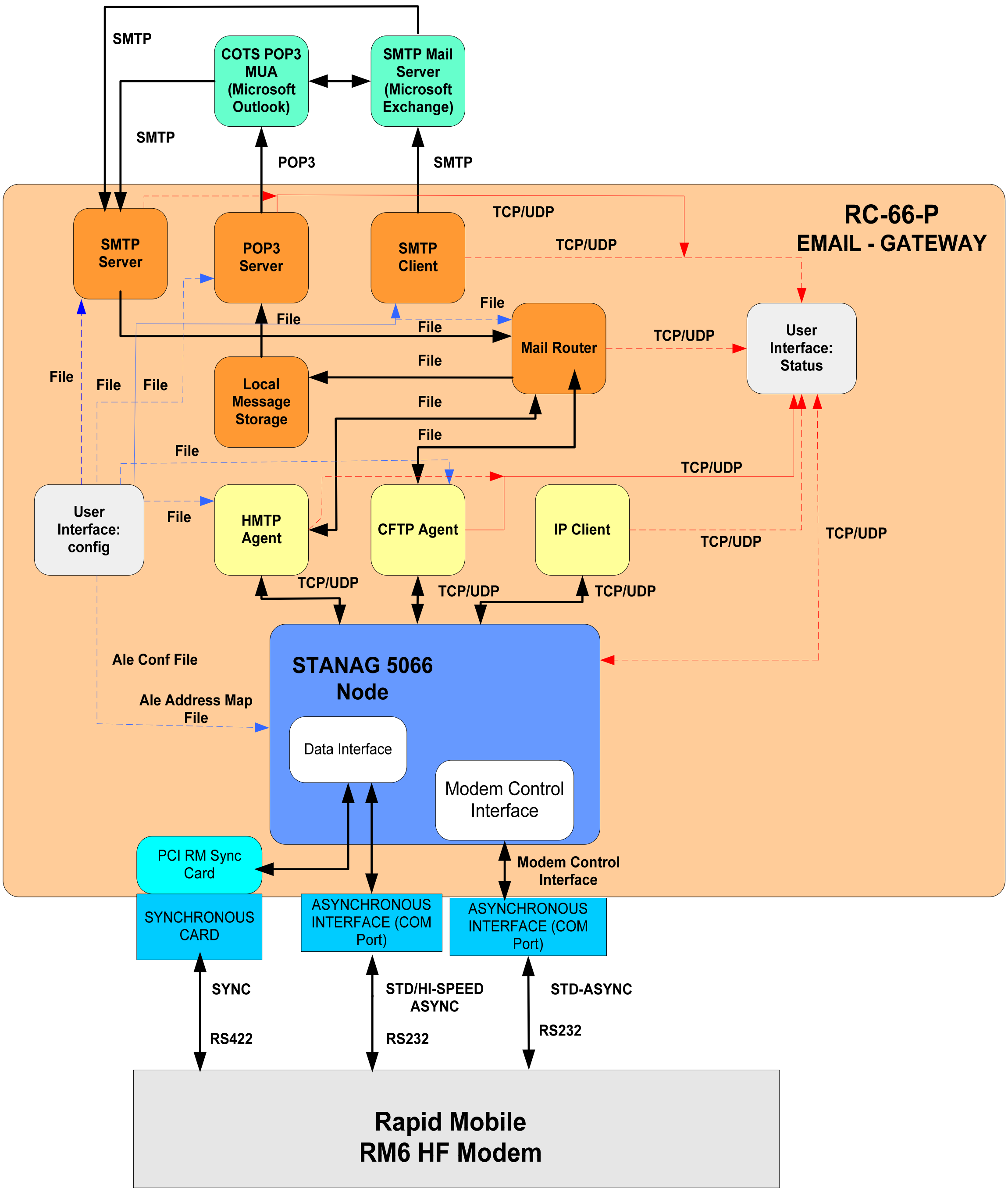

Figure 1.6: RapidM RC-66 STANAG 5066 HF Email Gateway solution.

1.6 RESEARCH METHODOLOGY

A DRC algorithm will be designed and simulated in a simulation environment and measured against other DRC algorithm implementations as found in literature and described in [12], [13] and [14].

The best algorithm design, based upon simulation, along with algorithms from literature, will then be implemented in the RC-66 STANAG 5066 node. A test system will be setup using two STANAG 5066 nodes and two STANAG 5066 clients communicating test data to each other. The two nodes will be connected using two HF data modems and a HF channel simulator. The performance of the DRC algorithm will be determined and compared to results of other DRC algorithms ([12], [13] and [14]).

The designed DRC algorithm will use channel information, such as the SNR, BER and FER (Frame Error Rate) to adapt the data rate and interleaver size to changing channel conditions.

1.7 RESEARCH CONTRIBUTION

There is no DRC algorithm implementation, currently in literature, that uses channel condition information such as SNR and BER for making data rate change choices (algorithm implementations in [12], [13] and [14] only use the FER). The designed DRC algorithm will make use of available channel information resulting in an algorithm implementation that will increase the overall throughput achieved by a STANAG 5066 system and an algorithm that can recognize and quickly adapt to changing channel conditions.

The DRC algorithm designed and implemented in this dissertation will also try to overcome two major disadvantages of algorithms presented in literature. These disadvantages are:

- data rate choice oscillation;
- algorithms that are slow to react to changing channel conditions.

1.8 STRUCTURE OF THIS DISSERTATION

Chapter 1 includes an introduction to the current subject of data communication over HF radio and also includes the problem statement, research objective, methodology and expected research contribution included in this dissertation.

Background information around the STANAG 5066 specification and STANAG 5066 client implementations are given in Chapter 2. The STANAG 5066 specification is examined and each of the client applications is defined.

The literature study on current STANAG 5066 DRC algorithms, interleaver size selection and frame size selection are included in Chapter 3.

Chapter 4 includes the design and simulation of the DRC algorithm.

The implementation of the DRC algorithm is discussed in Chapter 5.

In Chapter 6 tests will be designed to measure the DRC algorithm implementation performance and the results of the testing will be presented and discussed.

The conclusion to the paper will be presented in Chapter 7 where it will be determined whether the research did indeed achieve the goal of designing a DRC algorithm for the STANAG 4539 autobaud waveform. Problems experienced and further research options will also be discussed.

CHAPTER 2 THE STANAG 5066 PROTOCOL

2.1 INTRODUCTION

This chapter will serve to introduce the STANAG 5066 protocol and to explain the functionality of a HF data communication system that uses a STANAG 5066 node and associated clients. The STANAG 5066 protocol will first be examined based upon the OSI (Open Systems Interconnect) model, then the functionality of the STANAG 5066 node will be examined and lastly client applications that use the node will be identified.

2.2 THE OSI MODEL

When a HF data communications stack is examined in terms of the OSI model, the stack is made up of a transmission level, link level, network level and system level. Figure 2.1 shows the HF data communication stack layers when compared to the OSI Model.

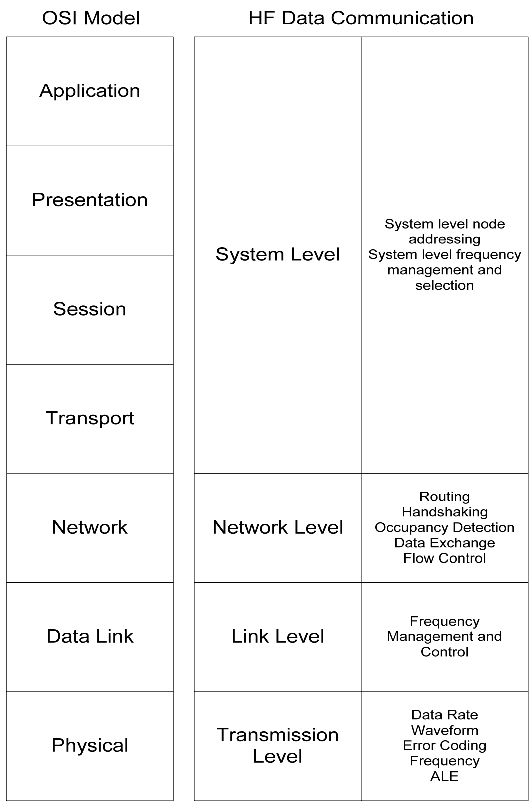

Figure 2.1: HF data communications stack layers compared to the OSI model.

2.3 MODEM TRANSMISSION DELAYS

The delays associated with communication over HF can be seen in Figure 2.2 and Table 2.1. These delays include the processing delay for the STANAG 5066 node and client, the data transfer delay between the client and the node, the transfer delay between the node and the HF modem, the buffering and processing delay of the HF modem and the OTA transmission time.

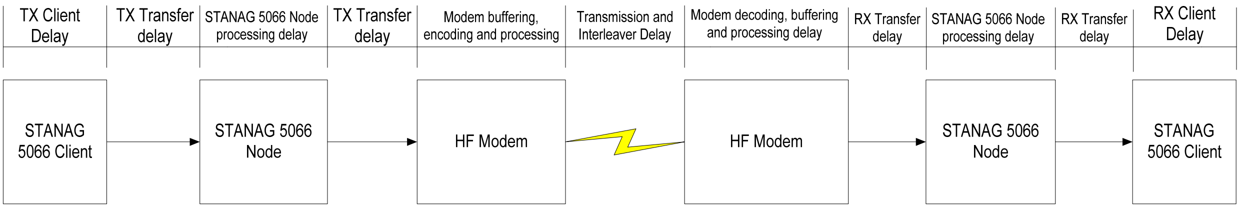

Figure 2.2: Transmission delays between two STANAG 5066 clients.

Table 2.1: Delays in ARQ data communication

STANAG 5066 processing and transmission delays	Waveform delays
Packet overhead	Fill TX Interleaver
Segmentation and reassembly of packets	Receive Interleaver at RX
ACK packet transmission	TX and RX Buffering
Selective packet acknowledgement (packet segment number not greater than 128)	TLC (Transmit Level Control)
	Waveform preamble
	Fill bits in last interleaver block
	TX and RX Waveform processing

2.4 THE STANAG 5066 PROTOCOL STACK

The STANAG 5066 protocol stack is made up of four layers, the SIS (Subnetwork Interface Sublayer), the CAS (Channel Access Sublayer), the DTS (Data Transfer Sublayer) and the MAN (Management Sublayer). In future a STANAG 5066 node implementation will only be referred to as a *node*.

2.4.1 Data Transfer Sublayer (DTS)

The main function of the DTS is to segment and reassemble D_PDUs and to ensure that the D_PDU has been received by the remote node. The DTS ensures error-free reception of D_PDUs over the link by using selective repetition of D_PDUs received in error. The DTS thus implements the ARQ function provided by an STANAG 5066 node.

2.4.1.1 Data delivery mode

A number of different data delivery modes have been defined in the DTS. Data can be sent in the ARQ or non-ARQ mode. In the non-ARQ mode the D_PDU is not acknowledged by the receiving node; thus there is no indication to the sender of correct D_PDU reception. In the ARQ mode the D_PDU can be set for:

- Client or node confirmation: in node confirmation the node confirms that the packet was received by the remote node, while in client confirmation mode the destination client also confirms correct reception of sent packets. This allows an application level acknowledgement between peer applications.
- Regular or expedited data service.
- Delivery C_PDUs to the remote peer C_PDU layer: the PDUs sent to the remote node are not intended for a client application but the remote node CAS. These messages are related to physical link setup and tear down.
- In-order or out-of-order D_PDU delivery: packets can either be delivered to the client application in the order received, this is called out-of-order delivery, or the node can wait for all packets in sequence to be received before delivering those packets to the client.

A C_PDU received by the DTS will be segmented into a number of D_PDUs with a D_PDU frame size no larger than 1023-bytes. A large C_PDU will be divided into a number of smaller D_PDUs. The sequence numbers for a segmented C_PDU placed in D_PDU frames will be consecutive, i.e. if the first C_PDU segment has sequence number *x* then the next C_PDU will use sequence number $x+1$. In order to reassemble a C_PDU the sequence number, C_PDU start flag and the C_PDU end flag will be used.

Segmented C_PDUs that are transmitted using non-ARQ mode will have the same C_PDU ID (Identification Number). Each D_PDU will contain an offset value that will indicate the data offset of the current C_PDU in the total C_PDU segment. The re-assembly will use the C_PDU ID, the C_PDU segment offset, segment-size and C_PDU size to reassemble the final C_PDU. When the C_PDU is completed the DTS will indicate which D_PDU data sections of the C_PDU were received in error and ask the remote node to resend those frames.

In the non-ARQ mode the D_PDU is set for:

- regular data service;
- expedited data service.

The D_PDU can be set to ARQ or non-ARQ on a per D_PDU basis.

2.4.1.2 D_PDU structure

The structure of a D_PDU can be seen in Figure 2.3.

SYNC Sequence (16 bits)	D_PDU Header	User data (max of 1023 bytes)	Data CRC

Figure 2.3: D_PDU structure.

The synchronization (SYNC) sequence used is a 16-bit Maury-Styles sequence: (MSB) 1110 1011 1001 0000 (LSB) 0xEB90. These bits are then mapped according to the standard CCITT V.42 and the least significant bit of the LSB of byte 0 will be transmitted first.

The D_PDU Header can be seen in Figure 2.4.

Type	EOW (Engineering Order Wire)	EOT (End of Transmision)	Size of Address	Size of Header	Header CRC	Address Information
4 bits	12 bits	8 bits	3 bits	5 bits	16 bits	0 - 56 bits

Figure 2.4: D_PDU Header fields.

The D_PDU header fields are:

- **Type:** the type represents a number that will indicate the type of D_PDU received (see Table 2.2).
- **EOW (Engineering Order Wire):** the engineering order wire bits are the first 12-bits of a MANAGEMENT D_PDU message. The EOW messages are most commonly used to carry the messages for data rate change or frequency change. The EOW message is also used to carry node configuration information. This information will inform a remote node of the capabilities of a local node, including whether the node is capable of performing data rate change. In order to perform data rate change the node has to be able to control the modem i.e. set the waveform type, data rate and interleaver size.
- **EOT (End of Transmission):** the EOT field is an approximation made by the local node to indicate to the receiving node, through a D_PDU, of how much time remains in the transmission.
- **Address information:** half of all address bits will indicate the local (source) 5066 node address while the other half will contain the remote node address (destination) address. The maximum address length for a STANAG 5066 address is 28-bits.
- **CRC (Cyclic Redundancy Check):** the header CRC will be calculated using a 16-bit CRC checksum.

Table 2.2 shows the D_PDU frame types and their functions. The DATA-ONLY, DATA-ACK and EXP-DATA-ONLY D_PDUs contain transmit sequence numbers that identify the D_PDU in that transmission window. The data portion of the D_PDU also contains a CRC that is only calculated on the data portion of the frame and not on the header. The ACK and

DATA-ACK D_PDUs contain a selective acknowledgement number that will acknowledge all correctly received D_PDUs of the current transmission window. The RESET/WIN-RESYNC D_PDU will control the synchronisation between nodes. The MANAGEMENT frame, used to send control information between nodes, has its own TX and RX frame numbers to control acknowledgement. WARNING D_PDUs are sent to indicate some kind of error in transmission. Reasons for an error could be:

- unrecognised D_PDU type;
- receiving a valid D_PDU in the wrong receive state.

Table 2.2: D_PDU Frame types

Type	D_PDU Frame	Function
0	DATA-ONLY	Simplex data transfer
1	ACK-ONLY	Acknowledgement of type 0, 2 data transfer
2	DATA-ACK	Duplex data transfer
3	RESET/WIN-RESYNC	Reset/Re-synchronise peer protocol entities
4	EXP-DATA-ONLY	Expedited simplex data transfer
5	EXP-ACK-ONLY	Acknowledgement of type 4 data transfer
6	MANANGEMENT	Management message transfer
7	NON-ARQ DATA	Non-ARQ data transfer
8	EXP NON-ARQ DATA	Expedited non-ARQ data transfer
9-14	-	Reserved for future extensions
15	WARNING	Unexpected or unrecognised D_PDU type

2.4.1.3 Node addressing

STANAG 5066 addresses are very similar to IP addresses and are expressed in the form MSB (Most Significant Bit) **w.x.y.z** LSB (Least Significant Bit), where there are only 28-bits available for the address. Thus the **w** value can be range from 0 to 15 while the **x**, **y** and **z** values can range from 0 to 255. Addresses do not necessarily have to be 28-bits and can be sent as 4-bits or as 1, 1.5, 2, 2.5, 3 and 3.5 (28-bits) bytes.

2.4.1.4 Sequence numbers

Each node implements a sending and receiving flow-control window that contains the sequence numbers of the sent, acknowledged and unacknowledged D_PDUs for both sending and receiving functions. Sequence numbers are assigned ascending, modulo 256, and a particular sequence number can only be reused once the original sequence number has been acknowledged. A node can therefore transmit a maximum of 128 unacknowledged frames.

2.4.1.5 Asynchronous vs. hi-speed asynchronous

In normal asynchronous communication the start and end of a data byte is indicated using a start and stop bit. Thus, when sending an 8-bit packet to a remote node one start bit and one or two stop bits will be sent along with the actual data bits to the receiver. This introduces a significant amount of overhead into each transmitted byte. Hi-speed asynchronous mode is where data is sent asynchronously to the HF modem for transmission but the start and stop bits are removed and not sent along with the data. The start and stop bits are then added again by the receiving modem when delivering the data to the receiving node.

2.4.2 Subnet Interface Sublayer (SIS)

There exists a client-server relationship between the users of the HF subnetwork and the HF subnetwork itself. In this case the STANAG 5066 node is the server and STANAG 5066 clients will request services from the node. These services are transparent from the server (node) perspective. All clients are equal as far as the node is concerned and a client can only be distinguished through his SAP number. Each client must connect to the node using a unique SAP ID that has not already been used by another client.

The SIS is responsible for managing the establishment and termination of sessions. There are three types of sessions:

- soft-link data exchange;
- hard-link data exchange;

- broadcast data exchange.

2.4.2.1 Soft-link data exchange session

A soft-link data exchange session will be started immediately when there is data available for transmission to a remote STANAG 5066 node. There is no soft-link setup procedure and the node can manage a number of soft-link sessions to the same or another node at the same time. A timeout is associated with the soft-link session and the session can only be terminated if this timeout value expires or if the physical link over which this soft-link exists and that is managed by the CAS of that node, is terminated. A client cannot request the setup or tear down of the soft-link data exchange session.

To explain the difference between a physical channel and a session: when a client wishes to send an email to a remote client a physical link is established by the CAS; then a soft-link session is set up over this physical link. The soft-link session is between the local and remote client while the physical link is between the local and remote node. Every client of the local node can establish a new soft-link session to a remote client over the same physical link.

2.4.2.2 Hard-link data exchange session

The hard-link data exchange session must be requested by a STANAG 5066 client. A client will use a hard-link to ensure that a particular link is maintained and/or to reserve capacity and quality of service on the link. There are three types of hard-link sessions:

- Type-0 (physical link reservation): this type of hard link will maintain the physical link between two nodes. Other clients can also use the current hard-link but soft-links to new nodes will not be created.
- Type-1 (partial bandwidth reservation): this type of hard-link will also maintain a physical connection between two nodes. Here the client that requested the hard-link at the local node will be the only client that can use the hard-link. Thus no other local client can use this link. Any of the remote clients can create a soft-link with a local client other than the local hard-link client.

- Type-2 hard-link (full-bandwidth reservation): here a physical link is maintained between two nodes. However only the local hard-link client and the specified remote client can use this physical link. No other local or remote clients can use the physical link.

2.4.2.3 Broadcast data exchange session

A broadcast session is started when a client wants to transmit non-ARQ data to a remote node where no physical link or session currently exists. In the broadcast data exchange session no physical link is required for communication.

2.4.3 Channel Access Sublayer (CAS)

The Channel Access Sublayer (CAS) will provide services to the Subnetwork Interface Sublayer (SIS) and the Data Transfer Sublayer (DTS). The CAS will perform the following functions for the SIS:

- The CAS will *make* or *break* physical links to remote nodes. All hard-links and soft-links have to be created over an existing physical link. The broadcast data exchange session does not require a physical link.
- The CAS has to inform the DTS and SIS as to the state of a physical link.
- The CAS will accept S_PDU (SIS Protocol Data Units) from the SIS for transmission over the physical link.
- The CAS will deliver S_PDU (SIS Protocol Data Units) received over the physical link to the SIS.

There are two types of CAS physical links:

- Exclusive physical links: these physical links are used to support the handling of hard-link data exchange sessions. A STANAG 5066 node can establish at most two exclusive physical links.

- Non-exclusive physical links: these physical links are used to support soft-link data exchange sessions. A node can have more than one non-exclusive physical link at a time.

There are two types of channel access protocols used by the CAS to establish physical links:

- CAS protocol type 1: Normal
- CAS protocol type 0: ALE

2.4.3.1 CAS protocol type 1

A type 1 channel access connection is a point-to-point connection between a pair of nodes on a common frequency where the frequency selection function is not under the control of the STANAG 5066 node or ALE. One of the main objectives of the CAS is the managing, i.e. *making* and *breaking*, of physical links between two nodes on a common frequency, where the frequency has been selected by an external frequency management and selection function. This external frequency selection function can be performed, in terms of the type 1 CAS protocol, by a human operator or any other external process. However, there is no management or interface between the node and the external frequency selection process.

2.4.3.2 CAS protocol type 0

CAS protocol type 0 is where there is an interface between the frequency selection function and the STANAG 5066 node, where the node is in control of the frequency selection. ALE [11] is the current standard used by the node to manage the frequency selection process.

2.4.3.3 ALE (Automatic Link Establishment)

Until a few years ago it was up to a human operator to determine the quality of a channel. Propagation models have also been used to try to predict channel conditions at particular times and so simplify the frequency selection function. This resulted in a need for an adaptive or *smart* mechanism to dynamically determine the best available channel to a remote node and to react to changing channel conditions.

ALE or Automatic Link Establishment ([11] and [15]) is a process where a controller controls a HF transceiver to dynamically establish the highest quality communications link with one or more HF radio nodes. In order to select the best operating frequency ALE must implement the function of HF frequency management. Frequency management involves the selection of the best operating frequency for an immediate message. The best operating frequency will depend on the following factors:

- **Path loss:** loss of signal strength due to path propagation characteristics.
- **Noise:** strength of interference and noise on the channel.
- **Dispersion:** dispersion is caused by the fact that frequency components have different propagation characteristics and are dependent on the radio frequency employed. Dispersion may lead to received signal distortion.
- **Multipath:** multipath arises when the received signal is non-dispersively distended by multiple reflections from the ionosphere with or without ground reflections or the received signal is made up of a superposition of multiple near-same amplitude signals. Multipath results is ISI (Intersymbol interference), which is the most serious effect of Multipath on a system. Multipath occurs due to scatter of ionization energy within the same layer, multiple ionosphere bounces from a number of layers or reflections from one or more layers of the atmosphere.

ALE controllers function on the basic principle of LQA (Link Quality Analysis) and *sounding*. LQA is the measure of signal quality between two stations based upon the BER, the SNR and distortion rate. Each ALE controller will have a set of predefined frequencies that it will use. Each of these frequencies, a TX and RX frequency, represents a channel. These channels are then scanned either two or five times a second. Each ALE controller also has its own call-sign called a *Self-Address*. The controller will *sound* a particular channel to determine current channel quality. ALE controllers will also detect and log calls from remote ALE stations. Channels are then continuously ranked according to channel quality.

When one ALE station wishes to create or *make* a link with another ALE station, a call is made over a particular channel. The ALE station will continue to try to *call* the remote station for a configurable number of times before failing. If the remote ALE station responds, a link is created between the two stations.

The main benefits of ALE are that:

- ALE can dynamically log current channel conditions;
- ALE can decide the best channel for communication;
- ALE can automatically search for a remote node and dynamically make and break the link without operator assistance.

A STANAG 5066 node that has data to transmit to a remote node will use the CAS to establish a physical link to the remote node. The CAS through the MAN layer will interface with an ALE process to create the physical link. The MAN layer will convert the 5066 address to an ALE address, which will be used by the ALE process to *call* the remote node.

2.5 STANAG 5066 CLIENTS

The interface protocol between clients and the subnetwork is TCP/IP (Figure 2.5). Each client will have a unique SAP ID and TCP/IP port number. A list of STANAG 5066 clients and their associated SAP IDs can be seen in Table 2.3.

The advantage of using TCP/IP is that a subnetwork client does not have to be located on the same PC as the STANAG 5066 node. TCP/IP also provides reliable connection-orientated communication to the subnetwork and traffic can be multiplexed to a subnetwork client because each client is uniquely identified by its SAP ID, IP address and port number.

The STANAG 5066 client requirements can be seen in [8] Annex F: HF subnetwork client requirements. For a discussion surrounding the implementation of client applications that use STANAG 5066, see [16].

Table 2.3: SAP IDs of STANAG 5066 clients

STANAG 5066 Client Application	SAP-ID
Subnet management client	0
Character-Oriented Serial Stream (COSS) Client	1
STANAG 4406 Annex E – Tactical Military Message Handling (T-MMHS) Client	2
HMTP (HF Mail Transfer Protocol)	3
HFPOP (HF Post-Office Protocol)	4
Operator order wire (HFCHAT)	5
Reliable Connection-Oriented Protocol (RCOP)	6
Unreliable Datagram Oriented Protocol (UDOP)	7
PPP (Point-to-Point Protocol) client	8
IP client	9
RESERVED	10-11
Compressed File Transport Protocol (CFTP)	12
UNASSIGNED	13-15

Figure 2.5 illustrates the STANAG 5066 clients and their connection to the subnetwork.

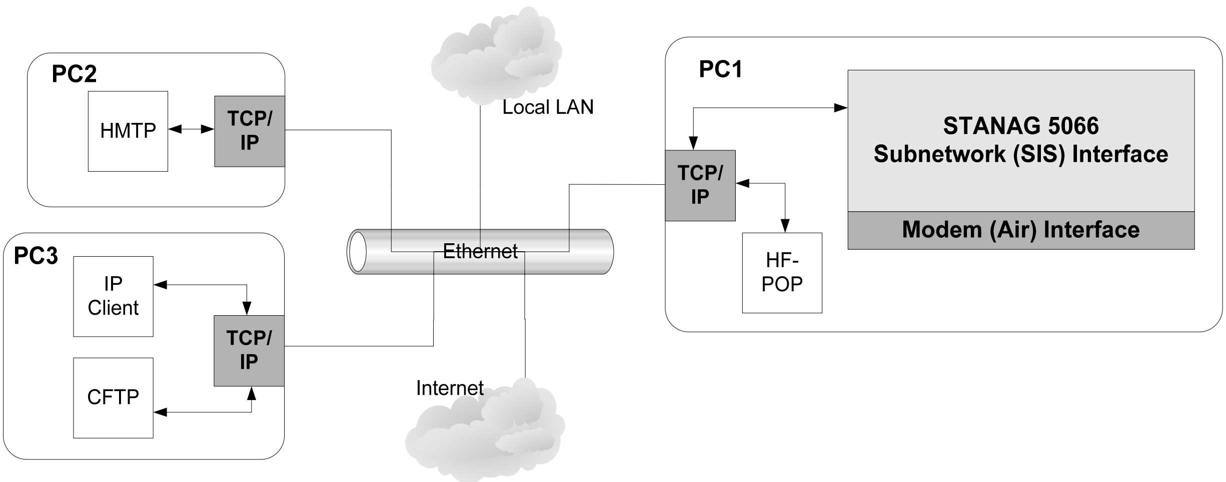

Figure 2.5: STANAG 5066 client connected to subnetwork via Ethernet LAN.

