Direct sequence spread spectrum techniques in local area networks

Smythe, Colin

DIRECT SEQUENCE SPREAD SPECTRUM TECHNIQUES IN LOCAL AREA NETWORKS

by

Colin Smythe, B.Sc.

A thesis submitted in accordance with the regulation for the degree of Doctor of Philosophy in the Department of Applied Physics and Electronics at the University of Durham.

August 1985

Direct Sequence Spread Spectrum Techniques in Local Area Networks

Colin Smythe

ABSTRACT

This thesis describes the application of a direct sequence spread spectrum modulation scheme to the physical layer of a local area network: subsequently named the SS-LAN. Most present day LANs employ one form or another of time division multiplexing which performs well in many systems but which is limited by its very nature in real time, time critical and time demanding applications. The use of spread spectrum multiplexing removes these limitations by providing a simultaneous multiple user access capability to the channel which permits each and all nodes to utilise the channel independent of the activity being currently supported by that channel. The theory of spectral spreading is a consequence of the Shannon channel capacity in which the channel capacity may be maintained by the trading of signal to noise ratio for bandwidth. The increased bandwidth provides an increased signal dimensionality which can be utilised in providing noise immunity and/or a simultaneous multiple user environment: the effects of the simultaneous users can be considered as noise from the point of view of any particular constituent signal. The use of code sequences at the physical layer of a LAN permits a wide range of mapping alternatives which can be selected according to the particular application. Each of the mapping techniques possess the general spread spectrum properties but certain properties can be emphasised at the expense of others. The work has involved the description of the properties of the SS-LAN coupled with the development of the mapping techniques for use in the distribution of the code sequences. This has been followed by an appraisal of a set of code sequences which has resulted in the definition of the ideal code properties and the selection of code families for particular types of applications. The top level design specification for the hardware required in the construction of the SS-LAN has also been presented and this has provided the basis for a simplified and idealised theoretical analysis of the performance parameters of the SS-LAN. A positive set of conclusions for the range of these parameters has been obtained and these have been further analysed by the use of a SS-LAN computer simulation program. This program can simulate any configuration of the SS-LAN and the results it has produced have been compared with those of the analysis and have been found to be in agreement. A tool for the further analysis of complex SS-LAN configurations has therefore been developed and this will form the basis for further work.

ACKNOWLEDGEMENTS

As in most cases of human endeavour the completion of this project is due to the help of many people and organisations. Firstly, I would like to thank the SERC for providing the financial support for the first year of this work, without which the work would never have started. Secondly, I am indebted to the scientists and technicians of the XCC division, based at ARE Portsmouth, who permitted me to use their computing facilities and then tirelessly provided answers and solutions to my many questions and problems.

I believe that the environment in which any work takes place is of primary importance to the success of that work. The academic staff, technicians and postgraduates of the Department of Applied Physics and Electronics have provided endless hours of entertainment and distraction which while thwarting my attempts to push back the frontiers of science have also provided that ideal environment.

Whilst there are many people to whom I am indebted, there are three of whom I am particularly appreciative. The first is Professor G.G. Roberts F.R.S. who during my years as an undergraduate, postgraduate and member of staff has always been a constant source of encouragement and who has been a guiding hand in my career. The second is my fiancee Christine, who has provided understanding, patience and support particularly when I have deserted her for the comforts of our study and the writing of my thesis, and whom I love very much.

The final person is in many respects the person to whom I owe the most. Professor C.T. Spracklen has always provided the right kind of guidance and criticism and was responsible for identifying the potential of the original idea. He has been an invaluable source of ideas and was always willing to act as "devils advocate" in our may discussions concerning the early development of the work. More importantly, he has become a valued friend whose views and outlook I will always respect.

I am indebted to all of these people and would like to give them my sincerest thanks and appreciation.

To Chris

CONTENTS

APPENDICES

Glossary of Terms

ACF	Autocorrelation Function
ADC	Analogue to Digital Converter
ASH	ASWE Serial Highway
AWGN	Additive White Gaussian Noise
BER	Bit Error Rate
CCF	Cross Correlation Function
CDM	Code Division Multiplexing
CDMA	Code Division Multiple Access
CPU	Central Processing Unit
CRC	Cyclic Redundancy Check
CSMA	Carrier Sense Multiple Access
CSMA/CD	Carrier Sense Multiple Access with Collision Detection
DEC	Digital Electronics Corporation
DPL	Data Processing Loss
DPR	Data Processing Resolution
DS	Direct Sequence
DSSS	Direct Sequence Spread Spectrum
EAROM	Electrically Alterable Read Only Memory
ECL	Emitter Coupled Logic
EOD	End of Data
Erfc	Complementary Error Function
ESSEX	Experimental Solid State Exchange
FDM	Frequency Division Multiplexing
FDMA	Frequency Division Multiple Access
FH	Frequency Hopping
FSK	Frequency Shift Keying
GDA	Global Data Area
IC	Integrated Circuit
IDN	Integrated Domestic Network
IEEE	Institute of Electrical and Electronic Engineers
I/O	Input/Output
ISO	International Organisation for Standardisation
LAN	Local Area Network
LANCE	Local Area Network Controller for Ethernet
m-Length	Maximal Length

MMI	Man/Machine Interface
NRZ	Non Return to Zero
OSI	Open Systems Interconnect
PCC	Polarity Coincidence Correlator
PG	Processing Gain
PN	Pseudorandom
PSD	Power Spectral Density
PSK	Phase Shift Keying
RAM	Random Access Memory
ROM	Read Only Memory
SCF	Simulation Command File
SGDA	Shared Global Data Area
SIK	Sequence Inversion Keying
SIL	Station Interface Logic
SILK	System for Integrated Local Communication
SNL	Station Node Logic
SNR	Signal to Noise Ratio
SS	Spread Spectrum
SSART	Spread Spectrum Asynchronous Receiver and Transmitter
SSF	Simulation Scenario File
SSMA	Spread Spectrum Multiple Access
SSNC	Spread Spectrum Node Controller
TDM	Time Division Multiplexing
TDMA	Time Division Multiple Access
TH	Time Hopping
VLSI	Very Large Scale Integration
VMS	Virtual Memory System
WAN	Wide Area Network

Mathematical Nomenclature

T=	Total Signal Duration
W=	Signal Bandwidth
D=	Signal Dimensionality
R_c=	Code Sequence Frequency
R_d=	Data Rate Frequency
C=	Channel Capacity
P=	Average Transmitter Power
N=	Average White Thermal Noise Power
N_o=	Average White Thermal Noise Power per Hertz
T_d=	Data Bit Period
T_c=	Chip Period
f_c=	Carrier Frequency
L=	Code Sequence Length
$C_{xx}(l)$=	ACF with a relative displacement of l bits
$C_{xy}(l)$=	CCF with a relative displacement of l bits
$C_1^2(t)$=	ACF in the analogue form for node one
τ_i=	Code time displacement between nodes 1 and i
B_{ss}=	Bandwidth of the Spread Spectrum signal
B_d=	Bandwidth of the data signal
m=	number of code sequences per data sequence
P_e=	Probability of Error
μ=	mean
σ^2=	variance
σ=	Standard Deviation
$\Phi()$=	Cumulative Gaussian Distribution Function
$Q()$=	Complementary Error Function
n=	Shift Register Length
f=	Code Sequence family size
N=	Maximum Number of Simultaneous Users possible
k=	Number of simultaneous actually transmitting
R_{tot}=	Total Network Wide Data Throughput
τ=	Propagation Delay along the total Channel
B=	Message Length in Bits
D_t=	Data Delay
S=	Data Supported by the Nodes for Transmission

$r(t)=$ Signal Received at the Front end of the Receiver
$SNR_c=$ The Chip SNR at the front end of the receiver
$\overline{SNR}_i=$ The Average SNR at the Output of the i^{th} receiver
$P=$ Probability of correct decoding by the Hardlimiter
$\overline{SNR}_c=$ The Average SNR after Hardlimiting
$\mu=$ Network Efficiency during Operation
$\delta=$ Code Sequence Orthogonality
$f(P)=$ Near-Far Power Ratio
$SNR_1=$ Single User SNR on the Channel
$SNR_d=$ Single User SNR after SS demodulation
$E_b=$ Signal Energy per Data Bit
$\eta=$ Processing Gain
$f(R)=$ Aperiodic ACF and Waveform function
$\overline{P}_i=$ Probability of Error from the i^{th} node
$G=$ Total Data presented to the complete network per second
$C_{xy}=$ Correlation Value
$C_t=$ Correlation Threshold
$AG=$ Number of Agreements in the Correlation
$DG=$ Number of Disagreements in the Correlation
$s_k()=$ Spread Spectrum signal from the k^{th} node
$c_k()=$ Code Sequence for the k^{th} node
$d_k()=$ Data Signal for the k^{th} node
$n(t)=$ Ambient Electrical Noise
$\tau_k=$ Relative Delay between the Reference and k^{th} nodes
$l=$ Discrete Displacement for a Code Sequence
$C(l)=$ Aperiodic Discrete Correlation Function
$d_{x,y}=$ Data Characteristics for Correlation
$ACF_e=$ Even (Periodic) ACF
$ACF_o=$ Odd ACF
$CCF_e=$ Even (Periodic) CCF
$CCF_o=$ Odd CCF
$\theta_i=$ Periodic ACF
$\widehat{\theta}_i=$ Odd ACF
$\theta_{x,y}=$ Periodic CCF
$\widehat{\theta}_{x,y}=$ Odd CCF
$\overline{v}=$ Signal and Noise Degradation per metre on the Channel
$l_s=$ Average Displacement between the Source and Destination
$l_n=$ Average displacement between the Noise and destination

$V=$	Bipolar Transmission Voltage
$\gamma=$	Propagation constant
$\alpha=$	Attenuation constant
$\beta=$	Phase constant
$\sigma=$	Conductivity
$\mu=$	Permeability

CHAPTER 1

Introduction

1.1 A History of Electronic Communications

One of the first communication systems to use electrical methods was proposed by Charles Morrison in 1753. He was a Scottish surgeon and suggested a system in which each character of the alphabet was allocated a wire (plus one wire for ground) and where the receiver was constructed from pithballs and paper with preprinted characters on it. Also during the mid 18^{th} century, a French monk, named Abbe Nollet, was demonstrating what was probably the first ring topology local area network. He connected together two hundred Carthusian monks with lengths of iron wire to form a circle of approximately 1.5 kilometres in diameter. To show that an electric signal travelled quickly around such a ring he then connected one of the lengths of wire to an electrostatic generator, promptly electrocuted all 200 monks and thus proved his point.

Throughout the rest of the 18^{th} century development of the telegraph system was slow due to the problems of charge storage and it was not until the invention of the battery by Galvani and Volta that electricity and not electrostatics was investigated. The demand for a telegraph system was stimulated by the development and growth of railway systems (the first public railway opening in England in 1830) and by 1837 the first true telegraph systems were invented by Morse in America and Cooke and Wheatstone in England. The first commercial telegraph was introduced in 1844 and this was further refined until 1880's, at which time the telephone was under development.

Bell, another Scotsman, invented the telephone in 1876 and this created the dominance of analogue signalling techniques over the digital methods - as used in telegraphy. In 1880 the Postmaster General obtained the rights to the telephone, after several lawsuits, which gave the Post Office licensing until 1911. Consequently this stopped the American companies, which had patented several telephone system designs, from producing any such system until 1912. From that time onwards the telephone has become the most important method of communication for people all over the world and its development has seen the introduction of digital switching techniques, satellite links, fibre optic cabling and, most recently, cellular personal radio telephones.

A development from the telegraph occured in the 1890s when a telegraph operator, named F.G.Creed, became frustrated by the use of stick perforators in the production of morse coded papertapes. Instead he connected a typewriter to a paper tape puncher and used it to generate the morse coded papertapes. He later extended this system to receive morse signals and to automatically punch the equivalent received papertape message. This was the embryo of the telex system which uses a separate public exchange network to link teleprinters in a fashion similar to that of the telephone network. An important facet in the telex system is the automatic acknowledgement, by the receiver, of the request for start of transmission sent by the transmitter. This is necessary because a receiver can operate without human intervention and consequently it is the precursor to the more sophisticated communication protocols used in computer systems.

The end of the 19^{th} century also saw the development of the first radio systems, particularly by Marconi who by 1897 was transmitting signals over a distance of nine miles through the atmosphere. Transatlantic radio communication first occurred in 1901 between Cornwall and Newfoundland and by 1910 the first radio systems were installed on commercial ships. Marconi spent the rest of his life modifying radio techniques and when he died in 1937 the world paid tribute to him with a two minute total radio silence. Development of the radio produced the standard, present day, modulation schemes such as amplitude modulation (single sideband, double sideband etc) and frequency modulation which provides improved noise rejection. The radio telephone was developed during the second World War, as was radar, and in 1947 it was introduced commercially for a fleet of tugs on the river Tyne.

During the 1950's two developments were to take place which would start a radical change in world communications. The first was the introduction of commercially available computers and the second was the launch of the first satellite, Sputnik, in 1957. The computers were digitally based and whilst they were not as compact as their present day equivalents they were based on semiconductor technology eg the 1952 Gamma 3 which used Germanium diodes and was built in West Germany. The American computers at that time were ENIAC, built in 1946, the EDVAC, built in 1949 and UNIVAC 1 which was built in 1952 and was the first commercially available computer. Transistorised computers were introduced in 1960 by Control Data Corporation with integrated circuits being first used in 1965. Digital technology was now well established and this encouraged the use of digital techniques in other systems which traditionally had been analogue based; as in the case of the telephone.

In 1959 the first digital telephone switching system was proposed by American Telephone and Telegraph company (ATT) and was named ESSEX (1), the Experimental Solid State Exchange. This provided the basis for all further work on digitally switched systems and in 1962 T1, the first commercial system of this type, was introduced by Bell Systems. By the end of that year 250 digital communication circuits had been installed and by 1976 that number exceeded three million installations.

After the shock of the Russian success with Sputnik the Americans had by 1960 launched the first communications satellite, Echo I, and in 1964 this was joined by Echo II. Both of these were passive satellites and were used to reflect radio signals across the world thereby solving the problems encountered with atmospheric reflection techniques (which had limited the range of point to point radio communications). Rapid development of active satellites followed and at the present time satellites must perform complex signal processing to efficiently share the available bandwidth and time between the ground stations. Satellites now provide blanket coverage of the globe and enable the transmission of television signals and telephone conversations from one side of the world to another.

By the end of the 1960's the computer was firmly established as a major data processing tool but there was a growing need for a means to interconnect physically distributed computers so that they could share common data. The interconnection requirement for computers produced a large number of problems which had not arisen in radio and telephone networks because the computer had to be instructed on error correction and conversation protocols (derived from the experiences gained using the telex system), which a human performs naturally. The first full computer network was ARPANET which used the existing telephone networks but which also implemented a packet switching protocol to provide both error control/recovery and a standard conversation protocol. These types of networks are now termed "wide area networks" (WANs) and can be categorised as low data rate, high cost networks.

The 1970's was a period when large mainframe computers were slowly replaced by minicomputers and then microcomputers. The emphasis changed from the use of a large central processor to the use of individual workstations which although they provided localised processing power could not overcome the cost of providing specialised peripherals at each microprocessor. The answer is the local area network (LAN) which provides high data rate communications at relatively low cost for a physically localised community of workstations. The first commercially available LAN was licensed by Xerox PARC, called the Ethernet, and even though many other forms of LANs are now commercially available this format is still the most widely used type of LAN. The various standards agencies across the world are now attempting to define the protocols for networks in general and are currently providing the detailed specifications: the IEEE 802 commitee has now published its final version of the standards for the lower protocol layers. These are being applied to LANs, using the Ethernet and Token passing LANs as the basic communications medium.

Ironically, the trends in computer processing have also turned full circle and the requirement for coordinated processing power is implemented by using distributed systems which are interconnected with LANs. LANs are also interconnected using bridges and gateways with WANs being used to connect the distributed LANs. Satellite links are also used in this interconnection, with Project Universe being the prime example in the UK. The current trends are concerned with the application of LANs to replace the telephone system, from which it has grown, and it is this voice/data mixing which is producing a challenge for present LANs. Unfortunately the simplicity of the first LANs with their lack of need of sophisticated routing algorithms, error correction and recovery, packet protocols etc is now being lost due to their application to WAN environments.

1.2 Information Theory

In 1947 C.E. Shannon changed the understanding of communication systems when he published his paper on channel capacity (2) and thereby created information theory. Not only did this paper give insight into performance limitations of communication systems it also introduced the concept of code division multiplexing as a means of transferring data and thereby rivalling the established multiplexing techniques: time division and frequency division. It is important to note that Shannon's paper did not provide methods by which to achieve the maximum performance for a system but defined the performance for a system with certain properties. This section will define the components of a communications system, it will define the channel capacity and will describe its relevance to the introduction of spread spectrum techniques.

1.2.1 A Communications System

A schematic representation of a general communication system is shown in figure (1.1) and can be described by three components: the transmitter, the channel and the receiver. The transmitter consists of a data source and the actual transmitter which converts the data signal to a format suitable for transmission over the channel. The channel is the medium which connects the transmitter and receiver and could be the atmosphere, a pair of wires or coaxial cable etc. The signal may be disturbed in the channel by noise or distortion. The noise may be from other transmitters or it may be part of the normal electrical environment through which the channel crosses. The difference between distortion and noise is that noise is statistical in nature whereas distortion is always fixed for a particular channel and may, in principle, be removed by applying the inverse operation. Noise perturbations are impossible to remove in total and as such it is the noise which limits the performance of a communications link. The third section is composed of the receiver, which performs the reverse operation to the transmitter and the data destination. The actual

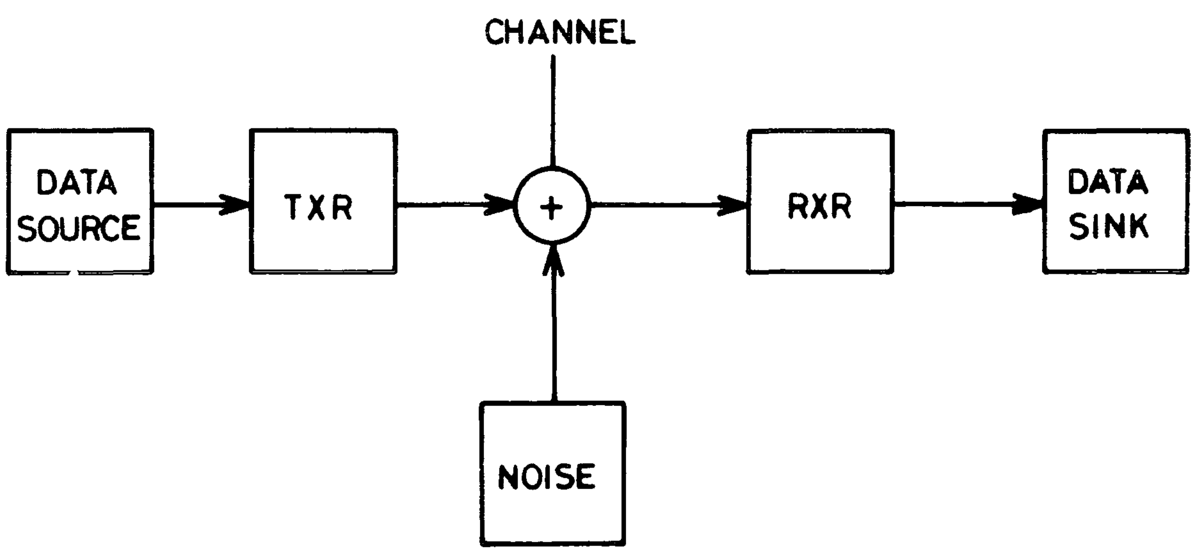


FIGURE (1.1): A General Communications System operation of the receiver will vary depending on the noise with which it must contend and consequently the data recovered from the received signal will contain some errors and will not be a replica of the original data. This representation of a communications system is true for all types of systems whether they be radio or cable, analogue or digital and single or multiple user.

1.2.2 Shannon's Channel Capacity Equation

The definition of the Channel Capacity equation is given by Shannon's second theorem which states, "Let P be the average transmitter power, and suppose the noise is white thermal noise of power N in the band W. By sufficiently complicated encoding systems it is possible to transmit binary digits at a rate:-

$$C = W \, \mathrm{Log}_2 \left[\frac{P+N}{N} \right] \qquad (1.1)$$

with as small a frequency of errors as desired. It is not possible by any encoding method to send at a higher rate and have an arbitrarily low frequency of errors." The capacity (C) is usually given in bits per second. The value of (P/N) is defined as the signal to noise ratio (SNR) of the system and this can be redefined in a different form if the noise is assumed to have the average power per hertz of N_o, thus $N=N_oW$ and the capacity equation becomes:-

$$C = W \, \mathrm{Log}_2 \left[1 + \frac{P}{N_oW} \right] \qquad (1.2)$$

By suitable manipulation and use of the identity $\mathrm{Ln}\,(1+x/n)^n = x$, as the limit of n tends to infinity then:-

$$C_{\lim W \to \infty} = \frac{P}{N_o} \mathrm{Log}_e 2 \quad = 1.44 \left[\frac{P}{N_o} \right] \qquad (1.3)$$

which gives the channel capacity as the bandwidth tends to infinity. What exactly does the equation mean with respect to the design of a digital communication system?

Several explanations of this equation are available with the Shannon and Weaver (3) text providing one in a form for the non-mathematicians, however a more technical approach is presented by Costas (4) who modifies this equation to make it more useful for practical systems and then explains how it revolutionised the thinking behind communications. The capacity equation shows how the SNR, bandwidth and transmission rate are related and limited by the desired error rate. If for particular radio system the capacity of the channel is calculated for the bandwidth and SNR (using equation (1.1)) then the actual transmission rate of the data must be less than this capacity value or else a coding system cannot be designed for a given small error rate ie the system will operate but it will produce a larger than desired number of errors.

1.2.3 Signal Space and Spread Spectrum Modulation

Shannon developed the channel capacity formula by considering a signal as a point in n-dimensional space, where the number of dimensions was defined by the period of the signal and its bandwidth. If a signal has a period T and is composed of frequencies less than W then the dimensionality is defined by the equation:-

$$D = 2TW \tag{1.4}$$

From the Shannon theorems the capacity may be increased for a fixed data rate by increasing the SNR or bandwidth. Equation (1.3) shows the capacity limit as the bandwidth tends to infinity thus the use of coding of the data to increase the bandwidth will increase this capacity. This can be shown if the value of T is set as the data rate reciprocal, ie $T=1/R_d$ and the bandwidth is set as the coding data rate of $W=R_c$, then equation (1.4) becomes :-

$$D = \frac{2R_c}{R_d} \tag{1.5}$$

The ratio of R_c/R_d is the number of bits required to code each data bit and is termed the code sequence length. This technique is termed direct sequence spread spectrum (DSSS) and uses a code sequence to spectrally spread the data over as wide a bandwidth as possible thereby enabling a higher channel capacity OR permitting a lower SNR at the receiver.

Each of the D-dimensions of the space is orthogonal to the others (cf three dimensions in physical space) hence the use of one signal leaves D-1 positions which may be occupied by other signals, provided they are orthogonal to each other. This provides the multiple user access capablility of SS modulation thus the maximum number of simultaneous users is limited by the length of the code sequence as well as the orthogonality of the signals. Real code sequence families are not truly orthogonal and so they cause mutual interference thereby reducing the number of possible simultaneous codes and hence signals. The effect of electrical noise is to convert the single point in signal space into a volume, an extension of a sphere whose radius is defined by the noise itself. This volume will now reduce the signal volume available hence the number of signals will again be reduced. An interesting article on the packing of spheres (Sloane, 5) states that the packing limit is only 75% of a given volume (the theoretical limit of 78% has never been proved) and this suggests that the maximum number of users can never be greater than 75% of the code sequence length.

1.2.4 Code Division and Spread Spectrum Multiplexing

Code division multiplexing and spread spectrum multiplexing are frequently referred to as one and the same. In many cases this is acceptable however there is one fundamental difference between the two. In a SS scheme the data must be transmitted over an increased bandwidth and the implementation of this produces the spectral spreading as explained in the previous section. Typical values of this spreading are of a ratio of between 100 and 10^4 or 20dBs to 40dBs. The spreading sequence does not represent the data directly ie it is used to change the form of the data on a cyclic basis. In the case of CDM the codes are assigned to the data types and it is the data itself which is transmitted in symbolic form. The length of the code sequence is now determined by the number of data types between which the receiver must differentiate and while the coding produces a SS effect the code sequences are, in general, shorter and the receiver must be able to distinguish between all of the codes for reliable data reception.

1.3 Communication Techniques in Cable Systems

Cable based communication systems may be categorised by the multiple access scheme applied at the transmissiom level and the type of channel across which they transmit. The distinction between cable systems and radio systems is dominated by the difference in the noise environments. For radio the atmosphere provides nonlinear noise which varies not only with the frequency of transmission but also with the physical location. In contrast cable systems generally have a low noise level but the channel bandwidth is restricted thus, as shown by Shannon, the noise could be of greater significance.

1.3.1 The Channel

The channel for LANs is a cable which is usually coaxial or twisted pair but which more recently has taken the form of an optical fibre. The choice of cable type is dependant on the frequency of transmission and subsequent channel degradation with length, the noise immunity and the cost per metre when the price of active repeaters has been included.

Twisted pair cabling is both cheap and readily available and many installations make use of spare wires in their telephone wiring. It is used primarily in point to point links eg the Cambridge Ring, but require shielding from noise and has limited range unless repeaters are available. Coaxial cables are by far the most popular LAN medium because they are comparatively cheap, robust,light weight, have good electrical isolation, reasonable range and provide the capacity for high data rates. Coaxial cable is also readily available in differing qualities and tap-in points for nodes are simple and reliable to install. The coaxial cables can be replaced by twinax and triax cables which have improved electrical properties but which are more expensive.

The use of optical fibres in LANs will eventually become universal once the price becomes competitive with the other interconnection technologies - this trend was reported by Polishuk (6) in his survey of fibre development in the USA. Optical fibres provide a considerably improved noise environment (immune to electromagnetic interference) coupled with the provision of a wide bandwidth (several hundreds of MHzs). Fibre optic systems are always point-to-point implementations because whilst they can provide a wavelength multiplexing scheme they cannot provide a multilevel system in which the signal level on the channel is proportional to the number of component signals. The move from electrical to optical channels does not require the redesign of the present LANs but does necessitate the inclusion of a logic to optical converter to replace the current logic to electrical converters and this may also have to incorporate some data buffering to compensate for the high data rates available using fibres.

1.3.2 Multiple Access Techniques

At present there are only three completely different techniques which can be employed to provide a multiple access capability: time division, frequency division and code division multiplexing (TDM,FDM and CDM). These three are a reflection of the representation of a signal and because code division is the basis for the new LAN this section will concentrate on the other two methods.

FDM is implemented by allocating frequencies to communicating nodes. Each frequency is orthogonal to other frequencies however when modulation schemes, such as FSK, are then applied and noise effects considered the individual frequency is transformed into a frequency bandwidth. For simultaneous multiple access these bandwidths are slotted alongside each other within the channel bandwidth with the number of users equal to the number of bandwidths. Frequency bands must also be introduced to curtail the intermodulation effects and within a bandwidth limited channel this raw allocation can be inefficient hence frequency sharing is usually implemented with a central controller; this allocates specific frequencies to nodes requesting permission to begin transmission.