2.5.1 Subnet management client

The subnetwork management client will use the S_MANAGEMENT_REQUEST, S_MANAGEMENT_CONFIRM and S_SUBNET_AVAILIBILITY [1] messages for communication with the subnetwork. The communication and exchanges between the management client and the STANAG 5066 node should be based upon SNMP (Simple Network Management Protocol).

2.5.2 HMTP (HF Mail Transfer Protocol)

SMTP (Simple Mail Transfer Protocol), specified in RFC 821 [17], defines a one-command/response type protocol used to transfer an email between a SMTP client and server. The structure of the SMTP protocol does not lend itself to implementation over HF because of the long delays between commands and responses. A normal SMTP command/response dialog could contain as many as fourteen message exchanges before an email is transferred. Internet Standard 60 (RFC 2920 [18]) "SMTP Pipelined Service Extensions" group these client and server messages together. This considerably reduces the number of data exchanges required to transfer an email. HMTP implements the pipelined extensions to exchange emails using the SMTP protocol over the STANAG 5066 subnetwork.

All HMTP client data will use the following service options:

- the transmission mode is ARQ;
- the delivery confirm option can be NODE CONFIRM or CLIENT CONFIRM;
- the D_PDU delivery option shall be IN ORDER DELIVERY.

2.5.3 HF-POP (HF-Post Office Protocol)

HF-POP is the HF version of the common POP3 (Post Office Protocol version 3 [19]) used on the Internet. HF-POP uses the optional features of RFC 1939 and RFC 2449 [20] for more efficient use of the current communications channel. These features include PIPELINING, POP3 Extension Mechanism (RFC 2449) and 8-BITMIME.

All HF-POP3 clients will use the following service options:

- the transmission mode is ARQ;
- the delivery confirm option can be NODE CONFIRM or CLIENT CONFIRM;
- the D_PDU delivery option will be IN ORDER DELIVERY.

2.5.4 RCOP (Reliable Connection Orientated Protocol)

The RCOP protocol is used to create a reliable connection between applications. The RCOP protocol thus sits on top of the SIS when viewing the STANAG 5066 stack and implements the transport layer function of the OSI layer. RCOP allows multiple applications to be multiplexed over the same STANAG 5066 client. The multiplexing is achieved by adding a header to the application data that is passed down to the STANAG 5066 node. The RCOP datagram can be seen in Figure 2.6 and a description of the fields of the RCOP header can be seen in Table 2.4.

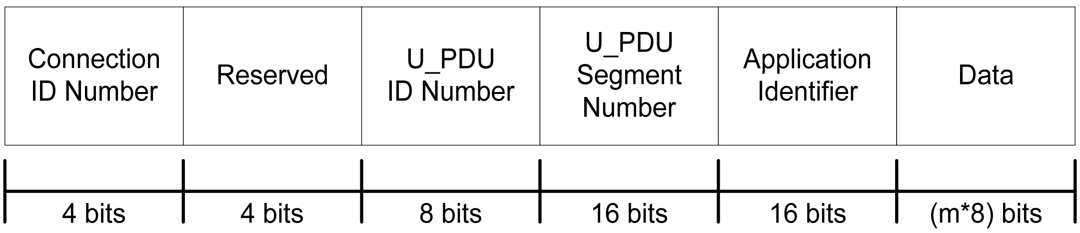

Figure 2.6: The RCOP datagram.

Table 2.4: RCOP datagram field description

Header fields	Description
Connection ID Number	RCOP will multiplex transport services for an application by providing multiple connections. Each connection will have a connection ID.
U_PDU ID Number	Each U_PDU (User Protocol Data Unit) created for the connection shall have successive connection IDs
U_PDU Segment Number	Segment numbers shall be assigned consecutively within one U_PDU
Application Identifier	Identifies the specific application that uses RCOP.

At each lower layer of the of the STANAG 5066 protocol stack a number of fields are added to the original application data before being sent by the subnetwork. The encapsulation process can be seen in Figure 2.7.

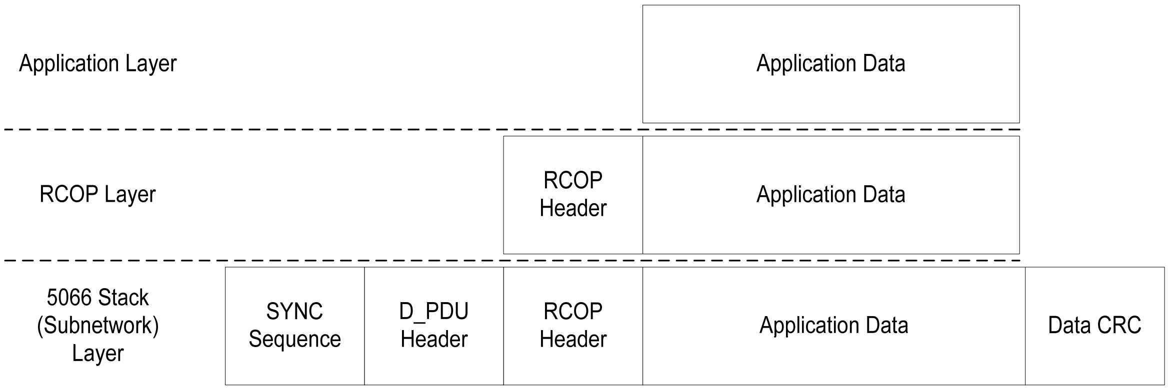

Figure 2.7: RCOP data encapsulation process.

All RCOP clients will use the following service options:

- the transmission mode is ARQ;
- the delivery confirm option can be NODE CONFIRM or CLIENT CONFIRM;
- the D_PDU delivery option will be IN ORDER DELIVERY or AS THEY ARRIVE.

2.5.4.1 RCOP segmentation and reassembly

Application data should be segmented so that the RCOP packet given to the subnetwork (at the RCOP layer in Figure 2.7) is smaller than the subnetwork MTU (Maximum Transmission Unit) size. The connection ID, U_PDU (User Protocol Data Unit) ID and segment number are used to reassemble the original application data. The source address, SAP ID along with the RCOP header fields uniquely identifies a RCOP datagram completely.

2.5.5 UDOP (Unreliable Datagram-Connection Orientated Protocol)

UDOP also provides multiplexing services over the subnetwork for multiple simultaneous client applications that use the broadcast exchange session. UCOP datagrams are similar to RCOP datagrams (see Figure 2.6).

All UDOP clients will use the following service options:

- transmission mode is non-ARQ;
- the delivery confirm option is none;
- the D_PDU delivery option will be IN ORDER DELIVERY or AS THEY ARRIVE;

2.5.6 BFTP (Basic File Transfer Protocol)

The BFTP can be used to transfer files over the subnetwork. BFTP makes use of the services provided by the RCOP to segment and reassemble file packets. The BFTP PDU can be seen in Figure 2.8.

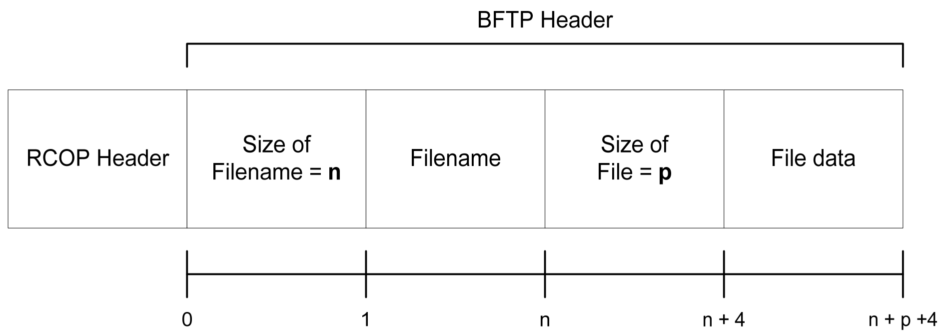

Figure 2.8: BFTP PDU structure.

2.5.7 FRAP (File-Receipt Acknowledgement Protocol)

To ensure that files sent using BFTP are indeed correctly received by the remote client, the FRAP can be used. The receiving client, of a file transferred using the BFTP, will when the entire file has been received; send a RCOP datagram with the application identifier set to 0x100B and the same connection ID as the original BFTP connection ID.

The FRAP has to two versions; the version described in the previous paragraph is known as version 1. FRAP version 2 also includes a copy of the received BFTP header in the RCOP PDU sent by the receiving node. This allows the sender to verify not only that a file was received by the remote node, but also that the correct file was received and that the received file is the correct size i.e. that no data was lost in transmission.

2.5.8 IP Client (Internet Protocol Client)

The IP client is able to send datagrams to and receive datagrams from the subnetwork in both ARQ and non-ARQ mode. The IP client is able to support point-to-point and point-to-multipoint connections and IP datagram packets are assigned service requirements on per packet basis.

An example of an IP client interfacing with a STANAG 5066 subnetwork can be seen in Figure 2.9. Datagrams from across the IP network will enter on the LAN (Local Area Network) interface card and datagrams destined for the HF network are then routed to the IP client driver using the local PC routing table and the ARP (Address Resolution Protocol [21]). The IP client will then segment received IP packets from the IP driver and deliver those packets for transmission to the node. On the receiving side, received packets from the node are reassembled by the IP client and delivered to the IP driver for routing by the PC TCP/IP stack. The IP client will also implement the ICMP (Internet Control Message Protocol [22]), which allows users of the IP client to determine network connectivity and identify network problems.

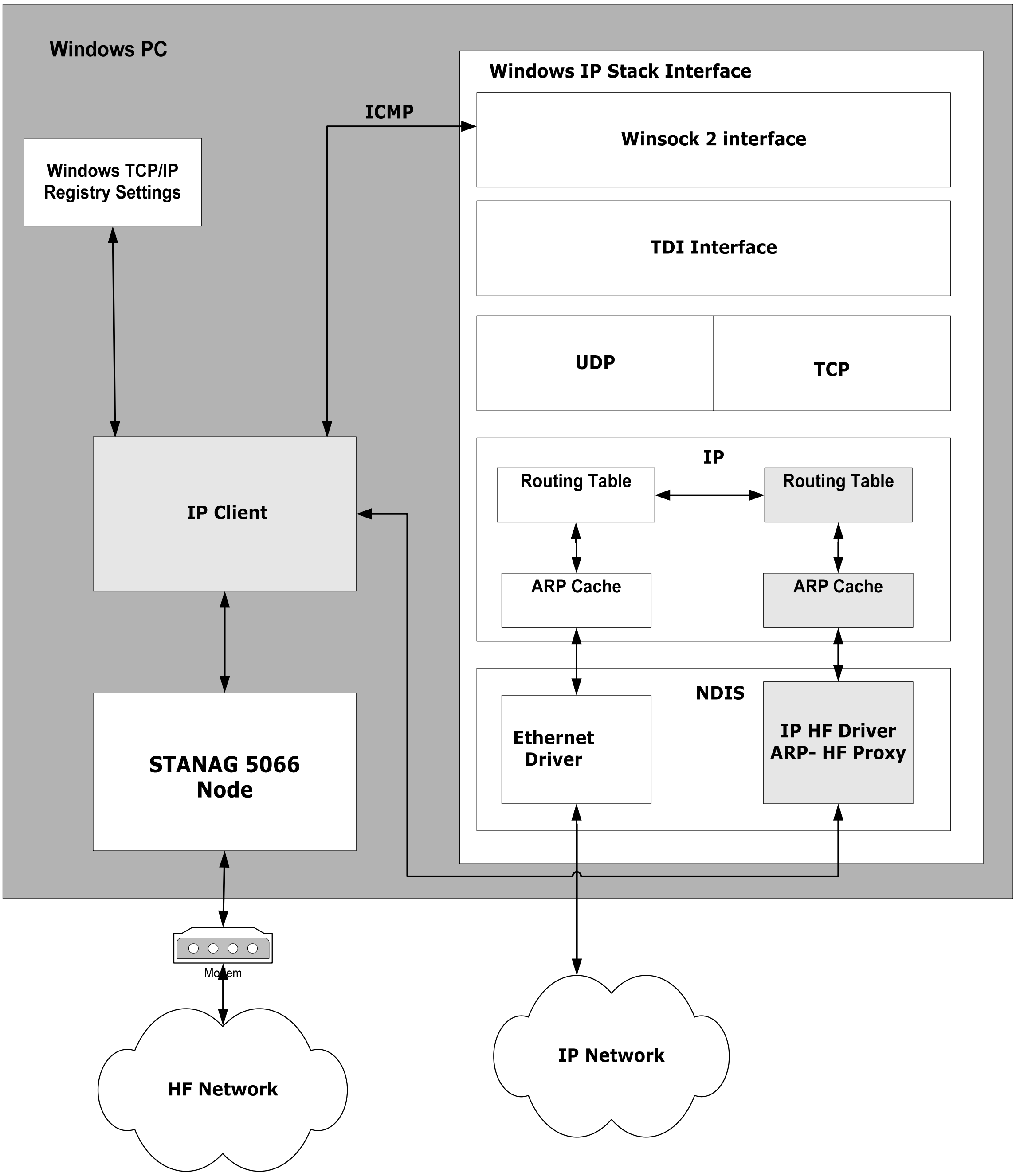

Figure 2.9: Example of an IP client implementation.

2.5.8.1 IP datagram QoS (Quality of Service) determination

The IP header TOS field (Type of Service, see Table 2.5) labels are used to determine the STANAG 5066 service options. IP clients are also able to define service options based upon DiffServ [23] (Differentiated Services Code Points or DSCP, see Table 2.6).

Table 2.5: IP TOS mapping to STANAG 5066 service options

IP-address Type	Delay	Throughput	Reliability	Cost	Delivery Mode
Multicast	X	X	0	X	Non-ARQ
Multicast	X	X	1	X	Non-ARQ
Unicast	1	0	0	0	Non-ARQ
Unicast	0	1	0	0	ARQ
Unicast	0	0	1	0	ARQ
Unicast	0	0	0	1	Non-ARQ
Unicast				X	
• **X – don't care state**					

Table 2.6: DiffServ DSCP mapping to STANAG 5066 service options

DSCP	IP Precedence	DSCP	5066 Priority Name	5066 Priority Value	Delivery Mode
111xxx	Network		Network	1110	ARQ/non-ARQ
110xxx	Internet		Internet	1100	ARQ/non-ARQ
101xxx	Critical	EF	Expedited	1010	ARQ/non-ARQ
100xxx	Flash-Override	AF4	Flash-Override	1000	ARQ/non-ARQ
011xxx	Flash	AF3	Flash	0110	ARQ/non-ARQ
010xxx	Immediate	AF2	Immediate	0100	ARQ/non-ARQ
001xxx	Priority	AF1	Priority	0010	ARQ/non-ARQ
000xxx	Routine	Default DSCP	Routine	0000	ARQ/non-ARQ
• **Reserved 5066 priority values not shown** • **Multipoint = non-ARQ** • **Point-to-Point = ARQ**					

2.5.9 PPP Client (Point-Point-Protocol Client)

PPP (Point-to-Point Protocol) is a well defined protocol often used as bearer for internetwork protocols. PPP is defined in RFC 1548 [24]. PPP PDUs will be constructed using the 2-byte protocol field as defined in RFC 1548, but HDLC (High-level Data Link Control) framing characters will not be included in the implementation.

PPP clients will use the following service options:

- the transmission mode is ARQ;
- the delivery confirm option is NODE CONFIRM or CLIENT delivery confirm;
- the D_PDU delivery option will be IN ORDER DELIVERY.

2.5.10 CFTP (Compressed File Transfer Protocol)

The CFTP is used to send a compressed email over the subnetwork from one CFTP Client to another, see Figure 2.10.

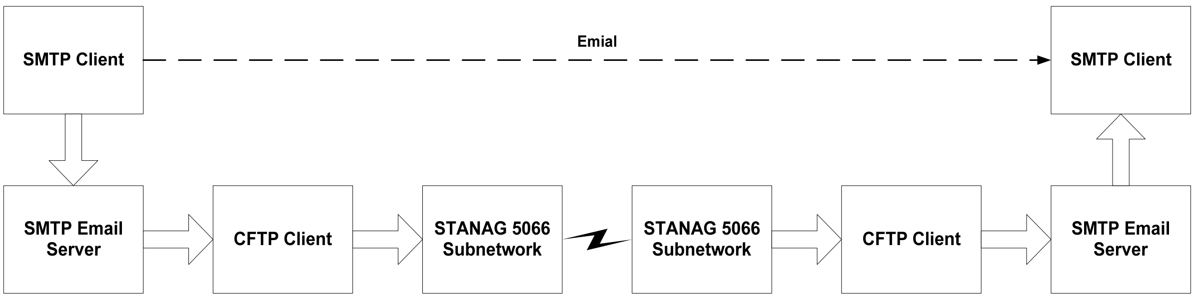

Figure 2.10: Email transfer using CFTP.

An email will be received, by the CFTP client from the SMTP server, compressed and delivered using the BFTP and RCOP to a remote CFTP client. CFTP shall use compression as defined in RFC 1950 [25], 1951 [26] and 1952 [27] (also called GZip compression). Figure 2.11 shows the CFTP email file structure.

Message ID	LF	Recipient List	LF	Message Size	LF	Message Data

LF - Line Feed

Figure 2.11: CFTP email file structure.

The CFTP email is made up of:

- **Message ID:** unique ID of the current email that will be used as the filename for the decompressed email file at the remote node
- **Recipient list:** this string will contain all the recipients of the email
- **Message size:** size in bytes of the message data
- **Message data:** email message body

The CFTP client receives the email from the SMTP server. The CFTP email file is constructed as in Figure 2.11. The email file is then compressed using GZip, encapsulated into BFTP packets, a RCOP header is added and then the RCOP data is submitted to the subnetwork for transmission.

CHAPTER 3 LITERATURE STUDY

3.1 INTRODUCTION

This chapter will detail the results of a literature study conducted to examine previous implementations of a STANAG 5066 data rate change algorithm, the implementation and workings of the DRC procedure as stated in STANAG 5066 as well as current research areas applicable to a STANAG 5066 HF data communication system.

3.2 STANAG 5066 DRC

3.2.1 Introduction

The objective of DRC is to select the modem data rate and interleaver size to maximize the data throughput on a communication link between two nodes. The following affects the data throughput: the selected waveform, the modem data rate and the interleaver size. The processing, buffering and encoding delays are ignored, because these are not under the control of the DRC algorithm.

3.2.2 DRC procedure

The DRC procedure is implemented in the DTS and requires that this sublayer include a method of modem control. The modem control should include methods to modify the modem waveform settings as well as receive current HF channel condition information including the SNR, BER, Doppler spread, etc.

It is up to the receiving node to start the DRC procedure because this node can determine the optimum rate that data should be sent to it. The DRC mechanism must be initiated and is controlled by the receiving node, not the sending node. To clarify: a DRC algorithm at a local node sets the TX data rate of the remote node for the next transmission interval, based upon events and information in the previous transmission interval. DRC messages are exchanged in the MANAGEMENT state and the initiating node is known as the DRC master. The DRC handshake procedure between two nodes using a non-autobaud waveform, like STANAG 4285, can be seen in Figure 3.1.

For autobaud waveforms, like STANAG 4539 and MIL-STD 110A, the DRC procedure only requires one step, called a quick data rate change. For an autobaud waveform, when one node issues the DRC request the receiving node, if it agrees with the new data rate and interleaver, immediately starts sending data at the new rate and interleaver setting, without using the handshaking mechanism.

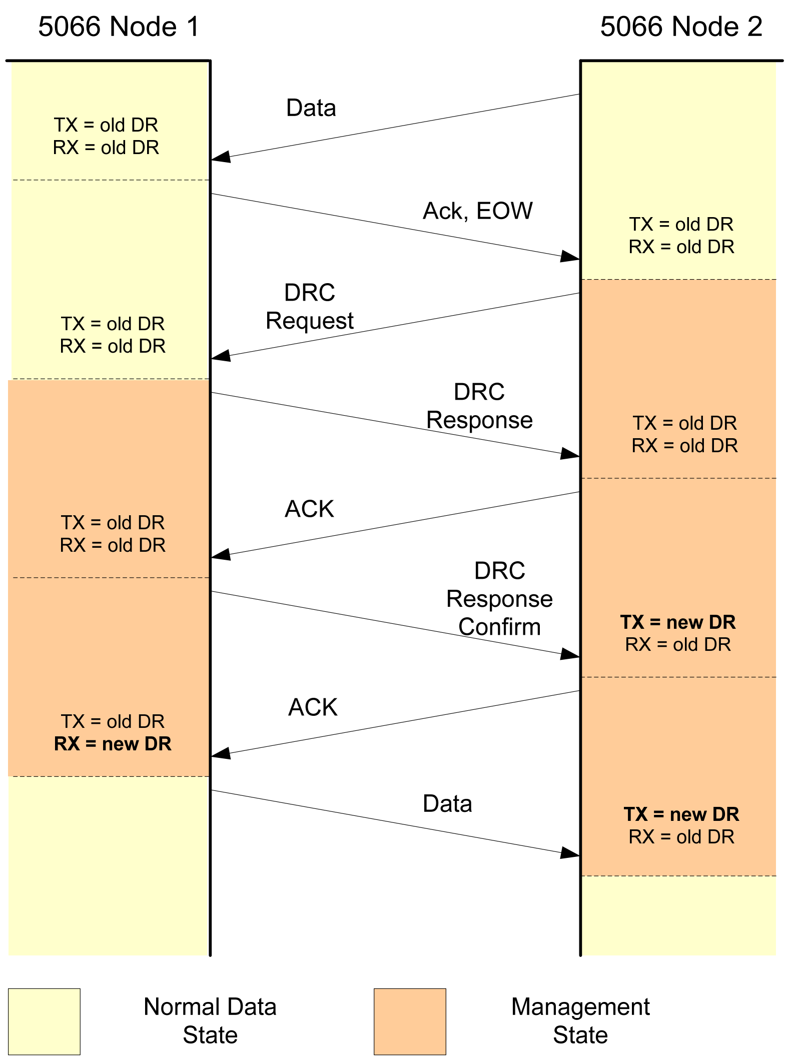

Figure 3.1: DRC procedure between two nodes using a non-autobaud waveform.

3.2.3 Interleaver selection

When selecting the data rate and interleaver setting for HF data communication, it is important to remember that the selection of the interleaver will affect the FER as well as the transmission latency. Latency is defined as *dead time*, or the time taken to move from transmission to reception. A longer interleaver will decrease the FER but will also increase the latency experienced on the communications link. Both of these parameters, the latency and FER, will also affect the data throughput.

3.3 DRC AND INTERLEAVER SELECTION ALGORITHMS

3.3.1 DRC algorithm requirements

In [12], Trinder and Brown describe the requirements for a DRC algorithm. These requirements are:

- the algorithm should facilitate data throughput maximisation;
- the algorithm should avoid unnecessary data rate changes;
- the algorithm should adapt to rapidly changing channel conditions;
- the algorithm should minimise the time taken to reach optimum data rate, this is especially important for non-autobaud waveforms that have to use a long handshaking procedure each time a data rate is changed;
- the algorithm should be robust i.e. a change to a new data rate should not break the current communications link.

3.3.2 DRC algorithms for non-autobaud waveforms

3.3.2.1 Introduction

Low data rate waveforms are waveforms with data rates that vary from 75 to 2400 bps. Waveforms that fall into this category are the non-autobaud STANAG 4285 waveform and the autobaud MIL-STD 110A waveform. Only the DRC algorithms for the STANAG 4285 waveform will be examined in this section.

One of the first data rate change algorithms designed was by Trinder and Brown [12] and the goal of their DRC algorithm or mechanism was to optimise the current modem data rate based upon current channel conditions as to maximise the data throughput. The primary goal of the article in [12] was to serve as a guideline for implementers of STANAG 5066. The data rate change algorithm described uses the FER to decide to increase or decrease the modem data rate and focuses primarily on the STANAG 4285 non-autobaud waveform.

Another goal of a DRC algorithm is to counter channel condition changes over a long time interval. It would thus be of little use to try to select the data rate for a channel condition change that takes place over a time that is similar to the time taken to change the data rate and exchange the necessary handshaking messages required for non-autobaud waveforms. The approximated time taken to change the data rate for a non-autobaud waveform using handshaking can be seen in Table 3.1.

Table 3.1: Approximate time taken to implement DRC per baud rate [12]

Data Rate (bps)	Modem Delay (s)	Transmission time (s)	Calculated DRC time (s)
75	3.2 - 5.6	0.853	20 - 32
300	1.9 - 2.2	0.853	14 - 15
1200	1.3 - 1.5	0.853	11 - 12

Thus the data rate change mechanism should only try to counter changes in the channel that take place over a longer time period than the time in Table 3.1

3.3.2.2 DRC algorithm design and implementation

The simple DRC algorithm designed by Trinder and Brown uses the measure of the received FER to select the optimum data rate ([12], [28] and [29]). This algorithm states that if the FER, i.e. the number of received frames in error over the total number of received frames, is above 50% then the data rate should be halved. In contrast, if the received FER is zero then the received data rate should be doubled, i.e. from 300 to 600 bps or 600 to 1200 bps. This is a very simple algorithm that is also very easy to implement. The DRC algorithm was evaluated using three different FER decision threshold values. These threshold values can be seen in Table 3.2. The FED STD 1052 [28] proposes that the rate increases when the FER is 0 and decreases when the FER is above 50%. Trinder and Brown do not express any opinion as to the mechanisms that should be used to determine the optimum interleaver size.

Table 3.2: FER threshold values used by Trinder and Brown.

ALGORITHM NO.	LOW THRESHOLD	HIGH THRESHOLD
Algorithm 1	0	> 50
Algorithm 2	< 20	> 50
Algorithm 3	< 40	> 60

Using three threshold pairs, see Table 3.2, the DRC algorithm was evaluated to determine which threshold pair would deliver the best performance based upon the following criteria:

1) **% of successful changes:** this criterion will attempt to determine whether the thresholds have been chosen successfully. Incorrect thresholds will result in a large number of failed DRC attempts. A failed DRC attempt is when the data rate is changed to a rate that results in a lost connection.
2) **% of DRC messages received without error:** as the channel quality degrades the error percentage for normal data frames as well as for DRC (MANAGEMENT) frames will also increase.
3) **Data rate as a function of SNR:** this criterion will attempt to determine how closely the chosen data rate will track the prevailing channel conditions.

3.3.2.3 Problems encountered

One of the major problems encountered by Trinder and Brown in their DRC algorithm implementation is one of modem data rate oscillation. This is where the modem data rate is increased because the FER is zero and in the next transmission interval the FER, at the higher data rate, is greater than 50%, which causes the modem data rate to be lowered. This oscillating effect can continue indefinitely if channel conditions remain constant. The effect is especially prevalent in a Gaussian channel, which has very steep BER curves and thus causes a very sharp change in FER with a constant SNR.

Another problem encountered in [12] involves the time required to gather enough data to accurately estimate the FER as well as the fact that even if the FER gives a fairly good indication that the data rate should be increased, it does not indicate by how much the data rate should be increased. This means that the data rate will only be increased in small steps, a very time consuming and inefficient approach.

3.3.2.4 DRC algorithm design recommendations

Trinder and Brown express the opinion that a system that uses current channel conditions and better statistical estimates should be able to make better DRC decisions. The greatest problem with using such an approach is the different number of COTS (Common off the Shelf) HF modem implementations and modem interfacing capabilities offered by vendors. For example, certain manufacturers would not return any SNR or BER information to the user. As the FER is calculated by the STANAG 5066 node itself and not by the HF modem, an implementation that uses the FER as basis for making DRC decisions will be able to create a vendor independent DRC algorithm solution.

3.3.2.5 Testing and results

The testing of the DRC algorithm included OTA tests using COTS HF modems for a path of approximately 180 miles.

3.3.2.6 Conclusion

The OTA test results indicate that the best throughput was achieved using thresholds that increase the data rate when the FER is below 20% and decrease the modem data rate when the FER is above 50%. Trinder and Brown concluded that their simple algorithm does indeed provide a fairly reliable estimate of the optimum data rate, with the two major problems they encountered, as discussed previously being:

- data rate oscillations;
- the robustness of the DRC algorithm.

The advantage of their algorithm is that it is:

- very simple to implement;
- independent of vendor HF modem implementation.

3.3.3 DRC algorithms for autobaud waveforms

3.3.3.1 Introduction

Trinder and Gillespie [14] and Nieto [13] investigated DRC optimization and STANAG 5066 performance using the STANAG 4539 waveform [1]. STANAG 4539 is a high-data rate autobaud waveform that defines a family of waveforms containing data rates from 75 to 12800 bps (Table 3.3) and interleaver lengths as defined in Table 3.4.

Table 3.3: STANAG 4539 modulation and coding

Data Rate	Modulation scheme	Coding
75	Walsh	1/2
150	2-PSK	1/8
300	2-PSK	1/4
600	2-PSK	1/2
1200	4-PSK	1/2
2400	8-PSK	1/2
3200	4-PSK	3/4
4800	8-PSK	3/4
6400	16-QAM	3/4
8000	32-QAM	3/4
9600	64-QAM	3/4
12800	64-QAM	None

Table 3.4: STANAG 4539 interleaver sizes

INTERLEAVER	NO. OF DATA FRAMES	LENGTH IN SECONDS
Ultra Short (US)	1	0.12
Very Short (VS)	3	0.36
Short (S)	9	1.08
Medium (M)	18	2.16
Long (L)	36	4.32
Very Long (VL)	72	8.64

3.3.3.2 DRC algorithm design and implementation

Trinder and Gillespie defined a formula to determine the ARQ throughput for the channel (see equation 3.1). This formula is used to determine the ARQ data throughput, taking into account interleaver and modem latency as well as data retransmissions.

$$\begin{aligned} &ARQThroughput = \frac{TX\_Data}{\left[\left(Data\_TX\_Time \times ACK\_Time\right) + 2 \times Latency\right]} \\ &TX\_Data = 128 \times Packet\_size(bytes) \times Bits\_per\_byte \\ &Data\_TX\_Time = \frac{TX\_Data}{Modem\_Data\_Rate(bps)} \\ &ACK\_Time \approx Interleaver\_Time(s) \end{aligned} \quad (3.1)$$

Trinder and Gillespie state in [14] that pervious work on DRC algorithms, as found in [12] and [29], indicate that:

1. a simple FER based algorithm provides a reliable DRC estimate;
2. an algorithm that has a min FER of 50% and max FER of 20% for lowering and increasing the modem data rate performed well during OTA testing;
3. more complex algorithms that use channel information like the SNR should provide better results.

Trinder and Gillespie further state in [14] that the selection of the minimum FER of 50% for lowering the data rate is intuitive, because when the FER at 2400 bps is 50%, it effectively means that only 1200 bps data is being received without error and the data rate should therefore be lowered to 1200 which would produce that same throughput with 0% FER. This is only true for waveforms that have data rates that increase by a factor of two each time, as MIL 110A and STANAG 4285. STANAG 4539 waveform data rates, however, do not increase in this manner. For rates 75 – 2400 bps STANAG 4539 does follow this model; the higher rates, however, are: 3200, 4800, 6400, 8000 and 9600 bps.

Trinder and Gillespie further studied the ARQ throughput as a function of SNR (see Figure 3.2). The resultant graph can be used to determine the optimum data rate choice for a particular SNR value when the channel exhibits CCIR Moderate (Consultative Committee for International Radio) characteristics.

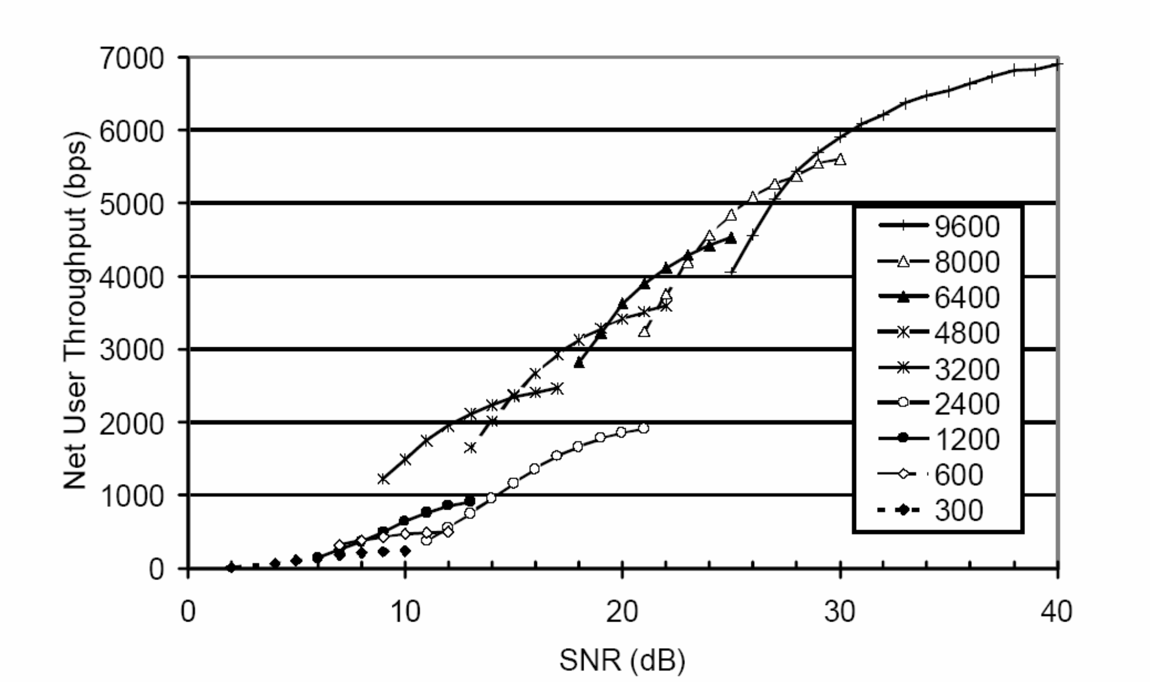

Figure 3.2: ARQ Throughput as a function of SNR (CCIR–Moderate) [14].

Table 3.5 indicates the optimum FER decision threshold values for DRC at every data rate. The min and maximum FER threshold values for data rates from 75 to 2400 bps remains the same (see Table 3.2). The 2400 bps data rate is not used in the algorithm implementation. When the data rate is increased from 1200 bps, the next higher data rate is 3200 bps, also when the data rate is lowered from 3200 bps the next lowest data rate is 1200 bps (see Figure 3.2).

Table 3.5: FER Threshold values used for DRC

DATA RATE	MINIMUM FER (Decrease Rate)	MAXIMUM FER (Increase Rate)
3200	50	10
4800	35	5
6400	20	5
8000	15	2
9600	5	N/A
12800	Not tested	Not tested

3.3.3.3 DRC algorithm design recommendations

Nieto [13] evaluated DRC using different packet sizes and varying SNR values over three types of channels. These channels were a CCIR Poor, Rician and AWGN (Additive White Gaussian Noise) channels. Nieto also states that the development of a DRC algorithm is quite complex due to the large number of variables involved. These include the message size, frame size, current channel conditions including SNR and BER, modem data rate and the interleaver size. Nieto produced a list of recommendations for the design of a DRC algorithm based upon his research (see Table 3.6).

Table 3.6: DRC algorithm recommendations

CASE STUDY	RECOMMENDATION
Packet Size and Interleaver	Smaller messages should be grouped together to increase message size. This increases channel utilization and throughput.
	For large messages it is better to use packet sizes that vary between 750 to 1000 bytes.
	Set all radios to the lowest AGC (Automatic Gain Control) setting.
	Only use long and short interleaver settings.
	Use long interleaver for fading channels and short interleaver for AWGN channels.
Data Rate Selection	Always be conservative when selecting data rates. Nieto found that a too optimistic data rate choice could result in significant performance loss.

Johnson [30] also produced a set of recommendations designed to improve the performance of a STANAG 5066 system. These recommendations are:

- select the initial data rate from the current measured SNR (see Table 3.7);
- adapt the packet size based upon the current modem data rate;
- track the data rate of the sending node, i.e. the receiving node modem TX data rate should be no less than half the sending node TX data rate.

Table 3.7: Initial data rate per SNR and message size

SNR	Size < 1500 bytes	Size > 1500 bytes
< 5 dB	75 bps	300 bps
< 15 dB	1200 bps	600 bps
< 25 dB	3200 bps	1200 bps
< 35 dB	3200 bps	4800 bps
> 35 dB	3200 bps	9600 bps

3.3.3.4 Interleaver selection

The length of the interleaver has an effect on the FER, as the interleaver will counter fades found in the HF communication channel. The choice of which interleaver to use is a trade-off between the latency due to the interleaver delay and the reduced FER. Trinder and Gillespie determined in [14] that the effect of reducing the FER has less significant effect on the ARQ throughput than the increase in the latency. Trinder and Gillespie recommend always using the short interleaver and only using the long interleaver in broadcast data exchange mode.

3.3.3.5 Frame size selection

Trinder and Gillespie in their paper on STANAG 5066 optimisation [14] examined DRC as well as the effect of frame size on throughput. The frame size for a STANAG 5066 D_PDU can be varied up to a maximum byte length of 1023 bytes. Increasing the frame size will improve bandwidth utilization by lowering the amount of protocol overhead. However, increasing the frame size will mean that if an error occurs in the data, the entire frame will have to be retransmitted, thus over a *bad* link, i.e. a link that has frequent bit errors, a lot of time will be wasted retransmitting longer packets instead of shorter packets. STANAG 5066 has a sliding window size of 255. Thus, a maximum of 128 unacknowledged frames can be transmitted [14].

At 9600 bps the frame size will have to be increased in order to fully utilize the entire maximum transmit interval time of 127s that each node has for data transmission. According to Trinder and Gillespie's findings, for a D_PDU size of 250 bytes a total of ±22 bytes is protocol overhead while the remaining 228 bytes is data. Trinder and Gillespie further recommend that the frame size be kept constant over all data rates because of the difficulty in obtaining correct channel condition information. They suggest a frame size between 250-350 bytes per frame would provide the best results over the widest range of data rates and channel conditions.

Johnson [30] proposed to adapt the packet size based upon the current modem data rate. His recommended packets sizes vs. data rate can be seen in Table 3.8.

Table 3.8: Packet size per data rate

Data rate (bps)	Packet Size (bytes)
9600	1023
8000	913
6400	726
4800	538
3200	351
2400	257
Lower rates	200

3.4 OTHER RESEARCH AREAS

3.4.1 Link-turn around time

The link turnaround time is defined in [31] as the time that the first data bit arrives at a node to the instant the response leaves that particular node. It is not the same as the propagation time.

The elements that affect this delay are the:

- communication security implemented, i.e. the processing time required to decrypt the data block;
- modem delay caused by signal processing (synchronization), modem data rate, interleaver and FEC implementation;
- radio delay caused by the modulation of radio signals;
- protocol overhead.

The STANAG 5066 node must estimate the value of this delay, in the DTS, in order to minimize the link turnaround time.

3.4.2 Collision detection

Collision detection deals with ensuring that all STANAG 5066 nodes communicating in a network share the channel fairly and equally. Collision detection protocols can be placed in two distinct categories, contention-free protocols and contention-based protocols.

3.4.2.1 Contention-free protocols

In an article by Johnson *et al* [32] and an article by Johnson and Balakrishnan [33] the authors researched contention-based protocols such as 802.11 (WiFi), TDMA (Time Division Multiple Access) and Token Passing. A MAC protocol controls the access to a channel that is shared among competing devices in the entire network. The use of a token to manage channel contention produces collision-free communication because only the node that "has" the token is allowed to transmit. All nodes that share the channel now form a ring and the token is passed from node to node in a circular manner. Communication in the token ring (see Figure 3.3) proceeds as follows:

- nodes can only hold, and therefore transmit, for a certain maximum time;
- nodes that have no data to transmit must pass the token immediately.

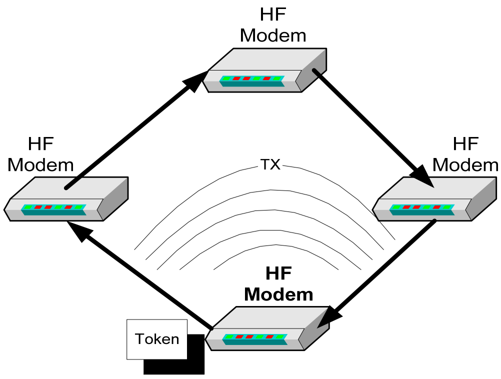

Figure 3.3: Token passing between HF Modems.

3.4.2.2 Contention-based protocols

Holcomb and Weston [34] researched ways to improve methods of channel contention and to avoid transmission collisions among a number of nodes sharing the same frequency. In [34] the authors define a STANAG 5066 node as being in one of three states: sender, receiver or listener node. Holcomb and Weston define three types of collision detection mechanisms:

1. **Carries Sense Multiple Access (CSMA):** This type of approach involves detecting whether any transmission is currently occupying the link and not transmitting until the channel is clear. The disadvantage of this approach is that the node does not interpret the communication that it is receiving; it therefore cannot determine when the communication will be finished and can therefore interrupt other nodes trying to finish a communications session.
2. **Active Monitoring Channel Access:** In this method the listening node is smart enough to understand the communication being undertaken between the sender and receiver. The disadvantage of this approach is that all client listeners implementing active monitoring will wait until the sender and receiver are finished and will then all try to transmit data over the medium at the same time, resulting in a number of collisions. To overcome this scenario a random back-off time can be chosen to wait before starting transmission.
3. **Optimized Active Monitoring Channel Access:** In this approach the Active monitoring session is optimized by attaching the initial link-request to the initial data as well as attaching the data termination request to the last data sent.

CHAPTER 4 DRC ALGORITHM DESIGN AND SIMULATION

4.1 INTRODUCTION

Chapter 4 will detail the design and simulation of a DRC algorithm that uses current channel information to determine the optimum modem data rate for the STANAG 4539 waveform.

The selection of the optimum interleaver size will not be discussed in this chapter, but will be included in the actual implementation of the DRC algorithm in Chapter 5. The use of the long interleaver is assumed for all simulations.

4.2 PURPOSE

The purpose of a DRC algorithm is to ensure the best possible data throughput on a link between a local and remote STANAG 5066 node. This is achieved by selecting the optimum modem data rate and interleaver size based upon current channel conditions and adapting the rate and interleaver to counter channel condition changes. During DRC the receiver (local node) is in the best position to determine at what rate data should be sent to it. The receiving node will inform the sending node, using EOW (Engineering Order Wire) messages, of the optimum data rate and interleaver settings. When using an autobaud waveform (like STANAG 4539 and MIL 110A) that automatically determines the data rate and interleaver setting used in communication, a node that receives a DRC advisory message can immediately change the sending rate and interleaver without having to go through a handshaking process, as would have been the case with a non-autobaud waveform such a STANAG 4285. Thus no time is lost changing and synchronizing the modem data rate and interleaver of both nodes when using STANAG 4539.

4.3 DESIGN

A remote node will first create a physical connection or link to the local node. After the physical connection has been created the remote node will start to send data to the local node; the local node will therefore be in the RX state. After a transmission interval lasting a maximum of 127s it will be the local node's turn to respond to data sent by the remote node. When the RX state is finished the local node has to make a decision as to the data rate and interleaver that the remote node will use in the next TX interval.

The input parameters to the DRC algorithm can be seen in Table 4.1. Also note the element (source) that calculated the parameter value.

Table 4.1: DRC algorithm input parameters

Parameter name	Description	Source
Interval time (ms)	Total time of the RX interval	STANAG 5066 node
Interval throughput (bps)	Data throughput achieved in RX interval	STANAG 5066 node
FER (%)	FER calculated from data in RX interval	STANAG 5066 node
BER	Estimated BER from data in RX interval	HF data modem
SNR (dB)	SNR value for the RX interval	HF data modem

The output of the DRC algorithm will be (as in Figure 4.1):

- new data rate (bps): the data rate can be one of the following data rates, 75, 150, 300, 600, 1200, 2400, 3200, 4800, 6400, 8000 or 9600 for STANAG 4539 (Only the coded data rates are used, not the 12800 bps un-coded rate);
- interleaver size: the long interleaver is assumed for all simulations.

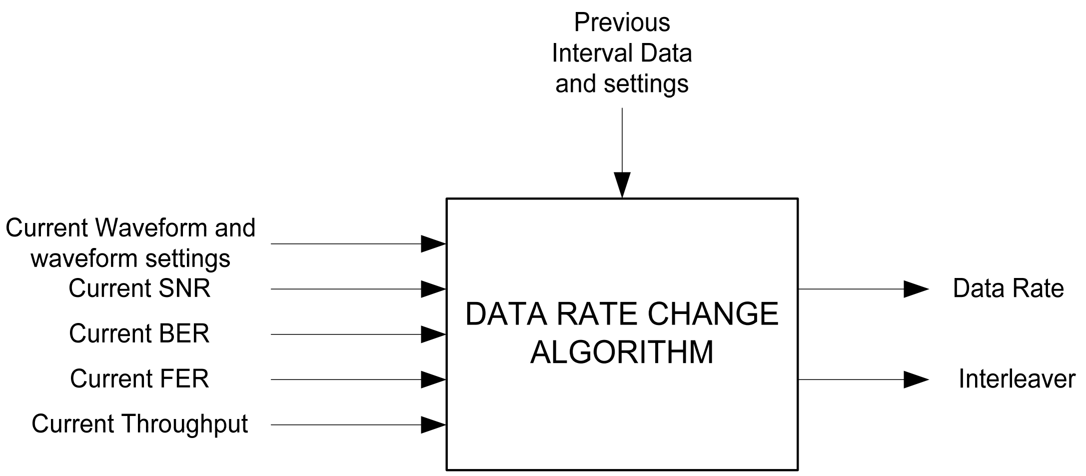

Figure 4.1: DRC Algorithm inputs and outputs.

A parameter needs to be chosen that accurately reflects current STANAG 5066 node performance and current channel conditions. Previous algorithms use the FER to determine the current data rate performance. The BER seems a better measure, because the BER combines the FER, SNR, Doppler spread and multipath effects into one measurable value.

Firstly, the relationship between FER and BER has to be defined. BER is measured as a running average on all data entering the HF data modem and is defined as the ratio of bits received in error as compared to the number of bits actually transmitted. BER is often expressed as the probability of a 1-bit error in a certain number of bits.

In order to determine the estimated FER from the BER, when the BER is 10^{-5} and the frame length is 250 bytes:

$$\begin{aligned} FER &= 1-(\%Number\_of\_frames\_correct) \\ &= 1-(1-BER)^{FrameLength(bits)} \\ &= 1-(1-10^{-5})^{(250\times 8)} \\ &= 0.0198 \\ &= 2\% \end{aligned} \tag{4.1}$$

The above equation gives the estimated FER based upon the estimated BER as measured by the HF data modem. The STANAG 5066 node is able to determine the actual FER from the data received. However, this FER measurement is subjective because fewer frames are received at a lower rate, like 75 bps, than at 9600 bps. Table 4.2 gives an indication of the number of frames received in a TX interval that lasts the maximum allowable time, for a transmit interval, of 127s and a frame length of 250 bytes. Also note that a STANAG 5066 node is only allowed to transmit a maximum of 128 unacknowledged frames in a TX interval. The confidence in the FER measurement would increase as the number of frames received also increases.

The time duration of the receive interval is also of concern when defining the confidence in BER, SNR and FER measurements, because the number of receive intervals and the duration of those intervals determine the accuracy of the BER and SNR measurements made by the HF modem.

Table 4.2: Number of frames received after 127s and frame length of 250 bytes.

Data Rate (bps)	Number of Frames
75	5
150	10
300	20
600	39
1200	77
2400	153
3200	204
4800	305
6400	407
8000	508
9600	610

Three DRC algorithms will be simulated: RapidM DRC algorithm 1, RapidM DRC algorithm 2 and the Trinder (QinetiQ) DRC algorithm or only the Trinder algorithm [14]. RapidM DRC algorithm 2 implements a subset of RapidM DRC algorithm 1. RapidM DRC algorithm 2 can only make one data rate step changes using the BER as parameter of interest, while RapidM DRC algorithm 1 can make larger data rate step changes depending on the DRC rule used. The reason for simulating RapidM DRC algorithm 2 is to evaluate the difference between the FER and BER as DRC decision parameter, when the algorithm can only make one data rate step changes.

4.3.1 RapidM DRC algorithm 1

BER measurements were made using the RM6 [35] HF data modem to determine the modem performance. The measurements describe the BER 10^{-5} points when examining a modem data rate vs. SNR graph. The tables below give the SNR value for each data rate where the BER is equal to 10^{-5} for different channels. The measurements for the CCIR Poor channel, using long interleaver, can be seen in Table 4.3 and for the AWGN channel in Table 4.4. As can be seen from the results the interleaver length has no effect in an AWGN channel (Table 4.4). The *estimated* SNR values for the BER=10^{-5} line for a CCIR Good channel using a long interleaver can be seen Table 4.5. All these measurements were made using the HFCS [36] (High-Frequency Channel Simulator) and BERT (Bit Error Rate Test) tool, supplied by RapidM. The BER measurement test setup can be seen in Figure 4.2.

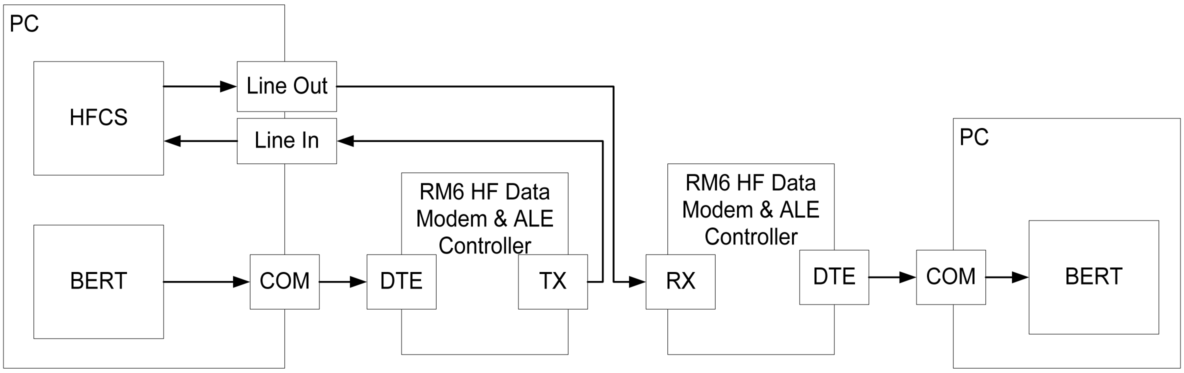

Figure 4.2: BER measurement test setup.

Table 4.3: The BER=10^{-5} SNR measurements (CCIR Poor).

CCIR Poor: SNR for BER 10^{-5}(dB)							
Interleaver		**US**	**VS**	**S**	**M**	**L**	**VL**
110A	**75**	-	-	5.00	-	-2.50	-
	150	-	-	6.50	-	-1.00	-
	300	-	-	8.50	-	1.00	-
	600	-	-	12.75	-	7.10	-
	1200	-	-	15.50	-	10.10	-
	2400	-	-	20.20	-	15.70	-
4539	**3200**	25.00	22.20	19.20	16.50	15.00	13.50
	4800	31.00	25.75	22.50	19.75	19.75	18.20
	6400	∞	29.25	25.00	23.50	22.20	21.30
	8000	∞	37.00	29.50	27.30	25.50	24.50
	9600	∞	41.00	34.00	31.20	29.75	28.30

Table 4.4: The BER=10^{-5} SNR measurements (AWGN).

AWGN: SNR for BER 10^{-5} (dB)		
Interleaver		**All**
110A	**75**	-6.75
	150	-4.00
	300	-1.50
	600	2.02
	1200	5.25
	2400	10.75
4539	**3200**	7.60
	4800	12.29
	6400	14.69
	8000	15.44
	9600	20.48

Table 4.5: The *estimated* BER=10^{-5} SNR measurements (CCIR Good).

CCIR Good: SNR for BER 10^{-5} (dB)		
Interleaver		**L**
110A	**75**	1.75
	150	2.00
	300	5.50
	600	12.18
	1200	14.95
	2400	19.45
4539	**3200**	21.40
	4800	25.21
	6400	30.71
	8000	35.56
	9600	42.02

The RapidM DRC algorithm 1 is based upon 4 rules. Rule 1 will estimate the best data rate based upon the 10^{-5} BER line on a data rate vs. SNR graph. Rule 2 estimates the data rate based upon the current BER and the average BER. Rule 3 will estimate the optimum data rate based upon the measured BER and is used when a specific data rate has been acquired and only small data rate changes are made, i.e. increase the rate or decrease the rate by one step. Rule 4 implements certain safety checks that will limit the change that a DRC rule can make to the current modem data rate, as well as limit the data rate when the SNR value is very low. The flowchart of the RapidM DRC algorithm 1 can be seen in Figure 4.3.