With few exceptions all LANs fall into the category of TDM. Here the aim is to ensure that only one node transmits data at any one time and whether this is implemented by collision detection, CSMA/CD, as in Ethernet or by time sharing slots, as in the Cambridge Ring, the aim is still the same. Time sharing is a method where each node is allocated the use of the complete channel, by some method, for a certain time. During that time, the node is responsible for transmitting the data in whichever manner it chooses and for relinquishing the use of the channel at the end of its alloted time (in many systems it is also responsible for error detection). In many systems the time sharing is provided by algorithms which are distributed in each node hence an individual node could become the dominant element and effectively control the network to the exclusion of all other nodes. In general this situation does not occur, however it does highlight an important failing of a TDM system which does not arise in the true simultaneous access systems of FDM and CDM. In contrast the receivers within CDM and FDM systems have to be more complex in order that they can differentiate between the required signal component and noise components of the channel's composite signal. Similarly, the transmitters must be capable of generating a range of signal constructs, each of which must also produce low mutual interference with the others - a tradeoff between performance capability and complexity.

1.4 Thesis Overview

This thesis decribes the operation of a new LAN which is based upon the concept of spread spectrum modulation for the provision of an unrestricted simultaneous multiple user system. As such the logical operation, theoretical analysis and computer simulation will be described in the following chapters. It is important to note that the work encompasses a wide range of research areas within which it would have been possible to concentrate on one of several more specialised aspects eg coding theory, protocol design and validation etc. The scope of this work is such that the concepts, properties and system implementation have been analysed in an attempt to provide a basic knowledge on the potential of this new LAN and consequently some of the work considers only one possible solution to a problem. In all instances the choice of solution was made in an attempt to simplify the system analysis whilst still maintaining the unique properties and general tenet of the original ideas.

Chapter two introduces the SS technique and describes how a direct sequence spread spectrum system operates - this method forms the basis for the SS-LAN. It also describes the types of LANs available and highlights their problems which the SS-LAN will solve.

Chapter three is the description of the new LAN and presents its physical structure, its properties and logical modes of operation. It also presents the limitations of the LAN and provides a qualitative comparison with other LANs.

Chapter four presents a review of the code sequences which are typically employed in SS systems. The ideal properties of the codes are defined and a selection of codes are compared with these individually and then with respect to each other. The final section discusses the types of applications for which each node could be used.

Chapter five is the system implementation of the logical description given in chapter three. It is concerned with the overall system construction and with the operation and functions of the individual nodes with the design produced using a "top-down" methodology.

Chapter six is a theoretical analysis of a simple general model of the SS-LAN. This will discuss data throughputs and delays, SNRs and BERs for all the SS-LAN and it will also compare these with results from other LANs.

Chapter seven is a description of the computer simulation developed to investigate the performance of complex SS-LAN configurations which cannot be modelled simply. The design is produced in a form which is both similar to that given in chapter six and which also highlights the accuracy of the simulation method.

Chapter eight is concerned with the comparison between the results obtained from the simulator and those predicted by the theoretical analysis. The discrepancies between the two are highlighted and the corresponding discussion is focussed on an explanation of their cause with appropriate modifications being suggested.

Chapter nine is a general conclusion and this will provide a guide to the development of the SS-LAN. It will also describe the potential applications for the SS-LAN, which will decide the ultimate fate of this new LAN.

CHAPTER 2

Spread Spectrum Communications and Local Area Networks

2.1 Introduction

A fundamental problem in LANs, as in the case of radio systems, is that of multiple access for a variable number of users. Traditionally this is overcome by the use of TDMA or FDMA systems but SSMA is now being applied in several radio systems - primarily those concerned with satellite communications. In principle this technique should also be applicable to cable bourne systems such as LANs, however one potential problem is the bandwidth limited nature of cable systems compared to the almost infinite bandwidth of satellite systems.

The application of SS modulation to a LAN will endow it with the unique properties of SS (these will be discussed later) such as noise immunity and simultaneous access, at a time when the established TDMA implementations have been found to be limited in some important areas of applications eg time critical environments. The intense attempts to standardise LANs coupled with the high funding in areas such as man/machine interfaces and knowledge based systems mean that any new LAN must conform and present itself as a very flexible communications medium for use in a large number of very different applications.

This chapter will present a brief background to both SS communications and LANs, both of which are rapidly expanding areas of technology. It will then discuss the important performance criteria for most communication systems and will concentrate on the problems of the currently accepted LANs. Finally the justification for a new form of LAN will be discussed and this will also highlight the properties necessary to solve the current LAN difficulties.

2.2 An Introduction to Spread Spectrum Communications

SS modulation has existed since the 1940s and consequently its history, development and technology is extremely complex - hence only a "feel" for the technology can be given in this section. The most important aspect is that the new LAN is based upon the direct sequence (DS) method (one of four pure methods employed in SS systems) thus it is this aspect on which the introduction will concentrate.

2.2.1 The History of Spread Spectrum

A comprehensive history of the development of SS since before the second World War is given by Scholtz (7) and this is further developed by Scholtz (8) and Price (9) who have both managed to locate and declassify much of the original work. The general flavour and tone of these three references is perfectly reflected by Bennett (10) and his presentation of the use of a simple SS scheme in a secure telephone implementation for use between Churchill and Roosevelt during WWII. The result is that once SS modulation was developed it was employed as the communications method for secure information and consequently most of the work involved with SS at that time became classified, which in turn resulted in SS being monopolised by military applications. Only the Americans were actively pursuing research into SS as the Europeans had several misgivings about its applicability: the British were worried about multipath effects and it is ironic that later applications in mobile radio systems made use of SS because of its immunity to multipath, Scholtz (8).

The seventies saw the gradual declassification of SS material and this, in turn produced more applications of SS - the majority of which were still American. The space shuttle (11) was provided with a direct sequence SS radio system for use as a backup in case of failure of the primary S-band system (and was consequently used in the first shuttle mission) whilst more recent applications have seen SS used in radio networks (12) and the development of cellular radio with SS modulation for taxis and vehicular telemetry (13). In Britain the Gas Board is investigating the use of SS for remote reading of meters and has installed a 1000 home test system in the south of England whilst some of the off-shore rigs use SS radio systems to transmit sensitive information to shore bases. Current research is investigating the use of SS in satellite systems where SS modulation can be twinned with the present TDM systems, as described by Masamura (14). The use of SS in such a system will reduce intermodulation effects which dominate FDM implementations.

Clearly the field of SS is extremely active with both military and industrial developments taking place. The Americans are still the major executors of SS but European industry has recently become aware of its potential. The logical step in the development of SS was its application to the area of local communications and, in particular, to LANs.

2.2.2 Spread Spectrum Modulation Techniques

There are four basic methods for SS modulation: direct sequence, frequency hopping, time hopping and chirp, all of which are described in a comprehendable fashion by Dixon (15) and Cook and Marsh (16). DS and FH modulation are described in greater detail by Harris (17). In each case the data is transmitted across a bandwidth which is far greater than is necessary, hence the terminology, because the data is "spread" out across the channel bandwidth.

In the case of DS this spectral spreading is achieved by generating a cyclic code sequence which is multiplied with the data signal. At least one cycle of the code is multiplied with each data bit or else the spreading efficiency of the system is reduced. In figure (2.1a) it is assumed that the data and code sequence are in a binary form (this need not be the case as SS modulation can take place before or after other forms of modulation such as FSK,PSK etc) and that T_d is the period of each data bit and T_c is the period of each code sequence bit; the latter is defined as the "chip period". The power spectrum of a square waveform is the sinc waveform where 90% of the power is contained in the central lobe. This waveform is also the power spectrum of DSSS signal, or more specifically, it has an average power distribution of the sinc curve superposed on which is a shaping fuction that is derived from the code sequence and its own modulation of the data signal (18) - this is shown in figure (2.1b). This diagram shows the carrier frequency f_c, if present, the code sequence frequency, R_c, the code sequence length, L, and the displacement between the frequencies in the power spectrum given as R_c/L. It can also be seen that the bandwidth of the SS signal is $2R_c$, or twice the code sequence frequency.

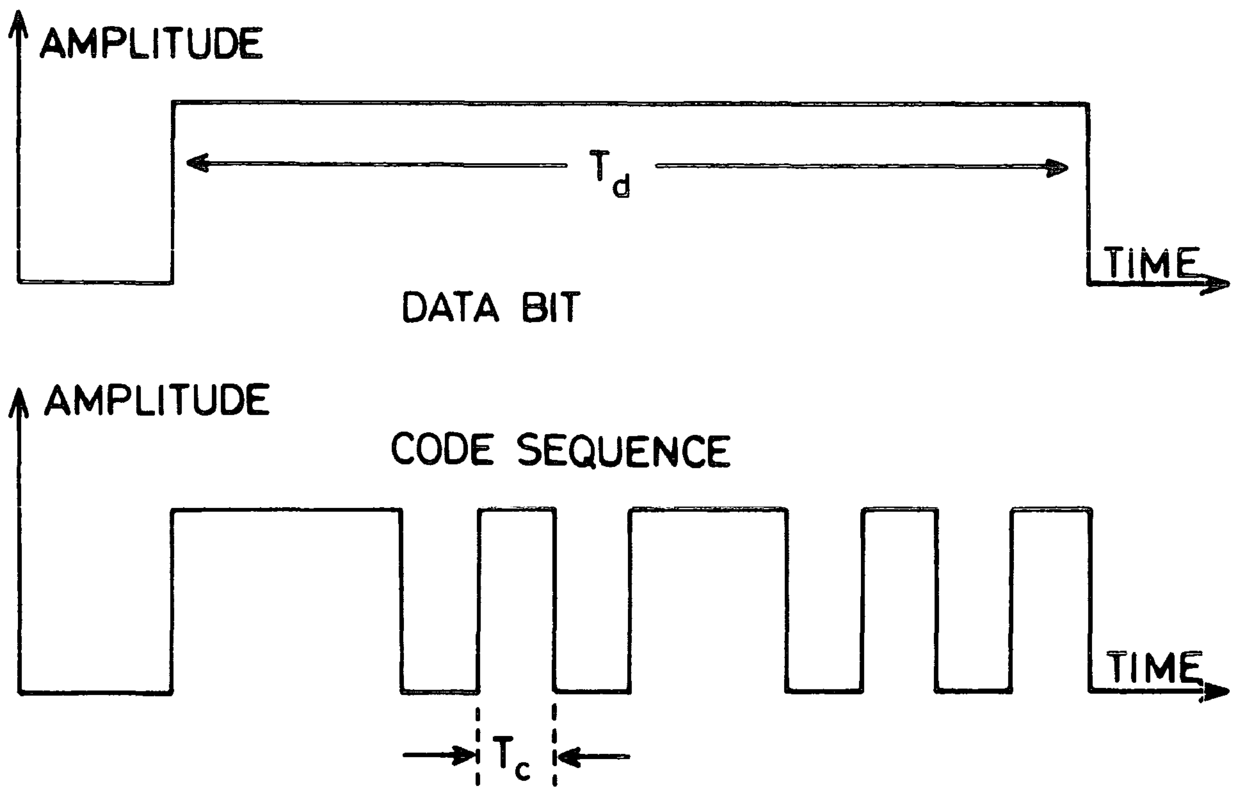


FIGURE (2.1a): Direct Sequence SS Modulation

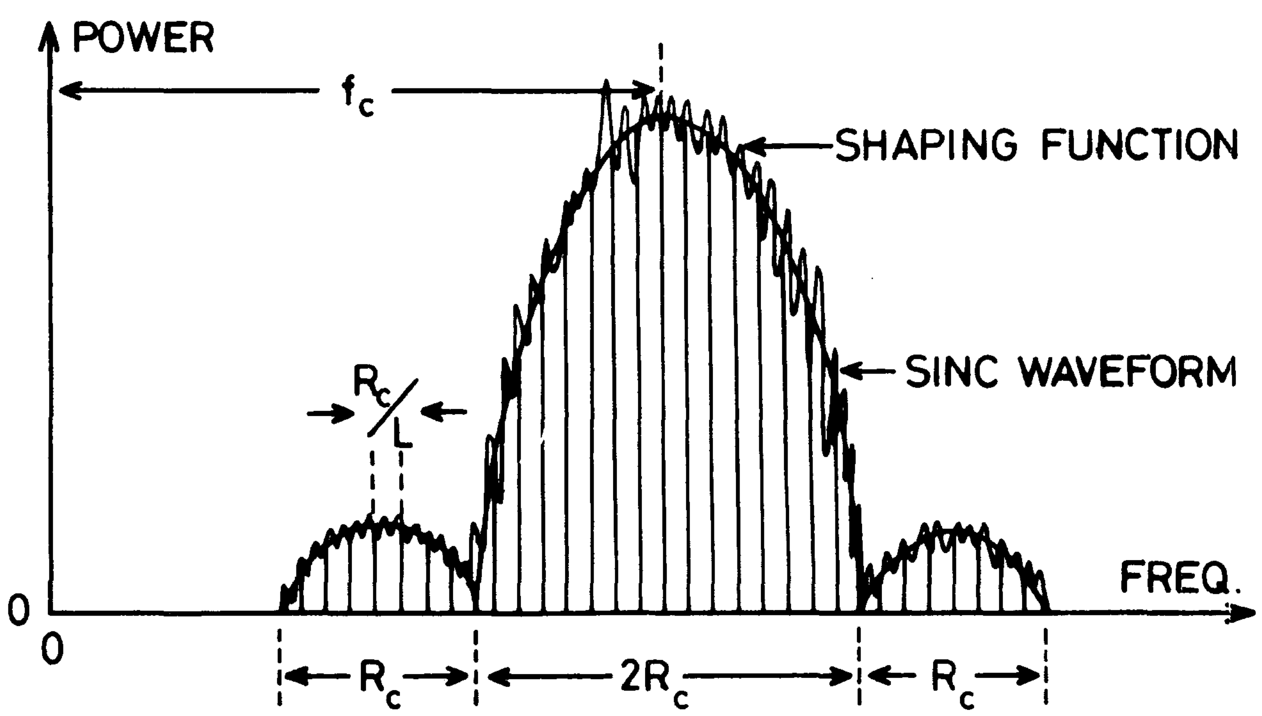


FIGURE (2.1b): A General DSSS Power Spectrum

FHSS modulation is performed by transmitting the data on discrete frequencies. The frequency order is usually generated via a code sequence identical to that used in the DS method. The number of frequencies on which the data bit is transmitted is dependent on the length of the code sequence whilst the bandwidth across which the data is spread is dependent on the frequency range allocated to the hop sequence . Figure (2.2a) shows a time map of the transmitted frequencies with the accompanying power spectrum shown in figure (2.2b). FH is similar to FSK except that several frequencies are used instead of just two. When transmitting the data for a zero it can be marked either by a frequency absence or by performing FSK modulation on the hopped frequency. Consequently at any one moment the data exists on one particular frequency and no other whereas in DS the signal exists on several frequencies at the same time: depending on the duration of the analysis. An extension of this principle produces "slow" and "fast" FH systems (16). Slow hoppers utilise one carrier frequency for several data bits whereas fast hoppers utilise several carrier frequences for one data bit (the latter is shown in figure (2.2a)). The advantage of the fast hopper is that it is less prone to jamming however its receiver is more complex than that for the slow hopper.

The third method is called time hopping (TH) whereby the data is transmitted in a time varied fashion. The generation of the order of transmission of time slots is usually provided by a code sequence (cf FH) and the frequency of transmission is fixed by the system. In this method the data is only present for a fraction of the duty cycle. This technique is rarely employed by itself but can be used in hybrid SS systems and can be considered as simple pulse modulation.

The final method is called pulsed FM or Chirp modulation. Again this technique is rarely used and most applications are in radar systems. A significant difference in chirp compared to the other techniques is that it does not use a code sequence of any form in either the transmitter or receiver. In fact, the transmitter is based on a common sweep generator with the receiver comprising of an appropriate matched filter. In most communication applications a multiple user environment is required and this is difficult to provide using chirp modulation - the reason for its use in radar systems is the low peak

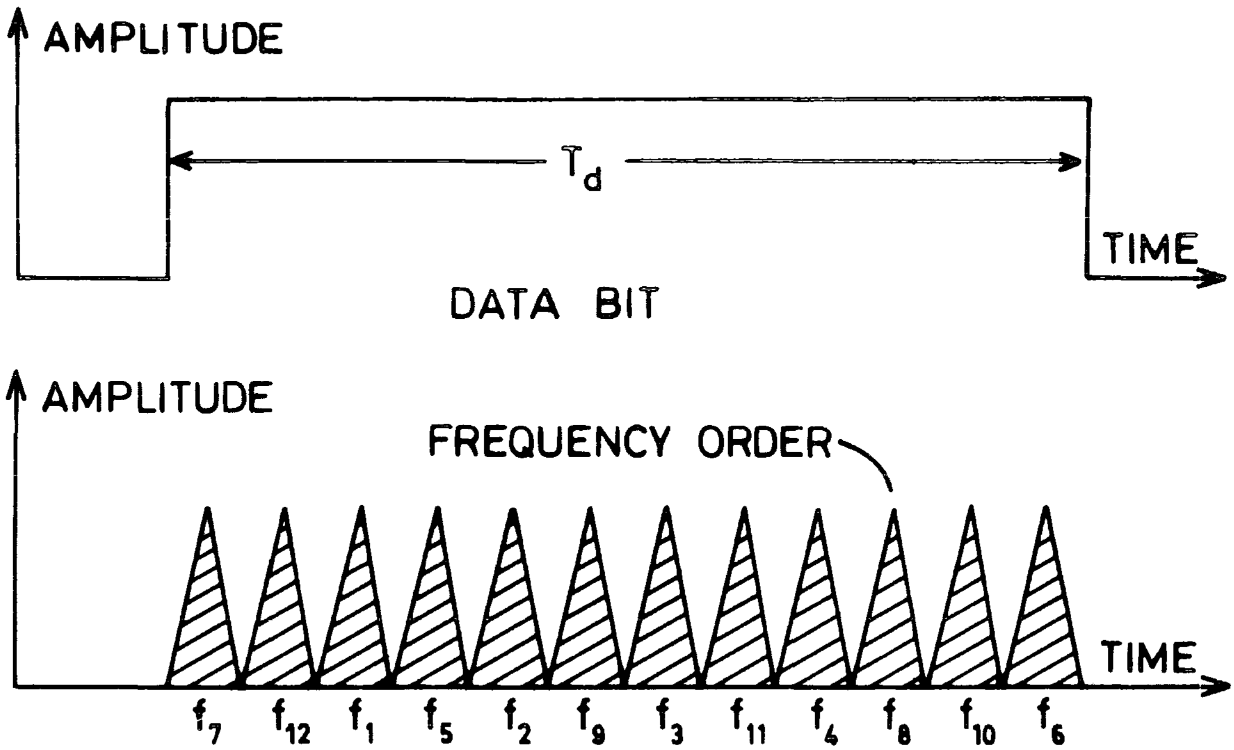


FIGURE (2.2a): Frequency Hopping Modulation

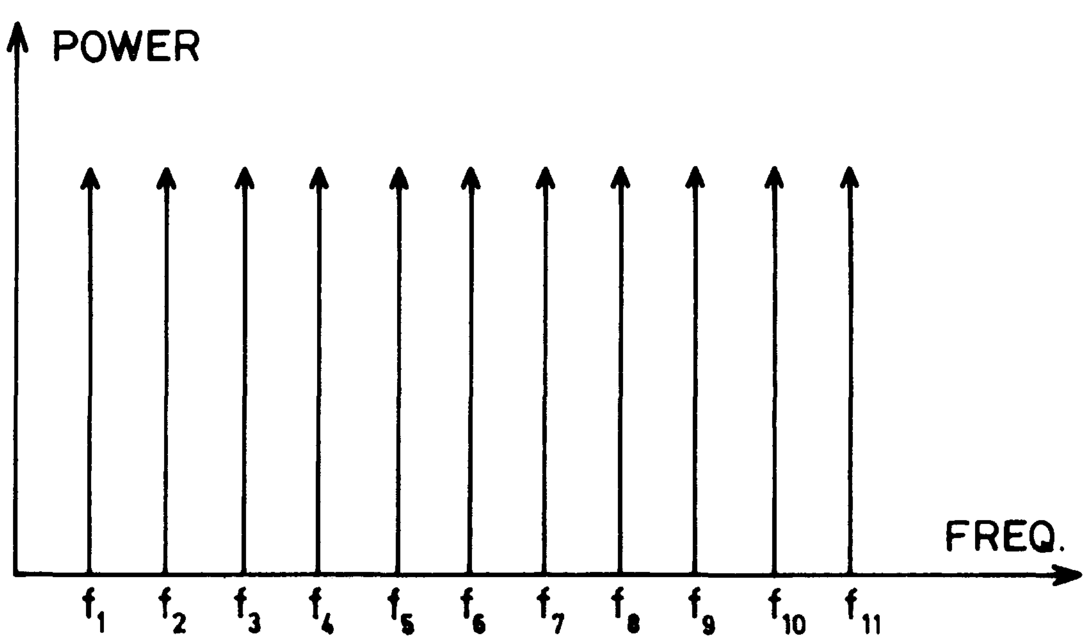


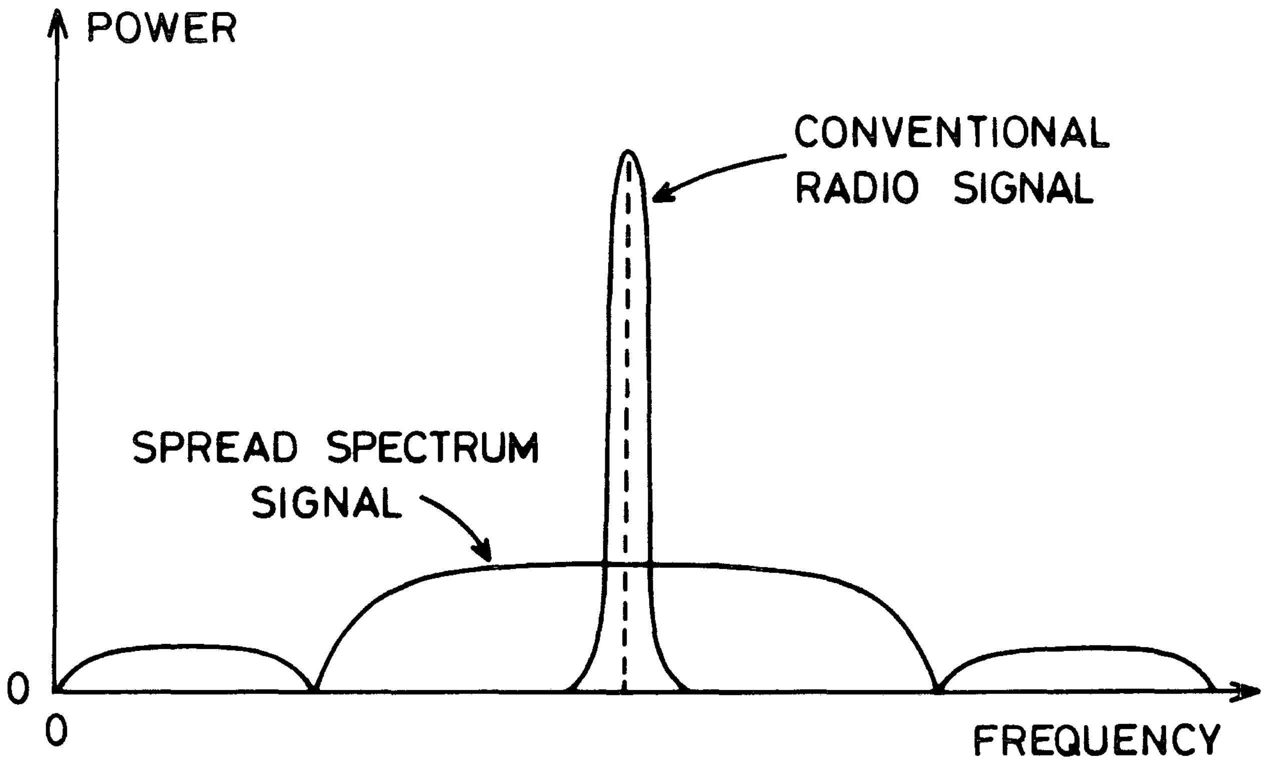
FIGURE (2.2b): Power Spectrum for FHSS Modulation power required when compared with other more conventional radars.

These four methods form the basic SS modulation techniques but it is possible to form hybrid systems based on two or more of these basic methods. Examples of such systems are FH/DS modulation, TH/FH and TH/DS but these will not be discussed as the SS-LAN is based solely, in its present form, on DSSS modulation. Each of the SS techniques provides the communication system with some interesting properties which, although they can be reproduced in other systems by the addition of complex hardware, are an intrinsic part of the modulation and it is these which will be discussed next.

2.2.3 The Properties of SS Systems

Figure (2.3) shows the power spectra for a conventional (narrowband) signal and an SS signal (wideband). The energy used in both techniques is usually the same the only difference being the distribution of the energy over the channel bandwidth. Many of the properties of a SS system can be appreciated from this diagram. The first is the "white noise" shape of the SS signal ie a constant signal energy for all frequencies, which gives rise to the second property; that of inherent noise immunity. Noise is in general restricted to small bands within a channel (dependent on the electrical environment) and should this band coincide with the frequency of transmission then for a conventional system the signal is lost, however the effect is minimal on the SS signal, as can be seen from the diagram. In fact in a SS system the signal can still be recovered when the signal to noise ratio is less than one.

The low energy density of the SS signal produces a situation where the SNR can be lower than unity at the front end of a receiver. This means that the signal cannot be separated from the noise ie there is a low probability of detection by conventional radio systems. This strange SNR situation is resolved in the SS receiver by integrating the distributed energy giving rise to the definition of the "Processing Gain (PG)" of the system. The PG will be defined later but a qualitative appreciation can be gained by considering it as the factor by which the SNR is improved via the process of energy integration - thus the larger



FREQUENCY (2.3): Comparison of Radio and SS Power Spectra the PG then the greater the output SNR. This wideband energy distribution also makes it more difficult to jam the signal ie SS modulation provides an antijam capability. Whilst it is not impossible to "jam" a SS signal (the method depending on the form of SS modulation in use) it usually requires detailed knowledge of the type of SS transmitter and expends a large amount of energy which can disrupt the jammer's own communications environment.

The final two properties which will be described are that of secure and/or private communications and the provision of a simultaneous multiple user environment. The first results from the fact that both the transmitter and receiver must be synchronised in a predefined cyclic manner which to all intents and purposes can be considered to be cyclically random! A casual "listener" would not be capable of decoding any transmitted information - a private communications link - and the use of standard encryption techniques and codes can be used to make the link secure ie immune to intentional non-authorised users. Another important property of a SS system is that of simultaneous multiple access which is derived from the sharing of the signal dimensionality as described in the previous chapter. This means that within the limits of the system specification any individual user can transmit information independent of the current state of the communications channel: an ability not possible in both TDMA and CSMA based systems. It is this final property on which the new LAN is based. In conclusion all SS systems (independent of their type) possess anti-jam, anti-interference, low probability of detection, secure/private links and a multiple user environment; a good review of these is provided by Pickholtz, Schilling and Milstein (19,20).