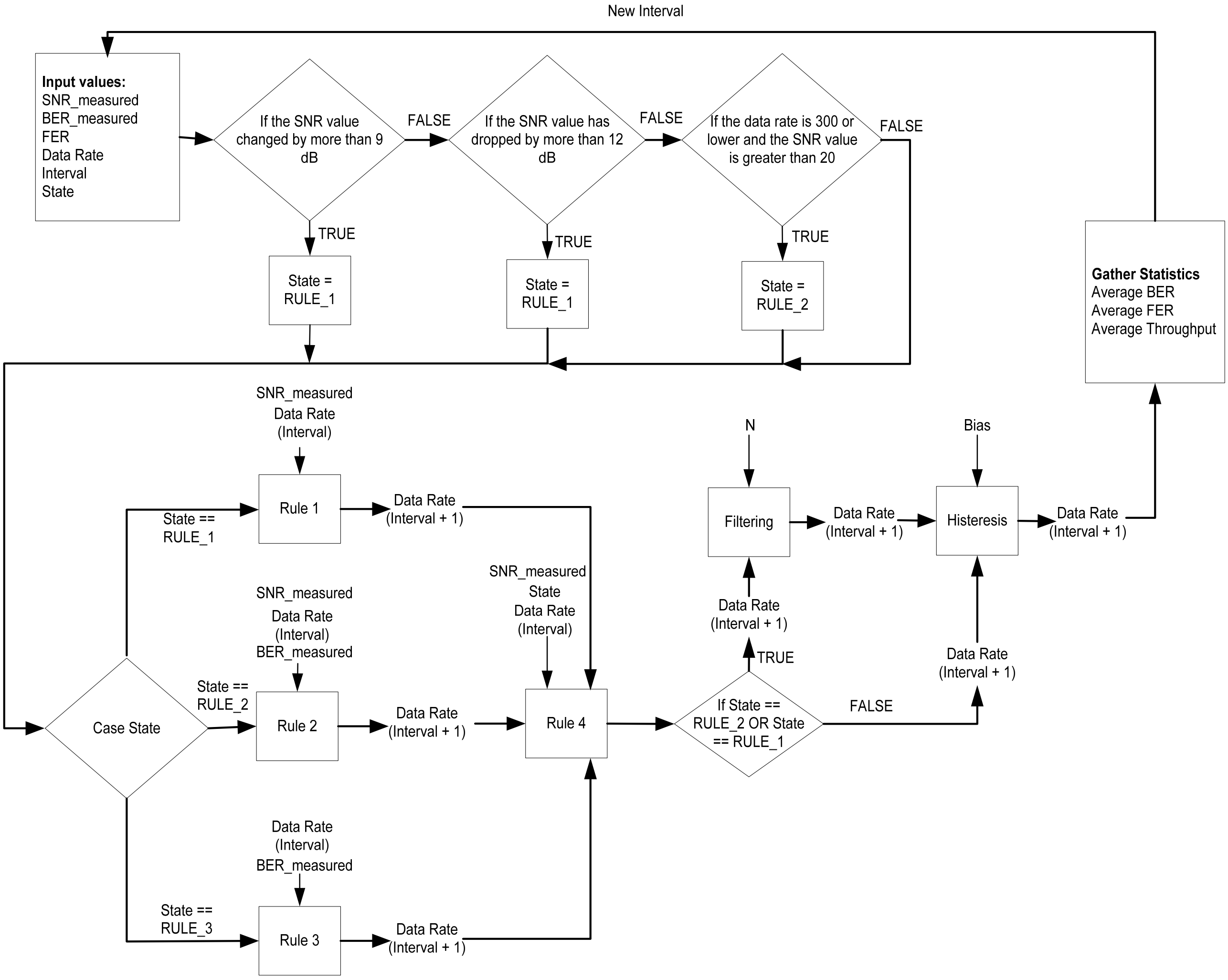

Figure 4.3: RapidM DRC Algorithm 1 flowchart.

The algorithm proceeds as follows, first, when the DRC algorithm is started, a data rate is chosen based upon the assumption that the current channel is a CCIR Good channel, the ***worst*** simulated channel found. Then the state is set to use rule number two. After the next receive interval, based upon the BER measurement, a new data rate is chosen and the state is set to use rule number 3. After the third receive interval, rule 3 will be used to determine the next data rate and the state remains in rule 3. The state can be reverted back to either rule 1 or rule 2 when an event is triggered (see Table 4.6 for rules to trigger a state change).

Table 4.6: State change trigger rules

New State	Criteria	Reason
Use Rule 1	If the SNR value dropped by more than 12 dB	We need to go back to the best data rate under the 10^{-5} Good Line, this is our fallback position
Use Rule 1	If the current data rate is 300 or less and the SNR value is greater than or equal to 20 dB	We need to go back to the best data rate under the 10^{-5} Good line, to go up to at least 1200 bps
Use Rule 2	If the SNR value either increased **or** decreased by more than 9 dB	The next data rate change can either move by one or more data rates. A data rate change by one rate up or down will have an 5-9 dB impact on performance

4.3.1.1 Rule 1

Rule 1 uses the data from Table 4.3, Table 4.4 and Table 4.5 to determine a line that will fit the measured SNR values. After a receive interval the inputs to rule 1 is the SNR measurement and the current data rate. Rule 1 then determines the best data rate choice that remains under the 10^{-5} BER line assuming a CCIR Good channel (see Figure 4.4).

The line through the data points, in Figure 4.4, is found as described in [37] with the *least squares method.* All the data points are plotted and the best line through the data points is calculated and the line equation in the form of $y = m*x + c$ is determined. The *least squares method* works on the basis of the *y-quantity* being the value likely to change and the *x-quantity* the value that is known with high precision. The best line can be found by minimising the sum of the square of the residual values, the residual is calculated as the difference between the observed y value and the calculated y value i.e. $\Delta y = y_{obs} - y_{calc}$.

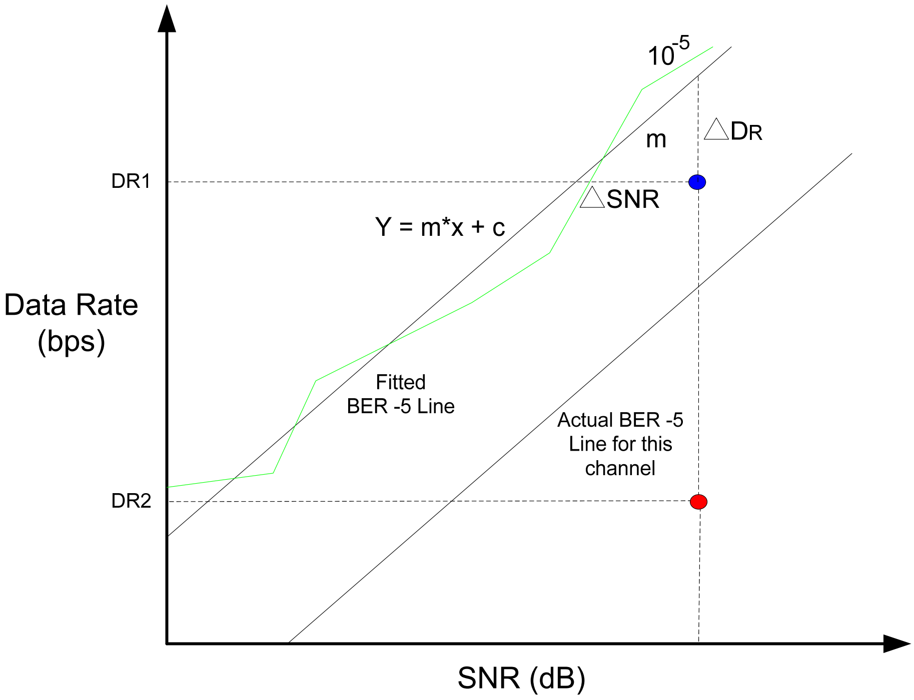

Figure 4.4: The fitted line through the 10^{-5} BER points.

Rule 1 uses equations 4.2, 4.3, 4.4 and 4.5 to determine the new data rate (*Data_Rate_Out*).

$$\Delta SNR = SNR - \left[\frac{(10 \times \log_{10}(Data\_Rate\_In) - c)}{m} \right] \quad (4.2)$$

$$\Delta DR = m \times \Delta SNR \quad (4.3)$$

$$Final\_Data\_Rate\_Log = 10 \times \log_{10}(Data\_Rate\_In) + \Delta DR \quad (4.4)$$

$$Data\_Rate\_Out = 10^{\left(\frac{Final\_Data\_Rate\_Log}{10}\right)} \quad (4.5)$$

The *Data_Rate_Out* is then converted to the closest actual STANAG 4539 data rate. Rule number 1 assumes that we are using a Good Channel and that the interleaver length is long. The data rate chosen is determined by the performance of the RM6 HF data modem. One assumption made, however, is incorrect:

- The current channel is not known. Therefore a method is required to determine how the actual channel differs from the assumed channel and how this difference affects the choice of the new data rate. Rule 2 is this method and is used to estimate the offset of the current data rate to the data rate that reflects the actual channel conditions.

Rule number 1 can change the rate from any rate to any new rate, thus it has a very large scope in terms of data rates that it can choose from.

4.3.1.2 Rule 2

The output of rule 2 is a ΔDR value that should be added to the current data rate to produce the data rate for the next receive interval. Rule number 2 can change the data rate a maximum of 2 data rate steps up or 3 data rate steps down. It is assumed that the change in BER is equal to **1 dB per decade**. Thus, if the BER is equal to 10^{-5} and the SNR increases by 1 dB the BER will change to 10^{-6} for the same data rate. The change in data rate ΔDR is proportional to the change in SNR (ΔSNR), which is determined by the measured BER value minus 4 because the minimum BER value must be 4, in order to keep the FER value low (for BER = 10^{-4} and frame length of 250 bytes the FER from (4.1) is 18 %), times the gradient m of the 10^{-5} BER line (Figure 4.4). Rule 2 uses equation 4.3, 4.4, 4.5 and 4.6 to determine the new data rate (*Data_Rate_Out*).

$$\Delta SNR = BER_{measured} - 4 \tag{4.6}$$

$$\Delta DR = m \times \Delta SNR$$

$$Final\_Data\_Rate\_Log = 10 \times \log_{10}(Data\_Rate\_In) + \Delta DR$$

$$Data\_Rate\_Out = 10^{(\frac{Final\_Data\_Rate\_Log}{10})}$$

4.3.1.3 Rule 3

Rule 3 is exactly the same as the RapidM DRC algorithm 2 that will be discussed later. The purpose of rule 3 is that when the algorithm starts there is not yet have enough information to make an adequate decision regarding the type of channel the nodes are communicating on. The BER gives us an accurate measure of the channel characteristics basic upon actual incoming STANAG 5066 data. Rule 3 is used to make small modem data rate changes. Rule 3 can only change the data rate one data rate step up or one data rate step down. Rule 3 uses equations 4.7, 4.8 and 4.9 to determine the new data rate.

4.3.1.4 Rule 4

The purpose of rule 4 is to make sure that if the current data rate is changed using rule 2 that the rate can only be changed by a maximum of 2 data rate steps upwards and a maximum of 3 data rate steps downwards. The seconds part of rule 4 sets the data rate to 75 bps if the SNR value is smaller than -2 dB.

4.3.2 Trinder (QinetiQ) DRC algorithm

The Trinder (QinetiQ) algorithm is the DRC algorithm described in [14] and uses the current FER measurement to change the modem data rate. The FER decision thresholds to increase and decrease the data rate as well as the corresponding BER values computed using equation 4.1 can be seen in Table 4.7. As stated earlier in Chapter 3, the 2400 bps value is never chosen when using this algorithm.

Table 4.7: Trinder FER and BER decision threshold

Data Rate (bps)	Reduce Rate		Increase Rate	
	FER (%)	BER	FER (%)	BER
75	N/A	N/A	20	1.116E-04
150	50	3.465E-04	20	1.116E-04
300	50	3.465E-04	20	1.116E-04
600	50	3.465E-04	20	1.116E-04
1200	50	3.465E-04	20	1.116E-04
2400				
3200	50	3.465E-04	10	5.268E-05
4800	50	3.465E-04	5	2.565E-05
6400	35	2.154E-04	5	2.565E-05
8000	15	8.126E-05	2	1.010E-05
9600	5	2.565E-05	N/A	N/A

4.3.3 RapidM DRC algorithm 2

The RapidM DRC algorithm 2 is a very simple algorithm and works on the same principle as the Trinder (QinetiQ) algorithm. It does not use the FER to decide to either increase or decrease the data rate, but the estimated BER measurement made by the HF data modem. The algorithm works as follows:

- increase data rate when $BER \leq 10^{-7}$; (4.7)
- data rate stays the same if $BER \geq 10^{-6}$ **OR** $BER \leq 10^{-5}$; (4.8)
- decrease data rate if $BER \geq 10^{-4}$. (4.9)

The BER is a much better estimate of the current channel conditions than the FER because the BER is a function of the current SNR, Doppler spread and the multipath and is measured on all incoming data bits and is not subject to change based on the number of frames received from the remote node. The estimated BER is returned by the HF data modem as a value between 0 and 7 and based on this returned value the data rate will either be increased, decreased or remain the same. The frame length will need to remain consistent for all data rates of the algorithm.

If equation 4.1 is used to determine the corresponding FER decision thresholds used to increase or decrease the data rate, the results can be seen in Table 4.8 (frame length is 250 bytes).

Table 4.8: Corresponding FER values for BER decision thresholds

	BER	FER
Decrease Rate	10^{-4}	18.1%
Increase Rate	10^{-7}	0.2%
$FER = 1-(1-BER)^{FrameLength(bits)}$		

4.4 SIMULATION

4.4.1 Purpose

The purpose of this simulation is to simulate each DRC algorithm in order to determine the best performing algorithm in terms of data throughput and to determine whether the use of BER as a decision metric delivers better results than using the FER. The three simulated algorithms are:

1. RapidM DRC algorithm 1
2. Trinder (QintetiQ) algorithm [14]
3. RapidM DRC algorithm 2

4.4.2 Parameters of interest

The following parameters of interest will be measured:

- **STANAG 5066 data throughput:** the non-erroneous (OK) data throughput in bps for a receive interval.
- **Number of oscillations:** this is a count of the number of times the data rate oscillated. This value counts the number of times the data rate was changed in one interval only to be returned to that data rate in the next interval. The number of oscillations can be determined using the following pseudo-code formula:

```
NumberOfOscillations = 0;
For interval =(2) to (End_of_intervals – 1)                    (4.10)
      If(DataRate(interval – 1) == DataRate(interval + 1) &&
      (DataRate(interval) != DataRate(interval – 1) &&
      (DataRate(interval) != DataRate(interval +1))
            NumberOfOscillations++;
      End if;
End for;
```

- **Robustness:** robustness is defined as a measure of the strength of the algorithm, i.e. an algorithm is robust if changing the data rate does not *break* the communication between two nodes. As the FER is determined on the actual STANAG 5066 data, when the FER value is above 80% the algorithm made a wrong choice, because in this interval no actual data could be sent between the two nodes. The robustness of the algorithms is measured as follows:

$$Robustness = Number\_Of\_Intervals\_FER > 80\% \qquad (4.11)$$

- **Average FER:** this measures the average FER over the entire interval duration. This is computed using equation 4.1.
- **Average BER:** this gives a measure of the average BER over the entire receive duration.
- **Acquisition time:** this is a measure of the number of time intervals taken by the DRC algorithm to acquire the correct data rate (see Figure 4.5).
- **Error Area:** this is *Undershoot* or *Overshoot* error area due to the change of data rates (see Figure 4.5).

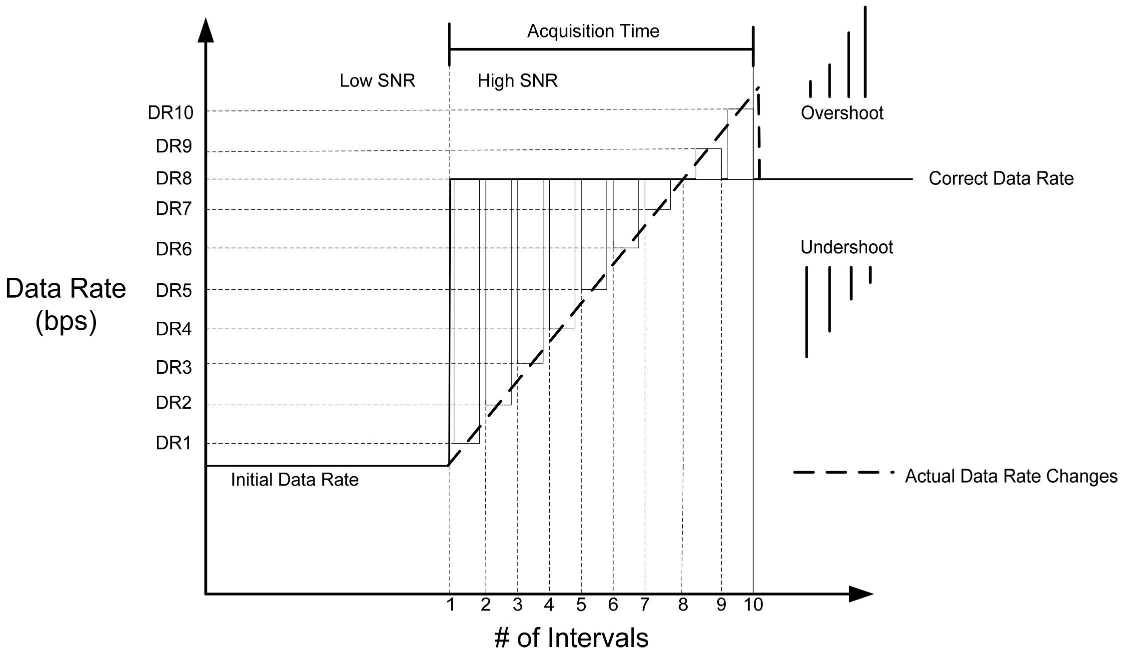

Figure 4.5: Illustration of overshoot, undershoot and acquisition time for a data rate change.

4.4.3 Simulation environment

Matlab 11.3 is used to create the simulation environment (see Figure 4.6) and simulate the three DRC algorithms. The three simulated channels can be seen in Figure 4.7. A normal *for*-loop simulates each receive interval. After each receive interval the simulated function has the following input variables:

- Current SNR
- Current BER
- Current data rate
- Current FER

Each of the three algorithms is then used to compute the data rate for the next interval. The performance for the current interval is also saved. The flow chart for the simulation can be seen in Figure 4.8.

The following assumptions are made regarding the simulations:

- the SNR remains **constant** during each receive interval;
- the BER remains **constant** during each receive interval;
- the interleaver length in **long**;
- there is **no** SNR or BER measurement errors;
- the channel can either be a **CCIR Good**, **Poor** or **AWGN** channel and the channel will be **constant** for the entire simulation;
- the channels will be differentiated by the **actual** BER measurement returned from Figure 4.6 based upon the modem data rate, channel and current SNR using the SNR and BER data from Table 4.3, Table 4.4 and Table 4.5.

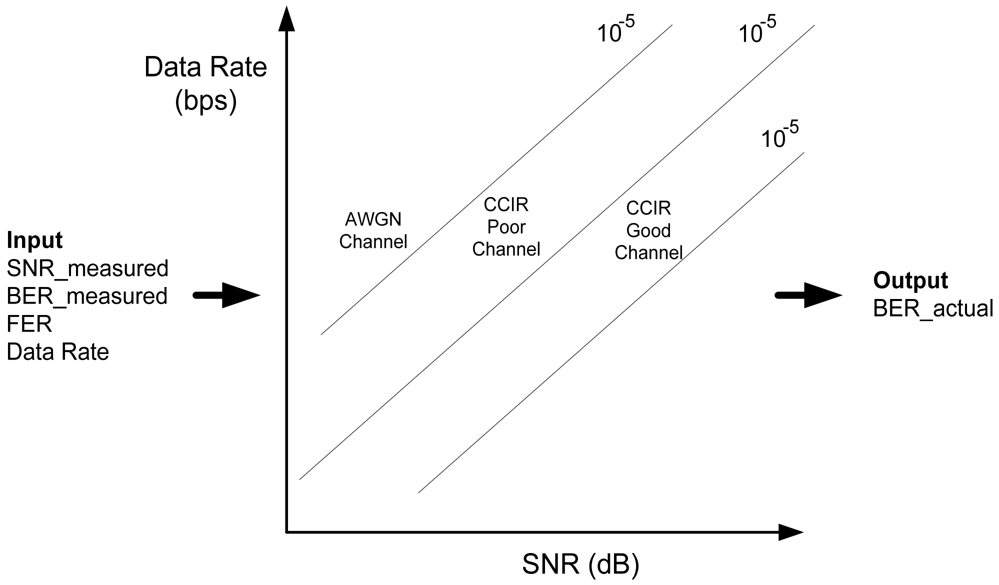

Figure 4.6: Input and output of the simulation environment.

The plot of the three simulated channels on a data rate vs. SNR graph can be seen in Figure 4.7.

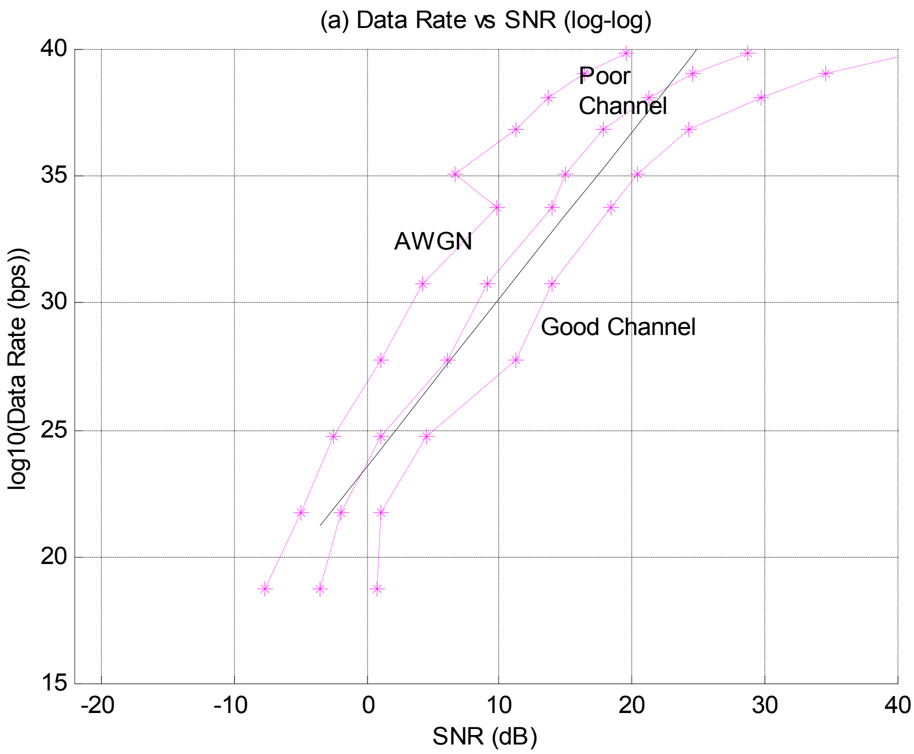

Figure 4.7: The three simulated channels (Long interleaver).

The constant values for the simulation can be seen in Table 4.9.

Table 4.9: Simulation constants

Description	Name	Value
Number of receive intervals	Intervals	100
Time of each receive interval	Receive time (s)	120
Number of bits in one byte of received data	Bits per byte (bits)	8
Number of bytes in one received frame	Frame Length (bytes)	250
Constant value that will be used to determine the filtering weight	Filtering Constant: N	3

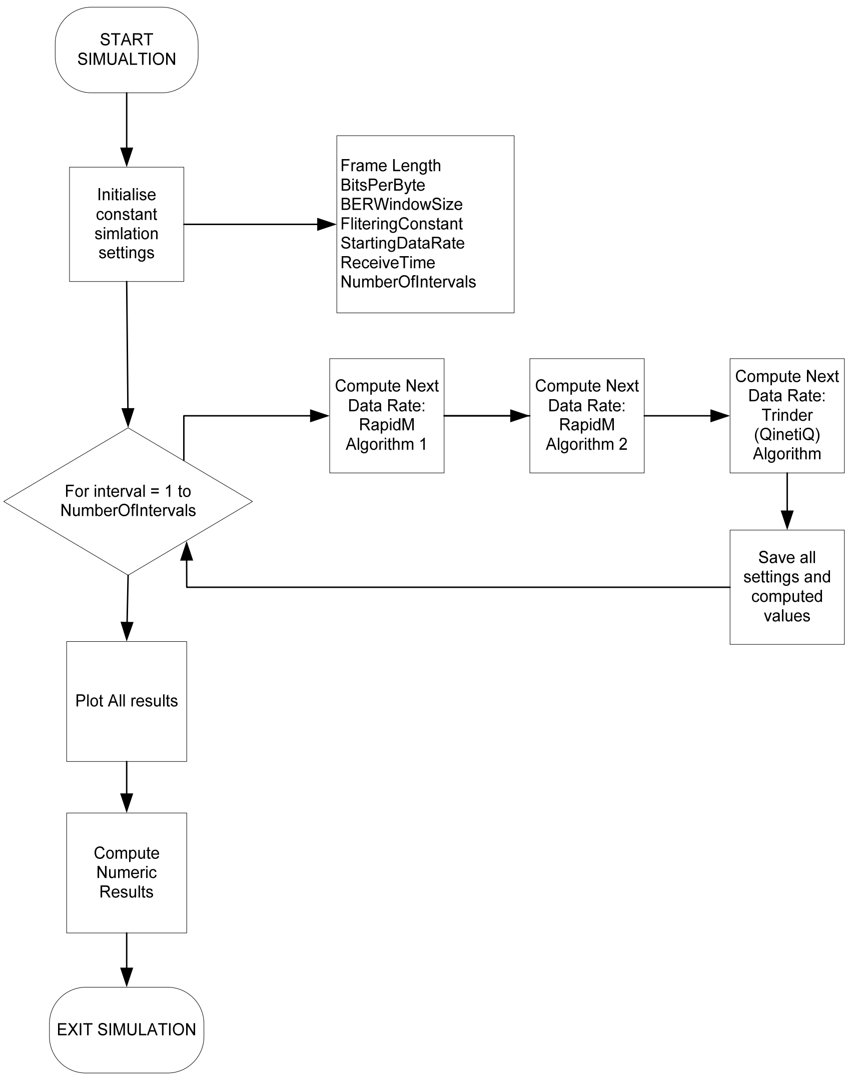

Figure 4.8: Simulation flowchart.

4.4.4 Simulation results

The following input signals were used to test the response of the three DRC algorithms:

1. upward changing inputs signal (see Figure 4.9);
2. downward changing input signal (see Figure 4.10);
3. constant changing input signal (see Figure 4.11).

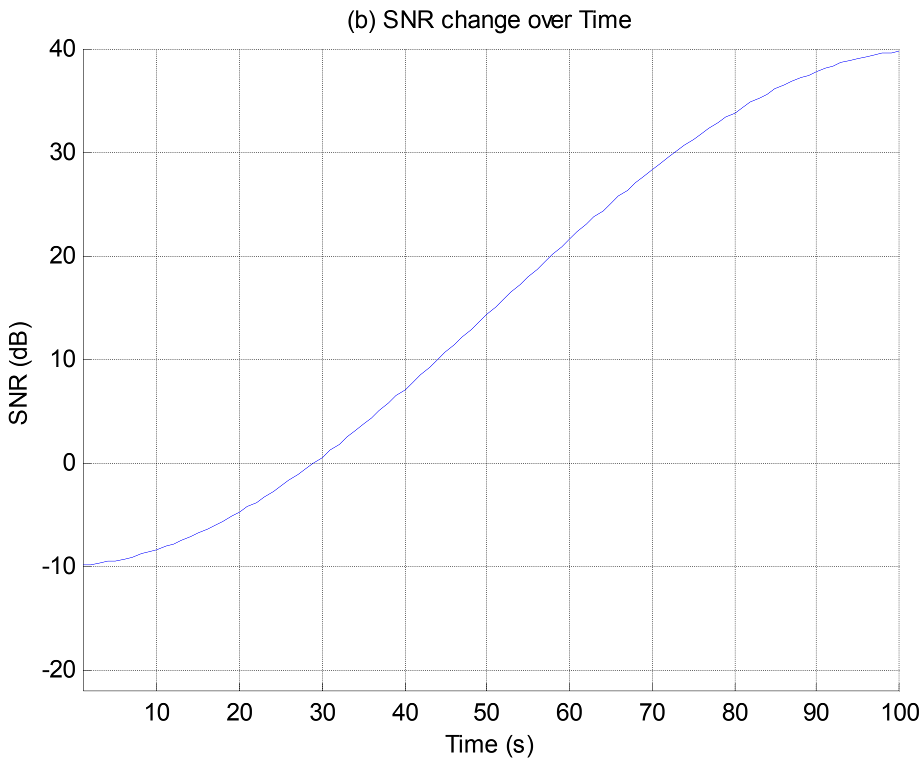

Figure 4.9: Upward changing input SNR signal.

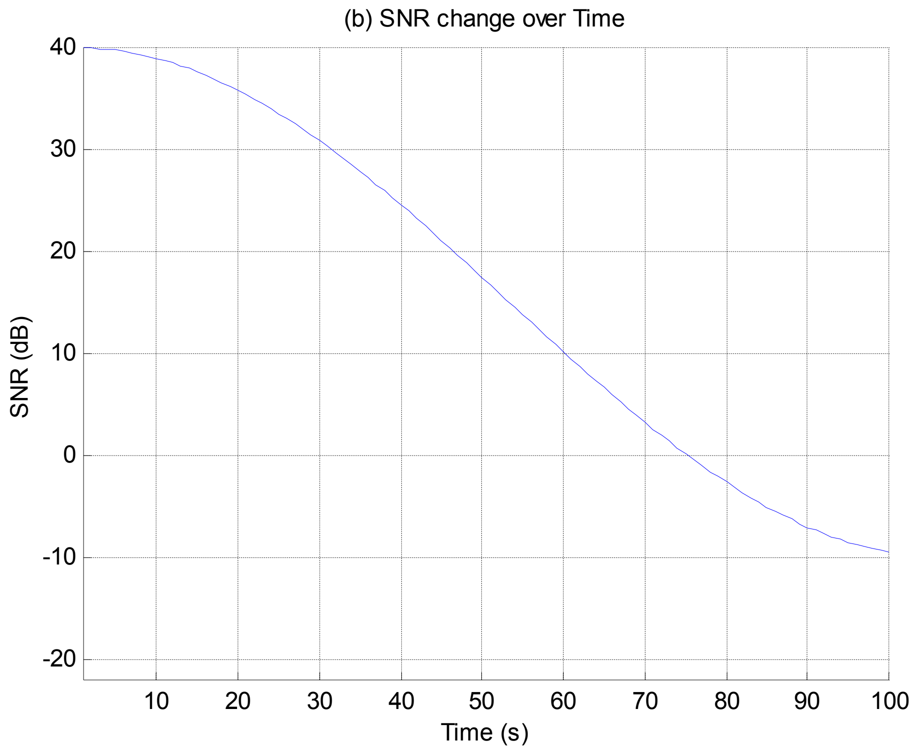

Figure 4.10: Downward changing input SNR signal.

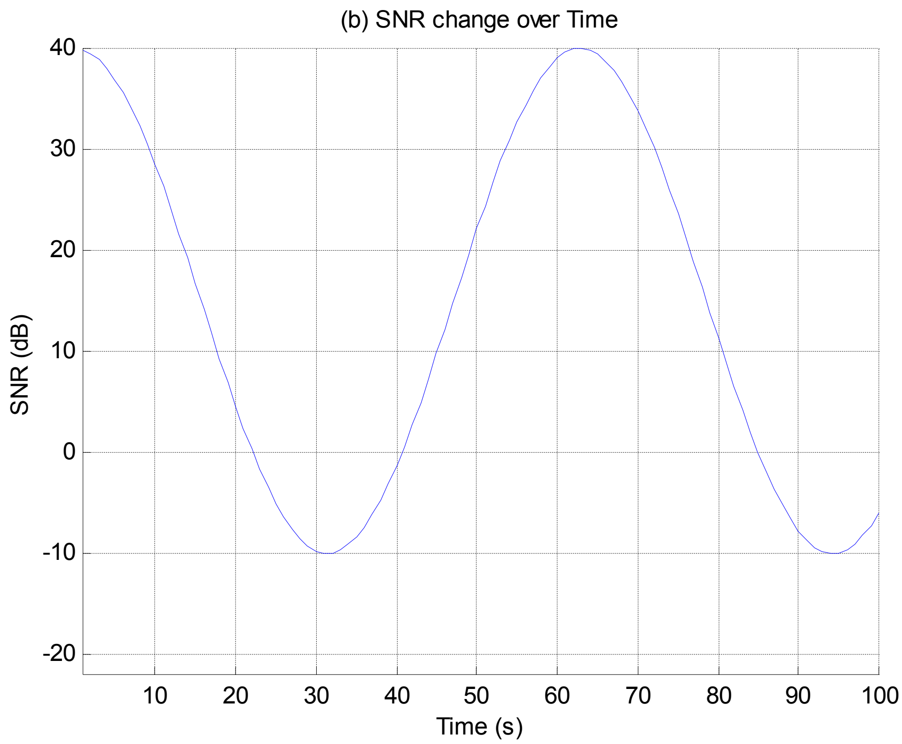

Figure 4.11: Constant changing input signal.

The results of the simulation of the three DRC algorithms using the input signals from Figure 4.9, Figure 4.10 and Figure 4.11 can be seen in the tables below.

Table 4.10: RapidM DRC algorithm 1 simulation results (AWGN)

Channel type	AWGN Channel		
Input signals	**Downward**	**Upward**	**Changing**
Average BER (10^{-x})	5.8911	6.0891	5.3564
Average FER (%)	8.5785	7.7502	24.2658
Average received bits	657810	618630	572310
Average throughput (bps)	5481.8	5155.3	4769.3
Average throughput without error (bps)	5011.5	4755.7	3612.0
Number of oscillations	8	8	0
Robustness	7	7	25
Average number of received frames	329.01	309.44	286.3

Table 4.11: RapidM DRC algorithm 1 simulation results (CCIR Poor)

Channel type	CCIR Poor		
Input signals	**Downward**	**Upward**	**Changing**
Average BER (10^{-x})	4.9901	5.1881	4.4752
Average FER (%)	24.8908	22.6235	37.3294
Average received bits	542910	519270	472.110
Average throughput (bps)	4524.3	4327.3	3934.4
Average throughput without error (bps)	3398.1	3348.3	2465.6
Number of oscillations	16	16	2
Robustness	24	22	37
Average number of received frames	271.56	259.76	236.2

Table 4.12: RapidM DRC algorithm 1 simulation results (CCIR Good)

Channel type	CCIR Good		
Input signals	**Downward**	**Upward**	**Changing**
Average BER (10^{-x})	4.3564	4.5149	3.9307
Average FER (%)	27.4245	28.2492	39.7996
Average received bits	262410	234960	232380
Average throughput (bps)	2186.8	1958	1936.5
Average throughput without error (bps)	1587.0	1404.9	1165.8
Number of oscillations	12	12	6
Robustness	25	27	38
Average number of received frames	131.3400	117.66	116.37

Table 4.13: Trinder DRC algorithm simulation results (AWGN)

Channel type	AWGN Channel		
Input signals	**Downward**	**Upward**	**Changing**
Average BER (10^{-x})	5	5.2673	4.6931
Average FER (%)	28.7140	23.2986	36.1925
Average received bits	655140	642.2390	569880
Average throughput (bps)	5495.5	5353.3	4749
Average throughput without error (bps)	3891.9	4106.0	3030.2
Number of oscillations	38	40	18
Robustness	28	23	37
Average number of received frames	327.6	321.24	285.01

Table 4.14: Trinder DRC algorithm simulation results (CCIR Poor)

Channel type	CCIR Poor		
Input signals	**Downward**	**Upward**	**Changing**
Average BER (10^{-x})	4.2475	4.5644	3.9307
Average FER (%)	42.6891	36.6370	50.1769
Average received bits	538320	540870	471390
Average throughput (bps)	4486	4507.3	3928.3
Average throughput without error (bps)	2571.0	2855.9	1957.2
Number of oscillations	40	40	16
Robustness	43	37	51
Average number of received frames	269.24	270.53	235.82

Table 4.15: Trinder DRC algorithm simulation results (CCIR Good)

Channel type	CCIR Good		
Input signals	**Downward**	**Upward**	**Changing**
Average BER (10^{-x})	3.2871	3.2871	3.1485
Average FER (%)	57.2729	53.6303	61.5754
Average received bits	297690	296940	259920
Average throughput (bps)	2480.8	2474.5	2166
Average throughput without error (bps)	1060.0	1147.4	832.2765
Number of oscillations	53	53	36
Robustness	57	52	63
Average number of received frames	148.96	148.600	130.12

Table 4.16: RapidM DRC algorithm 2 simulation results (AWGN)

Channel type	AWGN		
Input signals	**Downward**	**Upward**	**Changing**
Average BER (10^{-x})	5.7327	5.9406	5.2277
Average FER (%)	9.0182	8.1287	25.3594
Average received bits	630330	620160	545820
Average throughput (bps)	5252.8	5168	4548.5
Average throughput without error (bps)	4779.0	4747.9	3395.0
Number of oscillations	8	10	0
Robustness	7	7	26
Average number of received frames	315.23	310.16	273.02

Table 4.17: RapidM DRC algorithm 2 simulation results (CCIR Poor)

Channel type	CCIR Poor		
Input signals	**Downward**	**Upward**	**Changing**
Average BER (10^{-x})	4.9703	5.1782	4.4752
Average FER (%)	25.0683	22.6235	37.3294
Average received bits	514440	519360	443550
Average throughput (bps)	4287	4328	3696.3
Average throughput without error (bps)	3212.3	3348.9	2316.5
Number of oscillations	16	16	2
Robustness	24	22	37
Average number of received frames	257.32	259.8	221.92

Table 4.18: RapidM DRC algorithm 2 simulation results (CCIR Good)

Channel type	CCIR Good		
Input signals	**Downward**	**Upward**	**Changing**
Average BER (10^{-x})	4.3861	4.5149	3.9703
Average FER (%)	27.4192	28.2492	40.4357
Average received bits	245370	235050	217260
Average throughput (bps)	2.044.8	1958.8	1810.5
Average throughput without error (bps)	1484.1	1405.4	1078.4
Number of oscillations	12	12	7
Robustness	25	27	39
Average number of received frames	122.82	117.7	108.81

In order to determine the acquisition time and error area for a changing input signal. The following input functions will be used:

1. downward step function (see Figure 4.13);
2. upward step function (see Figure 4.12);
3. changing step function (see Figure 4.14).

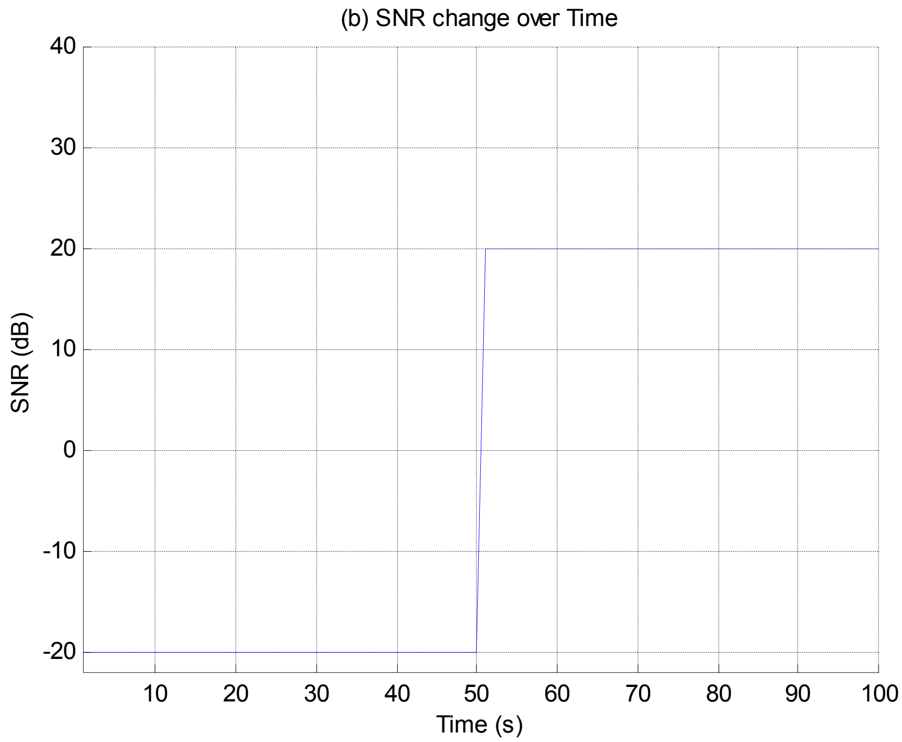

Figure 4.12: Upward step input function.

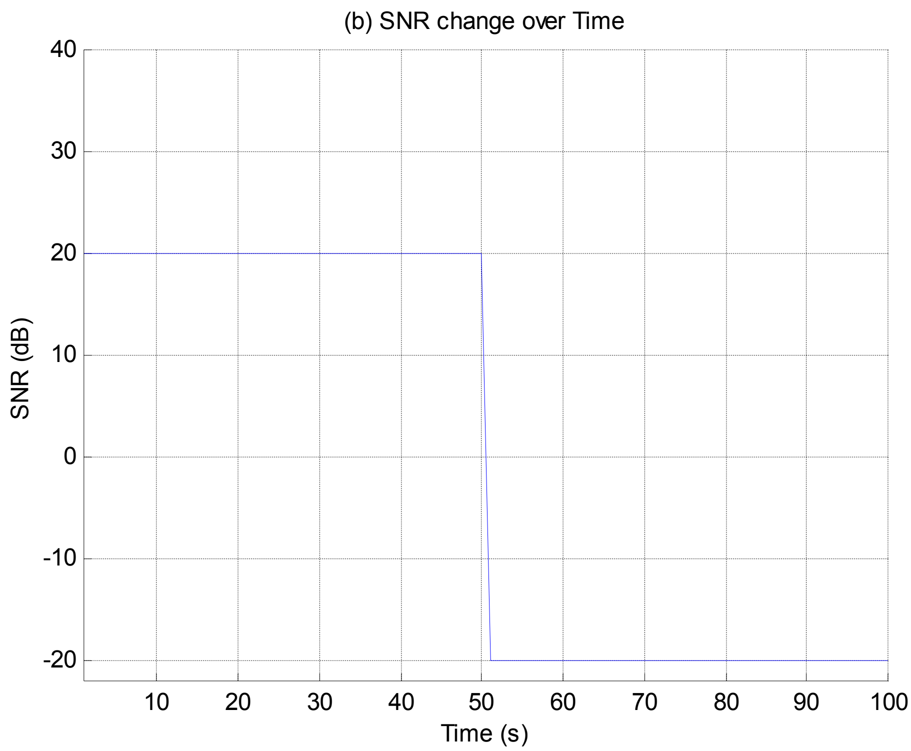

Figure 4.13: Downward step input function.

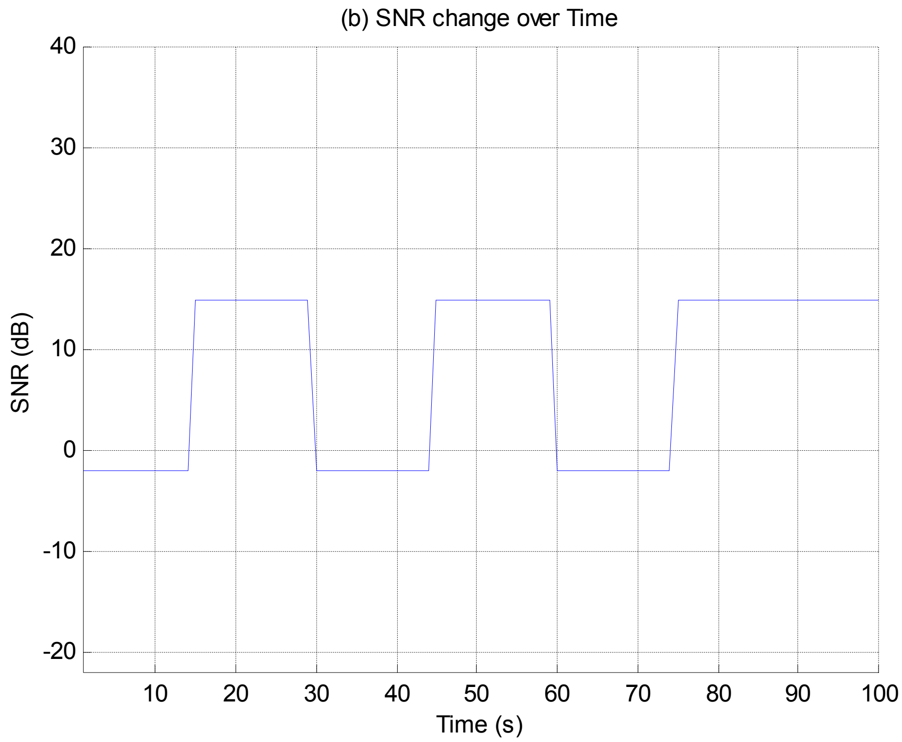

Figure 4.14: Changing step input function.

The results of the simulation of the two RapidM DRC algorithms using the step input signals from Figure 4.12, Figure 4.13 and Figure 4.14 can be seen in the tables below.

Table 4.19: RapidM DRC algorithm 1 step simulation results (AWGN)

Channel type	AWGN		
Input signals	**Downward Step**	**Upward Step**	**Number of Steps**
Acquisition Time	4	6	Not measured
Error Area	750	5525	Not measured
Average BER (10^{-x})	5.2376	5.427	5.0495
Average FER (%)	6.8915	3.0113	14.5336
Number of oscillations	0	0	0
Robustness	6	2	14

Table 4.20: RapidM DRC algorithm 1 step simulation results (CCIR Poor)

Channel type	CCIR Poor		
Input signals	**Downward Step**	**Upward Step**	**Number of Steps**
Acquisition Time	5	4	Not measured
Error Area	1200	750	Not measured
Average BER (10^{-x})	4.3970	4.2158	4.1782
Average FER (%)	9.9558	11.0148	14.6623
Number of oscillations	0	0	0
Robustness	4	2	10

Table 4.21: RapidM DRC algorithm 1 step simulation results (CCIR Good)

Channel type	CCIR Good		
Input signals	**Downward Step**	**Upward Step**	**Number of Steps**
Acquisition Time	4	4	Not measured
Error Area	750	750	Not measured
Average BER (10^{-x})	3.7208	3.8505	3.6297
Average FER (%)	32.9916	30.1549	39.8607
Number of oscillations	0	0	0
Robustness	51	50	44

Table 4.22: RapidM DRC algorithm 2 step simulation results (AWGN)

Channel type	AWGN		
Input signals	**Downward Step**	**Upward Step**	**Number of Steps**
Acquisition Time	10	7	Not measured
Error Area	14000	7250	Not measured
Average BER (10^{-x})	5.2376	5.505	5.1485
Average FER (%)	6.8915	1.0685	12.5873
Number of oscillations	0	0	0
Robustness	6	0	12

Table 4.23: RapidM DRC algorithm 2 step simulation results (CCIR Poor)

Channel type	CCIR Poor		
Input signals	**Downward Step**	**Upward Step**	**Number of Steps**
Acquisition Time	8	6	Not measured
Error Area	10500	5525	Not measured
Average BER (10^{-x})	4.0970	4.2554	3.99376
Average FER (%)	15.9558	11.0426	19.86865
Number of oscillations	0	0	0
Robustness	4	1	9

Table 4.24: RapidM DRC algorithm 2 step simulation results (CCIR Good)

Channel type	CCIR Good		
Input signals	**Downward Step**	**Upward Step**	**Number of Steps**
Acquisition Time	7	6	Not measured
Error Area	7250	5525	Not measured
Average BER (10^{-x})	3.5208	3.5001	3.5089
Average FER (%)	46.173	44.0017	45.6686
Number of oscillations	0	0	0
Robustness	50	50	44

4.5 CONCLUSION

4.5.1 Introduction

The purpose of this chapter was to design and simulate a DRC algorithm for use in the RapidM RC-66 system. The two designed algorithms are called RapidM DRC algorithm 1 and RapidM DRC algorithm 2. A third DRC algorithm was also implemented; this algorithm is the one designed by Trinder and Gillespie [14]. All three algorithms were simulated using the simulation environment as described earlier in this chapter.

4.5.2 Results assessment criteria

There main assessment points are:

1. Which algorithm performs the best in terms of data throughput?
2. Which algorithm produces the least amount of data rate oscillations?
3. Which algorithm is the most robust?
4. Which algorithm has the lowest average FER and thus the lower number of retransmission?
5. Does using the BER as DRC decision threshold give a better result than the FER?

4.5.3 Discussion

4.5.3.1 AWGN channel

Using Table 4.10, Table 4.13 and Table 4.16 for an AWGN channel the results of the three algorithms are similar in terms of data throughput. RapidM DRC algorithm 1 does have better data throughput values than RapidM DRC algorithm 2. The Trinder algorithm has marginally better data throughput results than the other two algorithms, except for the RapidM DRC algorithm 1, when the input signal is a changing sinusoidal input.

When the Trinder algorithm and the RapidM DRC algorithm 1 are compared, the data throughput for each of the three types of input channels is very similar. However, the Trinder algorithm has a much more data rate oscillations, 38, 40 and 18, than RapidM algorithm 1, which only has 8, 8 and 0 oscillations. Data rate oscillations were one of the potential problems, highlighted by Trinder and Gillespie, of their DRC algorithm.

Another area that the RapidM DRC algorithm 1 outperforms the Trinder algorithm is in terms of its robustness. The Trinder algorithm has unsuccessful data rate changes on 28, 23 and 37 occasions, while the RapidM DRC algorithm 1 only had 7, 7 and 25 unsuccessful changes.

The last important point to consider is the effect of retransmissions on an ARQ system. This is not clearly expressed in the simulation. A large number of retransmissions would decrease the actual message throughput. As can be seen from the results the RapidM DRC algorithm 1 has average FER values of 8.5, 7.7 and 24.2 for the three input signals, while the Trinder algorithm has average FER values of 28.7, 23.2 and 36.1. A lower average FER for the RapidM DRC algorithm would mean a lot less retransmissions than for the Trinder algorithm.

4.5.3.2 CCIR Poor channel

In an AWGN channel the results of data throughput for both the Trinder algorithm and the RapidM DRC algorithm is very similar. This is still the case for a CCIR Poor channel (see Table 4.11, Table 4.14 and Table 4.17) although now the RapidM DRC algorithm 1 has better throughput for both a downward and constant changing sinusoidal input signal.

The advantage the RapidM DRC algorithm 1 has in terms of less data rate oscillations is still present. The Trinder algorithm has 38, 40 and 18 oscillations, while the RapidM DRC algorithm 1 has 16, 16 and 2 oscillations, for the three input signals.

The advantage the RapidM DRC algorithm 1 has in terms of robustness is approximately 50% better than the results for the Trinder algorithm.

The RapidM DRC algorithm 1 again outperforms the Trinder algorithm in terms of average FER with values of 24.8, 22.6 and 37.3.

4.5.3.3 CCIR Good channel

In a CCIR Good channel the results in Table 4.12, Table 4.15 and Table 4.18 indicates that the RapidM DRC algorithm 1 outperforms the Trinder algorithm in all areas. The data throughput for the RapidM algorithm is almost double that of the Trinder algorithm and the number of data rate oscillations is also significantly less at 12, 12 and 6 when compared to the Trinder algorithm values of 53, 53 and 36. The robustness and the average FER of the RapidM DRC algorithm is also much less than the values for the Trinder algorithm.

4.5.4 Summary

A summary of the simulation results can be seen in Table 4.25, which indicates in what area each DRC algorithm performed well. The summary table indicates that the RapidM DRC algorithm 1 outperformed the Trinder algorithm in nearly all aspects simulated except for data throughput in an AWGN and CCIR Poor channels.

Table 4.25: Summary of DRC algorithm simulation results

DRC algorithm	RapidM algo 1	Trinder algo	RapidM algo 1	Trinder algo	RapidM algo 1	Trinder algo
Channel	AWGN		CCIR Poor		CCIR Good	
Data throughput	√	√	√	√	√	x
Number of oscillations	√	x	√	x	√	x
Robustness	√	x	√	x	√	x
Average FER	√	x	√	x	√	x

From the results presented in this chapter and based upon the fact that the RapidM DRC algorithm 1 uses the BER as decision threshold, it is concluded that using the BER gives better results than using the FER as the Trinder algorithm.

Another disadvantage expressed by Trinder and Gillespie in [14] is that an algorithm that only changes the data rate in rate steps can be very inefficient and time consuming. This effect can be seen in Figure 4.15. In the figure the RapidM DRC algorithm 1 is faster to reach the correct data rate when the SNR increases from -2 dB to 15 dB and the algorithm is also quicker to reach the correct data rate when the SNR decreases from 15 dB to -2 dB (AWGN channel).

The acquisition time of the RapidM DRC algorithm 1 and 2 were compared for three different channels and three different step input functions. The results can be seen in Table 4.19, Table 4.20, Table 4.21, Table 4.22, Table 4.23 and Table 4.24. For each simulation case the RapidM DRC algorithm 1 produced an acquisition time lower than algorithm 2 and consequently also produced a lower error area.