2.2.4 The Operation of a Direct Sequence SS System

A schematic block diagram of a typical radio communications DSSS system is shown in figure (2.4), as derived from references (19,21), and is assumed for the moment to be a single user system. Consider firstly the construction and transmission of the SS data signal. The raw data is presented in a digital fashion (the data source) and is multiplied by the binary code sequence generated in the transmitter. Binary multiplication is equivalent to a modulo two addition of the two binary

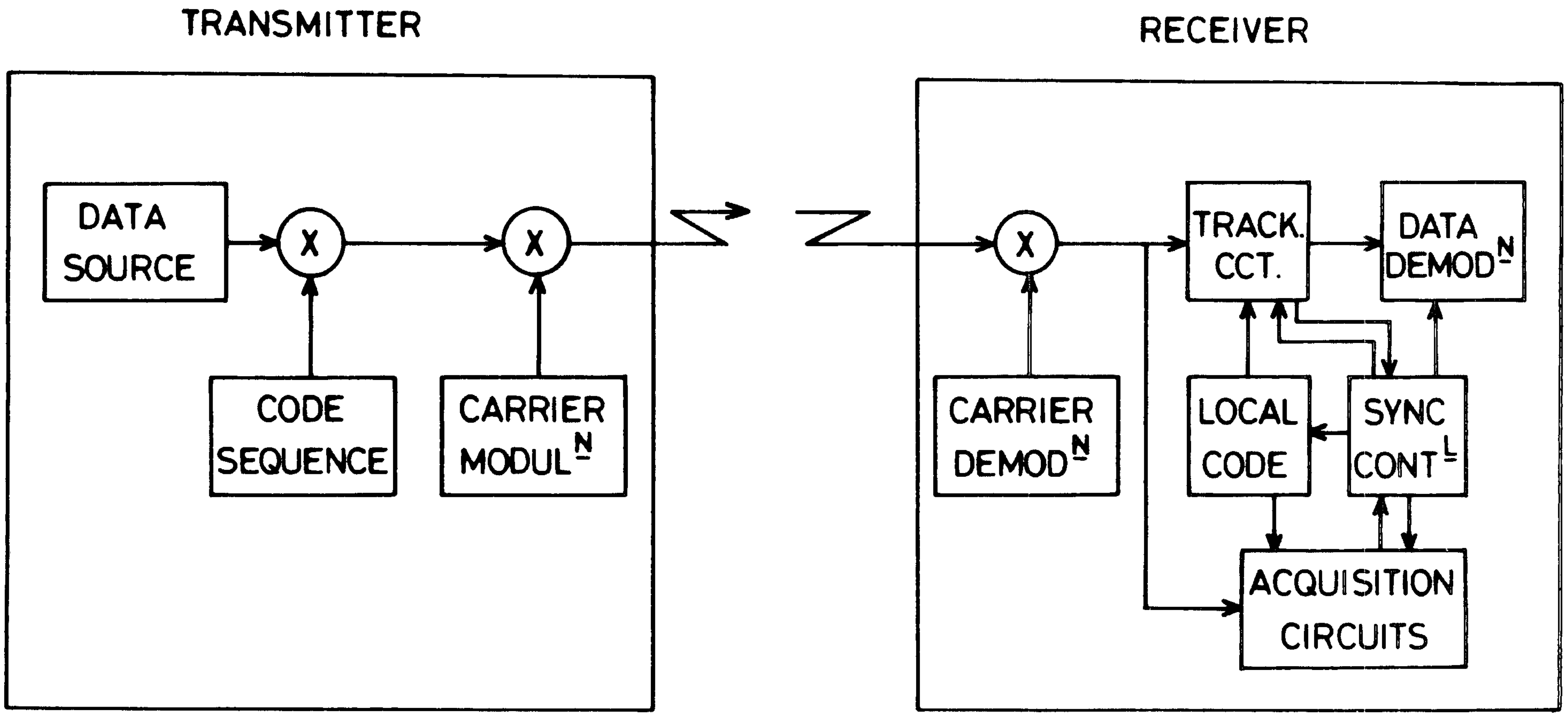


FIGURE (2.4): A General DS SS Radio System streams ie the exclusive OR operation. The timing between the data source and the SS code is such that a complete code sequence is modulated with a single data bit, (it is this ratio which will be seen later to effect the PG for a DSSS system) however this can be changed from system to system. The resulting binary stream is then used to modulate a carrier frequency, via the most appropriate broadband technique eg PSK,FSK etc, which is in turn transmitted via the communications channel.

At the receiver the signal undergoes carrier demodulation and the signal is then presented in a binary form to the tracking and acquisition circuits. The acquisition circuit performs a coarse synchronisation between the incoming modulated SS signal and an offset, locally generated code, identical to that used in the transmitter. This synchronisation is necessary due to the physical displacement of the transmitter and receiver, resulting in chip displacement of the incoming signal and the locally generated code. The form of acquisition in this system is called a sliding correlation technique whereby the two signals slide past each other, in a controlled fashion, until the correlation between the two signals exceeds a predefined threshold whereupon coarse synchronisation is achieved ie the transmitter and receiver are in chip alignment.

Once acquistion is achieved the synchronisation control instigates tracking, or fine synchronisation, of the signal. Here the correlation between the incoming signal and local code is constantly monitored so that it is always above the threshold. This includes a forward and backward correlation to check for single bit slips which can be corrected by chip displacing the incoming signal with respect to the local code. Tracking is necessary in many systems because there may be relative motion between the transmitter and receiver or more importantly the clocking frequencies of the transmitter and receiver will not be exactly stable and will drift relative to each other. If synchronisation is lost then coarse synchronistaion must be repeated to regain the lost signal. Assuming that fine synchronisation is achieved and maintained then the data can be demodulated via the modulo two addition of the local code and the incoming signal performed within the tracking circuit.

The performance of the DSSS system relies upon the quality and reliability of the acquisition and tracking circuits or more precisely the ability to perform fast digital correlation. The modulation technique relies upon the fact that ths double modulo two addition of synchronised code sequence and the SS signal will return the originally modulated data. This requirement will be formally specifed in the next section.

2.2.5 An Analysis of a DSSS Communications System

A simple analysis of a DSSS system is given in appendix one where the system is assumed to be a multiple user baseband configuration with the nodes distributed around the communications channel and with the environmental noise represented as additive white Gaussian noise in nature. It can be seen that if correlation is possible and synchronisation is achieved then the properties of the code sequences dominate the performance requirement of the system. The value $C_1^2(t)$ as given by equation (A1.5) is more commonly defined as the autocorrelation of the signal and is usually represented in the digital form, where L is the code sequence length, as given by Lynn (23),

$$C_{xx}(l) = \frac{1}{L} \sum_{m=0}^{L-1} X_m \cdot X_{m+l} \qquad (2.1)$$

where it is assumed the period of the ACF is finite ie L. In the DSSS this must be as close as possible to unity because it defines the clarity with which the original data signal is reproduced and represents the total signal energy when in synchronisation.

The second important feature in the analysis concerns the products defined as $C_1(t)C_i(t-\tau_i)$ given in equation (A1.5). These are defined as the cross correlation coefficients and are usually represented by (again as given by Lynn (23) in the digital format),

$$C_{xy}(l) = \frac{1}{L}\sum_{m=0}^{L-1} X_m.Y_{m+l} \qquad (2.2)$$

and it is assumed that these tend to zero in the analysis. The CCF is a measure of the interference, or noise, produced by simultaneous users and must be kept to a minimum or else it degrades the SNR at the detector thereby limiting the number of simultaneous users. Both the ACF and CCF have a larger impact on the system than has been detailed so far. The problem of synchronisation between the incoming signal and the local code was described briefly but it is in fact the most difficult facet of the receiver.

The ACF will vary as the sliding correlation is performed hence the codes must provide a clear distinction between synchronisation and non chip alignment. It is the difference between thses two levels which specify the synchronisation capability. The CCF then amends this synchronisation capability by introducing other user noise which decreases the gap between the two ACF levels, until a point is reached where the system cannot perform correlation ie acquisition is not possible.

The final consideration in the DSSS system is the definition and calculation of the processing gain. In the baseband system the bandwidth of the SS signal is defined as the code sequence rate, R_c, as shown in figure (2.1b). The general definition for PG at baseband is given as the ratio of the spread bandwidth to the original data bandwidth,

$$PG = \frac{B_{ss}}{B_d} = \frac{R_c}{R_d} \qquad (2.3)$$

and this can be deduced from the analysis in appendix one where, for equation (A1.6), the power of the signal in the data bandwidth, before demodulation, is $1/R_cT_d$, when compared with that after demodulation, which is the same as equation (2.3). A more usual definition for the PG in DS systems is to relate it to the code sequence length, L, and the number of code sequences, m, modulated with each data bit. This gives rise to the definition,

$$PG = mL \tag{2.4}$$

because $R_c = mLR_d$, which when substituted into equation (2.3) produces equation (2.4). The PG is a measure of the receiver's ability to resolve the data which has been transmitted to it. The larger the PG then the smaller the chip power necessary, at tranmission, for the receiver to successfully recover the original data. The PG is a measure for individual transmitter/receiver links and is independant of both the number of users present and the efficiency of the data transmissions but it is inversely proportional to the point to point data rate (the larger the PG the lower the point to point data rate for a given chip rate) and a later chapter will present the equivalent PG defintion for the network wide situation.

2.2.6 The Problems in a DSSS System

Most of the problems in a DSSS system are related to the code sequences and the digital correlation performed for tracking and acquisition. The code sequences must each have high ACF coefficients (tending to one) and low CCFs but must still be simple to generate in a programmable fashion and must exist in large families: essential for a large multiuser system. The stability of the transmitters and receivers is also important if data is to be decorrelated or else the acquistion of the signal will not occur: the two clocks must be stable for at least one whole cycle of the code sequence or else a bit in the code sequence will either be duplicated or lost at the receiver.

An added complication in the baseband system is that even though the code sequence transmitted is in a binary form the signal which is received is analogue in structure due to the simultaneous multiple users. The signal will in fact vary between the values of all nodes sending logic one and all nodes sending logic zero. This means that the receiver must perform an ADC before the digital correlation, in the form of a sliding correlator, can be attempted.

2.3 An Introduction to LANs

Since their introduction as a communications tool some 15 years ago the number of different types of LANs has grown profusely ranging from low data rate polled networks, such as the ASWE Serial Highway (24), to high speed fibre optic LANs, such as the new Cambridge ring under development by Acorn, which will operate in excess of 100Mbits/sec. The field of LANs is so vast that it is impossible to cover it in any depth at this time, however two excellent references on LANs covering not only the technology but protocols and standards are by Flint (25) and Lee (26). Flint (25), in particular, summarises the guidelines laid down by the IEEE in specifying what a LAN must provide in the form of physical and link characteristics, errors, failures and maintenance. The heart of the guideline is that a LAN should provide a highly reliable communications medium which can support upto 200 varied devices spread over a distance of not greater than two kilometres with a transmission rate of at least 1 Mbit/sec and with enough flexibility for the network to be totally reconfigured and interfaced to any other form of telecommunication system: a tall order and one which is not yet provided by any available LAN.

2.3.1 LAN Topologies

There are four basic LAN topologies as shown in figures (2.5a), (2.5b), (2.5c) and (2.5d): the ring, the bus or linear, the star and the mesh. There are subdivisions of each of these basic four and it should be noted that there are several other methods by which LANs can be categorised. There are also a wide range of networks which are formed from a hybrid of these basic four but all LANs have one basic characteristic and that is they enable many users to freely communicate between each other. It should also be noted that the topology need not give a guide as to how the LAN operates- for example the token bus which can operate similarly to a ring. Several other different topologies also exist such as the tree, daisy chain (active bus) and snowflake but these will not be covered in this summary.

Within the category of ring topologies there are token rings (as being designed by IBM (27)), slotted rings such as the Cambridge Ring

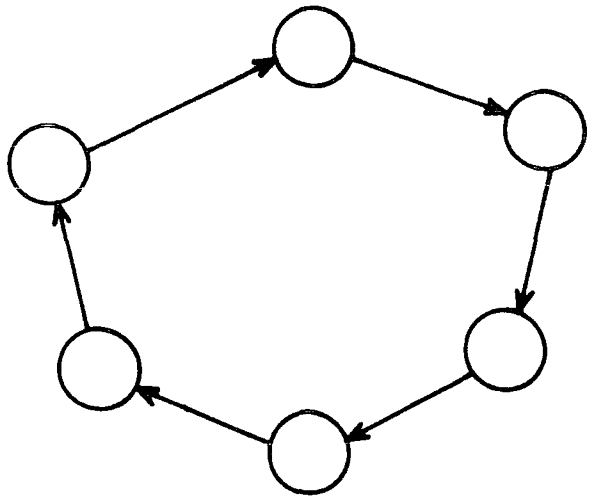


FIGURE (2.5a): A Ring Topology LAN

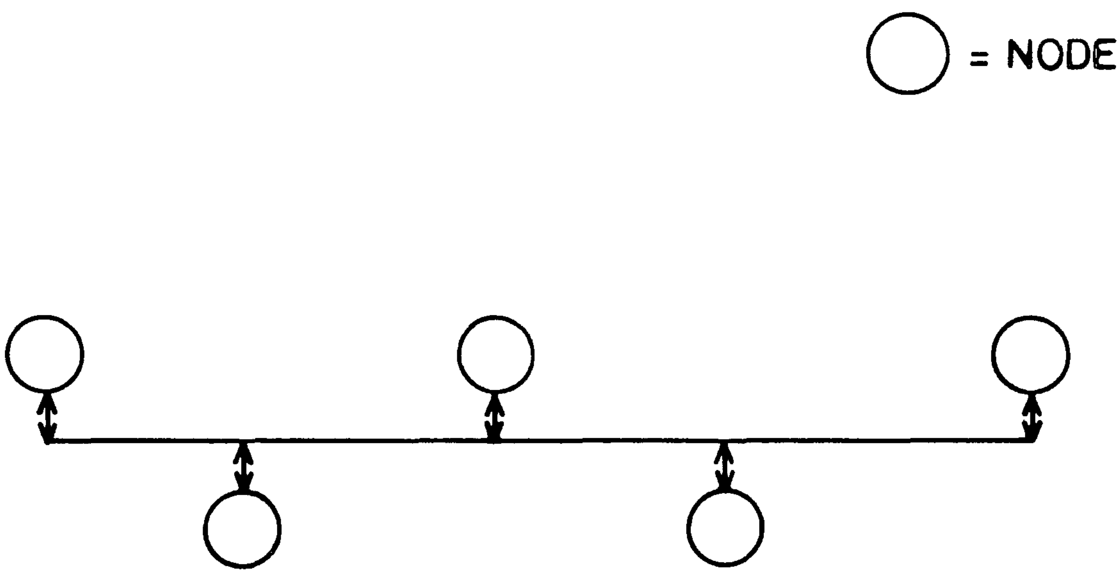


FIGURE (2.5b): A Passive Bus or Linear Topology LAN

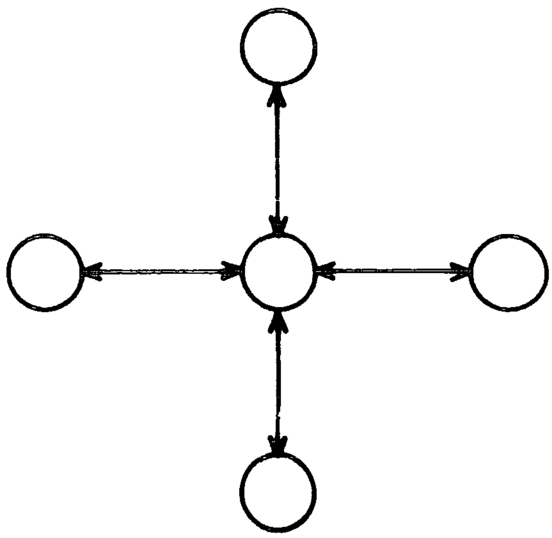


FIGURE (2.5c): A Star Topology LAN

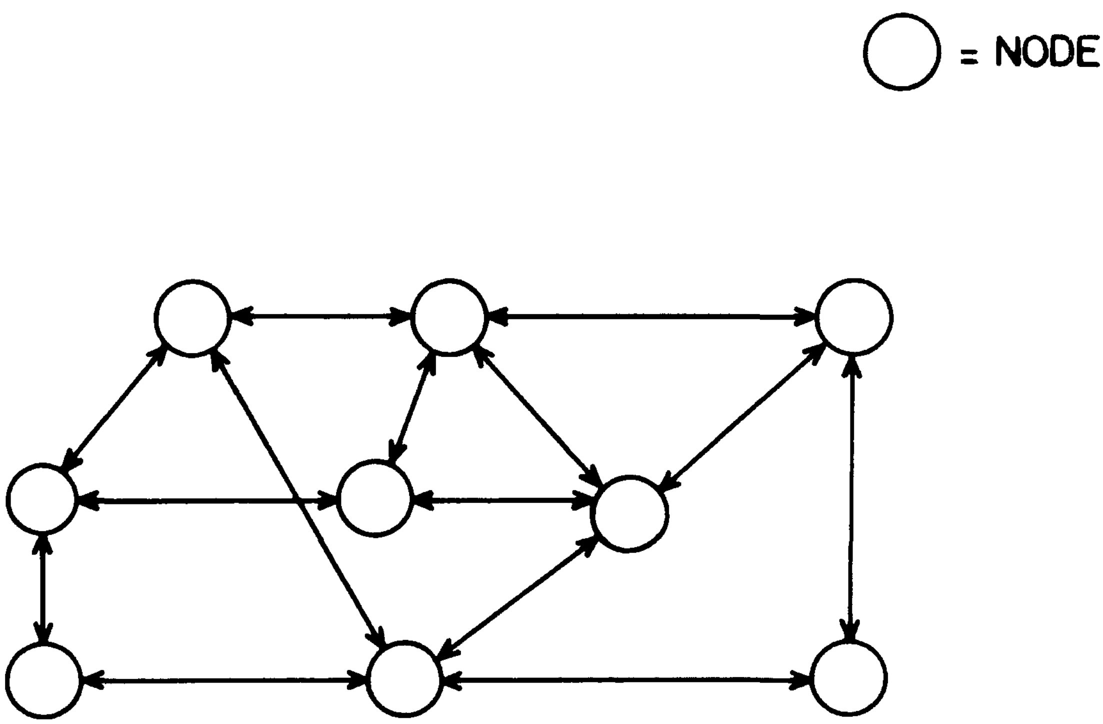


FIGURE (2.5d): A Mesh Topology LAN

(28) and register insertion rings such as SILK (29). In the case of the token ring a token is circulated around the ring and when a node holds the token it can transmit data. Once complete it sends the token onto the next node. The circulation of the token enables a form of variable data loading to take place but there is a high protocol overhead required to ensure ring recovery should the token become lost or corrupted. In the case of slotted rings the ring is split into a series of virtual slots or minipackets, with one empty slot. When a node wishes to transmit data it reserves the slot and sends the data. Various algorithms are implemented to ensure slot sharing and error recovery but its major advantage is a guaranteed maximum delivery time, essential in real time applications. The final example is the register insertion ring where a data packet is inserted into the ring by altering the ring path thus varying the size of the ring: in the case of the slotted ring the ring remains rigid. The advantage of the register insertion is that the slots may be used more than once per cycle of the ring thus the data rate is in excess of the ring transmission frequency making it ideal for voice-data systems where the loading is heavily distorted due to the two types of data.

The second category of bus or linear topologies includes LANs such as the Ethernet system (30) (a passive bus), active bus structures and token bus structures. In the passive bus architectures the nodes are linked to the communications channel via a spur such that a signal propogates to all nodes once it has been transmitted without relying upon relay action by nodes or repeaters i.e. a passive channel. The advantage of this compared to an active bus is that should a failure occur at a node then the LAN operates normally, however in the active case the LAN would become split. The Ethernet systems employ CSMA/CD where each node is permitted to transmit data at any time. For small systems with moderately long messages the probability of contention is small however as the size of the LAN grows it must be segmented and linked via a series of bridges in order to reduce the probability of contention. When a contention occurs i.e. two nodes attempt to transmit simultaneously, then the collision detection mechanism forces both conflicting nodes to "back-off" and implement a time delay and retry algorithm. The active bus LANs are very similar to the slotted ring systems where the signal is passed from one node to another and where all the links are point to point connections. Similarly the operation of the token bus and token ring are closely related. The token is passed from one node to another and when a node possesses the token it is permitted to transmit data.

The star networks and mesh networks are rarely implemented in present LAN systems. The star network is centred upon a controlling station around which are connected the nodes: each node has its own seperate channel to the controller. Whilst this is wasteful of cabling it enables the controller to act as the system arbitrator over requests to the same node and as such the system can be configured to operate as an Ethernet type system as shown by Schmidt et al in their Fibrenet II system (31). This last example demonstrates how it is possible to consider the star topology as an extravagent bus topology. The mesh network can be considered as the anarchic topology because the protocols for communication frequently involve the flooding of the network with a message in order for it to reach its destination. The high connectivity of such a LAN provides it with high fault tolerence because if there is a node failure then there is still another route to the destination node. In fact such a LAN can still operate when a comparatively large number of its nodes have failed.

2.3.2 Current LAN Systems

There is no doubt that the major LAN in the world today is the Ethernet system, whether it is in its natural state or with collision deterrent enhancement algorithms and is currently licensed by the Xerox Corporation. Within Britain the Cambridge Ring has firmly established itself and is marketed by companies such as Acorn and Logica VTS. Strangely, it is not noted with any interest in the USA or the rest of Europe but is actively exploited here as Project Universe (32) has demonstrated: this was the interconnection of several Cambridge Rings distributed across the country and linked via a satellite. The major funding of IBM in their token ring means that within the next few years this will undoubtedly become a strong rival to Ethernet based not only on the forceful sales and marketing it will receive but also because it will be provided with a series of interfaces to all standard IBM equipment. The LANs available today are predominantly either TDMA, CSMA or token access based systems with FDMA, CDMA and SSMA generally disregarded, indicating the extent to which time multiplexing has dominated LAN development.

The development of LANs is now based on two areas: the use of fibre optics to replace co-axial cabling and the implementation of the node hardware and software in VLSI circuits. Two tutorials on the effects of fibre optics in LANs given by Hensel (33) and Finley (34) show how most LANs must eventually become fibre based, if only for the sake of noise immunity and higher data rates. Fibre optic based LANs already exist in the shape of Fibrenet II (31) and Gamma-net (35) where the latter is used to provide a high speed link between local computers. Fibrenet II is an example of how present multiple access schemes will be manipulated to conform to the limits of fibre optics to provide the benefits without changing the modulation scheme.

Similarly there are already some VLSI chip sets for LANs with the LANCE chip set reducing the cost of the Ethernet nodes. The two major factors in implementing devices in VLSI chips are that the individual node costs and the physical size are greatly reduced. Current technology limits the operating speed of VLSI to approximately 20MHz however this will increase as it moves towards smaller circuit manufacture control. Even if this limit were not to be improved VLSI could still be implemented for 100MHz systems by using a specialised fast front end based on ECL which would package the bits and pass them to the chip system at a lower data rate; the fibre optic Cambridge Ring being designed at Acorn is based on this very principle.

At the present time there is no fully certified implementation of the seven layer OSI model thus each LAN operates on its own interpretation of what is required as a protocol. Naturally this has led to difficulties in internetworking of LANs which although it is contrary to their concept is nontheless acquiring great importance. Even the work of the IEEE 802 committee has not succeeded in producing a definitive standard as this has been forced to compromise between the virtual link and datagram concepts which favour the token passing ring and Ethernet type systems respectively. However attempts at standardisation are being made and this coupled with the use of optical fibres and VLSI techniques are the trends for development of LANs in the foreseeable future.

2.3.3 Wideband LANs

Whilst it is true that most LANs employ some form of time multiplexing there is now research involved in considering wideband techniques as a means of making more efficient use of the bandwidth. The first system of interest is based upon the early work by Cohen et al (36) who designed a TH system where a pseudorandom number was allocated to each transmitter and this provided a time slot in which each node transmitted data. The result was that only a two level signal was transmitted but the transmissions were arranged to be sequential across the complete network.

This work was recently expanded by Davies and Shaar (37) who implemented a similar scheme on a fibre optic based LAN. The intention was to distribute the, otherwise unused, bandwidth across the LAN nodes and so produce a data averaging of the load across the network, consequently enabling standard 1Mbit/sec data rate nodes to efficiently interface to a 300 Mbits/sec channel capacity.

The definition of wideband must be considered in relation to its sister terms of baseband and broadband. Baseband is where data is transmitted at its natural rate without the use of a carrier or associated modulation techniques. Broadband is where a carrier is employed to move the frequency of transmission into a particular bandwidth but which also requires some modulation of the data onto the carrier eg PSK or FSK etc. This uses a particular bandwidth and it is not unusual for several of these to be used to transmit the necessary data, as for example in television. Wideband transmission is where the bandwidth used to transmit the data is greater than necessary and as such is identical to the defintion of SS modulation though the latter usually requires a PG of at least 100.

The most recent use of wideband modulation, apart from the work in this thesis, is concerned with a mixed voice-data LAN which employs a DS code for each node's transmission. The problem with mixed voice/data systems is due to the very different data rates of the two types and the fact that voice data usually requires data buffering and delivery within specified time limits. The Elhakeem et al system (38) uses a downlink and uplink system connected to a controller which transfers data from one to the other and also provides synchronisation pulses. In concept it is similar to that of the SS-LAN however its performance capabilities are restricted due to its controller and separate links. An extension of this work has been concerned with the application of SS to cable television networks, Hafez et al (39), for the transmission of information from television sets. In conclusion, the current work on the SS-LAN is unique, as regards the open literature, in the world at the present time.

2.3.4 Present Day LAN Performance Characteristics

The performance characteristics of LANs in use at the present time are, in general, similar for all topologies with only slight variations due to this difference. For example, the data transmission rate is 10Mbits/sec for Ethernet and Cambridge Ring with the token ring prototypes operating at 4Mbits/sec. The actual point to point data rate for the Cambridge Ring is set at 1Mbits/sec due to the slot sharing mechanism and the control overheads (where only 16 bits in every 40 are data). Similarly, whilst the Ethernet is faster the access time must also be taken into account and this lowers the average data rate: it is not relevant to calculate this accurately now but it will be analysed in a later chapter. This access delay highlights the basic difference between Ethernet and Cambridge Ring which is that the Ring provides a guaranteed maximum delivery time for a particular system configuration whereas Ethernet depends on a probabilistic method which could be considerably faster or slower depending on the loading at any moment in time.

The number of nodes which can be supported by a single LAN is limited, in practice, to 30 in both the Ethernet and the Cambridge Ring: in Ethernet this gives each node only a 3% section of the total LAN performance and in the Ring this is derived from the maximum of seven slots which must be shared between all the nodes. Many advertised systems such as Wangnet (40) state an interconnection capability of 2^{16} nodes but which are in fact simply a collection of bridged LANs where the physical address field is 16 bits wide. A few minutes reflection on some of the interconnection statistics offered for Wangnet would give an average data rate in the region of a few bits/sec if it was compared with Ethernet statistics.

The BERs for LANs are traditionally low and of the order of 10^{-9} because the noise environment is good. This in turn reduces the necessity for sophisticated error corection codes and so many systems use a CRC to verify the state of an entire message. The detection of a fault then entails the retransmission of the entire message which means that the data must be stored internally, at some level, until a successful transmission acknowledgement has been received. Finally most LAN nodes are provided with a series of specialised interface boards which permit the station node logic to be driven by different types of host user computers eg IBM, DEC and CDC machines where the word sizes vary considerably. This reduces the work for a potential user and aids network compatibility to its devices.

2.4 Performance Criteria for Communication Systems

Over the years certain criteria have established themselves as the performance parameters which describe the operation of a communications system. These have been adopted for LANs and they will be calculated for the new LAN in a later chapter. These criteria are the number of multiple users which can be supported by the multiple access scheme whilst providing specified values for the other parameters, the BERs for the network wide and point to point instances, the SNRs and the data throughput and delay times for the network wide and point to point cases. The following sections describe how each of the above parameters is calculated and what environmental and system factors affect its values.

2.4.1 Data Throughput and Data Delays

Each system possesses many data throughput and delay values depending on whereabouts the reference points are made eg at the physical layer or applications layer etc, where there will be a large difference in the values. In the SS-LAN the reference point is the time between the request for transmission, made to the station logic and the destination node's passing of the received data from the station logic. The data delay can therefore be modelled by the time map shown in figure (2.6). The transmit overhead is the delay from request reception to attempting to access the communication channel, the access delay is the time which the transmitter must wait before being allowed this access to the channel, the propagation delay is the time it takes the signal to physically move from the transmitter to the receiver; the message propagation delay is the time it takes the entire message to arrive demodulated at the receiver and the receive overhead is the delay required to finish the receive housekeeping instructions. The sum of these values gives the data delay time between two communicating nodes at the physical layer of the OSI reference model. This value is representative of a general LAN and its analysis for the SS-LAN will be given in a later chapter, where the dependencies of the individual sections will be defined and calculated.