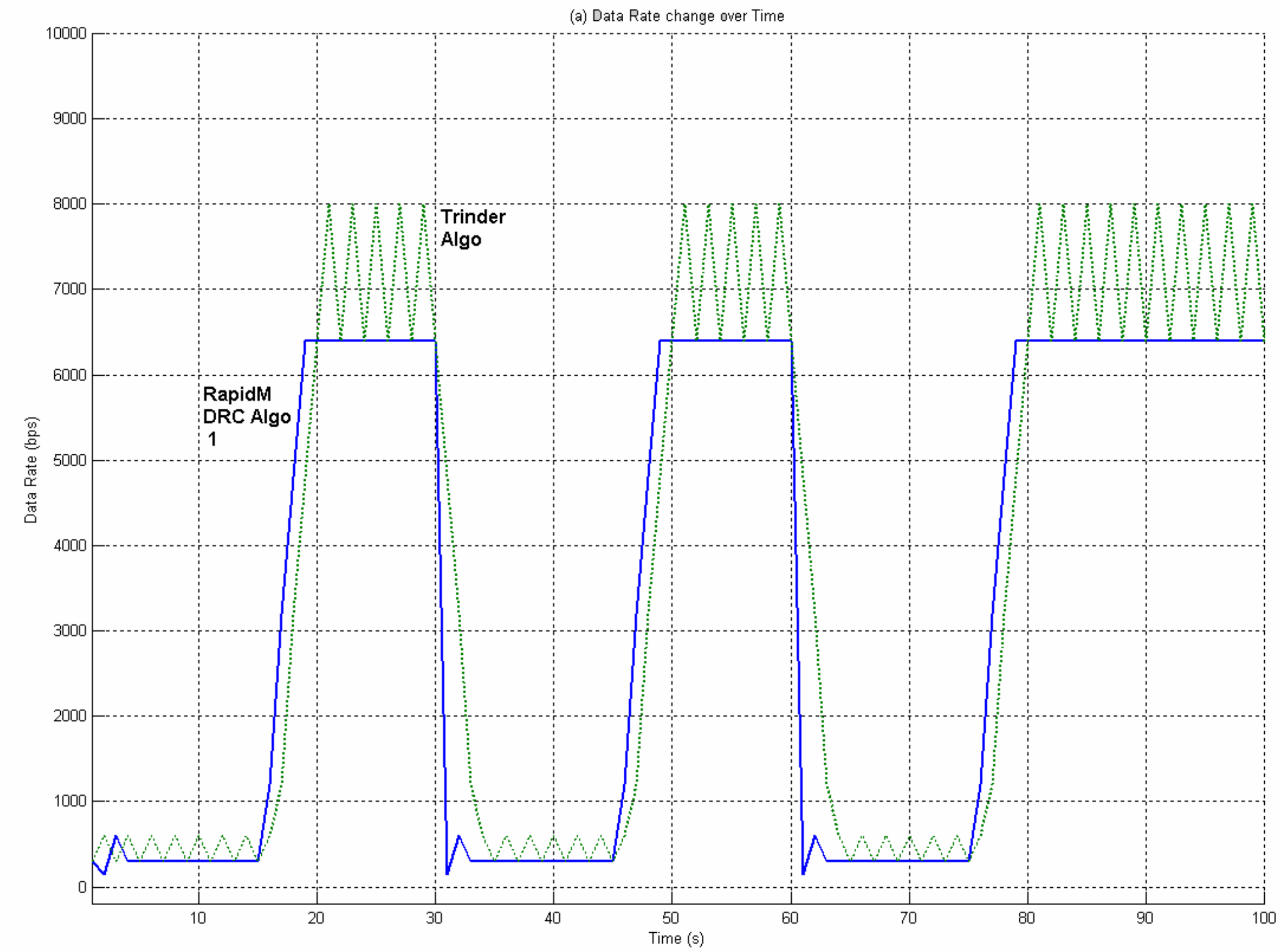

Figure 4.15: Simulation result when input step signal that varies from -2dB to 15dB.

Figure 4.16 shows the data rate vs. SNR graph for an AWGN channel and a downward changing sinusoidal input signal. The graph shows how the modem data rates chosen by the RapidM DRC algorithm 1 (solid blue line) remain under the 10^{-5} BER points for an AWGN channel. The figure also indicates that the Trinder algorithm (dotted green line) has more data rate oscillations than RapidM DRC algorithm 1. When the data rates chosen by a DRC algorithm remain under the 10^{-5} BER line the average FER is lower and the average BER higher.

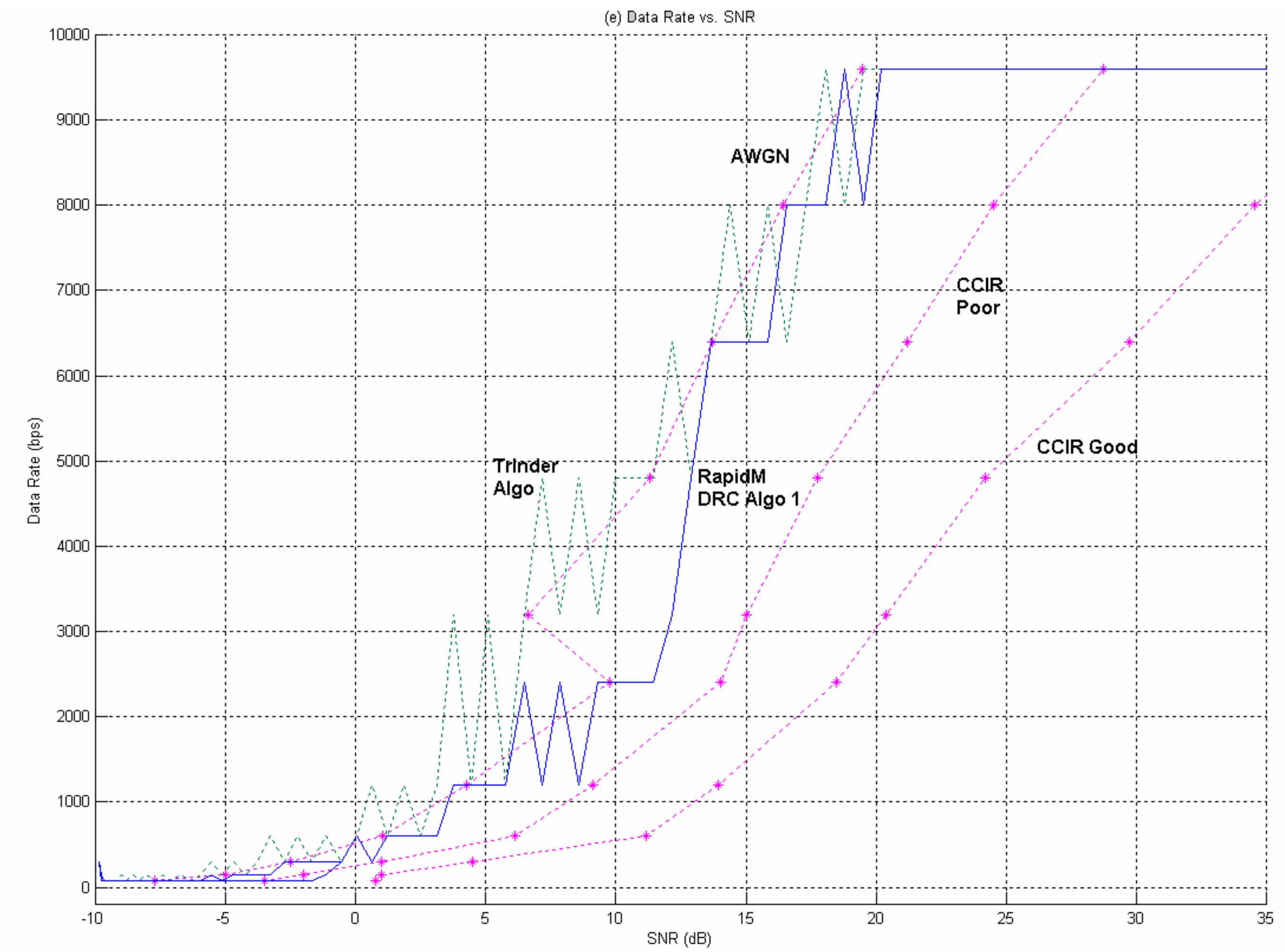

Figure 4.16: Downward sinusoidal signal graph of SNR vs. data rate.

CHAPTER 5 DRC ALGORITHM IMPLEMENTATION

5.1 INTRODUCTION

This chapter will detail the implementation of the RapidM DRC algorithm 1 in the RapidM STANAG 5066 node. The STANAG 5066 node is part of the RapidM RC-66 system (see Figure 1.6). The DRC algorithm is implemented in the DTS of the STANAG 5066 node.

The DRC algorithm will be executed at the end of each RX interval. During the RX interval the node queries the RM6 HF modem for measurements regarding the current channel conditions. The modem returns the current SNR and BER measurements for the RX signal. The measurement values are returned periodically to the node which will average these values. When the DRC algorithm executes, the average SNR and average BER together with the interval number, interval duration and current RX data rate constitute the inputs to the RapidM DRC algorithm 1 (Figure 5.1). The output of the RapidM DRC algorithm 1 will be the RX data rate for the next RX interval. The interleaver size and frame size will be determined using another method, that will be detailed later in this chapter.

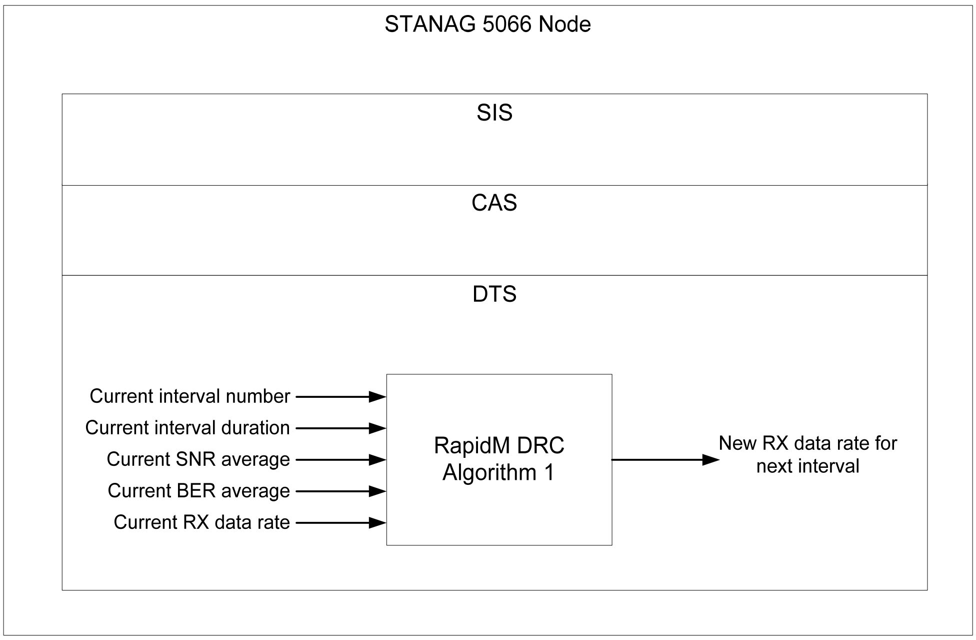

Figure 5.1: Inputs and output of the RapidM DRC algorithm 1 implementation.

5.2 EXECUTE DRC ALGORITHM DECISION THRESHOLD

It has to be decided when enough data has been received from the modem to execute the DRC algorithm and have confidence in the measurements returned by the modem. There are two possible thresholds that could be used:

1. the number of frames received > certain threshold number;
2. the number of seconds of the receive interval.

The Trinder algorithm is determined by the FER, and the less frames received the higher the variance of the FER measurement. This is the same for the BER average and the SNR average. The BER average is based upon actual measurements made on the received data.

A more realistic method to determine the threshold is to use the current Doppler spread and multipath and calculate the optimum time in seconds to wait before executing the DRC algorithm. This is done because the SNR average calculated for a small interval cannot be trusted to provide an accurate estimate of the actual SNR average for the channel. A method is needed that will adjust the amount of time, in seconds, over which accurate input information, i.e. SNR and BER can be calculated and have sufficient confidence in the accuracy of those measurements. If there is not enough confidence in the measurements, a DRC decision should not be made and the data rate should remain at the current rate and interleaver.

One proposed solution uses the Doppler spread and multipath values to determine the amount of time to wait for accurate channel measurements. This time value called *wait_interval* will then also be used as the minimum duration of the receive interval that will trigger the execution of the DRC algorithm. The *wait_interval* is calculated by taking the inverse of the current Doppler spread value times 10, up to a certain maximum value of 20 seconds (see Figure 5.2).

A decision threshold is also needed to determine if the channel is actually Gaussian in nature and not a CCIR Good channel with a low Doppler spread.

The simpler solution would be to decide on a minimum interval duration that would in most circumstances deliver accurate measurement data.

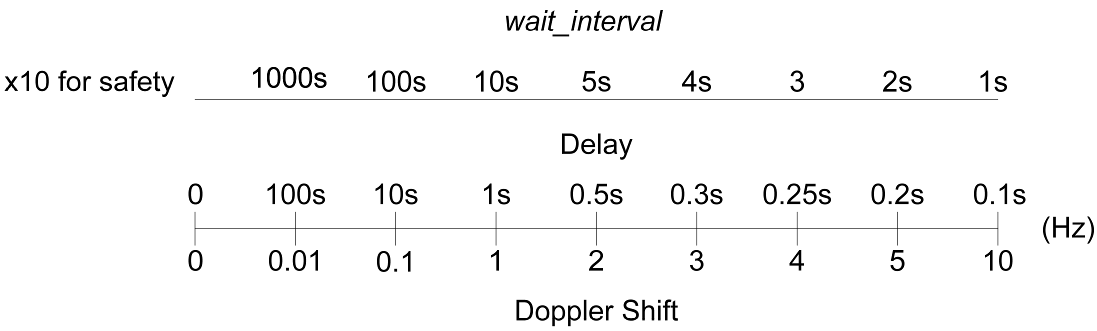

Figure 5.2: Execute DRC algorithm decision threshold.

SNR, BER, multipath and Doppler spread and values are returned by the RM6 every 300ms (Figure 5.3).

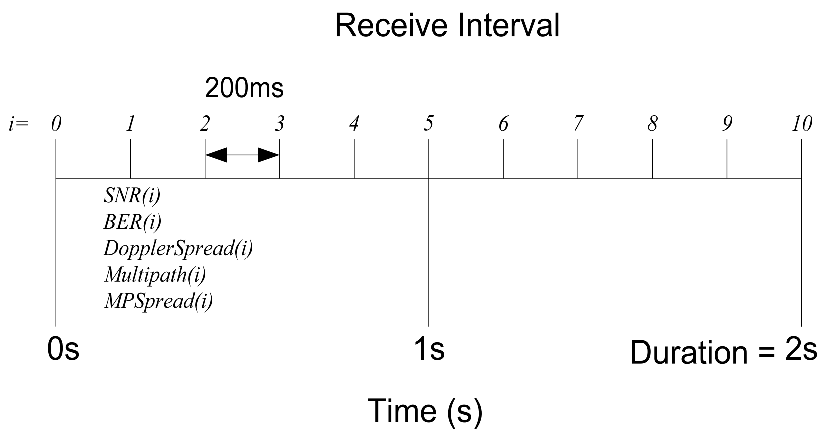

Figure 5.3: The receive interval

5.3 HF DATA MODEM MEASUREMENTS

The RM6 HF data modem ([35]) returns the following two measurements when queried:

- SNR
- Estimated BER

5.3.1 SNR average

The STANAG 5066 node queries the modem periodically for the current SNR measurement during the receive interval. The SNR value returned by the RM6 modem is calculated using a measurement period of 10 seconds. Ten seconds is also the minimum time that the RX interval has to last for the DRC algorithm to be executed. If the interval duration is less than 10 seconds then the DRC algorithm will not execute. The value of 10 seconds is used because the maximum time of sending one very long interleaver block with STANAG 4539 is 8.92 s. Thus the duration must be longer than this value to ensure that even if the data is sent with a long interleaver the entire block will be received, and therefore enough BER and SNR data can be calculated to make a correct estimate as to the actual channel SNR and BER values.

The STANAG 5066 node also has to average the queried SNR and BER values to reflect the measurements for the entire receive interval. This can be done through the use of one of the following equations:

- Calculate the average over the entire receive interval

$$SNR_{avg}(k) = \frac{\sum_{i=1}^{k}(SNR(i))}{k} \tag{5.1}$$

- Calculate averages every couple of seconds and average to a new average at the end

$$SNR_{\overline{avg}}(k) = \frac{\sum_{i=1}^{m}(SNR_{avg}(i))}{m} \tag{5.2}$$

- Calculate a filtered SNR average (first calculate the average as in equation 5.1)

$$SNR_{avg}(k) = \frac{(N-1)}{N} \times SNR_{avg}(k-1) + \frac{1}{N} \times SNR_{avg}(k) \tag{5.3}$$

- Calculate the average over the last timeslot (for example a timeslot could be 20 seconds)

$$SNR_{avg}(k) = \frac{\sum_{i=k-n}^{k}(SNR(i))}{n} \tag{5.4}$$

5.3.2 BER average

The same formulas used to determine the average SNR value (equations 5.1, 5.2, 5.3 and 5.4) can be used to determine the average BER measurement. The FER, throughput and interval duration information is calculated by the STANAG 5066 node.

5.3.3 Interval average

The BER and SNR interval measurements will be calculated using the average calculated over the entire receive interval using equation 5.1. The RM6 modem is queried every 300 ms for the current SNR and BER measurement for the duration of the receive interval.

5.4 INITIAL IMPLEMENTATION PROBLEMS

After the initial algorithm implementation a problem was encountered that is similar to the problem encountered by Trinder and Brown [12]. The modem data rates chosen by the RapidM DRC algorithm 1 tended to oscillate when in the rule 3 state (for a full description of the rule 3 implementation refer to the previous chapter). When the BER average is 10^{-7} the data rate is increased. During the next receive interval, at the same average SNR and new data rate, the BER average is 10^{-4} or greater, causing the data rate to be decreased. This oscillation phenomenon is especially prevalent at lower data rates (rates 75 to 2400 bps) on a Gaussian channel. The reason for this is that the Gaussian channel has a very sharp drop off, as can be seen in a BER vs. SNR curve, when compared to the drop off for a CCIR Poor and CCIR Good channel (see Figure 5.4). This can also be seen from Figure 4.7 where for lower data rates (smaller than 3200 bps) the gradient of the data rate vs. SNR graph for the AWGN, CCIR Poor and CCIR Good channels are very high. This means that there is a large change in the BER, when the data rate changes by either one data rate step up or down. For higher data rates the line for the CCIR Poor and CCIR Good channels diverge from the AWGN line, which means that the change in BER for a data rate step is not as severe as for an AWGN channel.

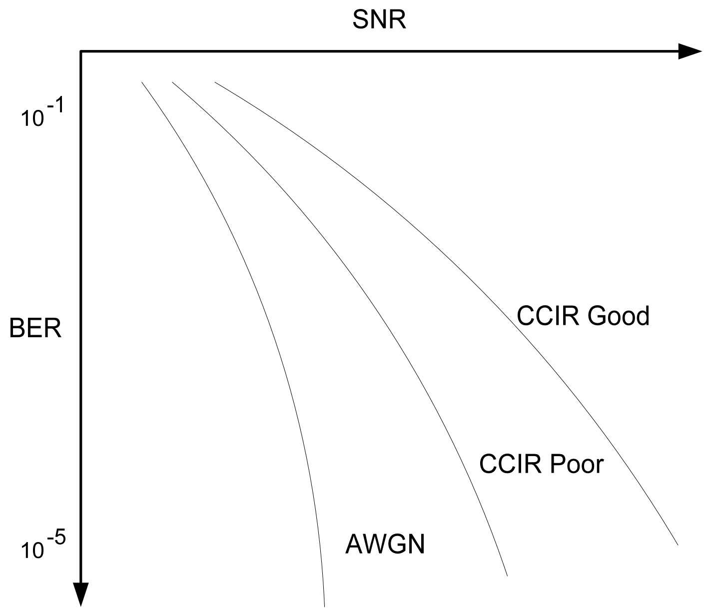

Figure 5.4: Example SNR vs. BER curve

Another problem seen in the implementation of the RapidM DRC algorithm 1 is that the algorithm does not use previous BER averages to predict the future behaviour of a data rate choice. Lastly, the change in SNR average from one interval to the next is also not used in the algorithm. The change in SNR average coupled with the current BER average could be used to determine and predict a new data rate that will deliver the required BER value.

The solution to this problem is to design and implement control logic that would estimate the BER for the next higher data rate. Based upon this estimate the next data rate can be chosen based upon previous BER averages saved in a table and changes in the SNR average. If the BER estimate for a data rate is not greater than a certain threshold value the data rate is not chosen. This would effectively eliminate the oscillation effect encountered.

5.5 REDESIGN OF RAPIDM DRC ALGORITHM 1

5.5.1 Description

The optimum solution would be to design and implement control logic inside the RapidM DRC algorithm 1 that would estimate the BER for each data rate by constructing a channel BER profile. A BER estimate table is constructed that contains the BER estimate for each data rate of the STANAG 4539 waveform, from 75 to 9600 bps. When the BER estimates are plotted vs. the data rate, the BER channel profile in Figure 5.5 is created. The implementation of this control logic means that the DRC algorithm structure for RapidM DRC algorithm 1 will be altered. The inputs to the algorithm and the outputs remain the same, however.

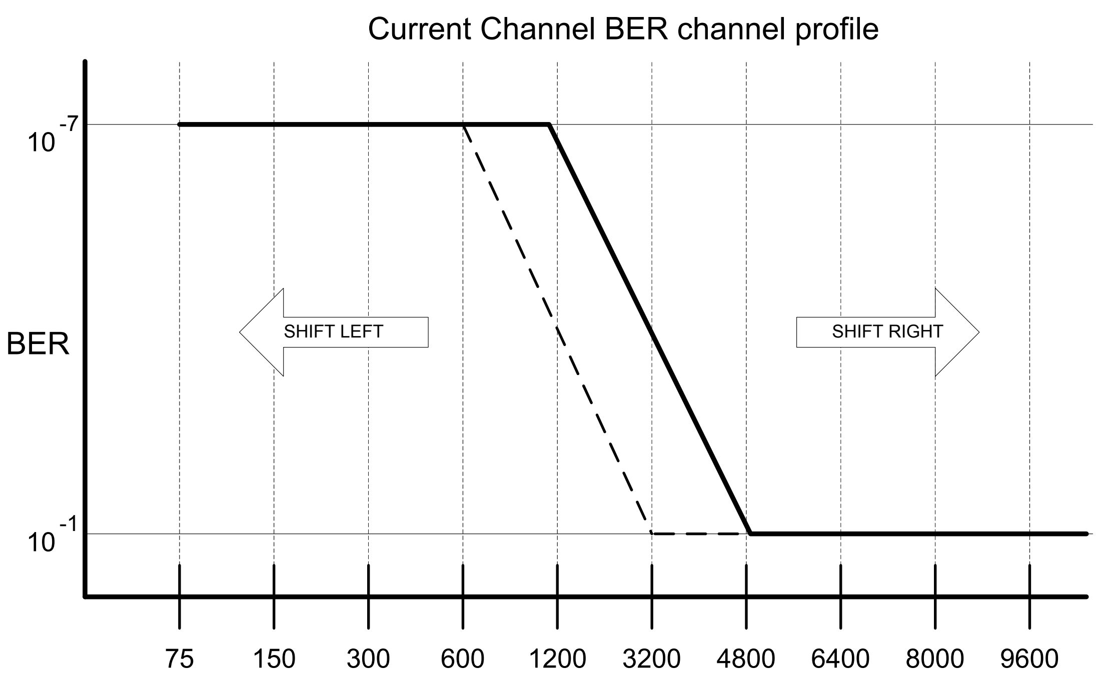

Figure 5.5: BER channel profile.

The control logic proceeds as follows: initially the BER estimate table is filled with CCIR Good channel BER estimates. After the first RX interval, based upon the current SNR, the first data rate is chosen assuming that the current channel is a CCIR Good channel. After the next RX interval a BER and SNR average is determined for the entire receive interval at a certain data rate (inputs to DRC algorithm, see Figure 5.1). The BER average is saved in the table and added to previous averages at that data rate, if any exist.

If the BER average value is 10^{-7} then the data rate can be increased by one rate step (this action is called PROBE, because the algorithm is probing the next higher or lower data rate to determine the BER average at that data rate, this action is similar to rule 3 of the previous chapter). This is done **only if** there has not yet been a BER average saved in the table at that data rate. Similarly, if the average BER is 10^{0} then the data rate can be decreased by one data rate step, also only if no previous BER average was saved in the BER estimate table. If a previous interval BER average has been saved at that rate the new BER average is added to the previous BER average to produce a new BER estimate and the BER estimate table is used to return the highest possible data rate that has a BER estimate that is higher than a certain threshold value (this action is called TRACK, because the algorithm tracking the current BER estimates for the channel).

The SNR change is calculated from the previous interval SNR average and the current interval SNR average. This change will change the BER estimates of data rates other than the data rate at which the BER average was calculated, i.e. the data rate of the current RX interval. The specific BER estimates affected depends on the size of the SNR change. For a small change only the rates one rate step higher and lower than the current data rate will be changed. For a larger SNR change the BER estimates for data rates that are more than one data rate step away from the current data rate will be affected. When a large SNR change occurs all BER estimates are reset, which means that the data rate can be probed again (the entire BER estimates table is shifted either left or right, this action is called ACQUIRE, because the algorithm is trying to acquire the channel BER estimate profile again because of a large change in SNR).

The following assumption is made:

$$\Delta BER / \Delta SNR = 1dB\_per\_decade \tag{5.5}$$

The change in BER for the current rate is directly proportional to a change in the SNR. This means that if the SNR increases by 1 dB the BER estimates for the rates will also increase by 1, 10^{-6} to 10^{-7}, to a maximum of 10^{-8} and conversely if the SNR decreases by 1 dB from one RX interval to the next, the BER will decrease by 1 i.e. from 10^{-7} to 10^{-6}. The maximum BER estimate value is 10^{-8} and the minimum value 10^{0}.

5.5.2 RapidM DRC algorithm 1 flowchart

The flowchart for the RapidM DRC algorithm 1 can be seen in Figure 5.6. The flowchart is made up of nine functions, which are:

- Fill Good Channel Data: fill in BER estimates of the CCIR Good channel.
- Measurement update: incorporate current channel BER average.
- SNR Measurement Update: update the SNR with new SNR average.
- SNR Change: change the BER estimate with the change in SNR average.
- Shifter: shift the BER estimates profile left or right (Figure 5.5).
- Limiter: limit the BER estimate based upon current BER average.
- Gauss Limiter: limit the BER estimate to maximum of AWGN channel BER estimates.
- DRC Rule 3: implement rule 3 of the previous chapter.
- Decision threshold: use the highest data rate with a BER estimate greater than a threshold value.

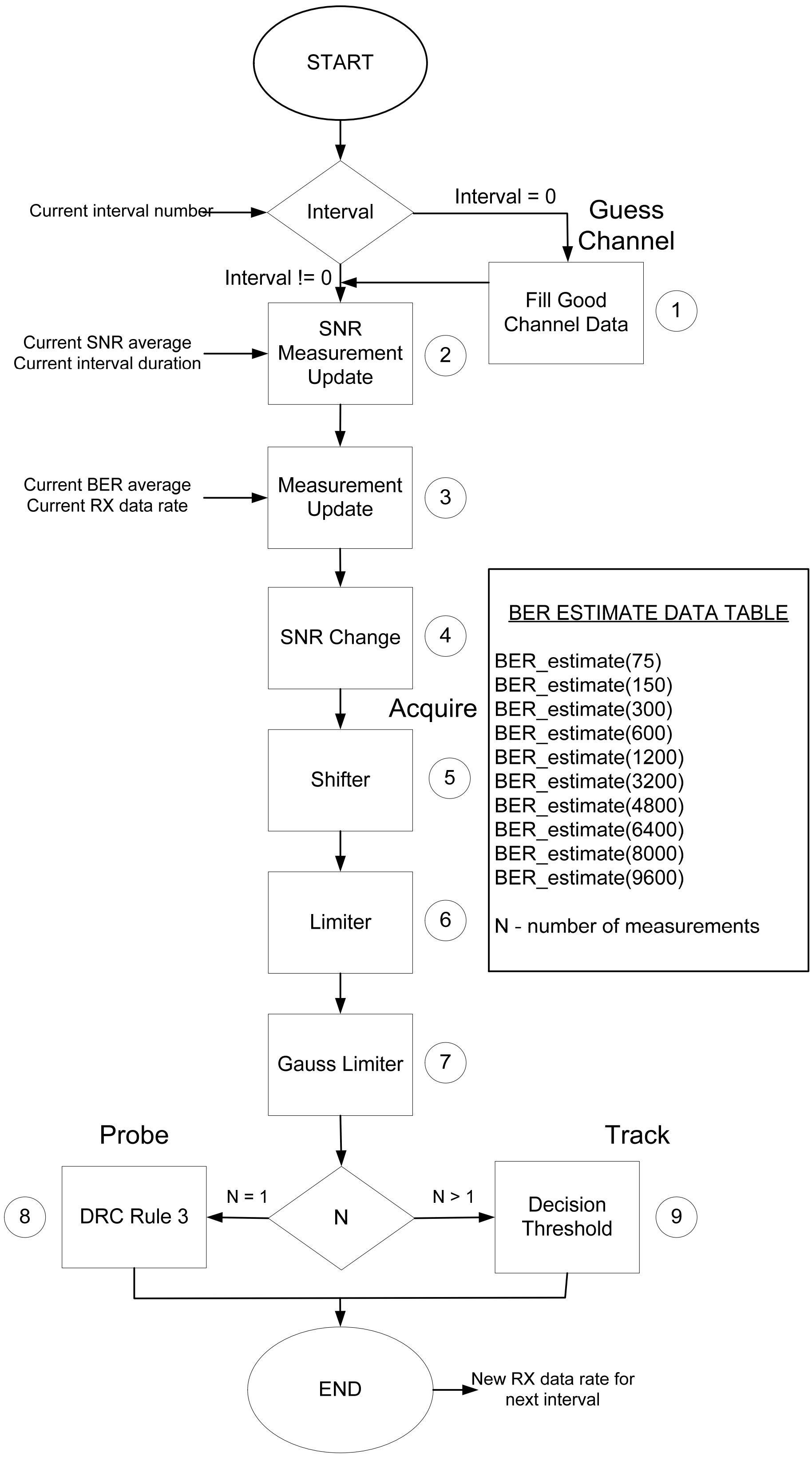

Figure 5.6: Control logic flowchart.

5.5.3 Fill Good channel data

This function is only executed for the very first interval or if the interval counter is reset to zero. The BER estimates table is filled with the BER estimates for a CCIR Good channel. This will ensure that the first data rate chosen will be the optimum data rate assuming that the channel is a CCIR Good channel.

5.5.4 SNR measurement update

In order to filter the current SNR average value that is calculated by the STANAG 5066 node to incorporate different interval durations and counter the effect that the interval duration has on the value of the current SNR average, the SNR value used to change the BER estimate is averaged a second time based upon the interval duration. This means that a SNR average calculated by the node over a longer interval counts more that another SNR average calculated over a smaller interval. *CD* is the duration of the current interval, *PD* is the duration of the previous interval and *k* the current interval number (equation 5.6). The SNR value is calculated using equation 5.6.

$$SNR_k = \left[\frac{CD}{CD+PD}\right] \cdot SNR_{average}(k) + \left[\frac{PD}{CD+PD}\right] \cdot SNR_{average}(k-1) \tag{5.6}$$

SNR change is calculated by the change in the SNR value between the previous and current interval (see equation 5.7).

$$SNR\_change = SNR_k - SNR_{k-1} \tag{5.7}$$

5.5.5 Measurement update

The measurement update will calculate the new BER estimate for the current data rate based upon the returned BER average from the HF modem and the BER estimate for the previous interval averaged by the number of measurements made (N). The maximum value for N is 2. The choice of N will reflect how fast the DRC algorithm will be able to adapt the changing channel conditions. A large value of N will mean that the average BER value will change slower than if N was smaller. The BER estimate is calculated using equation 5.8.

$$\hat{BER}_k(Data\_Rate) = \left[\frac{N-1}{N}\right] \cdot \hat{BER}_{k-1}(Data\_Rate) + \left[\frac{1}{N}\right] \cdot BER_{average} \tag{5.8}$$

The pseudo code for incrementing the N value and making sure that the value does not increase above the maximum value can be seen in equation 5.9.

$$\begin{aligned} &N++; \\ &IF(N > N\_MAX) \\ &N = N\_MAX = 2; \end{aligned} \tag{5.9}$$

5.5.6 SNR change

The SNR change function will change the BER estimate of data rates one step higher and one step lower than the current data rate, for which a BER measurement update has occurred using equation 5.10 and 5.11.

$$\hat{BER}_k(Data\_Rate+1) += SNR\_change \tag{5.10}$$

$$\hat{BER}_k(Data\_Rate-1) += SNR\_change \tag{5.11}$$

5.5.7 Shifter

The shifter will shift the current BER estimates table left or right by a certain number of data rates. The number of BER estimates that is shifted depends on the change of SNR from the previous interval to the current interval. The BER value can range from 10^{-7} to 10^{0}, and thus has a range of 7 values. The second important assumption made is that the average SNR difference between data rates is 3 dB (this can be seen from Figure 4.7 for an AWGN channel). That is to say that when a BER average value for a data rate is 10^{-6}, then it is assumed that if the rate is increased by one rate step, the next BER average at the same SNR will be 10^{-3}. So if the SNR changes by more than 3 dB the entire BER channel profile can be moved left or right by one data rate (see Figure 5.5). The change in SNR from the previous interval to this interval will determine the number of data rates to shift the BER estimates by (equation 5.12).

$$Number\_rates\_to\_shift = abs(SNR\_change/3) \tag{5.12}$$

The BER estimates table is shifted either left or right depending whether the SNR change is positive or negative (equations 5.13 and 5.14). Note that equations 5.13 and 5.14 are carried out for every *Data_Rate* from 75 to 9600 bps.

$$\begin{aligned} & if\,(SNR\_change < 0) \\ & \hat{BER}_k(Data\_Rate) = \hat{BER}_k(Data\_Rate + Number\_rates\_to\_Shift) \end{aligned} \tag{5.13}$$

$$\begin{aligned} & if\,(SNR\_change > 0) \\ & \hat{BER}_k(Data\_Rate) = \hat{BER}_k(Data\_Rate - Number\_rates\_to\_Shift) \end{aligned} \tag{5.14}$$

5.5.8 Limiter

The limiter will smooth the BER estimates table based upon the current BER average. If the BER average is high enough, all the BER estimates for rates lower than the current data rate will be set to the maximum BER value. If the current BER average is very low then all data rates higher than the current data rate will be set to the minimum BER value. The pseudo-code for the limiter can be seen in Figure 5.7.

```
if (current_BER_average >= 6)
      for data_rate < current_RX_rate to 75
            BER_estimate(data_rate) = BER_MAX_ESTIMATE
      end for
end if
if (current_BER_average <= 2)
      for data_rate > current_RX_rate to 9600
            BER_estimate(data_rate) = BER_MIN_ESTIMATE
      end for
end if
```

Figure 5.7: Limiter pseudo-code.

5.5.9 Gauss limiter

The Gauss limiter will ensure that the BER estimates table will never exceed the BER estimates for an AWGN channel. The function thus sets the maximum BER estimates based upon the current SNR average.

5.5.10 DRC rule 3

DRC rule 3 functions exactly as designed in Chapter 4 during the DRC algorithm simulations. The function works as follows (see equations 5.15, 5.16 and 5.17):

- increase rate when $BER\_average \leq 10^{-7}$; (5.15)
- rate stays the same if $BER\_average \geq 10^{-6}$ OR $BER\_average \leq 10^{-5}$; (5.16)
- decrease rate if $BER\_average \geq 10^{-4}$. (5.17)

5.5.11 Decision threshold

The decision threshold returns the data rate for the next interval by selecting the highest data rate that has a BER estimate value greater than the threshold value. The pseudo-code can be seen in Figure 5.8.

```
If (BER_estimate(9600) > BER_THRESHOLD) return 9600;
If (BER_estimate(8000) > BER_THRESHOLD) return 8000;
If (BER_estimate(6400) > BER_THRESHOLD) return 6400;
If (BER_estimate(4800) > BER_THRESHOLD) return 4800;
If (BER_estimate(3200) > BER_THRESHOLD) return 3200;
If (BER_estimate(1200) > BER_THRESHOLD) return 1200;
If (BER_estimate(600) > BER_THRESHOLD)  return 600;
If (BER_estimate(300) > BER_THRESHOLD)  return 300;
If (BER_estimate(150) > BER_THRESHOLD)  return 150;
If (BER_estimate(75) > BER_THRESHOLD)   return 75;
```

Figure 5.8: Decision threshold pseudo-code.

The selected *BER_THRESHOLD* value is **5.1**.

5.6 DRC ALGORITHM CLASS DESIGN

The UML (Unified Modelling Language) class design for the DRC algorithm implementation classes can be seen in Figure 5.9. The base class for the two DRC algorithms will the *DRC Algorithm* class. The two DRC algorithm implementation classes will inherit from this C++ class.

The *Interval Measurement* class will be used to save all the statistics information after each receive interval for each algorithm. The data saved in the statistics interval will then be used to determine the statistics of the algorithms and determine implementation performance.

The DRC classes will be implemented in Visual C++ and added to the RapidM STANAG 5066 node software running on a Windows system.

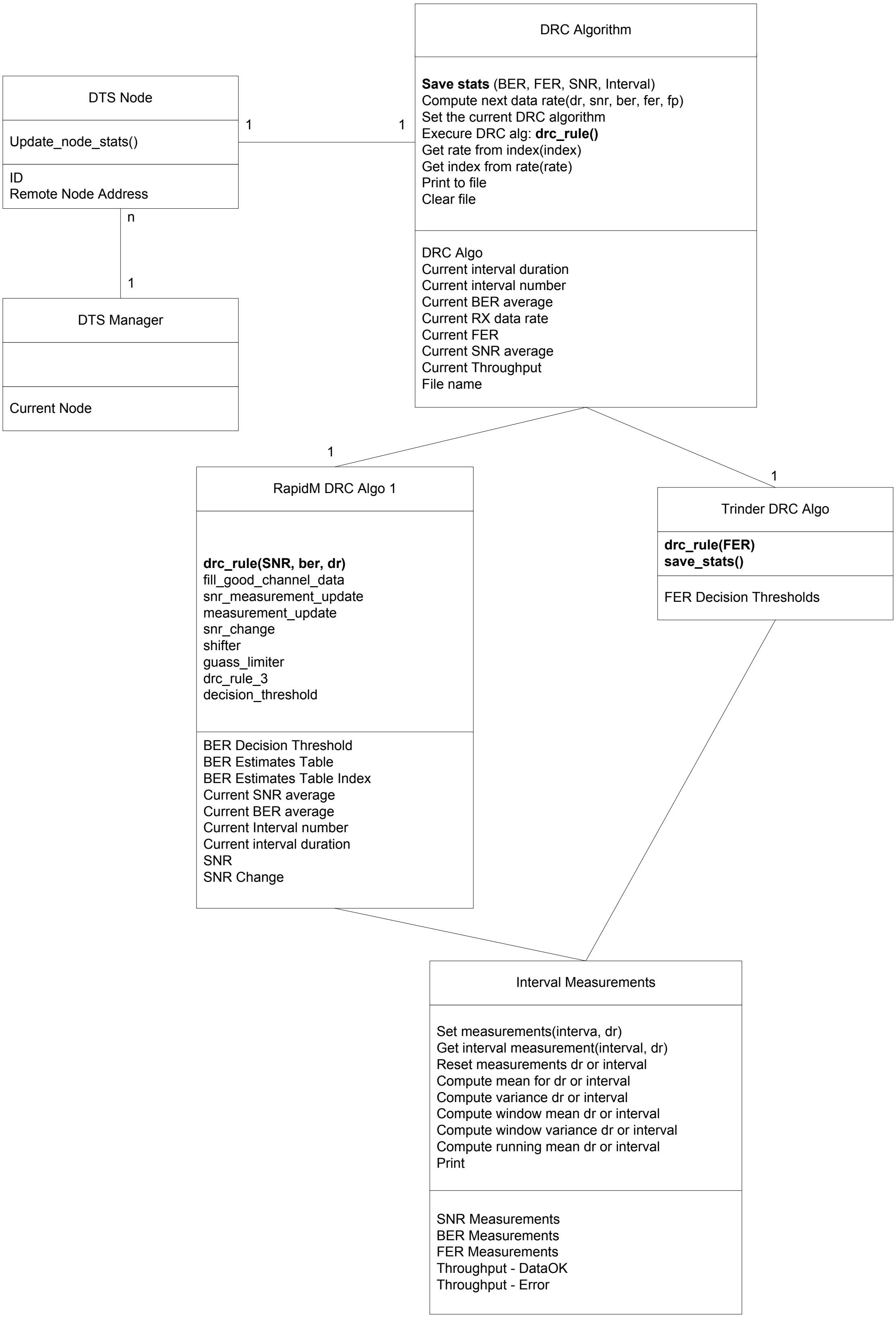

Figure 5.9: DRC algorithm class design.

5.7 TX DATA RATE AND INTERLEAVER SELECTION

All design decision for the retransmission strategy, TX interleaver selection and the selection of the best data rate for sending acknowledgements assume that the waveform used in the communication is the STANAG 4539 waveform.

5.7.1 Purpose

The purpose of this section is to define strategies for four decisions that will increase the robustness of data communication between two nodes. This section deals with the selecting of TX transmission settings, including data rate and interleaver for the transmit interval, i.e. the receive interval for the local node executing the DRC algorithm. It is assumed that the remote node is transmitting a message to the local node. The local node is the node that will execute the DRC algorithm. The four decisions are:

1. select data rate for retried data rate (by remote node);
2. select TX interleaver size based upon data size (by remote node);
3. select TX frame size (by remote node);
4. select data rate when sending acknowledgements (by local node).

For the implementation of the RapidM DRC algorithm 1 the remote node will determine the interleaver size and not the local node.

This solution is proprietary to the RapidM STANAG 5066 node implementation. For communication with other supplier nodes the DRC algorithm will recommend to the remote node to always use the long interleaver.

5.7.2 Implementation

5.7.2.1 Retransmissions

When the remote node cannot be reached retransmissions of data will take place every *retry_time* = 16 s + *random_time* (of between 0 and 8 seconds) (s). The total number of retransmission attempts is 8 and the maximum value of *retry_time* is 20s. Thus the maximum time taken for retransmissions is 24 * 8 = 192 s and minimum time is 16 * 8 = 128 s. The minimum time will thus be long enough for the remote node to finish with any current transmission that has a duration of 127s. The TX data rate to use depends on the number of retry attempts (see Table 5.1).

Table 5.1: Retransmission data rate based upon the number of retry attempts

Attempt number	Data rate to use
1 and 2	Same as current data rate
3 and 4	1 data rate lower than current data rate
5 and 6	2 data rates lower than current data rate
7 and 8	75 bps

5.7.2.2 Interleaver size selection

The interleaver size when using STANAG 4539 can be seen in Table 5.2. The interleaver size selection rule says that the data may only be transmitted using a long interleaver if the overhead of using that interleaver is below 50%. Thus for communication when using 75-2400 bps the data transmission time has to be at least 10.24 seconds long and when using 3200-9600 bps the data transmission time has to be at least 4.32 seconds long. Table 5.3 lists the amount of bytes needed to set the interleaver overhead to below or equal 50%.

Table 5.2: Interleaver sizes in seconds

Data Rate (bps)	Short (s)	Long (s)
75-2400	0.85	10.24
3200-9600	1.08	4.32

Table 5.3: Amount of bytes per data rate

Data Rate (bps)	Bytes
75	96
150	192
300	384
600	768
1200	1536
2400	3072
3200	1728
4800	2592
6400	3456
8000	4320
9600	5184

A better rule would, however, be to either send data using a long interleaver or send data at two rates lower than the current rate with a short interleaver. The advantage of using a data rate that is two data rates lower than the current rate is that a lower SNR is required to produce the same BER. Data will be sent a two data rate steps lower than the current data rate when the:

- interval duration (short interleaver, two data rate steps lower) = interval duration (long interleaver, current data rate).

Using equation 5.18 the number of bytes can be determined that will result in the interval duration to send data at rate2 (the current data rate) will be the same as sending data at rate1 (two data rate steps lower). Note that modem delays, processing delays, waveform EOM (End of Message), SOM (Start of Message) etc. are not taken into account. The decision whether to use the long interleaver or the short interleaver will thus be based upon the number of bytes to be sent and the current TX data rate (see Table 5.4).

$$\begin{aligned}
&TimeToSendAtRate1(s) = TimeToSendRate2(s)\\
&\frac{Bytes}{Rate1} + Short = \frac{Bytes}{Rate2} + Long\\
&\frac{Bytes}{Rate1} - \frac{Bytes}{Rate2} = Long - Short\\
&Bytes \cdot \left(\frac{Rate2 - Rate1}{Rate2 \times Rate1}\right) = (Long - Short)\\
&Bytes = \frac{(Long - Short) \cdot Rate2 \times Rate1}{Rate2 - Rate1}
\end{aligned} \tag{5.18}$$

Table 5.4: The estimated send time

rate2 (bps)	rate1 (bps)	Long (s)	Short (s)	Bytes	Duration rate2 (s)	Duration rate1 (s)
150	75	10.24	0.85	176.06	9.39 + Long (s)	18.78 + Short (s)
300	75	10.24	0.85	117.38	3.13 + Long (s)	12.52 + Short (s)
600	150	10.24	0.85	234.75	3.13 + Long (s)	12.52 + Short (s)
1200	300	10.24	0.85	469.50	3.13 + Long (s)	12.52 + Short (s)
2400	600	10.24	0.85	939.00	3.13 + Long (s)	12.52 + Short (s)
3200	1200	4.32	0.85	832.80	2.08 + Long (s)	5.55 + Short (s)
4800	2400	4.32	0.85	2082.00	3.47 + Long (s)	6.94 + Short (s)
6400	3200	4.32	1.08	2592.00	3.24 + Long (s)	6.48 + Short (s)
8000	4800	4.32	1.08	4860.00	4.86 + Long (s)	8.10 + Short (s)
9600	6400	4.32	1.08	7776.00	6.48 + Long (s)	9.72 + Short (s)

5.7.2.3 Adapting TX frame size

The frame size should be changed based upon the current data rate. This is due to the fact that if the frame size remains fixed, the amount of frames sent at different data rates will differ, for example, a fixed frame size of 250, a data rate of 9600 bps and a window size of 128 frames using *Hi-Speed Asynchronous* mode, only 27 seconds worth of data can be transmitted instead of the maximum of 127s (the interval duration per data rate can be seen in Table 5.5). The interval duration is approximately as calculated in equation 5.19 when the interleaver time is ignored. The approximate number of frames transmitted can be calculated using equation 5.20.

$$Time = \frac{BitsPerByte \times (NumberOfFrames \times FrameLength - Overhead)}{DataRate} \tag{5.19}$$

$$NumberOfFrames = Ceiling\left(\frac{\frac{(DataRate \times Time)}{BitsPerByte} + Overhead}{FrameLength} \right) \tag{5.20}$$

Table 5.5: Interval duration when frame size is 128 bytes

Data rate (bps)	Interval duration (s)
75	128 (max)
150	128 (max)
300	128 (max)
600	128 (max)
1200	128 (max)
2400	107
3200	80
4800	54
6400	40
8000	32
9600	27

The frame size is chosen so that the entire maximum interval duration of 128 seconds can be filled with data and the STANAG 5066 protocol overhead does not exceed 10% of the packet size, based upon a generic D_PDU header size of 22 bytes. In order to maintain the overhead to less than 10% of packet size the frame length has to be more than 220 bytes. The frame size used for each data rate can be seen in Table 5.6.

Table 5.6: Frame sizes chosen per data rate.

Data rate (bps)	Frame size (bytes)	Overhead (%)
75	220	10.00
150	220	10.00
300	220	10.00
600	220	10.00
1200	220	10.00
2400	300	7.33
3200	400	5.50
4800	600	3.67
6400	750	2.93
8000	900	2.44
9600	1000	2.20

CHAPTER 6 TESTING AND RESULTS

6.1 INTRODUCTION

This chapter will discuss the tests carried out to determine the performance of the RapidM DRC algorithm 1 and the Trinder DRC algorithm when implemented in the RapidM STANAG 5066 node. Later in this chapter the results of the tests will be presented.

6.2 DRC ALGORITHM TEST DESIGN

The DRC algorithms will be tested using the following two tests:

- Data throughput test
- Acquisition time test

6.2.1 Test 1 data throughput test

6.2.1.1 Description

This test will be between two nodes, called Node 1 and Node 2, each a STANAG 5066 node that will execute on its own PC. A STANAG 5066 test client will connect to each node. Test Client 1 will send messages containing random data of varying size to Test Client 2. The message size will vary between 200 and 1000 bytes. The data (D_PDUs) will be sent in ARQ mode, thus a soft-link will be set up between Node 1 and Node 2. Node 2 will only acknowledge the received data, with ACK_ONLY D_PDUs. Node 2 will execute the DRC algorithm under test, specifying to Node 1 the TX data rate. The test setup can be seen in Figure 6.1 and the settings can be seen in Table 6.2.

For each DRC algorithm the test will be repeated twice for each type of HF channel, which makes a total of 6 tests per DRC algorithm, a total of 12 tests each lasting 220 minutes. All the tests will be conducted using the same RM6 modems running the same version of RM6 modem code and on the same release version of the RC-66 system. The only variable that will change in the tests will be the type of DRC algorithm under test. Each test will be unique in terms of the HF channel used and the DRC algorithm under test. In test 1 the specific HF channel's SNR value will increase linearly with time.

Table 6.1: Test 1 settings

Test settings	Test setting value
HF channels used	AWGN, CCIR Poor and CCIR Good
D_PDU Frame length	250 bytes
Message size	Between 200-1000 bytes
Test Duration	220 min
SNR start value	-3 dB
SNR end value	35 dB

6.2.1.2 Objective

The objective of the tests is to measure the parameters of interest in Table 6.2. These parameters will then be used to compare the performance of the two implemented DRC algorithms, the RapidM DRC algorithm 1 and the Trinder algorithm. Secondly the parameters will be used to determine whether the BER is a better metric to use in DRC than the FER.

Table 6.2: Test 1 parameters of interest

Parameter name	Parameter description
Data throughput	Data throughput for the entire test duration, measured in bps.
Data rate oscillations	Number of data rate oscillations over the entire test duration. The formula for determining a data rate oscillation can be seen in equation 4.10.
Algorithm robustness	Number of times a data rate change resulted in loss of link during the entire test duration. This value counts the number of times the FER value due to a data rate change is greater than 80 % (equation 4.11).
Average BER	The average BER over the entire test duration
Average FER	The average FER over the entire test duration

The STANAG 5066 data throughput is calculated as the amount of correct bytes received over the total test duration. Thus the data throughput results include the ARQ turnaround time and the time taken by node 2 to send D_PDU acknowledgements. The acknowledgements are sent at 75 bps using a long interleaver. Also note that the frame length was set to 250 bytes for all data rates.

6.2.1.3 Setup

The test setup that will be used for each of the tests can be seen in Figure 6.1. The HF channel will be simulated using the RapidM HFCS [36] and the SNR scenario generator.

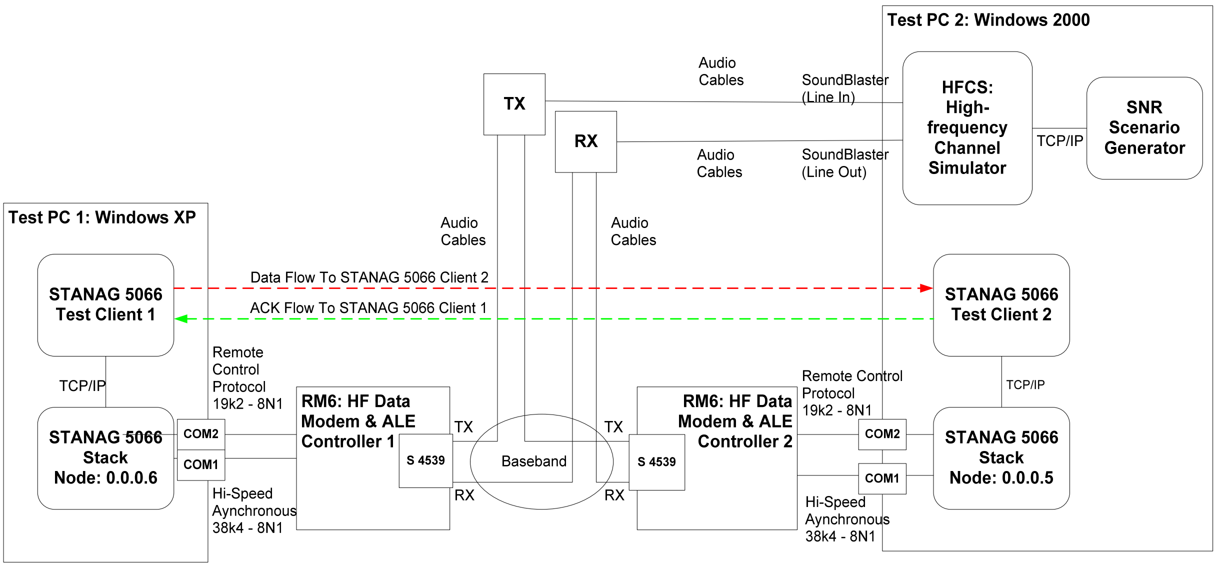

Figure 6.1: DRC algorithm test setup.

6.2.1.4 High-Frequency Channel Simulator

The RapidM HFCS is based upon the traditional Watterson-Coon HF channel model [38]. The HF channel is characterized by multi-path propagation and signal fading. The signal at the receiver may spread by as much as a few milliseconds. A HFCS is used to model the HF channel because it is nearly impossible to reproduce actual channel conditions for channel specific tests. The HFCS is a Windows based application running on a Pentium machine using the PC SoundBlaster for signal sampling. The HFCS includes the ability to handle a maximum of 5 different signal propagations paths. The user is able to adjust the power, Doppler spread and multipath delay of each signal path.