The point to point data throughput is now the reciprocal of the data delay multiplied by the number of data bits transferred during that delay. In many of the LANs there are two data rates, the point to point and the network wide throughputs which arise due to the multiple access scheme employed - as in the case of the Cambridge Ring. The network wide throughput is equal to the sum of the point to point throughputs which take place in one second. The point to point and the network wide throughputs compensate for systems which perform data load averaging with the network wide case being the data capacity of the complete LAN.

2.4.2 Signal to Noise Ratios (SNRs)

The SNR is the ratio of the signal power to the noise power and it is the basis from which the probability of error is calculated. The SNR is a physical value which can be calculated at different sections of the system eg at the transmitter or at the receiver input but values can also be extrapolated to produce SNRs at the output of the receiver where the noise and signal are not separable due to the signal processing within the receiver. Various factors affect the SNR not only in the electronics but along the channel: the channel will degrade the signal depending on the frequencies involved, the channel quality and the distance traversed by the components. The receiver input SNR can be used to calculate the required transmitter SNR and hence the transmitter voltage levels for the signalling method.

In most LANs the natural noise level is low and can be modelled using AWGN. In radio systems the noise environment is usually considered hostile because the atmosphere produces variable noise conditions. The AWGN can only be applied if the electrical environment is in general constant and includes no dominant high powered frequencies. This means that if the power is the same for all the frequency components of the power spectrum and if all the possible frequencies are present then it is defined as white noise. It is usual to specify the power as the two sided power density per hertz with the total noise dependant on the bandwidth concerned. This can also be derived from the normalised Gaussian distribution curve where the variance is equal to the PSD as defined earlier. The SNR is now used to calculate the probability of error which in turn defines the bit error rates for the network and it is this BER which will now be discussed.

2.4.3 Bit Error Rates (BERs)

The BER is the number of errors received per second in the point to point data stream and it is dependent on the probability of error and the data frequency. Consequently for a low BER the probability of error must be low (of the order of 10^{-9}) because the data frequency will be typically 1Mbit/sec according to the IEEE definition. The BER is the product of this frequency and the probability of error,P_e. P_e is related to the SNR and is a measure of the performance of the receiver hardware ie its ability to pick out a signal correctly from the noise. The relationship of P_e to SNR is therefore dependent on the modulation/demodulation scheme employed however the function is always linked to the cumlative Gaussian distribution function Φ, which has a mean, μ , of zero and a variance, σ^2, of one.

Figure (2.7) shows the density distribution of a curve which has a mean μ and variance σ^2. This is related to the $\Phi()$ and Q() curves as shown with the sum of $\Phi()$ and Q() being unity, Feller (41). If the value for analysis is, a, then the probability of the received value being

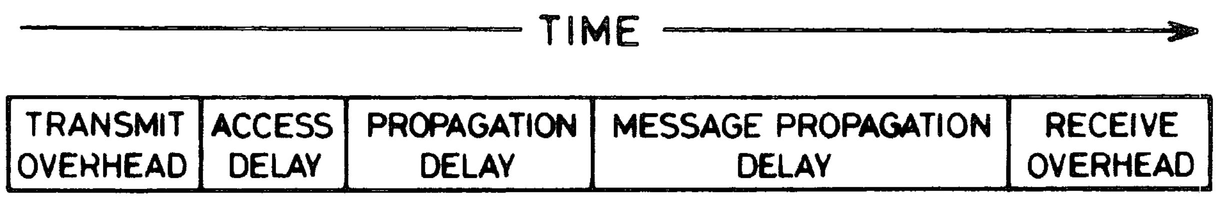


FIGURE (2.6): Components of the Data Delay

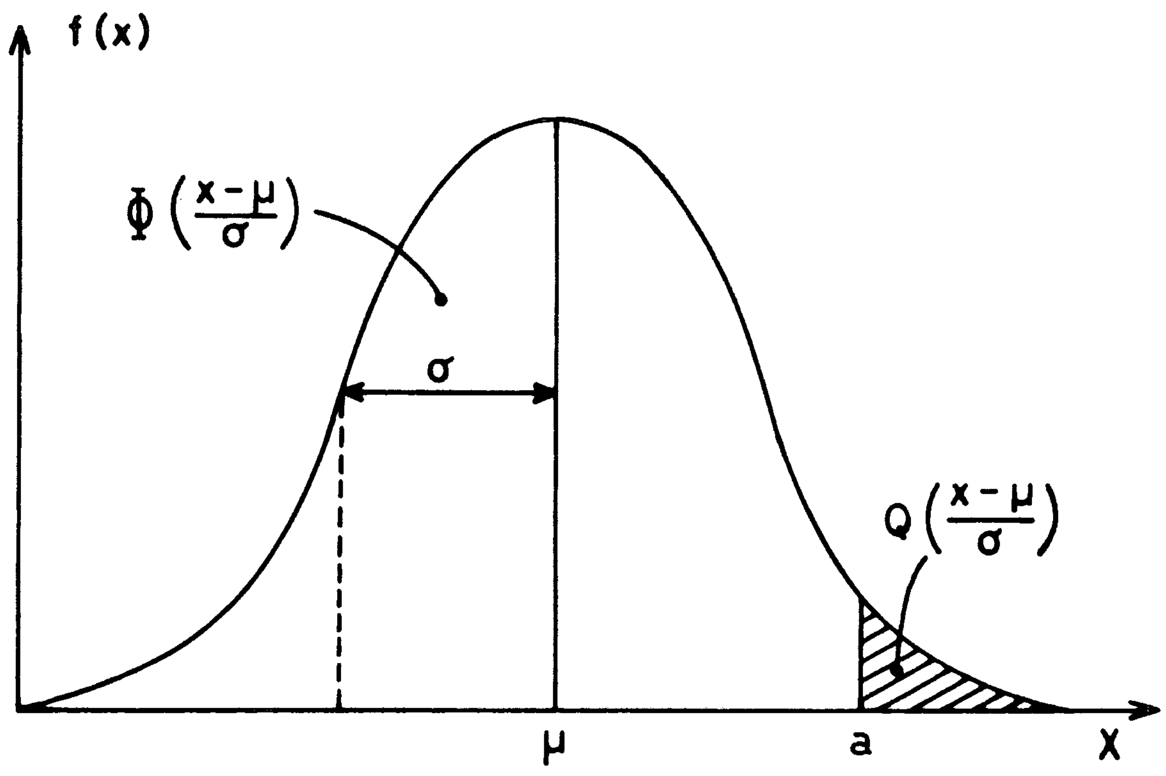


FIGURE (2.7): Probability of Error Curve outside of the range $-\infty < x < a$ is given by $Q((a-\mu)/\sigma)$ or $1-\Phi((a-\mu)/\sigma)$. Q() is called the complementary error function and the SNR analysis must derive a function such that $f(SNR) = \Phi((a-\mu)/\sigma)$ and so that the probability of error, P_e, or complementary error function Q() can be calculated from the corresponding SNR.

2.4.4 Multiple User Support

The final parameter is the variation of the SNR, BER, data throughput and data delay with the number of multiple users. This variation will not necessarily be linearly dependent on the number of simultaneous users and is a means by which the efficiency of the multiple access scheme can be investigated. The hoped for relationships are that the SNRs and BERs are unchanged as the number of users increase and that the data delay remains unchanged but has an associated higher network wide data throughput. Invariably all of these improvements are impossible thus it is important to see which are improved and which deteriorate.

2.5 LANs and Their Problems

The problems of LANs can be categorised into two areas; time access restriction and network controllability. The first is a direct result of the time multiplexing systems which are predominant and whilst in general they perform adequately they fail to provide unrestricted support in time critical and time demanding environments because they must consider the network as a whole and not from an individual node's requirements. The second, network controllability, is of increasing importance due to the confidential nature of the data which is now transferred via LANs. This is not just concerned with military data but more widely with personal information such as individual financial and medical histories where the information must be protected for decades instead of just a few hours. Improvements or replacements in the current LAN technology must be made in the near future because these problems now limit the applicability of many LANs to certain systems where such a network would be useful.

2.5.1 Time Access Restrictions

The problems of time access for LANs (excluding those of FDMA nature) are:-

(a) Actual access to the channel is dependent on the current activity being supported by that channel ie only one user may access a particular channel at any one time.

(b) The duration of continuous access to the channel must be limited or else the peformance of the other nodes is seriously degraded.

(c) When access is obtained the data must be transmitted at a fixed rate requiring buffering at both the transmitter and receiver to absorb node to network data rate mismatches.

The applications of LANs has become widespread hence the requirements for specific applications is more demanding, especially in real time environments. Such situations need to permit any node to send its data when it becomes available and to send it in its entirety instead of forcing fragmentisaion and packetisation - as do the current methods. The LANs also interface to a wide range of devices and it is now typical for a LAN to service both man driven and computer driven devices where there are several orders of magnitude difference in the generated data rates. This leads to complex software capable of interfacing the data types and is reflected in the voice/data networks which are now being produced.

The demands of real time and MMI LANs seem to contradict each other, however, it may be possible to use two different networks for each of the applications. A simpler solution, in the long term, is to employ a standard system which can be configured to specific applications. Also, the requirement to provide real time and MMI interfaces on a network is common thus a single system must be available or else the cost is high; the space taken by two LANs could be prohibitive and their efficient interfacing could be difficult to achieve. Consequently, a rethink of LAN systems is required to try and overcome the inherent limitations of time multiplexing for certain applications.

2.5.2 Network Controllability

The problems for network controllability are:-

(a) Many LANs cannot be simply extended and are limited to the number of nodes a single LAN can maintain. Extension can also mean a complete restart of the entire network.

(b) Data must be encrypted to protect it from unauthorised access which requires the use of standard encryption systems such as the DES (42) and RSA (43) algorithms.

(c) Networks must be controlled to limit access to certain data and to maintain efficient use of the devices linked to the network.

These problems are based on the need for more control on the flow of data. Previously there was little privilege checking in LANs however this and further restrictions must now become a part of the LAN architecture. LANs are becoming the communications medium by which distributed systems communicate hence they must maintain an organised structure for the passing of database information between several processes without the loss of data and whilst keeping track of the fault status of the LAN. Consequently a LAN must now provide a complete distributive facility within a fault tolerant environment of which the applications programs can take advantage.

2.6 Yet Another New LAN?

The previous section has highlighted the requirements for a new LAN which will provide a real time environment dedicated to each individual node and provide a method by which data can be communicated in a private/secure and reliable manner. Such a system could be employed in time critical applications such as control systems and where a complex system requires data encryption, noise immunity, flow control etc as part of the standard LAN properties. Whilst it is not impossible to provide much of this using present LANs by adding several components eg DES hardware, inevitably system incompatibilities arise due to a non consistent system design and more importantly the performance becomes seriously degraded due to duplicated buffer copying and increased control data requirements.

The new LAN proposed in this thesis is termed the Spread Spectrum Local Area Network (SS-LAN) and will provide a LAN capable of fulfilling all of the above requirements. It provides a time independent access mechanism, with flow control capabilities due to its SS modulation: direct sequence sequence SS modulation. The network can be expanded simply and provides a natural data load averaging whilst at the same time encrypting the data and providing inherent noise immunity. Its most significant feature is that it permits simultaneous multiple users each with asynchronous and unrestricted access to the channel which is especially independent of any activity being currently supported by that channel.

2.7 Conclusion

LANs have become invaluable for the interconnection of computers and devices in a cheap and flexible manner. They are, however, no longer limited to systems in which occasional data is transmitted but are being applied where the channel delay is a critical component in the overall system. Several LAN topologies have been developed over the years in conjunction with several multiple access schemes however these are almost totally biased towards one form or another of time division multiplexing, which is in itself the limiting factor in real time environments.

Traditionally, the access schemes applied to LANs have been taken from those already tried and tested in radio communication systems. The past ten years have seen an increase in the use of SS techniques in satellite systems, mobile radio systems and general radio applications, due to the advance of VLSI technology which has provided ICs capable of complex digital signal processing. Until these were available, the theoretical basis for the application of SS modulation was firmly established - however the technology was not sufficiently sophisticated to provide the appropriate hardware.

These developments now mean that a SS based LAN can be proposed as a solution to the problems which are now encountered by LANs: especially that of time restricted access to the communication channel. Whilst the node processing for the SS system is more complex and demanding than for systems such as Ethernet or the Cambridge Ring it does provide an integrated system with noise immunity and data encryption which can only be implemented in other LANs by the copious addition of specialised hardware. The basic concepts of SS techniques and LANs have now been introduced and these will provide the foundation from which the new SS-LAN will be designed, analysed and simulated in the following chapters.

CHAPTER 3

A New LAN using a SSMA Technique

3.1 Introduction

There is NO time sharing, time scheduling or carrier sensing in the spread spectrum local area network (SS-LAN). This removes the major system limitation of LANs such as Ethernet, Cambridge Ring and Token passing implementations and presents the SS-LAN as a potential network for use in real time systems where the access time is the critical aspect of the system performance. The previous chapter introduced the concept of SS and briefly described its properties and applications. It is apparent that the history of SS is dominated entirely by the requirements of radio communications whereas this thesis is concerned with the application of SS to a bandwidth limited medium, the coaxial cable.

Fortunately, the properties of SS modulation still apply within a LAN but there is an important distinction in the noise component between the radio system and the LAN. Within a radio communications system the atmosphere provides the most important noise component and the multiuser interference is usually limited by the large physical displacement of the node distribution and the relatively low simultaneous transmission aspect. In contrast the LAN has low environmental noise but the multiuser noise is predominant, with a high probability of most nodes transmitting simultaneously. Consequently the operation of the new SS-LAN must be analysed with this significant difference in mind.

This chapter is split into six further sections - all of which describe a different facet of the logical operation of the SS-LAN. The next section presents a brief description of the LAN and defines the logical location of the SS modulation. The following two sections describe the properties of the SS-LAN, many of which arise from the SS modulation, and introduce a suite of logical configurations for the LAN, a unique aspect in LAN technology which permits tuning of the system performance for individual applications. The drawbacks of the new LAN are then described, followed by a qualitative comparison between this LAN and implementations of the other multiple access techniques. The final section concludes the chapter stressing the important points introduced within it.

3.2 The System Description

In its present structure the SS-LAN is a bus topology network capable of supporting at least 100 simultaneously active nodes whilst maintaining a network wide data rate of at least 10Mbits per second. A direct sequence spread spectrum modulation technique is employed at the physical layer, as defined by the OSI reference model (44) with the data being encrypted by the DS code sequences in the normal fashion for DSSS systems. The system is organised such that the nodes operate independently and transparently to other nodes as far as the user is concerned but they interact regularly to provide each other with the most reliable picture of the system and to operate the system at its most efficient.

3.2.1 The Data Modulation Scheme

All data which is to be transmitted must undergo DSSS modulation with the resultant signal being transmitted at baseband over the communications channel (a coaxial cable at the present time). The code sequence frequency will be at least 10MHz with a preferred value of at least 100MHz which is a figure in keeping with the present day trends of LAN operational frequencies. The code sequence length will be at least 127 bits long upto a maximum of 2047 bits - the actual value depending on the types of codes in use and this will stipulate the maximum point to point data rate for any particular virtual link. The network wide data rate is a function of the code sequence frequency.

The actual interfacing of the nodes to the coaxial cable will be via an isolating transformer arrangement so that the number of simultaneous users will not alter the characteristic impedance of the channel. The receiver will perform a simple thresholding on the input signal to provide the ADC. A bipolar signalling method will be employed with a corresponding zero threshold (hard limiting) in the ADC. This specifies the signal mean as zero but gives a larger variance than the case for unipolar signalling which requires a variable threshold detector (a soft limiter) and gives a lower variance. It is this variance which defines the noise at the receiver input.

3.2.2 The SS-LAN Topology

The SS-LAN is a bus topology network but the modulation scheme can be applied equally well to star and ring configurations; application to a ring structure requires the use of complex filtering which will reduce the maximum achievable data throughput. The choice of the bus configuration with spurs for each of the nodes gives a more reliable network because in both the star and ring topologies the nodes act as active relays hence the failure of a node isolates parts of the LAN - not the case in the passive bus LAN. The LAN supports at least 100 nodes - the actual number being sensitive to the code sequence type and length - over a distance of not more than 2kms, as specified in the previous chapter. SS-LANs will be linked together via the specialised bridges which will perform any code sequence translation. Interconnection between dissimilar LANs will be performed by specialsed gateways.

3.2.3 SS-LAN Operation

Assume that a SS-LAN configuration consists of ten nodes, where each receiver is assigned a unique code sequence and where eight of the nodes are presently communicating with each other - the final two nodes now start to communicate. The data progresses through the following stages. At the transmitter a request to transmit is received from a user which passes the logical identifier destination identifier to the host node station CPU. The CPU "looks up" the destination code sequence configuration instructions within ROM and then generates the code sequence and stores it in RAM. Again the code sequence generation instructions will vary from code type to code type but would include parameters such as shift register length and start state etc. The CPU will then request all further information such as the length of data and add the necessary protocol information ie message header and trailer, and will provide the transmitter data in word packages. The transmitter now reads the code sequence from RAM and modulo two adds an entire sequence to each data bit. The binary values are converted to voltages and sent along the bus via an isolation transformer interface.

On the channel there are now at least five signals - one for each pair of communicating nodes. If the code sequences are of the correct structure then the composite signal ressembles additive white Gaussian noise and it is this signal which is received at each of the nodes.

The receiver, at each node, is continuously operational and attempts to receive information from any transmitter. The first operation is to perform a hard limiting on the signal with the threshold set at zero - for a bipolar transmitter. The hard limiting is performed on a sampled version of the signal. Sampling is necessary because it is not possible to recover the clock from the signal on the channel because of its analogue nature. This binary sequence is now loaded into a digital correlator which compares this signal with the locally generated code sequence. The local sequence was preloaded into the receiver when the node was first "powered up" and it is the code sequence allocated to that receiver. The incoming signal is allowed to slide past the local code with each chip alignment being inspected to see if the correlation threshold is exceeded. Once correlation is achieved the data is determined according to the number of agreements and disagreements produced by the modulo two additions of the correlation. The acquisition phase is continually repeated to overcome the affects of the clocks drifting in between individual data bits hence a tracking scheme is not necessary. Any clock drift which occurs during a data bit is not compensated for at the present time and the resolution of this could require the introduction of a tracking circuit.

The above operations are repeated for all data transmissions with the receiver responsible for obtaining frame synchronisation, as well as chip synchronisation, and for the removal of the sections of the protocol which were added at the equivalent level in the transmitter. This scheme of operation is valid no matter which codes are used and it is also independent of the way in which the codes are mapped to the node physical addresses: this mapping will be discussed in section (3.4).

3.2.4 Network Organisation

It was stressed in the previous chapter that the properties of the code sequences were important and further importance is ascribed to them when their distribution across the network defines the suitability of the SS-LAN for particular applications. The assignment of the codes to particular nodes is not static and the dynamic allocation must be performed by a distributed network management system which resides at the physical layer and is transparent to higher levels of protocol. The reorganisation of the network will be instigated due to several factors such as the logical reorganistaion of the node functions, better data load distribution or as a periodic function to increase data protection etc.

Not only can the nodes be redistributed but the codes themselves can be reconstituted either in their type or their physical properties, such as length. Also the code sequence to data bit ratio may be altered or different code sequence frequencies can be assigned. The system is in fact highly tunable which provides a corresponding flexibility. A single hardware configuration can be tuned to fit many applications by the suitable choice of the code sequence types, the code distribution, the code lengths and the code sequence/data ratios. In situations where the LAN will be required to provide a general communications channel for several different types of applications it is possible to produce a hybrid SS-LAN in which a variety of code families may be employed - one for each application perhaps.

3.3 Properties of the new SS-LAN

The properties of the SS-LAN system are derived, primarily, from the code sequences employed for the DSSS modulation but these are enhanced by the logical allocation of the codes to the nodes: the modes of distribution are discussed in the following section. The SS-LAN possesses unrestricted access to, and use of, the communications channel, asynchronous communication, contentionless access, total decentralisation, data rate loading according to the individual node, private/secure communications, parallel processing and a priority mechanism, noise immunity and dynamic reconfigurability. It should be noted that whilst all SS-LAN systems possess these properties to some degree it is more realistic to assume that certain ones will be emphasised in different applications which may lead to a corresponding degradation of some of the others.

3.3.1 Unrestricted Medium Access

Chapter 1 introduced the concepts of CDMA and SSMA and chapter 2 described the point at which the DS codes were employed in SS systems. When considering multiple access systems the codes are considered as members of a family and not as individuals. The main criteria for the choice of a family is that the mutual interference between the codes is as low as possible which permits a larger number of codes to be used simultaneously for a fixed interference level. The important point is that within the system limits, the transmissions occur simultaneously and if the design is such that all the nodes can transmit simultaneously then the action of each node is totally independent of the others as regards access to the channel.

The simultaneous multiple user aspect provides each node with the ability to access the channel whenever it wishes ie no access delay, and to continuously transmit data for as long as it requires. This situation is true for all the nodes at the same time and because there is no requirement for polling or waiting, the communications link can be defined from a system viewpoint as asynchronous (communications between nodes is usually protocol oriented and this implements a handshaking sequence which imposes a form of synchronisation at a higher level).

It was stipulated earlier that the system was designed to operate within the specification when all the nodes were transmitting simultaneously. If the specification had defined a 90% simultaneous access load support and the actual load was actually greater than this then the system would not fail but would operate at a higher BER and a lower correlation rate, that is a soft degradation would occur. This means that a system need only be designed for the average environment of its application because it would still operate in adverse conditions but with a degraded performance.

3.3.2 Contentionless Access

The common definition of contention is based upon Ethernet (CSMA) systems and is phrased as "a contention arises when two or more nodes attempt to transmit data during the same packet cycle". Thus should a node be transmitting data then no other node should attempt to do so until the first has completed otherwise a contention arises and both nodes have to execute a recovery algorithm. In true TDMA and FDMA contention is not possible, however, in the former this advantage can be lost with the associated decrease in data throughput and access time.

For the SS-LAN this definition must be amended to become, "a contention arises when two or more nodes attempt to transmit data using the same code sequence and when the code sequences are chip aligned". The chip alignment is a necessary addition because a SS receiver will lock onto the first correlation position and will ignore further positions if they are not within the region of the cyclic correlation point - this ability counteracts the problem of multipath (which seriously degrades other systems) and makes it ideal for mobile radio taxi systems. If the codes are uniquely allocated to the transmitters then the SS-LAN is truly contentionless (however this will not necessarily be the best allocation scheme) but other allocations can lead to the situation of a contention; this probability can be minimised by using long code sequences, short messages and by reducing the need for nodes to use the same codes. The last method is implemented by the careful choice of the logical allocation of the codes.

3.3.3 Total Decentralisation

Many systems rely upon one form or another of centralised control eg the monitor in the Cambridge Ring, the controllers in the ASH or the token passing scheme used in token rings and buses, where if one element fails then it can either cause a total failure of the system or require a complex recovery algorithm to revive the network. Reliable applications therefore require either several backups for each device in the LAN or use a decentralised technique where the nodes are totally independent and the failure of one node has localised effect on the system. The SS-LAN is totally decentralised and does not require any form of common clocking system. The only limit on the nodes is that their own clocks should be as stable as possible otherwise correlation becomes degraded.

3.3.4 Variable Data Loading

The unrestricted access nature of the SS-LAN produces a beneficial effect on the data loading of the LAN. Whilst it would be usual for the code sequences to be common across the LAN it is not the case for the data frequencies. This is because the code sequence length and the number of sequences per data bit can be altered hence the data rate varies accordingly. From a network wide point of view this effectively distributes the bandwidth where it is required most ie where the highest data rate is located. This can be readily understood by considering, once again, Shannon's capacity equation in which the channel can only support a certain capacity for a given bandwidth and SNR. In most systems the bandwidth allocation is fixed for each node usually by time sharing channel access (TDMA) or frequency allocation (FDMA). In the case of TDMA the nodes must transmit data at a fixed rate independant of the amount of data to be transmitted or of how important it is for delivery to be achieved within a certain time.

The code sequences perform true bandwidth sharing: the longer the code the lower the data rate thus the lower the capacity requirement hence the lower the node point to point resultant bandwidth. An added advantage to sending data at the lower rate is that as the code sequence length increases then the rejection levels increase hence more simultaneous users can be supported, the link becomes less likely to accidently reveal its information, the contention rates are reduced and the noise immunity is increased.

3.3.5 Security and Privacy of Data

One method of interpreting DSSS modulation is as a form of data encryption. The data itself is not sent but instead a representative waveform is transmitted whose decryption requires the presentation of the exact code sequence and the correct chip alignment. If the code is randomly generated then for a long sequence the probability of accidental decryption is low and intentional unauthorised decryption becomes time consuming. The use of codes endow the SS-LAN communication with "privacy" ie a casual "listener" cannot receive or understand the data being transmitted on the channel unless it is specifically sent to it.

A private system is not a secure one as the latter must protect the data from the intentional system intruder. A secure system requires the use of cryptographically secure codes - these must be of a nonlinear structure - which make the task of "cracking" difficult. It is impossible to make a useful code which cannot be broken given enough time and computer power but it is possible to design them such that it takes excessively large amounts of both and also to make the lifetime of the data as short as possible. Secure codes could be used in the SS-LAN however a restriction on their use is created due to the size of their family; in the case of nonlinear codes these tend to be small.

Communication between nodes is only possible if both the transmitting and receiving nodes have knowledge of the other code sequences hence "breaking" of the system requires the failure of more than one code. An extension of this is that the codes themselves need not be stored in any non-volatile medium. Only the configuration instructions are necessary to a valid node hence these could be openly transmitted provided the code generation algorithm was kept secure. The encryption also provides a pointer to the source of the information and should a node start to transmit randomly without reason then it would be possible to trace it if the logical allocation of the codes was related to the transmitters.

3.3.6 Parallel Processing and Priority Mechanism

Once simultaneous communication is possible then a node can be designed to receive more than one message at a time, resulting in a parallel processing capability. If two nodes were to transmit data to a common node then unless that node possessed duplicate receivers it would miss the later message. This is overcome by supplying each node with a series of receivers - all of which perform independently - with the limiting factor being the physical space available for the number of receivers. The data throughput is significantly increased for an individual node by the use of such a scheme however there is an increase in the complexity of the node control. This is because the requests for transmissions have to be mapped to the available and idle resources, unless the node acts as a data concentrator with channels dedicated to different devices.

One application for this parallel processing is in the form of a priority mechanism whereby the code sequences are allocated priorities and the reception of data is dependent on its associated sequence priority. This means that the data from high priority nodes could be received in preference to lower priority data, guaranteeing the maximum data delivery time which is essential in real time systems. One important priority mechanism is that used in interrupt schemes: traditionally these must be handled sequentially. However in the SS-LAN they will be received immediately without, necessarily, causing the loss of other data.

3.3.7 Noise Immunity

Chapter two described the "legendary" noise immunity of the SS systems and naturally the SS-LAN also possesses it. This noise immunity however is traded with the number of simultaneous users which can be supported by the system. For a particular PG the noise limit is defined and both environmental noise and multiple user noise affect this limit. SS systems are better equipped to withstand certain types of noise according to their modulation technique - DS modulation is supposedly more resilient to pulse noise than it is to continuous wave noise, Dixon (15). This immunity makes SS techniques attractive in conditions of extreme electrical noise such as in power stations and industrial control applications.

3.3.8 Dynamic Reconfigureability

The SS-LAN can be readily extended to include more nodes provided there are enough codes to facilitate the extension. The length of the LAN is also inconsequential, unless the propagation delays start to affect the protocol time out algorithms, which is highly unlikely.

In contrast the length affects access delays in ring structures and also Ethernet systems; in the latter the length must be no greater than a 50nsec equivalent. The individual nodes have been described as possessing a distributed network management system and this is responsible for co-ordinating the code sequence allocation to these nodes, using several criteria such as data load, data type and node function. This operates transparently to the higher levels of protocol and provides a situation where no user is aware of the actual distribution and status of the codes. The network management is completely independent, which produces difficulties in the enrolling of new nodes to the network but which also means that the SS-LAN is in a constant state of flux in an attempt to operate in the most efficient manner.

3.4 Logical Configurations of the SS-LAN

The use of code sequences at the physical layer of communication provides an extra mapping interface which can be altered to produce different logical systems. The actual use of the codes is independent of the physical allocation of the codes to the nodes - the operational formats are concerned only with the logical allocation of the codes ie the codes may not always be assigned to individual receivers but would be mapped to the type of data to be transmitted. The choice of code allocation method used is dependent on the system's application but it will be shown later, that the final mode specification can be made in such a way that a single version of the node can implement whatever code allocation scheme is appropriate. The distribution formats which will be described are the point-to-point mode, the broadcast mode, the node group zone mode, the function mode, the message type mode and the protocol mode. Many of the modes are closely related to each other and usually define a more restricted allocation which could be used in specialised systems.

3.4.1 The Point-to-Point Mode

In this mode each receiver is allocated a single unique code or a set of unique codes - the important point is that only one receiver can

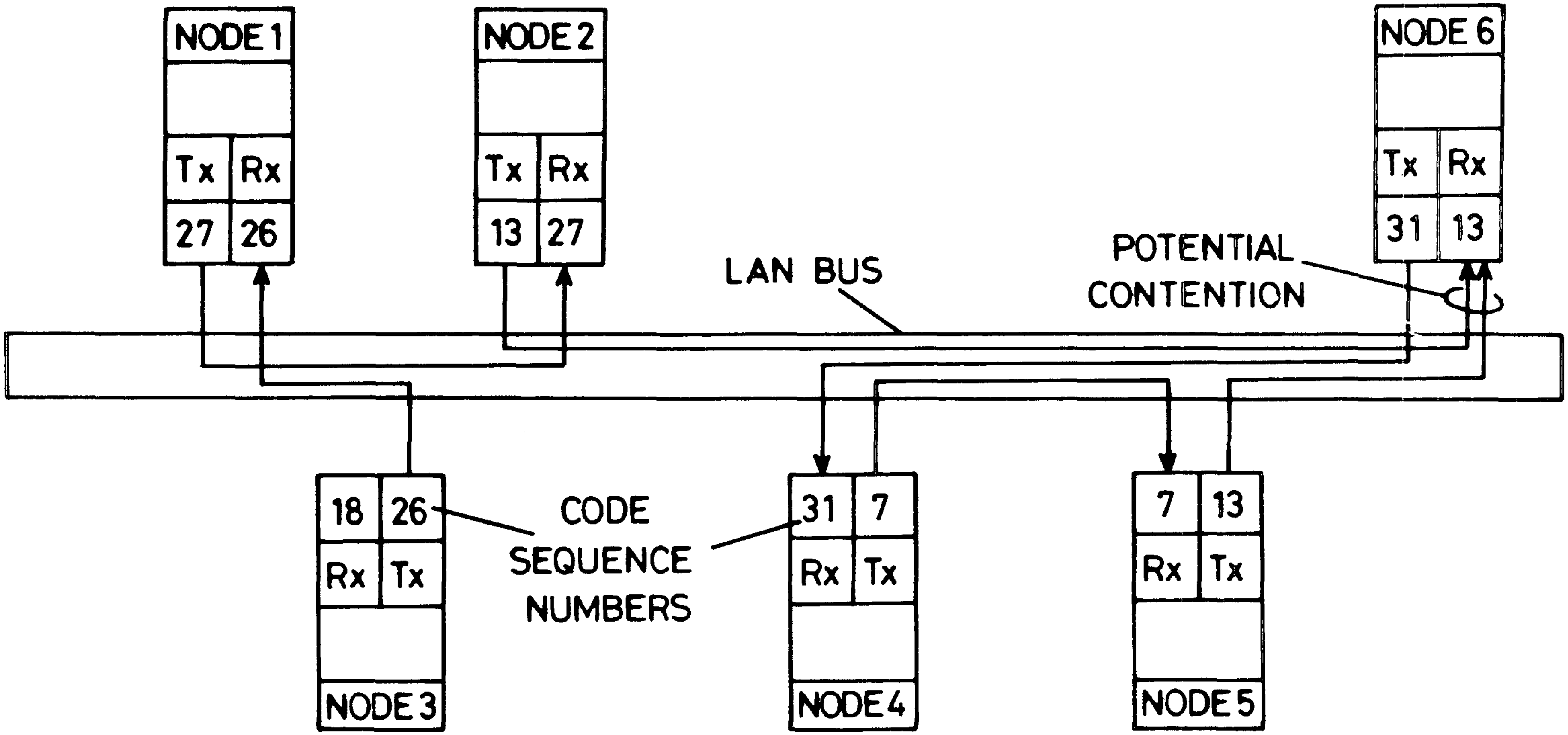


FIGURE (3.1): Unique Receiver Distribution receive data from any one code. When a transmitter requires to send data to a node then it must first generate the destination node's code sequence. This code is used to perform the SS modulation and to transmit the data. The receiver at each node need now only "listen" on the unique codes allocated to it (in fact it cannot "listen" for any others) because all the data intended for it has been encrypted using these sequences. The properties of the SS-LAN now apply - however there is a low probability of contention which occurs when two or more nodes attempt to transmit to the same node using the same code and when they are chip aligned. In the event of non chip alignment and if a single receiver was operating at the node then the data would be missed however this is rectified by using multiple receivers (with co-ordinated search algorithms) and relying upon higher levels of protocol.

Figure (3.1) shows an example of the unique receiver allocation or, more descriptively, the point-to-point mode. The codes on which each node receives data are contained in the "RX" box and the code which is being transmitted is given in the "TX" box. The diagram shows node 3 transmitting using code 26, which has been physically allocated to node 1, hence transmission is made logically to a node number and mapped to its physical address which is the code sequence. Similarly node 4 transmits to node 5 and node 5 to node 6. The last example highlights a contention because node 2 is also transmitting to node 6 hence data would be missed in this single unique code allocation implementation.

3.4.2 The Broadcast Mode

The reverse of the previous mode is used in this case. Each transmitter is allocated a single unique code or a set of unique codes where again the uniqueness is the most important factor. When a node wishes to transmit information it uses one of its own unique codes to spread the data. The responsibility for the data being received by the correct node is now the receiver's which must "listen" to all the code sequences (and hence nodes) from which it expects to receive data. This is more commonly termed a broadcast facility because a single message can be received by several, if not all, other nodes. Again the SS-LAN properties are provided however a major advantage over the point-to-point mode is that there is no possible contention whatsoever. The

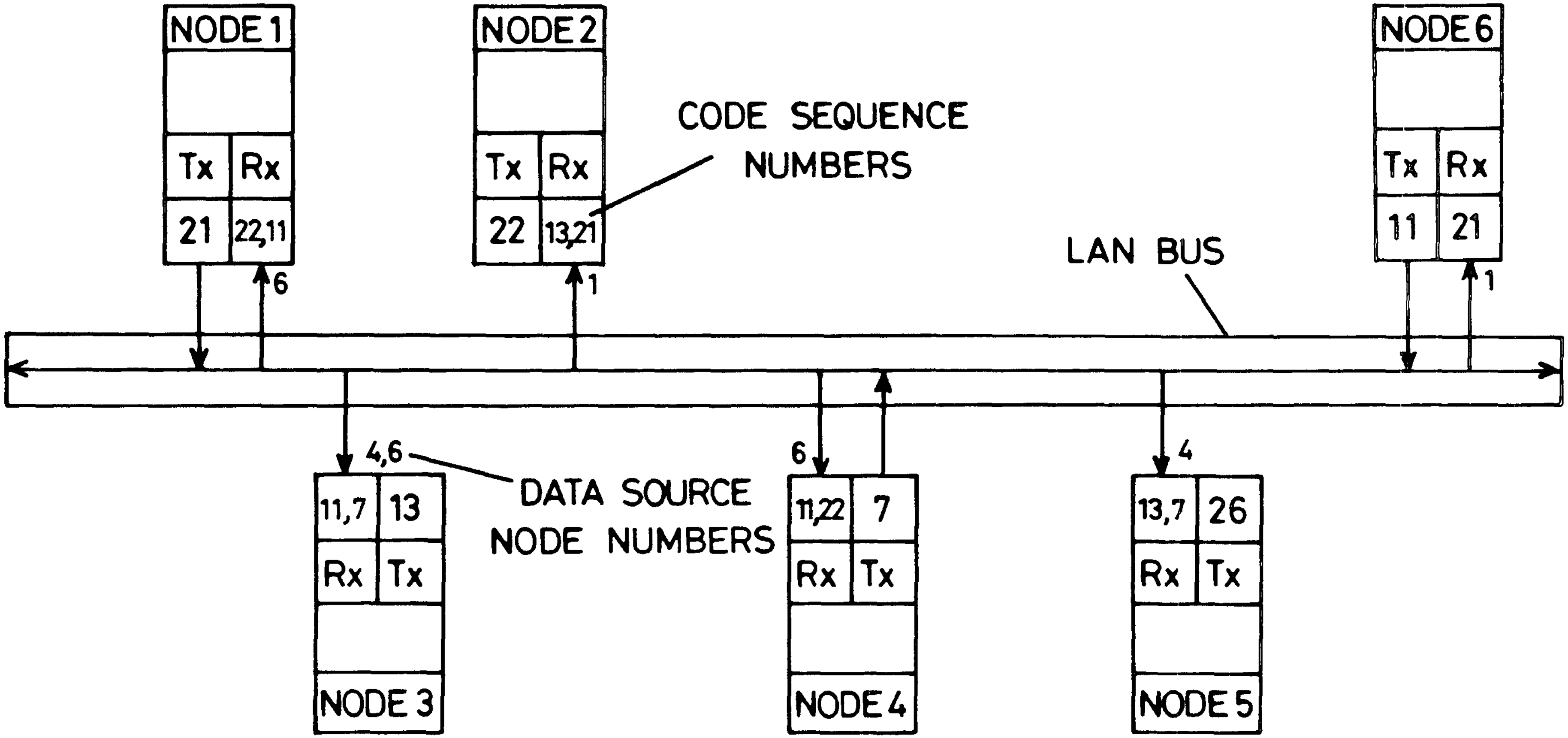


FIGURE (3.2): Unique Transmitter Distribution drawback is however in the sophistication of the receiver because it must be capable of receiving more than one code sequence or else it will only receive data from a single transmitter.

Figure (3.2) shows an example of this mode where the codes which each node is capable of receiving are listed in the "RX" box and the node on which it is transmitting is listed under the "TX" box. In the case of node 6 it transmits data using code 11 hence nodes 1,3 and 4 can receive this data ie node 6 has broadcast a message to them. A similar situation is shown between node 1 and nodes 2 and 6. Node 3 is receiving data on both of the codes on which it is "listening" hence it requires a parallel processing capability to prevent the loss of data. Should it be impossible to allocate multiple receivers to a node then one solution to the above problem is to perform a sequential search on a list of codes using one receiver. Whilst his does not solve the problem of multiple nodes sending data to a receiver simultaneously it does allow a receiver to obtain data from more than one transmitter.

3.4.3 Node Group Zone Mode

A different approach to the previous two modes is to group the nodes into logical sets, called zones or domains as defined by a recent ICCC report (45). A zone is defined as a group of nodes which regularly communicate with each other to perform a common function, which can be physically distributed across the network but which form a small subset of the total number of nodes in the network. Within each zone a set of code sequences are allocated and communication amongst zone members is limited to these sequences. There is no restriction on the zones as to how its codes are distributed thus different zones could use different modes within the overall domain mode. A node could be a member of more than one zone thus the network management system must keep a record of the zone organisation. A problem within this system is that external nodes could not communicate with a zone however this could be resolved using a higher level mode on top of the zones or by using common code sets amongst certain zones. The latter produces a probability of contention and is not therefore a preferred solution to interzone communication.

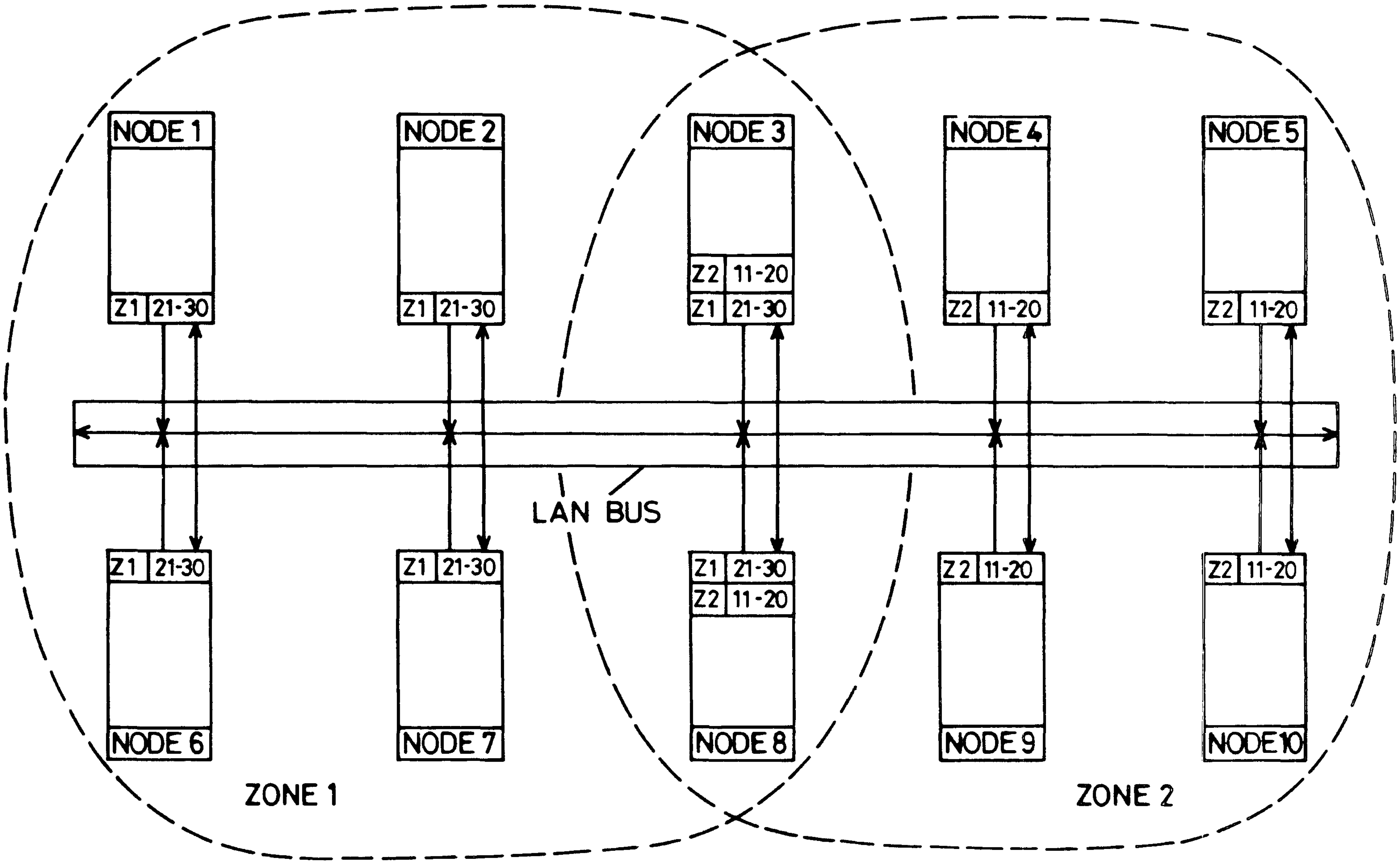


FIGURE (3.3): Node Group Zone Distribution

Figure (3.3) illustrates an example of two zones located in a 10 node SS-LAN. The zones to which a node can communicate are given in the small boxes adjacent to which are the code numbers assigned to that zone. Nodes 3 and 8 are common to both zones hence they could be used for routing information between the zones if necessary. The zone codes are uniquely allocated hence there is possibility of contention between the two zones but there may be contention within the zones depending on the allocation mode.

3.4.4 The Function Mode

A logical extension of the node group zone mode is to specify clearly defined functions for the nodes. Typically these functions should utilise two or three nodes but the important fact is that any single node communicates unidirectionally with each node and uses one code for transmission and one for reception ie the network is constructed from a series of long term virtual links, all of which can operate simultaneously. This mode is a restricted form of the point-to-point mode and as such there is no possibility of contention if unique code allocations are used. Also there is increased system security because knowledge of the nodes is logically localised and the receiver requires no parallel processing capability.

Figure (3.4) illustrates a typical set of virtual links which could be established across a SS-LAN. The code numbers for transmission and reception are as marked in previous diagrams and each code is employed unidirectionally and uniquly across the network. In the diagram nodes 1 and 3 are functinally linked, as are nodes 2,4,5 and 6. The linkage of the latter is similar to a ring structure implemented on a bus topology LAN where the data circulates in a single direction.

This type of system is useful when nodes are paired together in sensor and control applications where it is important that no single function should dominate the operation of the network. A typical application could be in an industrial environment where a large number of sensors report to their own controllers indicating temperature, humidity, the speed of conveyor belts etc. It is simpler to provide a standard communications channel than to supply individual trunking,

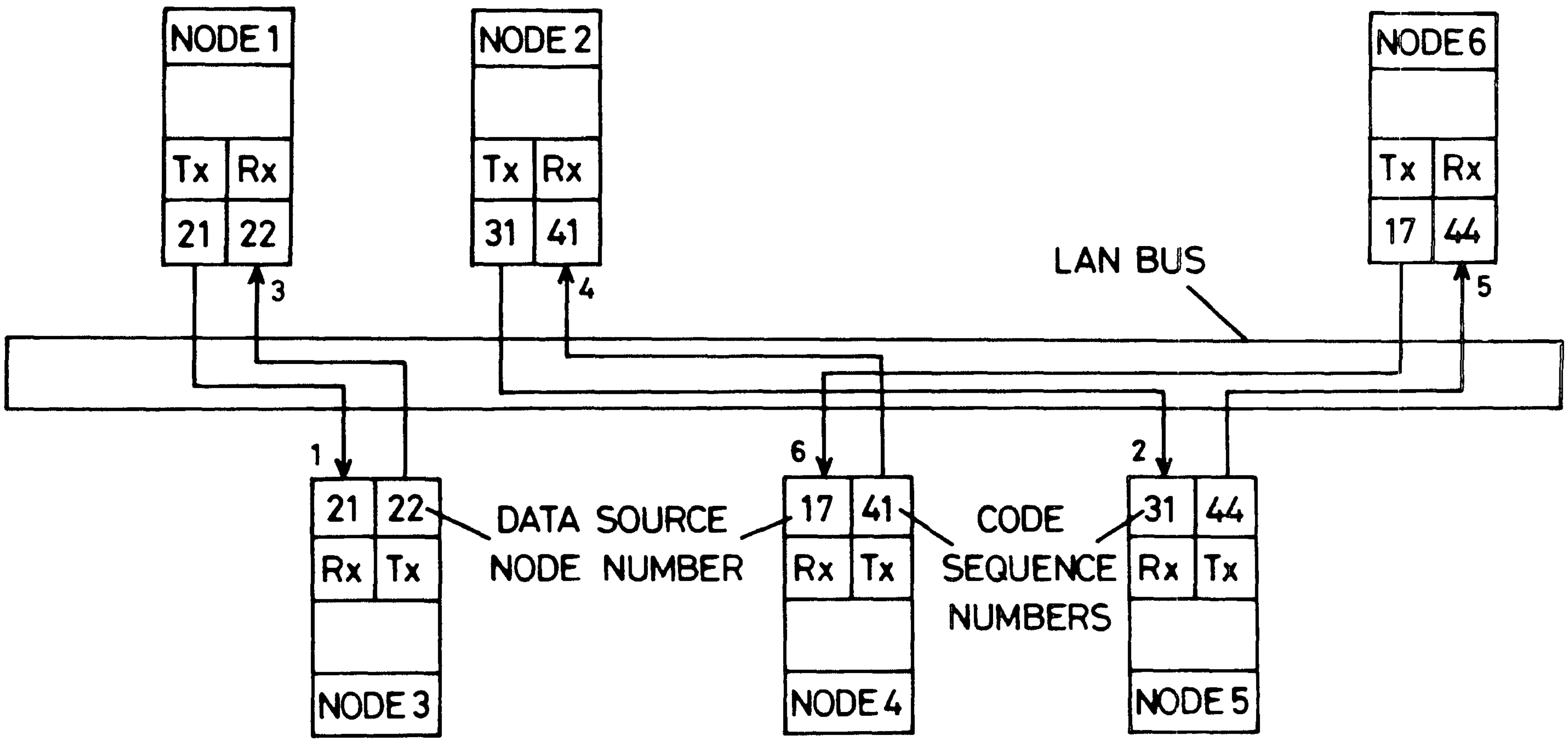


FIGURE (3.4): Functional Distribution

especially if there is to be a management monitor which requires information from all other nodes.

3.3.5 The Message Type Mode

This mode assigns the code sequence to the data according to the data type. Hence when a node transmits data it must first access the code pool and find the code assigned to that type. The receiver must therefore "listen" on all data types from which it expects to receive information; a situation similar to the broadcast mode allocation. Care must be taken when using this mode because if there are only a few data types then there is a high probability of contention, especially if the messages are of long duration. The ideal situation for an SS-LAN implementing this mode is for a large number of short messages, datagrams or short control messages, such as in a remote procedural call system (46).

An example of the message type mode is shown in figure (3.5). Each transmitter has a list of all data types it can send and these are allocated code sequences - each type has its own unique code sequence which is the same throughout the entire network. The receivers are capable of receiving all the codes by using either a sequential search or by parallel message processing. A contention is shown between nodes 3 and 4, both of which are using code 7, hence there is a potential loss of data in the system. All of the other communications occur simultaneously with no transmission problems.

3.4.6 The Protocol Mode

The final distribution scheme is the protocol mode, which can be implemented in two ways. The first is to take the ISO-OSI reference model and to allocate a code for each level and stipulate that the level at which the data originates is the mapping algorithm to the code sequences. The second method is to assign the codes according to the states within a protocol level ie one code for request to link another for request rejection or request acknowledgement etc. Errors within the protocol exchange would be reflected by the loss of code sequence synchronisation providing a method to simplify message loss and error

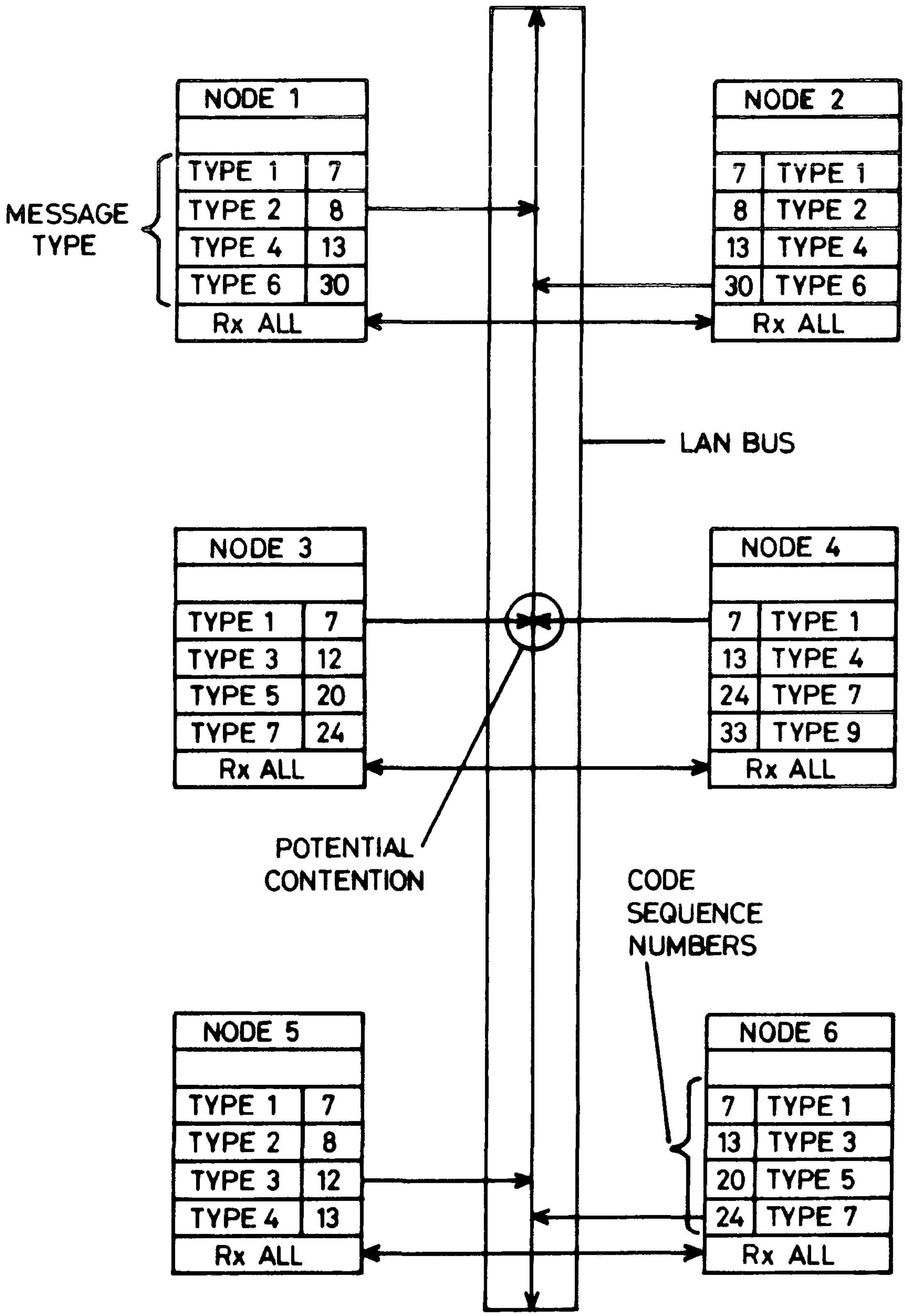


FIGURE (3.5): Message Type Distribution

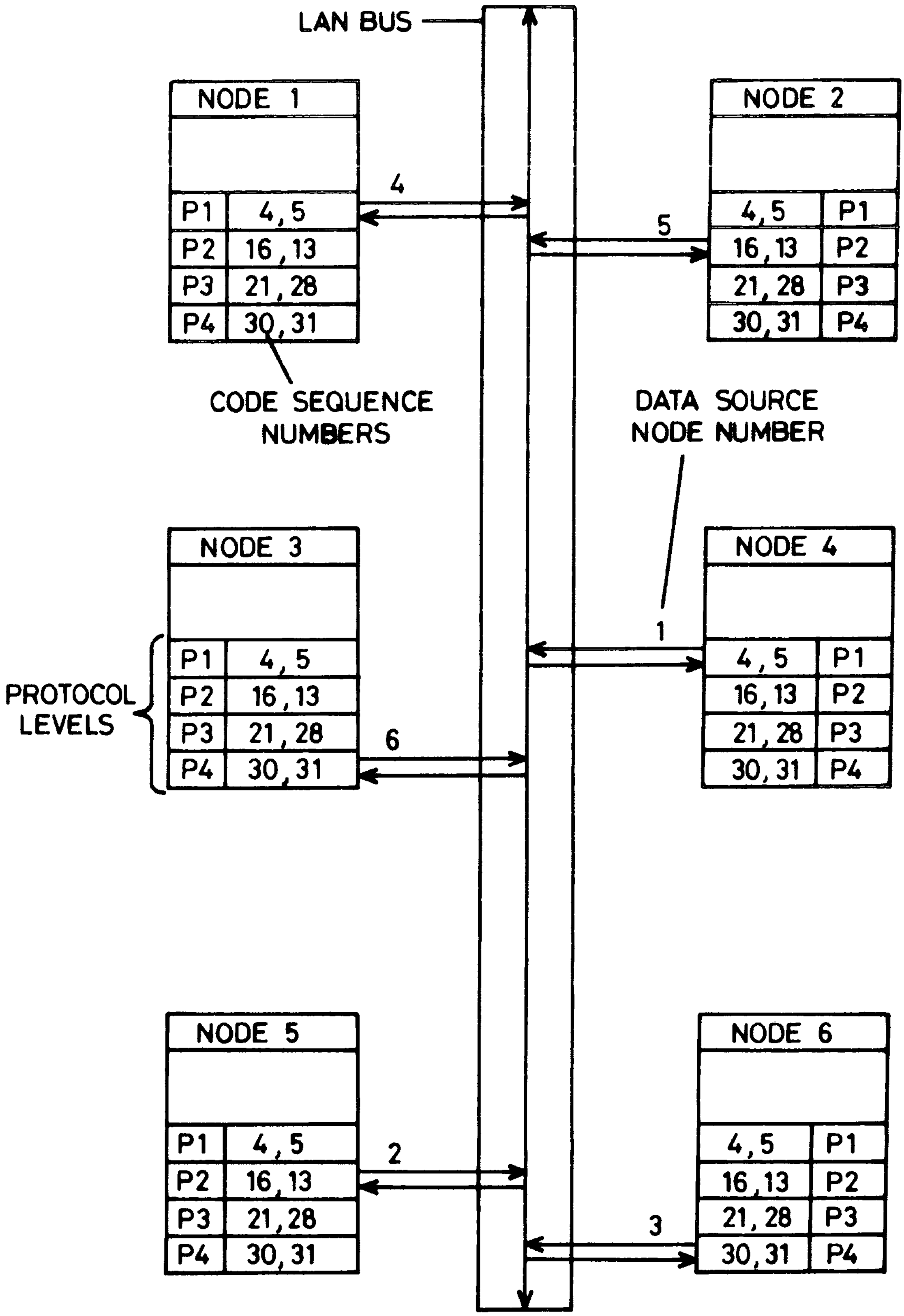


FIGURE (3.6): Protocol Distribution Level recovery. The receivers of such nodes need only "listen" to one code sequence at a time (for each virtual link they maintain) but there is a low probability of contention should more than one node step through the protocol at the same moment in time.