6.2.1.5 SNR scenario generator

The function of the SNR scenario generator is to change the actual channel SNR value of the HFCS based upon the current scenario under test. Currently the scenario generator only supports the linear SNR changes. The SNR scenario generator allows the user to select the settings, as described in Table 6.3.

Table 6.3: SNR scenario generator settings

Scenario settings	Description
Start SNR value	The SNR at the start of the test
End SNR value	The SNR value at the end of the test
Channel	Type of channel for test, this can either be AWGN, CCIR Poor or CCIR Good
Test time	Total time, in minutes, of the test

6.2.1.6 Channel setup

The settings for each HF channel used in test 1 can be seen in Table 6.4.

Table 6.4: Test 1 channel setup

HF Channel	AWGN	CCIR Poor		CCIR Good	
Path number	Path 1	Path 1	Path 2	Path 1	Path 2
Doppler spread (Hz)	N/A	1	1	0.1	0.1
Power (dB)	0	0	0	0	0
Multipath (ms)	N/A	N/A	1.979	N/A	0.521
Input signal modulation	64-QAM	64-QAM		64-QAM	
Noise bandwidth (Hz)	3000	3000		3000	
N/A – not applicable					

6.2.2 Test 2 acquisition time test

6.2.2.1 Description

The setup for tests 1 and 2 are similar, the goals of the tests, however, are different. Test 2 will try to determine how quick the algorithm is to adapt to changing channel conditions. The test will only be conducted using the CCIR Poor channel. Test 2 will be run two times for each DRC algorithm. In total test 2 will be conducted four times. The settings for test 2 can be seen in Table 6.5.

Table 6.5: Test 2 settings

Test settings	Test setting value
HF channels used	CCIR Poor
Message size	Between 200-100 bytes
Test Duration	220 min

The input signal used in test 2 can be seen in Figure 6.2. The input signal was chosen to implement large and small SNR step changes, where large SNR step changes are between 10 and 20 dB and small changes between 0 and 5 dB.

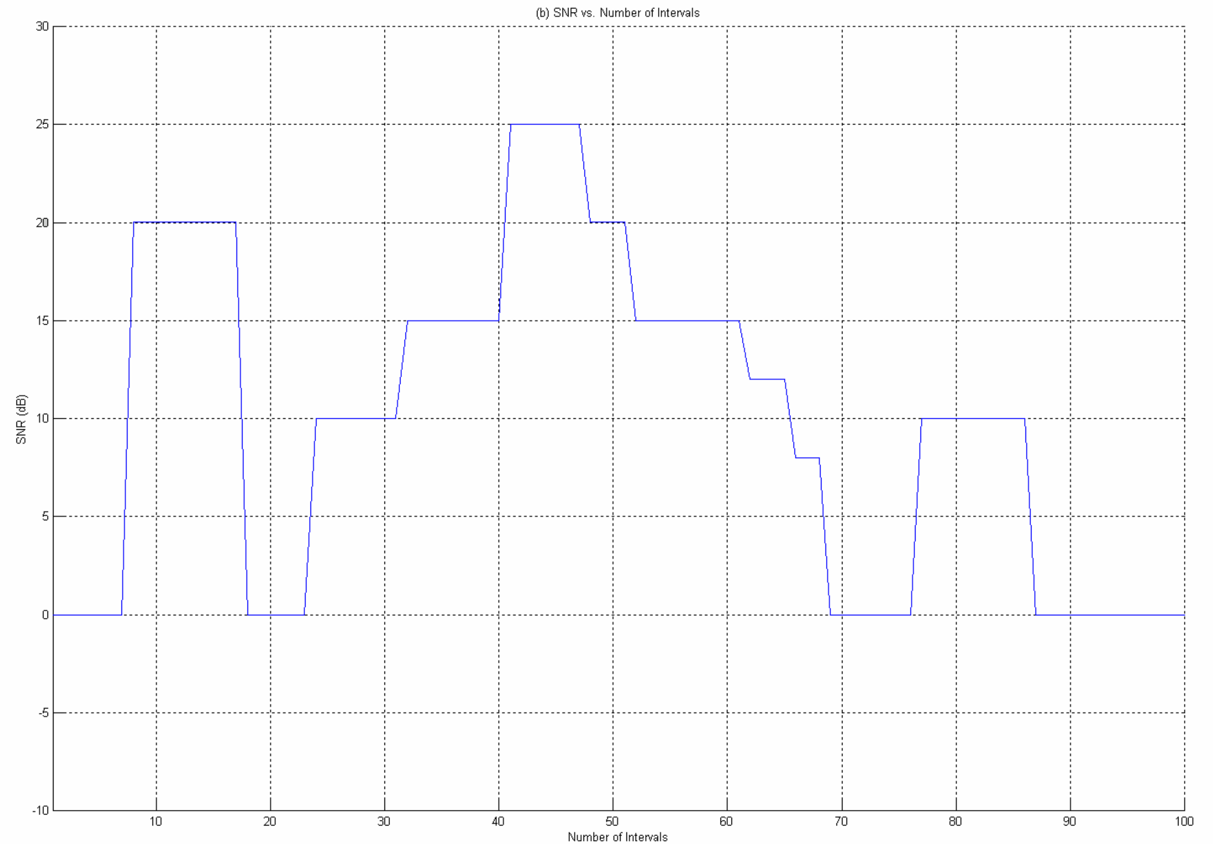

Figure 6.2: Test 2 input signal.

6.2.2.2 Objective

The objective of test 2 is to determine the DRC algorithm response to random changes in channel conditions. The parameters if interest for test 2 can be seen in Table 6.6.

Table 6.6: Test 2 parameters of interest

Parameter name	Parameter description
Algorithm robustness	Number of times a data rate change resulted in loss of link during the entire test duration. This value counts the number of times the FER value due to a data rate change is greater than 80 % (equation 4.11).
Average BER	The average BER over the entire test duration
Average FER	The average FER over the entire test duration
Total acquisition time	The sum of the interval count to reach the optimum data rate for that particular SNR, due to the change in the SNR value

6.2.2.3 Setup

The test 2 setup can be seen in Figure 6.1.

6.3 TEST 1 RESULTS

6.3.1 AWGN channel

The test 1 results for the AWGN channel are reflected in Table 6.7.

Table 6.7: Test 1 results for the AWGN channel

Parameter	Trinder Algo	RapidM Algo 1
Number of intervals	160	203
Average BER (10^{-x})	5.4375	6.740196
Average FER	15.74375 %	1.431373 %
Number of oscillations	61	8
Robustness	17	2
Data throughput	2030.167 bps	2435.536 bps

6.3.2 CCIR Poor channel

The test 1 results for the CCIR Poor channel are reflected in Table 6.8.

Table 6.8: Test 1 results for the CCIR Poor channel

Parameter	Trinder algo	RapidM algo 1
Number of intervals	201	180
Average BER (10^{-x})	5.120775	6.442975
Average FER	18.80889 %	5.768116 %
Number of oscillations	52	10
Robustness	18	2
Data throughput	1239.916 bps	1776.474 bps

6.3.3 CCIR Good channel

The test 1 results for the CCIR Good channel are reflected in Table 6.9.

Table 6.9: Test 1 results for the CCIR Good channel

Parameter	Trinder algo	RapidM algo 1
Number of intervals	210	201
Average BER (10^{-x})	4.9556	6.101695
Average FER	23.80889 %	8.227119 %
Number of oscillations	36	9
Robustness	21	4
Data throughput	912.583 bps	1191.149 bps

6.4 TEST 2 RESULTS

6.4.1 CCIR Poor channel

The test 2 results for the CCIR Poor channel can be seen in Table 6.10. The data rate choice vs. the number of intervals can be seen in Figure 6.3 (the dotted red line is the Trinder algorithm and the solid blue line is the RapidM DRC algorithm 1). The total acquisition time also counts the intervals that data rate choice oscillated between the optimum data rate and the next higher data rate.

Table 6.10: Test 2 results for the CCIR Poor channel

Parameter	Trinder algo	RapidM algo 1
Average BER (10^{-x})	5.306931	5.980198
Average FER	23.58416 %	11.94059 %
Robustness	6	4
Total acquisition time	49	19

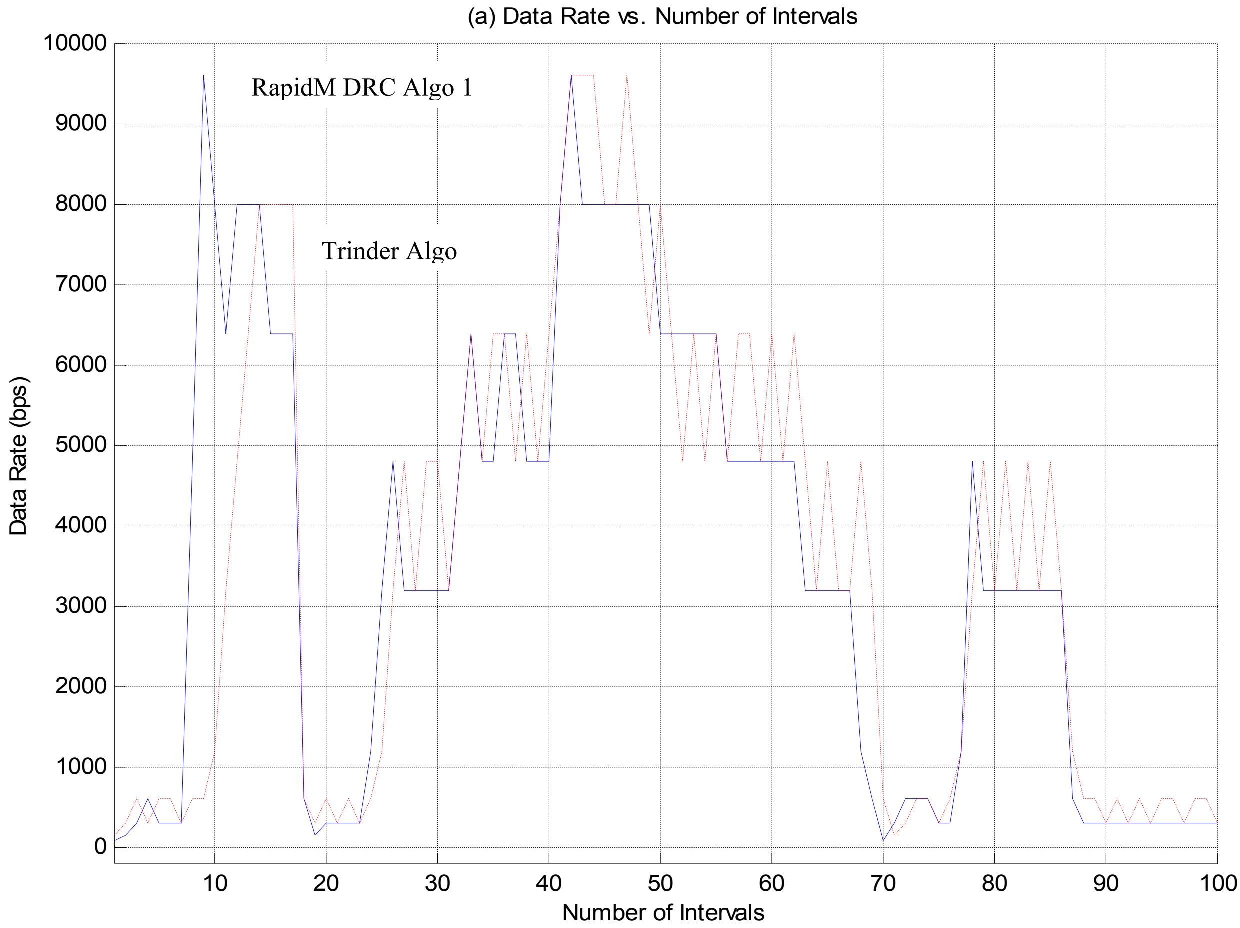

Figure 6.3: Data rate vs. number of intervals for test 2.

6.5 DISCUSSION

6.5.1 Discussion of test 1

Test 1 results can be seen in Table 6.7 for the AWGN channel, in Table 6.8 for the CCIR Poor channel and in Table 6.9 for the CCIR Good channel.

From the results of the data throughput test (test 1) it can be seen that the RapidM DRC algorithm 1 has better throughput for every type of channel when compared to the Trinder algorithm. In fact, the RapidM DRC algorithm 1 has on average 400 bps higher throughput than the Trinder algorithm for all channels.

The RapidM DRC algorithm 1 average BER remains above 10^{-6}, while for the Trinder algorithm the average BER is always below 10^{-6} and goes so low as 10^{-4} for the CCIR Good channel. When the RapidM DRC algorithm 1 average BER is converted to the equivalent FER using equation 4.1 and the frame length is an average of 250 bytes, the results can be seen in Table 6.11.

Table 6.11: Comparison of the converted FER and the actual FER

	Average BER (10^{-x})	Converted FER (equation 4.1)	Actual average FER
AWGN	6.740196	0.036 %	1.431373 %
CCIR Poor	6.442975	0.072 %	5.768116 %
CCIR Good	6.101695	0.158 %	8.227119 %

The reason the actual and converted FER differ from each other is due to the frame errors the RapidM DRC algorithm 1 makes when probing new data rates. This usually causes a very large FER. The difference could also be due to the measurement accuracy of the BER measurements made in the RM6 modem, and a third reason is that the BER average values for each interval have equal weight, this not correct, because the BER for a longer interval should have a larger weight than the BER for a smaller interval. The algorithm could thus be improved by calculating the average BER as the SNR is calculated, weighted by the interval duration.

The RapidM DRC algorithm 1 produces an average FER over the entire test duration that is below 10 %, while the FER for the Trinder algorithm is between 15 and 25% for the three HF channels. This means that the Trinder algorithm message throughput would be decreased due to more retransmissions and the extra link turnaround time lost.

The number of data rate oscillations experienced during DRC algorithm execution is also an important factor in determining performance. As the number of data rate oscillations increase so does the average FER. The average data rates oscillations for the three HF channels for the Trinder algorithm are 49.667, while for the RapidM DRC algorithm 1 the average is only 9. It seems as if the number of data rate oscillations experienced with the Trinder algorithm is dependent on the type of HF channel, while for the RapidM DRC algorithm 1 the oscillations are independent of the channel, but dependent on the amount of data rate probing done when using rule 3 (see chapter 5).

Lastly, when examining the robustness of the three algorithms, the average robustness over all three HF channels for the Trinder algorithm is 18.667 and for the RapidM DRC algorithm 1 it is only 2.667. The RapidM DRC algorithm robustness average is a tenth better than the results for the Trinder algorithm. It seems that the robustness results are channel independent, which is in contradiction to the data rate oscillation results for the Trinder algorithm. It would be expected that the robustness would also be channel dependant, because the robustness of the algorithm is dependant on the number of data rate oscillations due to the fact that it is usually during a data rate oscillation that a FER of greater than 80 % is experienced. For the Trinder algorithm it would be expected that the robustness should decrease like the number of data rate oscillations for each HF channel, where the AWGN channel would have the highest robustness. The reason that this is not the case could be because the Trinder algorithm does not use previous measurements to determine the data rate for the next interval, but only current FER measurements. This means that a data rate choice could be based upon a good FER measurement in a CCIR Good channel that delivers a high SNR during only a short period of time before the channel becomes worse. Also the FER measurements made by the Trinder algorithm is not filtered with receive interval duration or frame length. One FER measurement has the same weight as any other FER measurement independent of frame length and interval duration.

The results of test 1, the data throughput test, is summarised in Table 6.12. The table indicates which algorithm has better results for each parameter of interest. The implementation summary is very similar to the results summary for the simulations (see Table 4.25), except that the data throughput results for the Trinder algorithm does not match the results for the RapidM DRC algorithm 1. This is due to the fact that the RapidM DRC algorithm 1 was redesigned and improved during the implementation phase to decrease data rate oscillations and use changes in the SNR average to predict the BER estimates for each data rate.

Table 6.12: Summary of DRC algorithm implementation results

DRC algorithm	RapidM DRC algo 1	Trinder algo	RapidM DRC algo 1	Trinder algo	RapidM DRC algo 1	Trinder algo
HF Channel	AWGN		CCIR Poor		CCIR Good	
Data throughput	√	x	√	x	√	x
Number of oscillations	√	x	√	x	√	x
Robustness	√	x	√	x	√	x
Average BER	√	x	√	x	√	x
Average FER	√	x	√	x	√	x

From Table 6.12 it can be seen that the RapidM DRC algorithm 1 delivers better performance for every parameter of interest, than the Trinder algorithm.

6.5.2 Discussion of test 2

The RapidM DRC algorithm 1 has a lower average FER over the entire test duration than the Trinder algorithm. The FER average is supported by the average BER value of the RapidM DRC algorithm that is higher than the result for the Trinder algorithm. The robustness results for the two algorithms are very similar. The robustness errors by the RapidM DRC algorithm were mostly made while probing the next higher data rate. This can be seen from Figure 6.3 as the spikes of the solid blue line just after an SNR change.

The initial data rate spike to 9600 bps was due to the BER estimate table being shifted right (see Figure 5.5) based upon the change in SNR of almost 20 dB. It seems as if the BER estimate table has been shifted too far. The assumption of a 3 dB difference between data rates, while being true for an AWGN channel is not a very good assumption for a CCIR Poor channel and definitely not for a CCIR Good channel. From Figure 4.7 it can be seen that the line for the CCIR Good and CCIR Poor channel diverse from the AWGN channel at higher data rates and the difference between data rates for the CCIR Poor channel appears to be an average of 4 dB and for a CCIR Good channel appears to be as high a 5 dB.

The ability to adapt to changing channel conditions is one of the advantages of the RapidM DRC algorithm 1. From Figure 6.3 it can be seen that the RapidM DRC algorithm tracks the

20 dB upward change faster than the Trinder algorithm and is also faster to track the downward SNR changes. For smaller changes in the input signal SNR the time taken by the algorithms are almost similar. From the figure it can also been seen that the RapidM DRC algorithm 1 does not oscillate between data rate choices. The total acquisition time for the RapidM DRC algorithm 1 is 19 receive intervals, while for the Trinder algorithm it is 49 receive intervals (note that the total acquisition time also includes data rate oscillations intervals).

CHAPTER 7 CONCLUSION

7.1 DESIGN AND IMPLEMENTATION OF A DRC ALGORITHM

The author succeeded in designing, simulating implementing and testing a DRC algorithm, called the RapidM DRC algorithm 1 that solves the problem presented to the author, which was to implement a DRC algorithm for the STANAG 4539 waveform. The algorithm will be used in the RC-66 email gateway system to facilitate the highest throughput on a communications link between 2 nodes. The DRC algorithm implemented succeeds in doing the following:

- selecting the optimum data rate for current channel conditions;
- avoiding data rate oscillations;
- adapting rapidly to changing channel conditions.

7.2 COMPARISON WITH OTHER DRC ALGORITHMS

The RapidM DRC algorithm 1 has also produced better results than current DRC algorithm implementations found in literature. The algorithm has solved the data rate oscillation problem and the problem of how to rapidly adapt to changing channel conditions. The algorithm is more robust than current implementations found in literature due to the fact that it can adapt to changing channel conditions, use previous channel measurements and channel information to make better data rate choices.

The DRC algorithm presented in this dissertation is also the first implementation that uses current channel condition information such as BER and SNR to determine the optimum data rate.

The solution proposed for interleaver size selection puts the remote node (node not executing the DRC algorithm) in charge of selecting the interleaver size. The selection of the interleaver size is dependent on the amount of bytes being transmitted at a certain data rate.

7.3 DIRECTIONS OF FUTURE RESEARCH

The main disadvantage of the RapidM DRC algorithm is that it is proprietary, i.e. only interoperable with the RM6 modem. In order to ensure that the system is interoperable with other manufacturer equipment, the FER should be used in the DRC algorithm. Instead of using a BER estimate table as in chapter 5, a FER estimate table can be constructed to determine the optimum data rate for the next receive interval. A method needs to be determined that can convert SNR changes to changes in the FER.

Another possible area of research is the relationship between FER, BER, data rate and frame size in a STANAG 5066 system. From this dissertation it can be seen that the four aspects are related to each other, but no clear mathematical relationships have as yet, as far as is known by the author, been presented in literature.

Other channel parameters such as the SIR (Signal-to-Interference Ratio), Doppler spread, multipath and multipath spread could also be used to determine the current channel profile more clearly for use by a DRC algorithm in selecting the optimum data rate and interleaver size.

A further area of research is to redesign of the current RapidM DRC algorithm 1 to incorporate the principles of fuzzy-logic and/or non-linear control systems to improve the BER estimation and data rate selection process.

REFERENCES

[1] STANAG 4539, "Technical Standard for Non-Hopping HF Communications Waveforms", North Atlantic Treaty Organisation, 2000.

[2] MIL-STD-188-110-A: Interoperability and Performance Standards for Data Modems, U.S. Department of Defence, September 1991.

[3] MIL-STD-188-110-B: Interoperability and Performance Standards for Data Modems, U. S. Department of Defence, April 2000.

[4] P.D. Reynolds and A.F.R. Gillespie, "Interim Profile for Maritime HF Data Communications", IEE 7th International conference on HF Radio Systems and Techniques, Nottingham, UK, July 1997.

[5] F. Eken and D.A. Clark, "A Connection Orientated OSI Data Link Protocol for HF Packet Radio, NATO STC Technical Note TN-492, December 1992.

[6] F. Eken and H. Voordouw, "Laboratory and Field Tests of High Frequency Open Systems Interconnection Data Link Protocol", NATO STC Technical Note TN-506, August 1993.

[7] P. Reynolds, "Maritime Gateway: Extending Terrestrial Network Services over Radio Links", HF 95 Nordic Shortwave Conference Proceedings, Vaxjo, Sweden, August 1995.

[8] STANAG 5066, "Profile for Maritime High Frequency (HF) Radio Data Communication", North Atlantic Treaty Organisation, 2000.

[9] STANAG 4285, "Characteristics of 1200/2400/3600 bits per second Single Tone Modulators/ Demodulators for HF Radio Links", North Atlantic Treaty Organisation, 1989.

[10] STANAG 4529, "Characteristics of Single Tone Modulators/Demodulators for HF Radio Links with 1240 Hz Bandwidth", North Atlantic Treaty Organisation, 1988.

[11] MIL-STD-188-141B: Interoperability and Performance Standards for Medium and High-Frequency Radio Systems, U.S. Department of Defence, 1999.

[12] S.E. Trinder and D.J. Brown, "Algorithms for the Adapation of Data Rate using STANAG 5066", IEE Colloquium on Frequency Selection and Management Techniques, London, UK, March 1999.

[13] J.W. Nieto, "An investigation of Throughput Performance for STANAG 5066 High Data Rate HF Waveforms", Proceedings of IEEE Military Communications Conference (MILCOM) 2002, Anaheim, CA, October 2002.

[14] S.E. Trinder and A.F.R Gillespie, "Optimisation of the STANAG 5066 ARQ Protocol to Support High Data Rate HF Communication", Proceedings of IEEE Military Communications Conference (MILCOM) 2001, Washington, DC, October 2001.

[15] High-Frequency Radio: Automatic Link Establishment (ALE) Application Handbook, NTIA Report 99-360, September 1998.

[16] A.F.R Gillespie, S.E. Trinder and D.J. Brown, "Client Application Considerations for Low Bandwidth Communications using STANAG 5066", Proceedings of IEEE Military Communications Conference (MILCOM) 2001, Washington, DC, October 2001.

[17] J. Postel, "Simple Mail Transfer Protocol", Network Working Group Request for Comments, RFC 821, 1982.

[18] N. Freed, "SMTP Service Extension for Command Pipelining", Network Working Group Request for Comments, RFC 2920, 2000.

[19] J. Myers, "Post Office Protocol - Version 3", Carnegie Mellon, Network Working Group Request for Comments, RFC 1939, 1996.

[20] R. Gellens, C. Newman and L. Lundblade, "POP3 Extension Mechanism", Network Working Group Request for Comments, RFC 2449, 1998.

[21] D.C. Plummer, "An Ethernet Address Resolution Protocol", Network Working Group Request for Comments, RFC 826, 1982.

[22] J. Postel, "Internet Control Message Protocol", Network Working Group Request for Comments, RFC 792, 1981.

[23] K. Nichols, S. Blake, F. Baker and D. Black, "Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 headers", Network Working Group Request for Comments, RFC 2474, 1998.

[24] W. Simpson, "The Point-to-point Protocol", Network Working Group Request for Comments, RFC 1548, 1994.

[25] P. Deutsch and J-L. Gailly, "ZLib Compressed Data Format Specification Version 3.3", Network Working Group Request for Comments, RFC 1950, 1996.

[26] P. Deutsch, "DEFLATE Compressed Data Format Specification Version 1.3", Network Working Group Request for Comments, RFC 1951, 1996.

[27] P. Deutsch, "GZIP File Format Specification Version 4.3", Network Working Group Request for Comments, RFC 1952, 1996.

[28] U.S. Federal Standard 1052, "Telecommunications: HF Radio Modems", 1995.

[29] S.E. Trinder, A.F.R Gillespie and M. Street, "Data Rate Change Algorithms for Maritime HF Subnetworks. HF98 Nordic Shortwave Conference, August 1998.

[30] E.E. Johnson, "HF Radio ARQ Protocol Analysis", 2002, online presentation, available at: http://www.hfindustry.com/jun02/presentations/eej_6_02.ppt. Last accessed on 12 May 2005.

[31] E.E. Johnson, M. Balakrishnan and Z. Tang, "Impact of Turnaround Time on Wireless MAC Protocols", Proceedings of IEEE Military Communications Conference (MILCOM) 2003, Boston, MA, October 2003.

[32] E.E. Johnson, G. Anaya, Z. Tang, M. Balakrishnan, H. Zhang and S. Sreepuram, "Performance of the HF Token Protocol", Proceedings of IEEE Military Communications Conference (MILCOM) 2004, Monterey, CA, Nov 2004.

[33] M. Balakrishnan and E.E. Johnson, "Queuing Analysis of DCHF and HF Token Protocol with Varying Turnaround Time" Proceedings of IEEE Military Communications Conference (MILCOM) 2004, Monterey, CA, Nov 2004.

[34] M.T. Holcomb and J.H. Weston, "Enhancing Channel Utilization and Performance of STANAG 5066", Proceedings of 9th International Conference on HF Radio Systems and Techniques, pp.180-184, June 2003.

[35] RM6 HF Data Modem & ALE Controller, Users Manual, revision 0L, RapidM (Pty) Ltd., 15/04/2005.

[36] HFCS: High Frequency Channel Simulator, User's Manual, revision 0B, RapidM (Pty) Ltd., 14/07/2004.

[37] L. Kirkup, "Experimental Methods: An Introduction to the Analysis and Presentation of Data", Queensland, Australia: John-Wiley & Sons, 1994.

[38] C. Watterson, J. Juroshek and W. Bensema, "Experimental confirmation of an HF channel model", IEEE Transactions on Communications Technology, vol. COM-18, no.6, pp. 792-803, December 1980.

TABLE OF FIGURES

TABLE OF TABLES

Table of Tables