Figure (3.6) illustrates the operation of this mode. Each protocol level has associated with it two code sequences depending on which side of the protocol it is involved ie transmit or receive. Whilst the node pairs, such as 1 and 4, operate at different protocol levels there is no contention - however should these coincide there is a potential loss of information. The usefulness of this system is restricted to systems which employ "heavy" protocols with many layers and sublayers but it could also be implemented in distributed systems where the interaction is set near the operating system layer, thereby handling all the interprocessor traffic.

3.5 Criticisms and Failure Modes of the SS-LAN

No system is perfect and totally failure free but the important point is that the limits must be carefully specified and designed so that the system is not submitted to a situation in which it would fail. Similarly for the SS-LAN there are situations in which it will either fail or not perform as well as some other LANs. This section will look at these criticisms and failure modes and will discuss their importance and likelyhood, concluding in a discussion of applications in which the SS-LAN should not be used. These criticisms and failure modes are the comparatively low point-to-point data rates, the receiver instability (due to clock instability) and protocol implementation problems (for a system which is radically different from those about which LAN standards are being specified).

3.5.1 Low Point-to-Point Data Rates

It was stated in chapter two that a LAN should provide a minimum point-to-point data rate of at least 1Mbits/sec. Most present LANs operate 10Mbits/sec (those which are non fibre optic based) but only provide an average point to point data rate of 1Mbits/sec because of the time sharing algorithms for channel use. In the case of the SS-LAN a data rate of 1Mbits/sec implies that the code sequence operates at a minimum frequency of 100MHz in order to achieve a modest spectral spreading or PG. Whilst this is not difficult on fibre optic systems or impossible on coaxial systems it does make implementation more difficult, especially as a code sequence rate of at least of 1GHz would be preferred.

The use of a fibre optic bus is not being implemented at this time because of the problem of the multilevel signalling which is required in the passive implementation - an active fibre system is a point to point chain linking each node in turn hence each node acts as a repeater. Consequently in this type of bus a new transmission algorithm is required to ensure two level transmissions independent of the four possible states of two nodes. Similarly a 1GHz signal could be passed along a coaxial cable but this would require repeaters placed only a few metres apart, in order to overcome the channel effects: it is interesting to note that the Japanese are in fact investigating 1GHz coaxial systems at the present time.

As technology improves the transmission rates will gradually increase from 10MHz to 100MHz, even along standard coaxial cable and using VLSI techniques, which means that the standard LANs must also provide their data at that frequency - difficult for many data sources, especially those which act as man/machine interfaces (MMIs), without the use of internal data buffering. Consequently the data load must be more evenly spread as it would be in a SS-LAN system with the point-to-point rates allocated where necessary. In fact, is a high point-to-point data rate strictly necessary? Whilst it is important to transmit the user data between the application layers as quickly as possible it has been found that this data transfer rate is almost independent of the physical layer speeds (as reported by Ellis et al (47) and the CERN implementation of their DATABUS protocol (48)) due to the delays imposed by the higher level protocols (it is the packetisation and fragmentisation of the messages which cause this extra delay overhead). This therefore suggests that whilst it is advantageous to transmit the data as rapidly as possible in time sharing systems, this is non critical within the SS-LAN and a data rate of some 100kbits/sec, point-to-point, would in general be sufficient within a layered protocol implementation.

3.5.2 Clock Stability

The SS-LAN performance relies upon the quality of the digital correlation method emloyed in the receivers. In an ideal world the relative states of the transmitters and receivers communicating with each other would remain constant, as in a LAN, especially if there is no relative motion between the nodes. However, the consequences of not having a common reference clock (decentralisation) and of not being able to clock in the data by using it as the time reference itself (a standard clocking technique described by Stremler (49), as for example in Manchester encoding) because of the analogue form of the channel, are that the clocks in the receiver and transmitter must remain stable and must not drift relative to each other and that the signal must be sampled to provide a digital signal for the sliding correlator: this digital conversion will be discussed in a later chapter.

Clock stabilities can usually be expected to be in the region of 1Hz in 10^6 however if the code sequence is 1000 bits long then the drift will occur within 1000 data bits - a typical message would have at least 160 data bits however in the SS-LAN it is likely to be effectively endless ie an open and continuously maintained data path. The effects of the clock drift are that during the sampling, a chip will either be missed altogther or it will be clocked in twice. The position within the code sequence at which this occurs is also very important because this will define the effect on the correlation scheme. Figures (3.7a),(3.7b) and (3.7c) show the effect of chip duplication when it occurs at the end, start and centre of a code cycle respectively. In the first two examples synchronisation can still be achieved if the major section of the code is left intact however in the case of duplication in the centre the effective maximum correlation is only half of its original value and this would not be sufficient to give synchronisation - in this situation the data would be missed. A similar situation is shown in figures (3.8a), (3.8b) and (3.8c) which are the corresponding diagrams for the loss of a chip. Again the critical area for chip loss is at the centre of the code sequence.

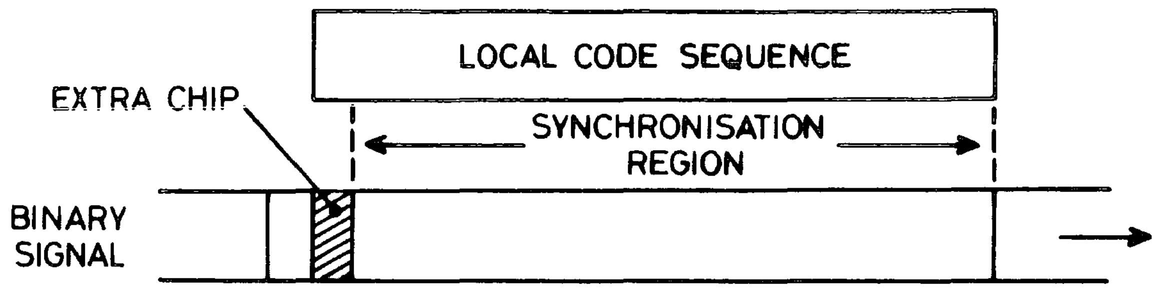


FIGURE (3.7a): End of Sequence Chip Duplication

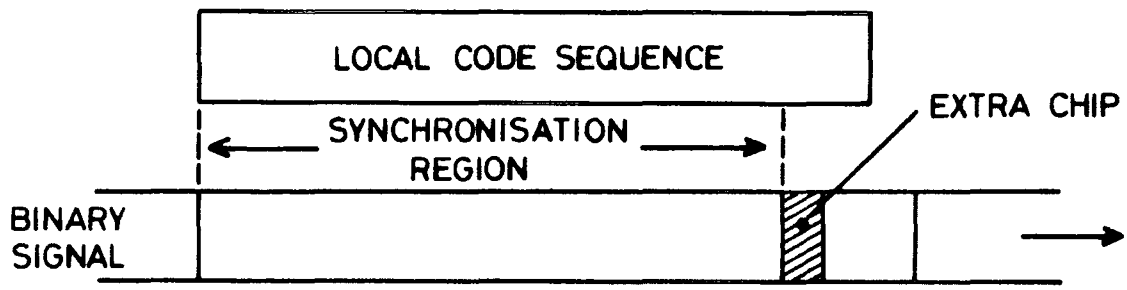


FIGURE (3.7b): Start of Sequence Chip Duplication

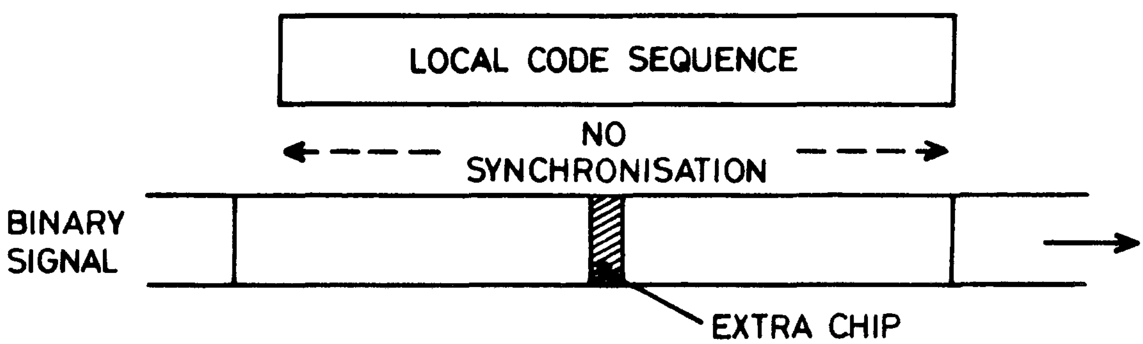


FIGURE (3.7c): Centre of Sequence Chip Duplication

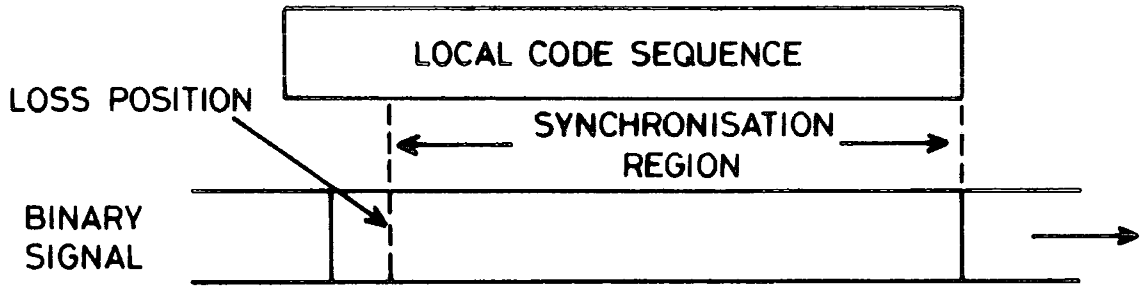


FIGURE (3.8a): End of Sequence Chip Loss

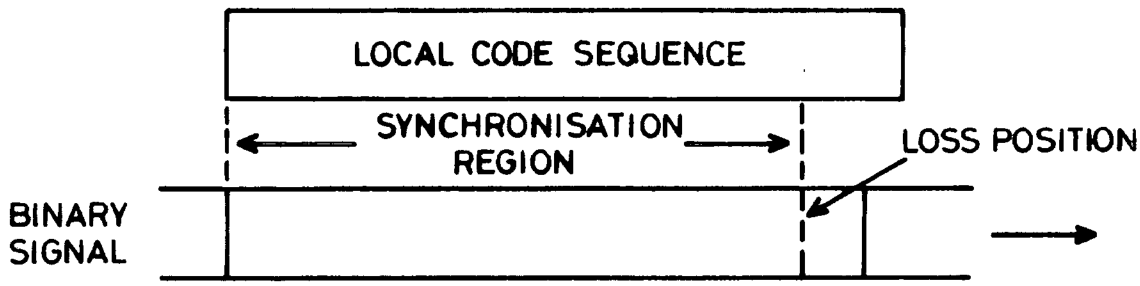


FIGURE (3.8b): Start of Sequence Chip Loss

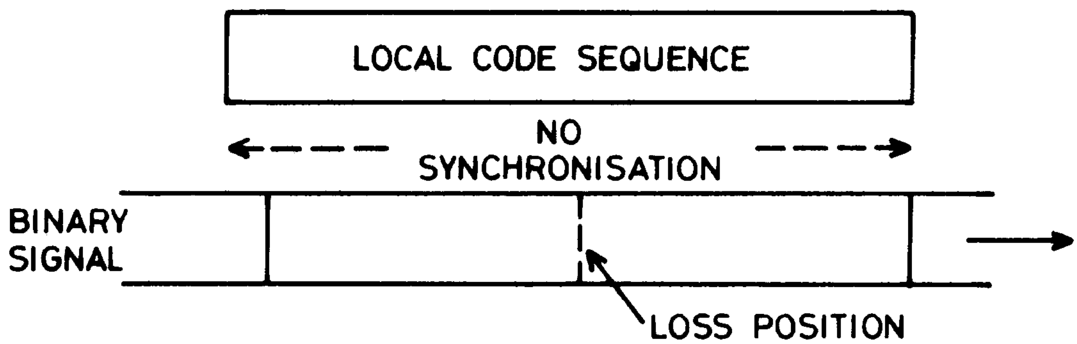


FIGURE (3.8c): Centre of Sequence Chip Loss

These effects are demonstrated in figures (3.9a) and (3.9b) which show the variation in the maximum achievable ACF with the duplication or loss of a chip in the received code sequence respectively. The code sequences are m-length PN codes, 127 bits long, with the feedback taps mask set as 2,62 and 118 for a modular shift register configuration. In both figures the maximum ACF is seen to decrease as the position of the loss or duplication approaches the central region of the code sequence. If the correlation threshold is set as shown then correlation will not be achieved if the loss or duplication occurs between chip positions 20 and 110 in the received signal.

At the present time other researchers are developing solutions for this effect which may include the use of tracking to minimise the area of effect around the central region of the code sequence. It is however outside of the scope of this thesis and as such will be assumed not to occur in future discussions of the receiver.

3.5.3 Protocol Implementation Difficulties

The present standardisation practices, as defined by OSI (44) and the lower level specifications given in the IEEE 802 standards (50), have layered the node functions with the limitation that the functions of the lower levels should be transparent to the higher levels. In the case of the SS-LAN certain modes require different types of information from the higher levels hence a flexible node, which can operate in any mode, will have a feedback effect on the applications drivers. One particular example is in the case of the message mode where the type of data maps to the code sequence. This means that the applications layer would have to pass the data type down through the protocol layers so that the physical layer could make use of it ie the physical layer requires knowledge of the type of data being transmitted instead of needing only descriptive information, such as destination and length.

Again the protocols required for the SS-LAN are outside of the scope of this thesis however care is required to design ones which are not only compatible with the present layer type structures but which also make full use of the SS-LAN's unique properties.

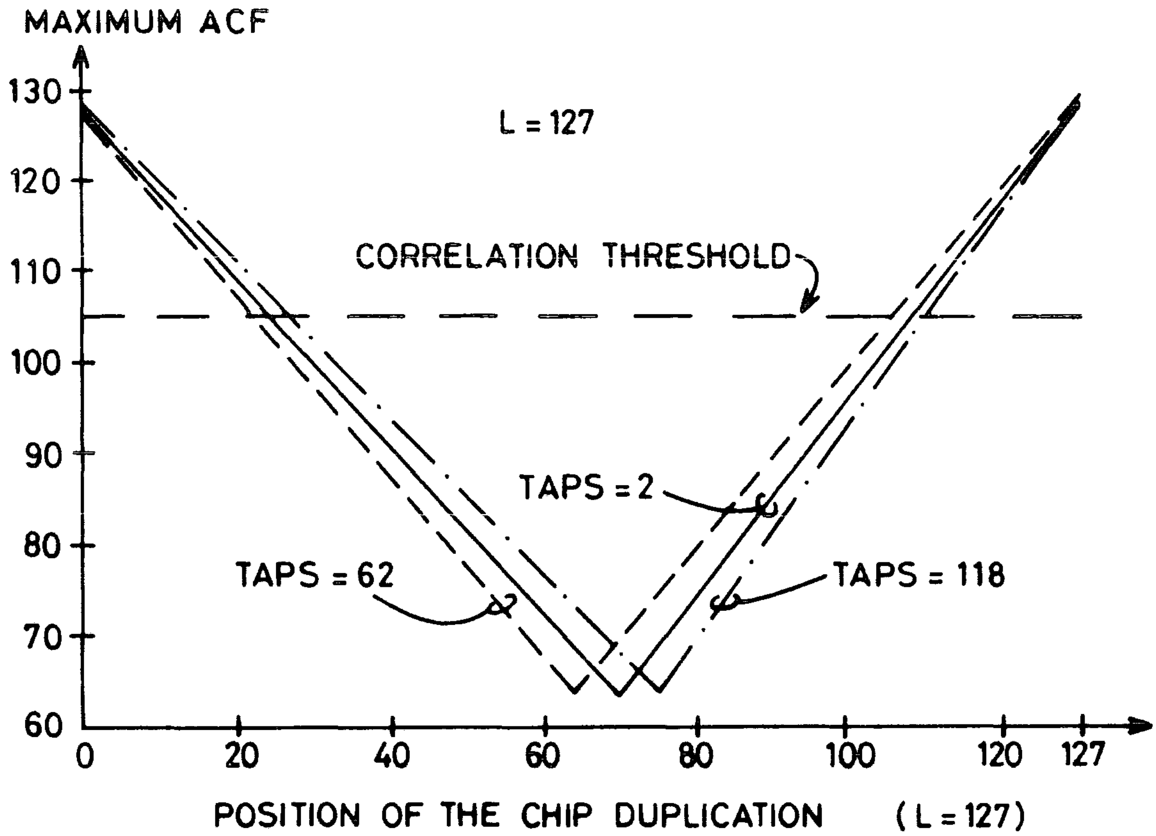


FIGURE (3.9a): Chip Duplication Effects on the Maximum ACF

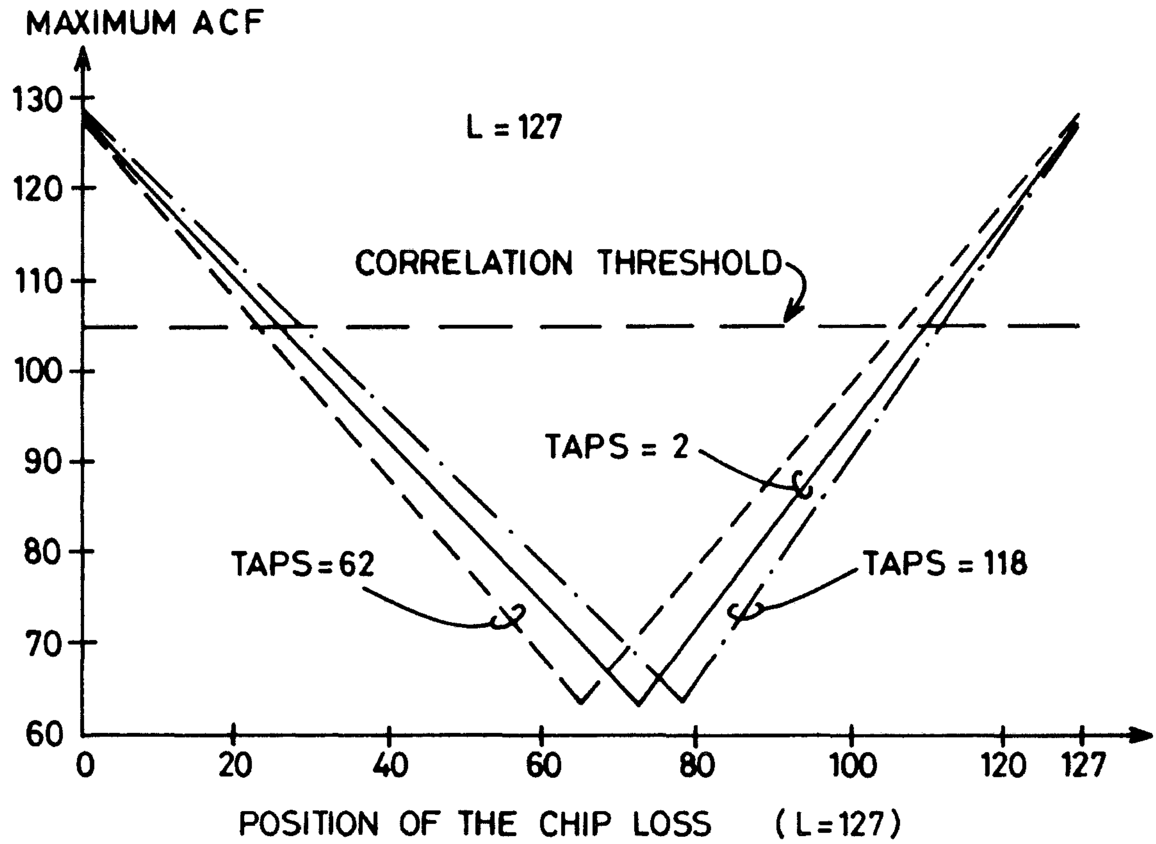


FIGURE (3.9b): Chip Loss Effects on the Maximum ACF

3.6 Comparisons of the SS-LAN with other Multiple Access Systems

This section will present a qualitative comparison between the SS-LAN and a LAN from each of the competing techniques ie CSMA, TDMA, FDMA and Token passing systems. A similar but quantitative comparison will be made in chapter six when a theoretical analysis for the SS-LAN is produced.

3.6.1 CSMA Systems

This comparison will be founded upon the Ethernet LAN utilising the basic CSMA/CD scheme. Both the SS-LAN and the Ethernet operate in a decentralised format however this produces problems in the CSMA system. The major difference between these two systems is that the Ethernet allows only one node to transmit at any one time and should a contention occur than all the nodes concerned must perform the retry algorithm. Consequently the performance of the CSMA system is seriously degraded with a large number of nodes and when the probability of transmission is high for the node. The data throughput and delay calculations in Ethernet must reflect this statistical access mechanism and while the basic data rate is 10Mbits/sec this is not achieved by the system over a long period. The use of an acknowledgment protocols further decreases these data parameters. The CSMA possesses good noise immunity (due to the coaxial cable) and it can be easily extended within its system limit. The point-to-point data rate is higher in the Ethernet system once data transmission is successful however it cannot provide a parallel processing mechanism as it can only operate sequentially and any data encryption must be implemented separately before the data is transmitted.

3.6.2 TDMA Systems

A distinction has been made between CSMA and TDMA by stipulating that the time segmentation should be predefined and constant, as in a slotted mechamism, and not of a probabilistic nature. A comparison will now be made between the Cambridge Ring and the SS-LAN. The ring provides a point-to-point data rate of approximately 1Mbits/sec which is guaranteed for every node. There is therefore a guaranteed maximum delivery time for the data. The differences between it and the SS-LAN are that the Ring uses a monitor station which if it fails will halt the ring's operation, it limits access to the medium and it does not permit parallel processing. The Ring contains no form of security because the destination of the data is held in the minipacket and this travels the entire ring before the data field is overwritten; thus all nodes could read the data if the station logic was altered. The advantages of the Ring are that its SNR and BERs are high and low respectively (provided the twisted pair cable is correctly shielded) and the data loading is distributed across the network to the nodes which require it - this capacity is lowered as the number of minipackets on the ring are increased. The data rate of the ring is further decreased by the large control overheads in each minipacket (only 16 of the 40 bits are for data) thus its equivalent data rate is comparable with the point-to-point rates of the SS-LAN.

3.6.3 FDMA Systems

The FDMA system for comparison will be an FDMA/ALOHA network (38) in which a central controller allocates frequencies to requests for transmission made by other nodes. The number of frequencies available is dependent on the channel bandwidth and the intermodulation effects at the receiver - when this is coupled with the frequency of the noise it defines the SNR and BERs for the FDMA system. Many of the properties between this and the SS-LAN are common eg parallel processing, variable data loading and contentionless access however there is a data throughput delay due to the frequency request mechanism and the system will fail if the controller fails. The most significant reason for using the SS-LAN in preference to the FDMA/ALOHA system is that the SS-LAN can support more simultaneous users when compared on similar terms, as shown by Elhakeem et al (38). The SS-LAN requires different codes using a single carrier whereas the FDMA/ALOHA technique requires different carrier frequencies but these will provide a higher data rate for the point-to-point transmission.

3.6.4 Token Passing Systems

The most significant difference between these and the SS-LAN is that no node can transmit data unless it possesses the token hence access to the channel is obtained on a cyclic basis. When a node acquires the token its data throughput is set as the transmission rate, which is 4MHz in the IBM token ring. Data delay is therefore higher in the token ring due to the access time. The SNR gives a low BER because the noise environment is fixed; there are no simulutaneous users. Both systems are decentralised and the token passing can be recovered, should the token become lost, by implementing the recovery algorithm which is independently located in each node. There is no natural data security hence data must be encrypted before transmission and error correction must also be applied though this usually takes the form of a CRC due to the low BER. The advantage in employing a token system is that it provides a maximum guaranteed data access time for each node however this is not an advantage over the SS-LAN. The significant advantage of the token system compared to the SS-LANs is the higher point-to-point transfer rate <u>once</u> transmission has actually started.

3.7 Conclusion

The SS-LAN is a bus topology LAN which employs DSSS modulation at the physical layer. The use of the code sequences endow the LAN with many of the properties of SS radio systems: primarily that of noise immunity which enables a simultaneous multiple user environment. The simultaneous multiple user property provides a totally unrestricted access mechanism to the channel for all of the nodes and this provides a solution to the time restriction implicit in most of the present LANs. The other properties of the SS-LAN, such as data privacy and decentralisation etc, can be "mimicked" by many current LANs however they are an intrinsic aspect of its operation and are not appended as and when needed. This does not mean that all of the system should be this flexible, but that the SS-LAN can be configured to emphasise the desired properties, using a single LAN version, for a wide range of applications. The application of CDM to fibre optic based systems is currently being investigated by Healy (51) who has created a special code family to overcome the two level signal amplitude restriction and in the future this technique may be applied to the SS-LAN.

The LAN can also be configured in a number of logical operational modes which reflect the use of the network. Again a single incarnation of the nodes can perform these different modes with the actual configuration being defined by a network manager, which is distributed within all the nodes on the LAN. These modes can reflect the data structure, the functional organisation etc and can provide a general communications environment - each contains many of the general properties of the SS-LAN but some are enhanced while others are diminished.

The SS-LAN compares favourably with LANs of other formats eg Ethernet and Cambridge Ring, when compared in a qualitative level however it is not without some drawbacks. The most severe limitation is the point-to-point data rate which is lower than other networks - this is only true once data transmission has successfully occured - and must be traded with the number of simultaneous users which the network must support. A further complication is that concerning the individual node clock stabilities. The relative clock drift must be limited else the BERs, become unacceptably high. In conclusion the SS-LAN provides an excellent alternative system for applications where there is a time critical and time demanding environment or where there is an extremely hostile noise environment with low data rate requirements.

CHAPTER 4

Code Sequences for a DS SS-LAN

4.1 Introduction

Chapter two showed how the code sequences in a DSSS system must possess high differential ACF coefficients (high in-phase and low out-of-phase values) and low CCF coefficients to enable reliable demodulation and reduce interuser interference respectively. Chapter three then described how the code sequences could be mapped to the logical address of the LAN to provide a variety of logical to physical addressing mappings, each of which would suit particular applications. The conclusion from these descriptions is that the choice of the code sequences is of fundamental importance to the operation of the LAN. This is not a unique situation - the choice of sequences is important in encryption, testing and CDMA systems in general and it is not uncommon for code sequences to be created with specific properties for a particular application. This chapter will look at the field of code sequences in general and will then choose and analyse the properties of codes which are regularly employed in CDMA and SSMA systems; such as pseudorandom sequences, composite sequences, Gold codes, Bent sequences etc.

The next section will discuss the properties required for a code sequence family before it can be considered for use in the SS-LAN, with the following section describing a selection of code families which fit these criteria. The final two sections will compare the code sequence families and their properties and will then describe the types of applications for which each family could be used. Whilst the selection of the code families is not exhaustive it will be representative of the families about which open material is available, however it is almost certain that several suitable codes exist in the classified material, to which access was not possible.

4.2 Desirable properties of the Code Sequences

Chapters two and three demonstrated the importance of the code sequences with the conclusion that they must possess high differential ACFs and low CCFs. The codes require other properties also such as a good linear spreading function, being a member of a large orthogonal family and providing a simple programmable generation method. Whilst these are a necessity for the operation of the SS-LAN other properties are also desirable for specific applications eg long linear spans to reduce the likelihood of intentional security violation of the codes, but only the former list of properties will be discussed.

4.2.1 High Autocorrelation Coefficients

Autocorrelation is the comparison of a signal with a time shifted version of itself. Consequently for a binary sequence it should consist, for our purposes, of a two level function where the in-phase value is unity and the out-of-phase value is zero. The variation of the phase and ACF coefficients is given in appendix two where it is shown that the aperiodic ACF is modified by the modulation of the data onto the carrying SS code sequence. The conclusions to be drawn from this analysis are:-

(a) The periodic ACF should have a high in-phase value (unity) and a low out-of-phase value (zero if possible).
(b) The odd ACF should also have a high in-phase value (unity) and a low out-of-phase value (zero if possible).

4.2.2 Low Cross-Correlation Coefficients

The CCF is the measure of similarity between two different signals. As shown in appendix one it is this function which represents the mutual interference in the simultaneous multiple user environment. As the value of the CCF increases then the synchronisation performance at the receiver will be degraded ie the resulting bit errors will lower the maximum achievable ACF. Welch (53) has shown that there is a theoretical non zero lower limit to the CCF for any two codes. This means that there will ALWAYS be some mutual interference and consequently there is only a finite number of simultaneous users which can be supported by the system. Appendix two provides a brief analysis of the variation of the CCF against code phase and data contents with the following conclusions:-

(a) the individual periodic CCFs should be as low as possible (the Welch bound) for all phase shifts.
(b) the individual odd CCFs should be as low as possible (the Welch bound) for all phase shifts.

These will produce a minimised noise environment which will therefore permit more simultaneous users for a given noise threshold.

4.2.3 A Linear Spreading Function

SS modulation performs spreading of the data signal across a bandwidth defined by the frequency of the code sequence. Ideally the spread signal should take the form of AWG noise for two reasons:-

(a) The channel capacity is bounded by the noise environment of the channel. In non AWGN the capacity equation is also dependent on the noise entropy, as opposed to just the average noise power and therefore the channel capacity is increased ie the noise is tending more towards a distortion form of interference which, theoretically, can be removed.
(b) The power spectrum of the code sequences has been described as composing of two parts: the ideal sinc curve and the shaping function formed by the overall code sequence. A nonlinear shaping of the spectrum will produce dominant frequencies which destroys the ressemblence to AWGN and which therefore lowers the number of simultaneous users that can be supported by the system.

The spreading linearity of a code is a spectral consideration (ie in the frequency domain) of the CCF which is a time domain entity. The more linear the spreading the lower the CCF and vice-versa.

4.2.4 Ease of Generation

This topic is concerned with a more practical problem for the SS-LAN. In a realistic system the maximum number of nodes which must be supported is apporoximately 100 (these need not be simultaneously active but may still require unique codes) and so there must be at least 100 codes (depending on the distribution mode). Each of these codes must be generated from a standard code sequence generator which is configured according to specific code sequence instructions. Generation of the sequences in software is preferable because this enables a flexible generation method without physical changes to the node. The requirements for a simple generation scheme are:-

(a) the code sequence family should be as large as possible with each node possessing the specified properties.

(b) Generation of the individual codes should be possible by implementing a standard technique used in conjunction with specific code sequence parameters.

4.3 Code Sequences for use in SSMA Systems

The theory of coding is well developed and is based upon set and group theory. The mathematical nature of the codes for discussion will not be developed (Lin and Costello (54) have written an excellent reference text for code theory) and the properties of these codes will be stated with appropriate references. The naming of code sequence types vary from text to text but for this work combinational sequences will be defined as described by Milstein and Ragonetti (55) and composite sequences as described by Beale and Tozer (56). Composite sequences are also known by the names concatenated (Maskara and Das (57)) and syncopated (by Dixon (15)) and the difference between these and the combinational sequences (eg Gold codes) is that they maintain the information concerning the codes from which they were generated - this is due to the frequency mismatch between the inner and outer codes.

4.3.1 Pseudorandom Sequences

All of the codes sequences for discussion can be termed pseudorandom but in the present context this name will refer to sequences which are generated via a single shift register with appropriate combinational logic. It is possible, as shown by Shaar (58), to generate all code sequences from a single shift register of variable length and feedback taps but in this context it will be assumed that only m-length PN codes are generated from this single shift register structure. For a shift register of length, n, this provides a maximum length sequence of 2^n-1 chips, termed a "linear maximal length PN code" in the literature, which is generated by feeding back various terms of the shift register. M-length PN codes possess several of the ideal properties - all of which have been extensively analysed in the literature (these types of codes are useful as test sequences for error correction). Dixon (15) provides a compact defintion of m-length PN codes properties where:-

(a) The ACF is a two step function due to the single one/zero imbalance.
(b) Simple modular programmable generator.
(c) Poor CCF - limiting the number of simultaneous users.
(d) Short linear span - providing a non secure code.
(e) A relatively small family size for each sequence length.

These sequences will be used as the reference against which all the other codes will be compared; their properties are well known and understood. Two further types of code can now be derived, from the basic shift register generation technique, which are closely related to the linear m-length PN codes. The first are non-linear PN codes which are an attempt to produce cryptographically secure codes ie codes which have long linear spans. These codes can be produced by the addition of combinational logic to the outputs of each shift register and by using these connections to produce the output from the generator (the m-length equivalent was from a single shift register element).

There are several drawbacks to the generation of non-linear codes which result from this complex feedback connection. The production of long sequences is difficult due to the problem of degenerative states ie the all zero state in the shift registers, at which point the sequences become constant. Associated with this is the small family size of the codes which can be generated from a fixed size shift register; consequently they cannot be employed in CDMA systems where a large number of codes are necessary. In the context of the SS-LAN however they are ideal for the transfer of secure data thereby enhancing the existing system security. Unfortunately little information is available in the open journals concerning non-linear codes and so figures for autocorrelation and crosscorrelation coefficients cannot be given.

The second type of code produced by a single PN generator are called "punctured PN" codes and are defined by Han and Hemmati (59). These codes are an attempt to lower the high CCF coefficients produced in m-length PN codes by removing the chips which cause most of the mutual interference ie the PN codes are deliberately punctured. The punctured code is used only in the receiver and Han and Hemmati have shown that the most effective environment for the use of these codes is one in which only short codes are required and where they also have high cross correlation coefficients. The effect of this puncturing is to increase the number of simultaneous users. Unfortunately, at the present time there is no fully documented list of the punctured codes and their generation algorithms.

4.3.2 Gold Codes

Gold codes (60) are the generally accepted choice for CDMA systems and are a class of the combinational sequences defined earlier. They can be generated by the modulo two addition of two m-length PN codes producing a non-maximal length PN code of the same length as the two generating codes. The difference in phase between the two source codes defines the particular Gold code to be generated thus if they are both 127 bits long then there are 127 Gold codes which can be generated from the combination of the source codes. Gold codes produce a large family size for a given length, essential for large CDMA systems, but what are the coefficients for their ACFs and CCFs? Both of these correlation functions are well defined for each code sequence length and, as will be shown later, are superior to those of similar length PN codes. Gold codes cannot be used in secure systems because they have a small linear span producing codes which can be decrypted after the reception of a relatively small number of chips.

The other classes of combinational sequences can be generated by altering the algorithm by which the two codes are multiplexed. One method is to use the Kronecker product matrix producing Kronecker codes (61) (these codes facilitate rapid synchronisation because acquisition can be achieved over a subset of the code chips) or the Hadamard product matrix etc. In fact this principle may be extended to include the combination of more than two codes accompanied by a suitable algorithm designed to generate codes with particular properties. It is this flexibility which provides such a vast number of codes which, if they do not contain the information to reproduce their component codes, are classed as combinational sequences.

4.3.3 Composite Sequences

There are many classes of composite sequence; defined by the method by which they multiplex the root codes. They must however still maintain the information to generate their root codes. The Beale and Tozer method (56) employs sequence inversion keying (SIK) of one code by another - SIK is the same as the modulo two addition in the DSSS modulation technique. The root codes employed are m-length PN sequences which limit the performance of the composite code. In general, for the same root codes there are twice as many composite sequences as combinational sequences but this does not mean that composite sequences will necessarily be capable of supporting a large number of simultaneous users in a practical SS system. Beale and Tozer do not provide a detailed list of their combinational codes and their generation indices however they do provide a means by which an exhaustive test of codes may be analysed in an attempt to find those with the minimum CCFs and highest ACFs.

4.3.4 Bent Sequences

Bent sequences are an example of a code which has been constructed so that it possesses certain characteristics: in this case the primary property is one of a long linear span which is necessary for a cryptographically secure code. The definitive work on Bent sequences for use in SS systems is provided by Olsen, Scholtz and Welch (62) with modifications to this work supplied by Lempl and Cohn (63) in which the correlation properties are found to be dependent on the pairwise orthogonality of the bent functions: a bent function is a function in which the fourier coefficients of the polynomial are +1. The properties of the Bent sequences are:-

(a) the sequences are balanced ie a difference of one between the number of ones and zeroes.
(b) large linear span equivalent - due to the nonlinearity of the codes.
(c) easily programmable using a standard generation technique.
(d) three valued ACF and CCF coefficients with clearly defined maximum.

The claim for Bent sequences is that they provide almost ideal CCF properties while remaining cryptographically secure and can be generated by hardware whose complexity is similar to that of Gold code generators: if this is true then they would appear to be the ideal sequences for the SS-LAN.

4.4 Comparison of the Code Sequences

A brief comparison of four types of code sequences will now be made. This is not an analysis of the properties themselves but is a statement of some numerical values that are important to the present work and which are obtained from the literature references (54) to (63). At this point two further papers must also be referenced. The first is a comparison by Shaar and Davies (64) of six code sequence construction techniques and their aperiodic CCF coefficients. This paper presents a quantitative review of the chosen constructs in an attempt to collate their analysis in a single review. The second is concerned with the computer analysis by D.Lin (65), a Durham graduate, of several code constructs and their parameters. His computer programs provided verification of the theoretical values presented in figure (4.1) and the author is indebted to him for this section of the thesis.

Figure (4.1) is the comparsion between four types of sequences for ten different criteria. For each criteria there is a statement of a

CRITERIA	M-LENGTH LINEAR PN (n=8,n=12)	GOLD CODE (n=7,n=10)	COMPOSITE SEQUENCES (a=4,6 b=5,7)	BENT SEQUENCES (n=8,n=12)
LENGTH (L)	2^n-1	2^n-1	$(2^a-1)(2^b-1)$	2^n-1
n=low	255	127	465	255
n=high	4095	1023	8001	4095
FAMILY SIZE	$1/n\Phi(2^n-1)$	2^n-1	$1/ab\Phi f(a)\Phi f(b)$	$2^{n/2}$
n=low	16	129	12	16
n=high	144	1025	108	64
SEQUENCE IMBALANCE	1	$1\rightarrow 2^{(n+1)/2}+1$	2^a-1 or 2^b-1	1
n=low	1	7	15,31	1
n=high	1	65	63,127	1
MAXIMUM CCF	CD	$1+2^{(n+2)/2}$	CD	$1+2^{n/2}$
n=low	35	17	-	17
n=high	1407	65	-	65
MAXIMUM ACF (in,out phase)	$-1,2^n-1$	$1+2^{(n+2)/2},2^n-1$	CD	$1+2^{n/2},2^n-1$
n=low	-1,255	17,127	-	17,255
n=high	-1,4095	65,1023	-	65,4095
CORRELATION PARAMETER				
n=low	35	17	-	17
n=high	1407	65	-	65
WELCH BOUND				
n=low	15.5	11.2	20.6	15.5
n=high	63.8	32	89	63.4
INDEX OF DISCRIMINATION	2^n	$2^n-2^{(n+2)/2}-2$	CD	$2^n-2^{n/2}-2$
n=low	256	120	-	238
n=high	4096	958	-	4030
SYNCH TIME	2^n-1	2^n-1	2^a-2^b-2	2^n-1
n=low	255	127	46	255
n=high	4095	1023	190	4095
LINEAR SPAN	n	2n	a+b	CD
n=low	8	14	9	-
n=high	12	20	13	232

CD = Code Dependent

FIGURE (4.1): Comparison Table of the Code Sequences general definition accompanied by numerical results which are presented for both a short code sequence and a long sequence (these lengths are not standard across the four code types). The linear m-length PN code is used as the reference and these are also employed as the source codes for the Gold codes. The limitation on the shift register length for Gold codes is that n=mod 4, as specified by Gold (60). For the composite sequences the root codes are m-length PN sequences also and two of them are used to generate each composite code. For this analysis it will be assumed that a>b with the restriction on the general analysis that a $\neq$ b. For the Bent sequences the shift registers must be of the length n=mod 4 (62).

The chosen criteria provide a comparison base for the properties of the codes and their performance characteristics. The "length" is the code sequence length, the "family size" is the number of orthogonal code sequences that can be generated with the fixed length and the "sequence imbalance" is the diffence between the number of ones and the number of zeroes in the generated code sequence. The sequence imbalance is a guide to the randomness of the sequences (the internal order of the bits is of equal importance in the construction of random like codes) and should therefore be as close to unity as possible (for odd length codes).

The "index of discrimination" is the difference in amplitude between the first (in phase) ACF peak and the second (highest out of phase) ACF peak. Ideally this difference should be equal to the code sequence period or else the effects of noise could cause equivalence between the first and second (or even lower) ACF peaks. The "maximum CCF" is the highest "non-normalised" correlation coefficient between two codes of the same family and length - this should be as close to the Welch lower bound as possible. The "ACF" represents the two criteria of the in-phase value and the maximum out-of-phase values.

The most frequently used criteria by which code sequence families are compared is termed their "correlation parameter" (58). This is a measure of their most significant interference component and consequently this is a comparison between the maximum out-of-phase ACF coefficient (considered for all members of the family) and the maximum CCF coefficient (considered for all pairings in the family). The family with the lowest correlation parameter is deemed the "best" family. A further measure of the performance capability of a family is the comparison between its correlation parameter and the Welch lower bound - as defined by equation (4.1) in which L is the code sequence length and f is the family size.

$$\text{Welch lower bound} = L\left(\frac{f - 1}{Lf - 1}\right)^{1/2} \qquad (4.1)$$

The Welch lower bound is the theoretical minimum value for the CCF of any family given a fixed code sequence length and family size. If a family is found to have a correlation parameter which is equal to the Welch lower bound then another family with a lower value for the same criteria will not be found. The Welch lower bound has been found to be a weak limit for binary sequences and consequently it has been superceded by the Sidelnikov bound which provides a tighter lower bound however this analysis will refer to the more commonly quoted Welch bound. Figure (4.1) has separated the CCF and ACF coefficients in order that the clarity of the comparisons can be maintained , particularly when these values can be related to the analyses shown in appendices one and two.

The "synchronisation time" is the number of bits required before synchronisation can be achieved. The final criteria is the "linear span" and this is a measure of the number of code bits required before the rest of the sequence can be accurately predicted.

The length of all but the composite sequences are the standard maximum for an, n, bit shift register. The composite sequences are slightly shorter for comparable register sizes. However, the family sizes differ dramatically with the Gold codes offering the largest family and the Bent sequences offering the smallest. When the code sequence balance is analysed it is the Gold codes which provide the worst case with both m-length PN and Bent sequences providing the best values. A similar trend is noted when the "index of discrimination" is evaluated with the m-length PN codes providing the highest value once again. The index is not yet available for the composite sequences but it has been shown to be dependent on the partial aperiodic autocorrelation functions (56).

Both Bent sequences and Gold codes produce correlation parameters which are close to their Welch lower bounds and consequently their mutual interference approximates towards the lowest achievable limit. Bent sequences produce an almost perfect value. In contrast m-length PN codes produce a correlation parameter which is signficantly larger than the Welch bound and therefore their codes have a relatively large mutual interference. The difference between the in-phase and maximum out-of-phase ACF coefficients is represented by the index of discrimination and it can be seen that the m-length PN codes produce the most distinct index value with the Gold codes producing a significantly lower value.

In general the synchronisation times are the equivalent of the sequence length, except in the case of the composite codes; these sequences were created for rapid synchronisation hence their general application to "ranging" systems. This rapidity is obtained due to parallel correlation which must be applied to the two codes, giving rise to the three levelled ACF. If there are three source codes then three correlation values are required and these are accompanied by four ACF levels etc.

The linear span demonstrates the non-secure nature of the other codes when compared to the Bent sequence. The Bent sequences can be constructed to have particular linear spans, depending on their degree (three in the example shown), limited by the maximum as defined by the shift register length. The other codes are not secure and the reception of only a few chips is necessary before the entire sequence may be predicted - if the m-length PN sequence is 127 bits long then only 8 bits are required for the accurate prediction of the entire code sequence.

In general, the Bent sequences provide the best performance parameters, particularly if a secure implementation is required, however their family size is a severe limitation to the size of network which they can accomodate. The Gold codes appear to be the ideal codes for systems in which a large family size is required but where the maximum number of simultaneous users is non-critical. Their application to secure systems would require the inclusion of data encryption because SS modulation using these codes does not provide a cryptographically secure environment. Unfortunately the application of m-length PN codes is limited - more is known about these sequences than of any other type - and their only significant advantage is their index of discrimination value. This coupled with their poor correlation parameter implies that m-length PN codes should only be employed in systems where the electrical environment is extremely hostile and where there are relatively few simultaneous users. Unfortunately there is little numerical information available on combinational codes however their application would appear to be limited unless the time for synchronisation is the critical factor.

4.5 Application of the Codes to the SS-LAN

The applicability of the codes to each of the six SS-LAN operational modes will now be discussed. The unique receiver and unique transmitter modes are similar in their requirements of the codes ie family size large enough for unique node allocation, low CCFs to permit large numbers of simultaneous users and a high index of discrimination. Clearly only Bent sequences support this specification.

The node group zone and the functional format also form a pairing of similar operation; communication is logically localised with no interaction between nodes external to the logical group. In the case of the functional mode the required number of nodes will be lower, in general than for the unique allocations due to the broadcast operational nature of logically ordered groups. Bent sequences are therefore the most attractive codes for implementation providing low CCFs which would facilitate long term open virtual link environments. The node group zone format is difficult to assimilate because of its subnetwork nature. If the network traffic is likely to be low then the Gold codes are ideal because they provide a large number of codes which can be liberally allocated to each of the subnetworks. Privacy of communication is therefore maintained and the codes can be organised in different ways within each zone, depending on the function of the zone.

The last pairing is that of the message mode and the protocol mode.

In each of these there is a requirement for a large number of message types, with a potentially small message length and where the individual node traffic will be low compared with the network load. In this situation the Gold codes provide all the properties as long as the number of simultaneous users is kept within the strict system limit; in this datagram type system the Gold codes permit a large number of codes, using the minimal sequence length (therefore highest point-to-point data rate), while still providing a simultaneous multiple user environment.

The lack of information on Composite sequences means that their applicability to any of the modes cannot be ascertained and before this would be possible their correlation parameters must be determined. M-length PN codes have no long term application within the structure of the SS-LAN because they do not provide a simultaneous user environment which can support a large enough number of users. They can however be employed for the calibration of the SS-LAN because they have well founded properties by which the performance of a practical system can be determined.

4.6 Conclusion

The selection of the types of code sequences for the SS-LAN has been discussed in this chapter. The analysis in chapter two highlighted several important properties required from the codes:-

(a) High in-phase ACF coefficients - unity if normalised.
(b) Low out-of-phase ACF coefficients - zero if possible.
(c) Low CCF coefficients - as close to the Welch bound as possible.
(d) A linear spreading function ie high noise entropy.
(e) Ease of generation of a large orthogonal family.

A selection of codes commonly used in SSMA were then described and compared with one another, using a list of criteria which reflected the above properties. From this list it was clear that Bent sequences offered the most flexible codes that could be used in many different types of systems. The primary advantage of these over Gold codes was their long linear span which is essential in cryptographically secure codes. The dominance of these codes was reflected in their application to the unique allocation modes of the SS-LAN however it was noted that for systems which were either datagram oriented or in which a large number of sequences were necessary then Gold codes were best suited because of their large family and corresponding short sequence length. M-length PN codes were found to be useful in systems which had only a small number of users where the noise environment was extremely hostile. There was insufficient data to fully discuss Composite sequences and these should become a topic for further investigation.

There are many other types of code sequences which must eventually be considered for use in the SS-LAN. Their selection will depend upon their applicability to particular systems and so the system realisation must provide a flexible code sequence generation method. One final point is that as yet no analysis has been made of the effects of hybrid SSMA systems ie one in which different code sequence families are distributed across the system. In the SS-LAN this introduces an extra degree of flexibility because the node group zone format can now implement code families according to their logical function instead of relying upon a single family. This also applies to some of the other modes and so hybrid SS-LAN systems will have to be analysed at a later stage of development.

CHAPTER 5

The System and Node Hardware Specification

5.1 Introduction

The previous chapters have described the logical operation of the SS-LAN whereas this chapter will present the specification of the system as a unit as well as that of the nodes themselves. This network will act as the communications layer for a larger system eg the control system for a ship and as such its interface must be flexible to reflect the different types of applications to which it it will be connected. Not only must the SS-LAN interact with other hardware and software it must also maintain its own integerity and this requires interaction of the nodes independent of user requirements. The design of the system and the nodes does not include the production of the circuit diagrams however there is sufficient detail to provide a basis for both an accurate theoretical analysis and a computer simulation of a simple configuration (these will be presented in following chapters).

Design standards for system and hardware specification, which rely upon the formal specification of the requirements, are currently being developed in an attempt to produce a more predictable design methodology. The SS-LAN design has however been implemented using a top down methodology as introduced by Comer (66) and is presented as such in the rest of this chapter. The next section describes the requirement specification followed by the functional and design specifications. The final section is concerned with the validation of the design with respect to the previously defined requirements.

5.2 Requirement Specification

The hardware requirement specification will define the facilities which are desired by the user and network manager in the context of system use and system maintainence respectively. This specification is in two sections: the first defines the overall system and the second concentrates on the nodes themselves.

5.2.1 The System Requirements

The SS-LAN will be of bus topology and will maintain at least 100 nodes over a maximum distance of 2kms using a passive channel; this is in line with current LAN physical specifications. There will be NO central controller, monitor or clocking system and communications between nodes will only occur using the communications channel. Nodes will not be required to transmit data to different types of networks as a matter of course however internetworking will be available through specialised dedicated "gateways" and linkage to other SS-LANs will be via specialised dedicated "bridges". The nodes on the network will not necessarily all transmit data from the same code family but the type of codes in use should be transparent to the general operation of the system.

The SS-LAN should permit the interfacing of a range of unspecified computers and devices where several may be connected to a single node and which may not necessarily operate sequentially or in co-operation with one another. The failure of an individual node should not cause a catastrophic failure of the system and once a node failure has been detected the network should logically reconfigure itself so that communication to the failed node is impossible until a resubmission request has been received from that node. The system should also aid with self error reporting and should provide a system self test capability to detect the occurrence and position of failed nodes.

Addressing at the physical layer is via code sequences, thus the individual transmitters and receivers require only the information to generate codes. The operational mode of the codes is programmable and may be changed at any time with a request to the network management supervisor which must be capable of implementing a validated request of this type. The operational mode mapping is a high level organisation and as such will not be changed frequently thus the SS-LAN may be reinitialised for this static change - no data should be lost. In the cases where the logical to physical mapping is other than by a logical node identifier then the destination node information will be carried at the data link protocol level and not as part of the physical layer information ie there is no physical address as in the accepted sense for LANs.

5.2.2 The Workstation Requirements

An individual logical position on the SS-LAN is defined as a workstation. Each workstation will consist of one, or more, nodes and each workstation may support one or more devices each of which may use more than one node; this is to provide a parallel processing capability for individual devices. The nodes will permit instantaneous access to the channel with unlimited duration and a data loss may only occur when a parallel processing system is in operation and is implementing a priority mechanism. The loss will only be temporary as all transmissions must be acknowledged with a success of fail status code. Transmissions must be attempted several times with timeout facilities for each failure and comprehensive status codes must be returned via the appropriate protocols. The nodes will also handle low level error detection/correction and will ensure the integrity of the data they receive with no data being lost in the system.

The data rate will be determined by the code sequence length and the code sequence/data bit ratio. Each node will be capable of generating any code sequence family in any phase and of any length. Information flow will be bidirectional for all nodes with the code sequence families not necessarily being the same for a node's receiver and transmitter. The node will remove any physical protocol and will not allow physical layer control information to pass to the higher protocol levels. The nodes will also coordinate the distribution of the codes and will always operate from an accurate configuration map of the network; this map need only be of the local SS-LAN.

Each node will provide loopback test facilities at several levels each of which will totally bypass the lower levels of the system. The nodes will perform self checking at initialisation and will immediately request configuration information from the rest of the network. A test interface will be provided to enable off-node testing and it will also provide status information on the internal operation of the node. The nodes will not contain any information on the purpose of the logical mapping and will only take this data and use it to find the configuration instructions for the code sequences. The network organisation must be maintained by the physical layer but it must be a function distributed between all of the nodes on the network. This must also be the situation for the transfer of any network management information.

5.3 Functional Specification

This is the technical definition for the operation of the whole LAN and the individual workstations. It includes the definition of the input and output protocols and the interface specification for the physical layer of the OSI reference model.

5.3.1 Definition of the LAN System

A block diagram of the logical separation of the SS-LAN is shown in figure (5.1). Each workstation is composed of three parts: the user devices, the station interface logic (SIL) and the station node logic (SNL). The communications channel is a passive bus using standard coaxial cable onto which the nodes are linked via "spur" connections - a node failure should not therefore cause a bus failure. The actual node interface to the channel will be through a multidrop transformer arrangement which isolates the varying node impedance from the characteristic impedance of the channel. The channel will also be "terminated" to stop voltage reflections (these would act as multipath signals to which SS is resistant however the additional noise reduces the number of maximum of simultaneous users).

5.3.2 Functions of the Workstation

The SIL is a specialised interface board designed to convert the general SNL interface to a host specific interface and so the SNL can be designed independently of host devices. The SNL is the engine for the SS-LAN workstation which will contain one or more physical nodes. A schematic diagram of the IEEE 802 and OSI protocol specifications imposed on the SS-LAN workstation is shown in figure (5.2). The rest of

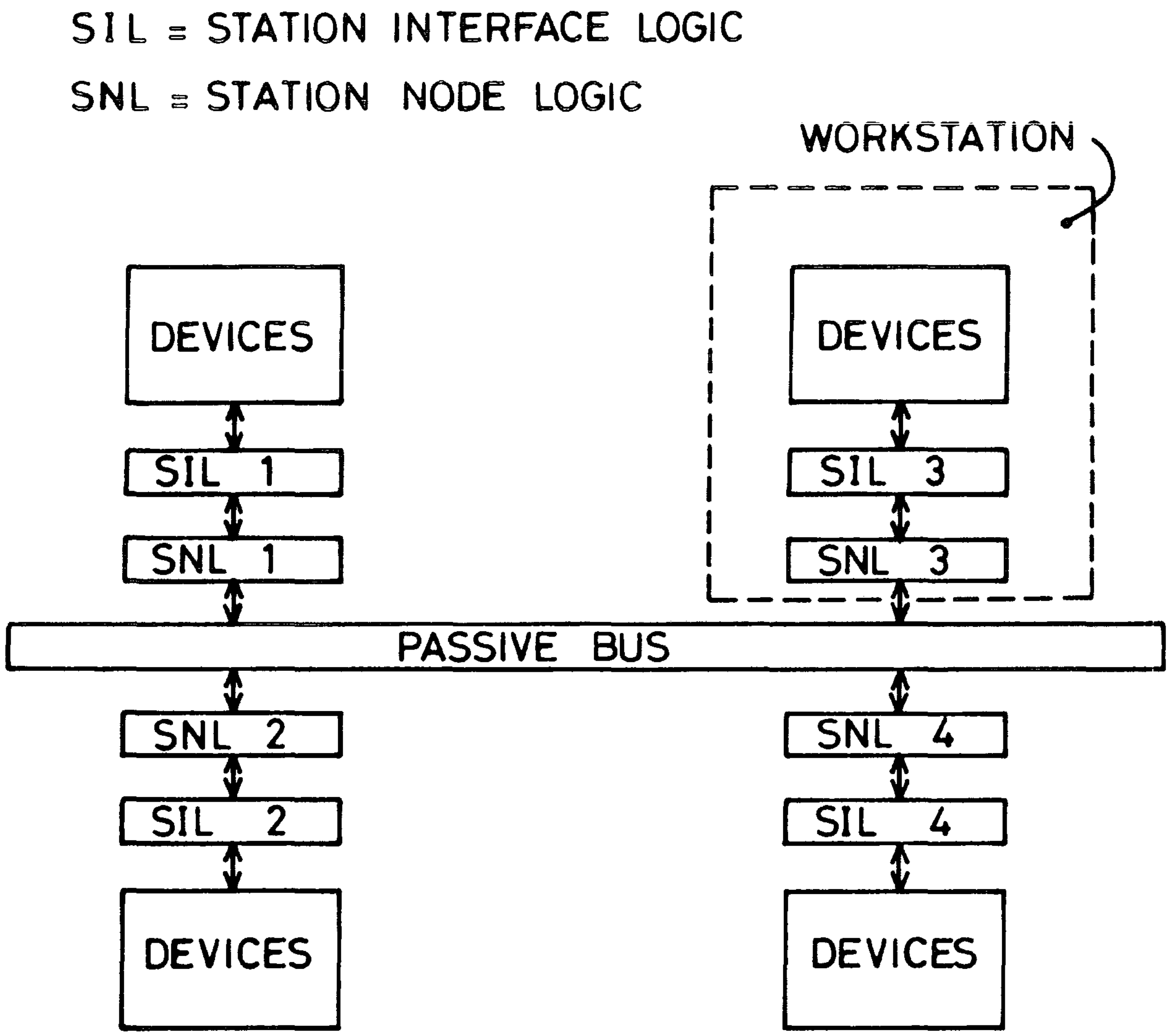


FIGURE (5.1): The SS-LAN System Block Diagram

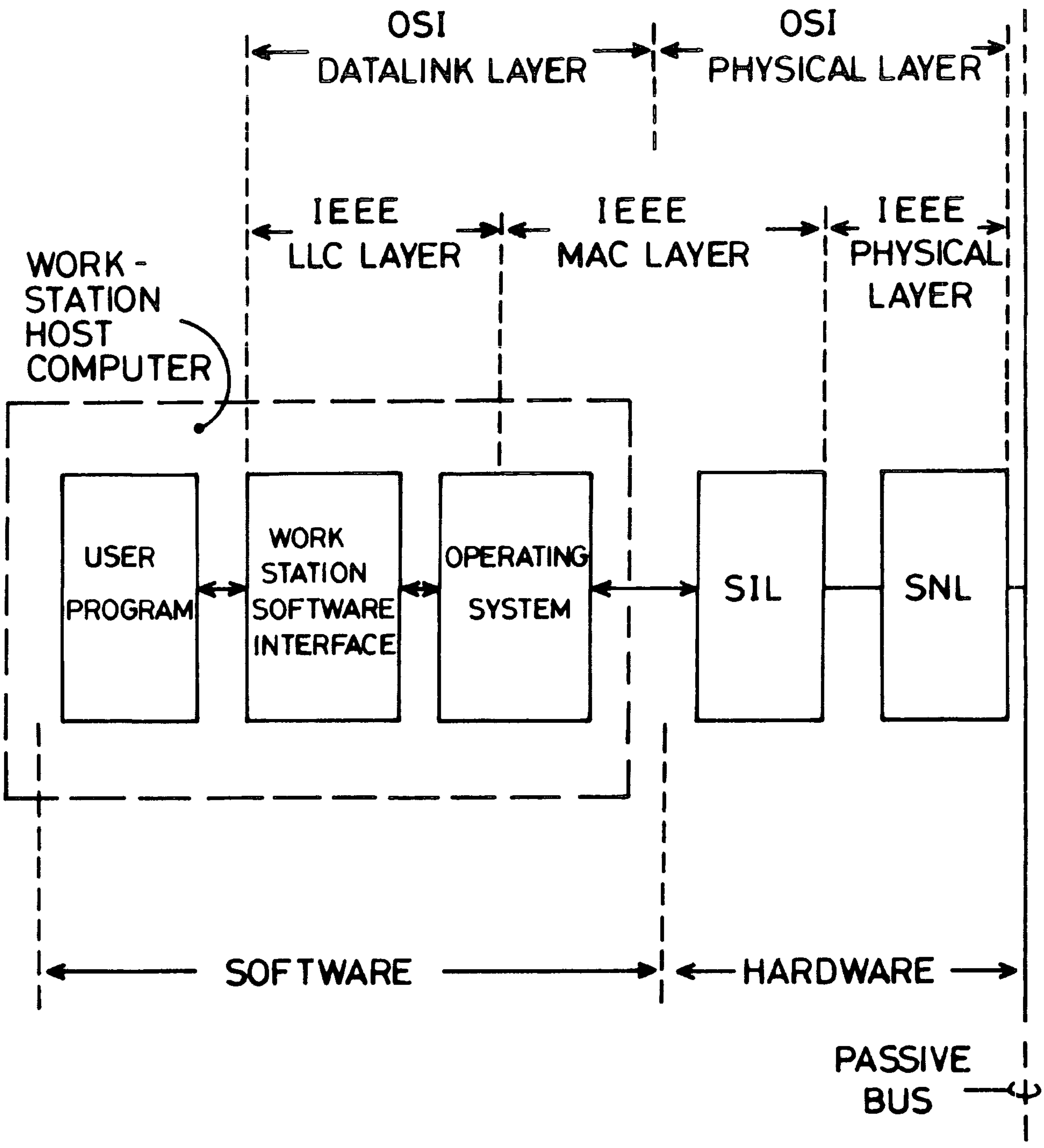


FIGURE (5.2): Workstation Protocol Configuration the function and design specifications will not be concerned with the SIL hence the protocols involved are as defined by the IEEE physical layer specifications.

Each node within the SNL will implement low level failure and error recovery algorithms with a general network management facility driving the nodes and interacting with all the nodes on the SS-LAN. Each SNL will also implement a loop back facility where data will be transferred directly from the node transmitter to receiver - this provides a test in isolation capability. The individual nodes will also perform self test diagnostics which will provide information on the individual components within the node, and this will be instigated on instructions received via the standard node interface.

5.3.3 Physical Layer Protocols

The data format for the transfer of data between the SIL and the SNL is given in figure (5.3). The ISO-OSI and IEEE 802 standards define a more sophisticated equivalent interface for peer to peer interaction and peer to subordinate interaction however the aim at this point is to investigate the general principles of this interface and not its detail: the problems of OSI and its implementations are discussed in a special issue of the Proceedings of the IEEE (67). The information required for the transmission of data is, the logical mapping identifier (maps to the code generation instructions), the length of the message, the data and end of channel (EOD) marker. For data reception the logical mapping identifier is unnecessary (this is contained within the code sequence used for transmission) but the rest of the information is received as transmitted. The figure shows the data in 16 bit words but this would be upgraded to 32 bits if this was to be the standard word size.

Figure (5.4) shows the binary data stream which is passed between communicating SNL nodes. These mesages are shown in their state representation by figure (5.5). In the first instance the transmitter is assumed idle when a user request for transmission is received. The transmitter will generate a "request for communication" and send this to the destination node/nodes using the appropriate code sequence - this request is necessary because the target nodes may be busy which would

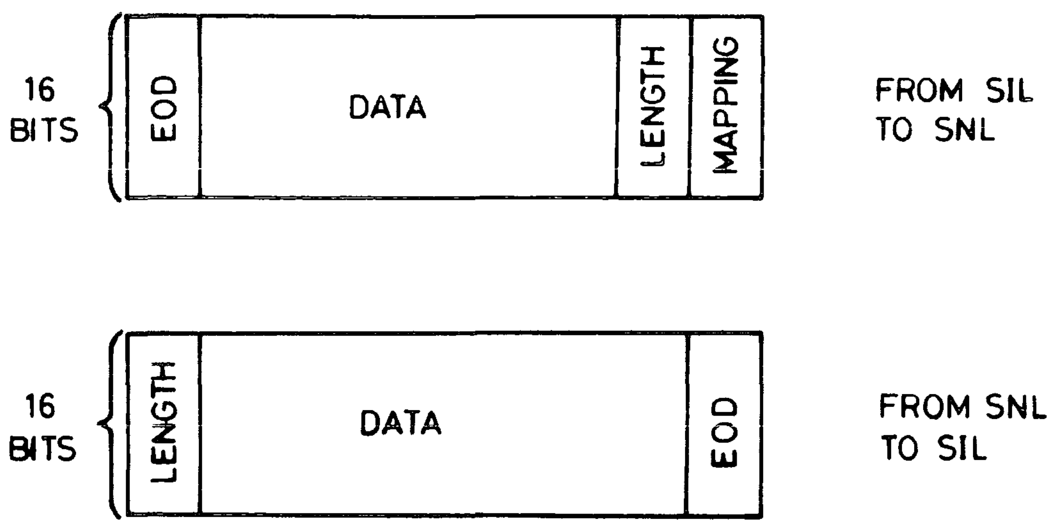


FIGURE (5.3): SNL   SIL Data Format

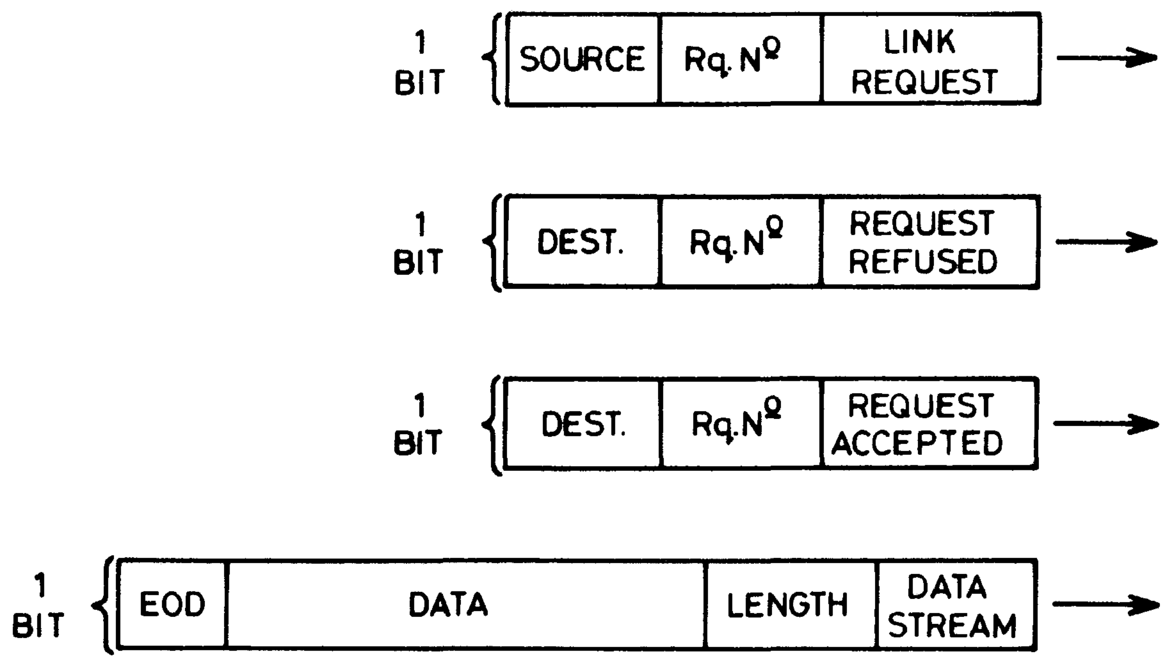


FIGURE (5.4): Binary Stream Protocol

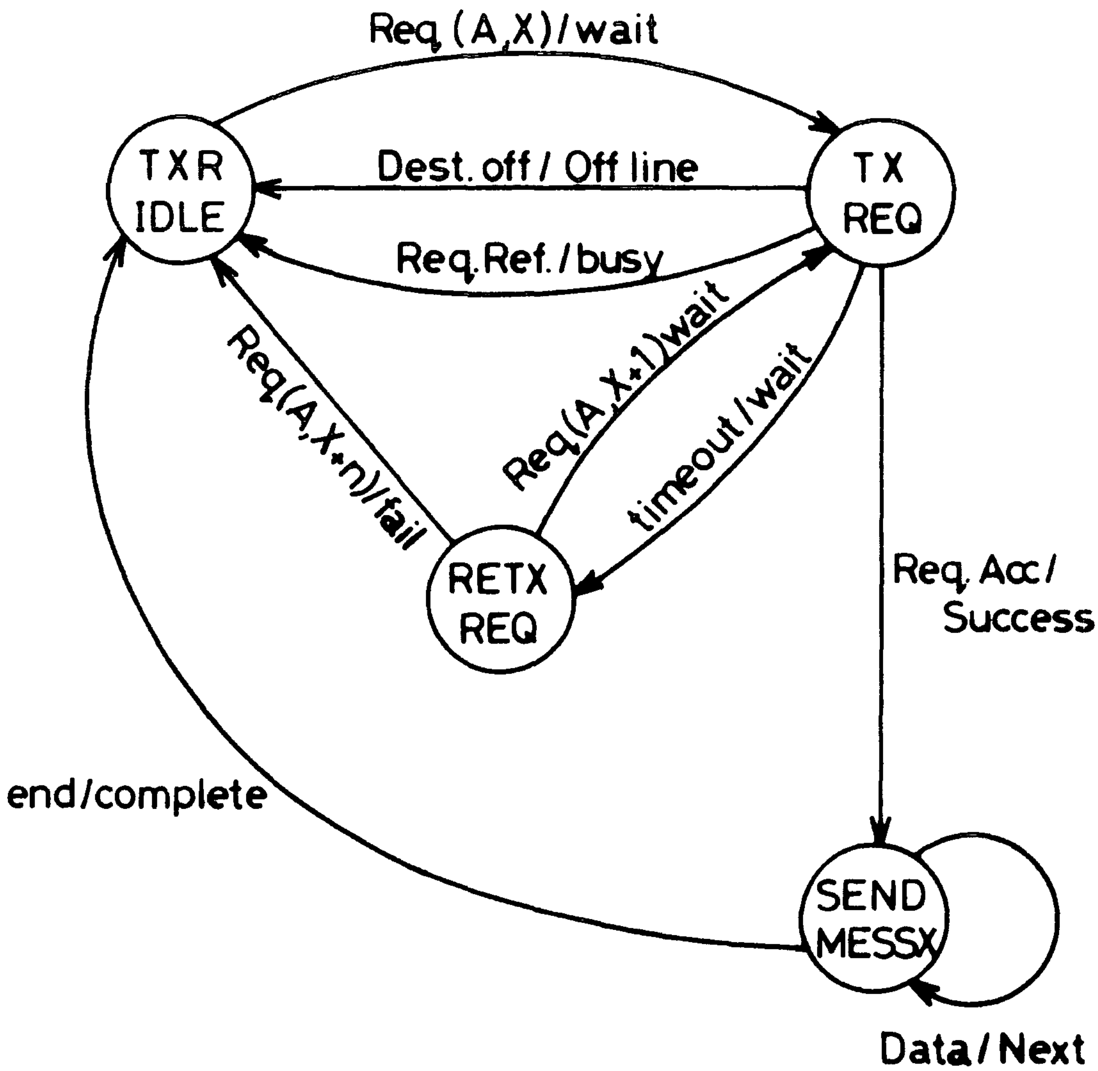


FIGURE (5.5): State Diagram for the Transmitter result in a long message using unnecessary bandwidth. Should the attempted logical to code mapping fail then this is interpretted as the off-line state of the destination node. The transmitter will now move to one of three states. The first is the result of the reception of a "request refusal" code (identified by its destination number and reflected request number) which returns the transmitter to its idle state with the returned states of "destination busy" - the receivers are busy. The second is a timeout whereupon retransmission is attempted using an incremented "request number" and returning to the "transmitter request" state and message. Should the timeout occur a predefined number of times then it is assumed the destination nodes have failed and a fail state is returned to the user. The final state is if any "request accepted" code is received - identified by its request number and source number. The transmitter will then send the data and eventually return to the idle state with a "complete" status for the user.

The protocols introduced here are simple and will require further development (which is outside of the scope of this thesis) however they do show the general nature of the message interaction. The difficulty for all the protocols, at this and all levels, is the broadcast nature whereby one message may go to several destinations. The awareness of just one destination must result in the transmission of the data thus "request refusals" must be stored until some timeout period. Similarly, the control information transmitted within each message has been kept to a minimum however there is no error correction implemented and this is necessary even if it is only present in the form of a checksum at the end of the message.

5.3.4 The Technology Specification

If it is assumed that the code sequence rate is to be as high as possible (as near to 100Mbits/sec as is possible using a passive coaxial arrangement) then the range of available hardware is limited. The codes cannot be generated, in real time, above a frequency of 12MHz therefore they will be generated and stored in ECL RAM. The data is transmited in a serial fashion hence when it is reformed and packaged in word sizes the speed requirement is dramatically reduced. The larger the word size the slower the processing requirements for the rest of the hardware. The "state of the art" microprocessor is the latest Motorola 32 bit machine, the 68020, which is one of the first true 32 bit microprocessors. It is comparatively expensive however this will gradually decrease in the coming years and will therefore become suitable for use in the SS-LAN.

Both ROM and RAM will be necessary to link with the CPU. The ROM could be provided as an EAROM which holds the mapping tables for the code sequence generation instructions. These could be changed but it would be infrequent and the use of an EAROM would permit updates without physical access to the SNLs.

5.3.5 The Interface Specification

Two interfaces must be defined; the SIL/SNL interface and the SNL/bus interface. The latter is the logic 0 and 1 voltage levels and the physical mnodulation system - some LANs use differential Manchester encoding etc. The difference between using a bipolar and unipolar scheme has significant affects on the type of ADC required in the receiver. At this time a simple bipolar NRZ scheme is to be employed ie +ve for logic one and -ve for logic zero.

The signal level in the channel will now be the difference between the sum of the nodes transmitting logic one and the sum transmitting logic zero, with an average value of zero. The threshold will therefore be a hard limiter set at zero and which will act as a simple one bit ADC.

The SIL/SNL interface will provide three types of information; control, data and test lines. The data and test lines will be coincident with the bus action defined by the control lines. The control lines will specify initialisation of nodes, termination of nodes and general handshaking. The test lines will provide as their output, the transmitter code sequence and clock, the receiver code sequence and clock, the modulated code sequence and data sequence and the received binary sequence before correlation.

5.4 The Design Specification

This design does not provide the detailed circuit diagrams however it will describe the general functions of the hardware and will provide a basis for the detailed design.

5.4.1 The SS-LAN System

Figure (5.6) shows a schematic diagram of the SS-LAN composition. Each workstation is split into four sections: the user devices, the SIL, the spread spectrum node controller (SSNC) and the Spread Spectrum Asynchronous Receiver and Transmitter (SSART) - the last two comprising the SNL.

USER DEVICES - Each device may access one or more SSARTs simultaneously but this will be transparent to the device. All devices will access the SS-LAN via the specialised SIL board.

SIL - This will interface user devices to the general SS-LAN nodes or the SNL. It will translate all data formats and will provide bidirectional communication.

SSNC - This will allocate free SSARTs to transmit requests received via the SIL. It will hold and maintain the mapping tables for code sequence generation and it will hold this information. It will also act as a network manager for the codes such that all the SSNCs form a distributed network controller for the SS-LAN. The SSNC has the ultimate control over the SSARTs and will close them down in the event of failure.

SSARTs - Any number of these may be connected to the SSNC subject to the limitations given below. Once an SSART has been allocated to a transmission/reception request then it operates independently of the SSNC. Each SSART can generate any type of code sequence, supported by the SS-LAN, once it has obtained the configuration instructions from the SSNC. The physical layer protocols are implemented within this block including the low level error correction.

The number of SSARTs which can be supported concurrently by the

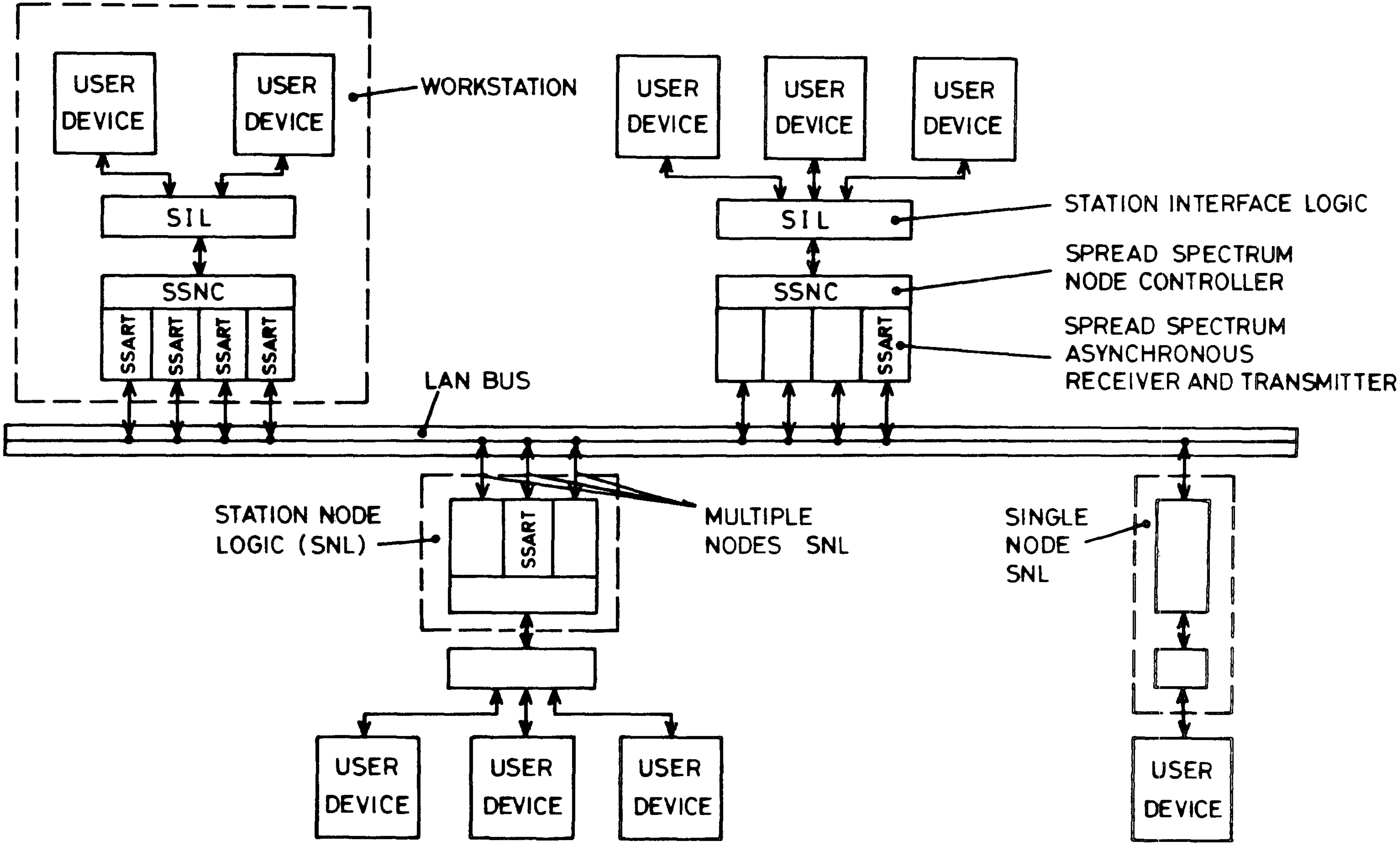


FIGURE (5.6): Logical SS-LAN Configuration

SSNC is dependant on the amount of control information which flows between the two. In general this will be low (unless network information is being transmitted) and occurs only at the initialisation of the SSART linkage to the user request. Each SSART is termed a node in the conventional sense and therefore the workstation can implement a parallel processing scheme - this must be supported and controlled at a higher protocol layer.

5.4.2 The SS-LAN Station Node Logic (SNL)

The next level of design for the SNL is shown in figure (5.7) where the SSART and SSNC have been represented by their constituent modules. The functions for these components are:-

SSNC-CPU: Perform the network management and SSART allocation

SSNC-ROM: Contain the code sequence generation instructions and mapping elements

SSNC-RAM: Contain the status information on the codes across the complete SS-LAN

SSART-CPU: Generate the code sequences and implement the low level protocols

SSART-RAM: Hold the code sequences used for transmission and reception.

SSART-ROM: Store the read only data for the SSART-CPU

TRANSMITTER: Modulo two add the data bits and the code sequence.
Provide test information on the transmitter design.

RECEIVER: Convert the analogue input to the digital equivalent.
Provide coarse synchronisation and tracking (if implemented).
Perform search algorithms for a set of code sequences.
Return raw data to the SSART-CPU.
Provide test information on the receiver action.

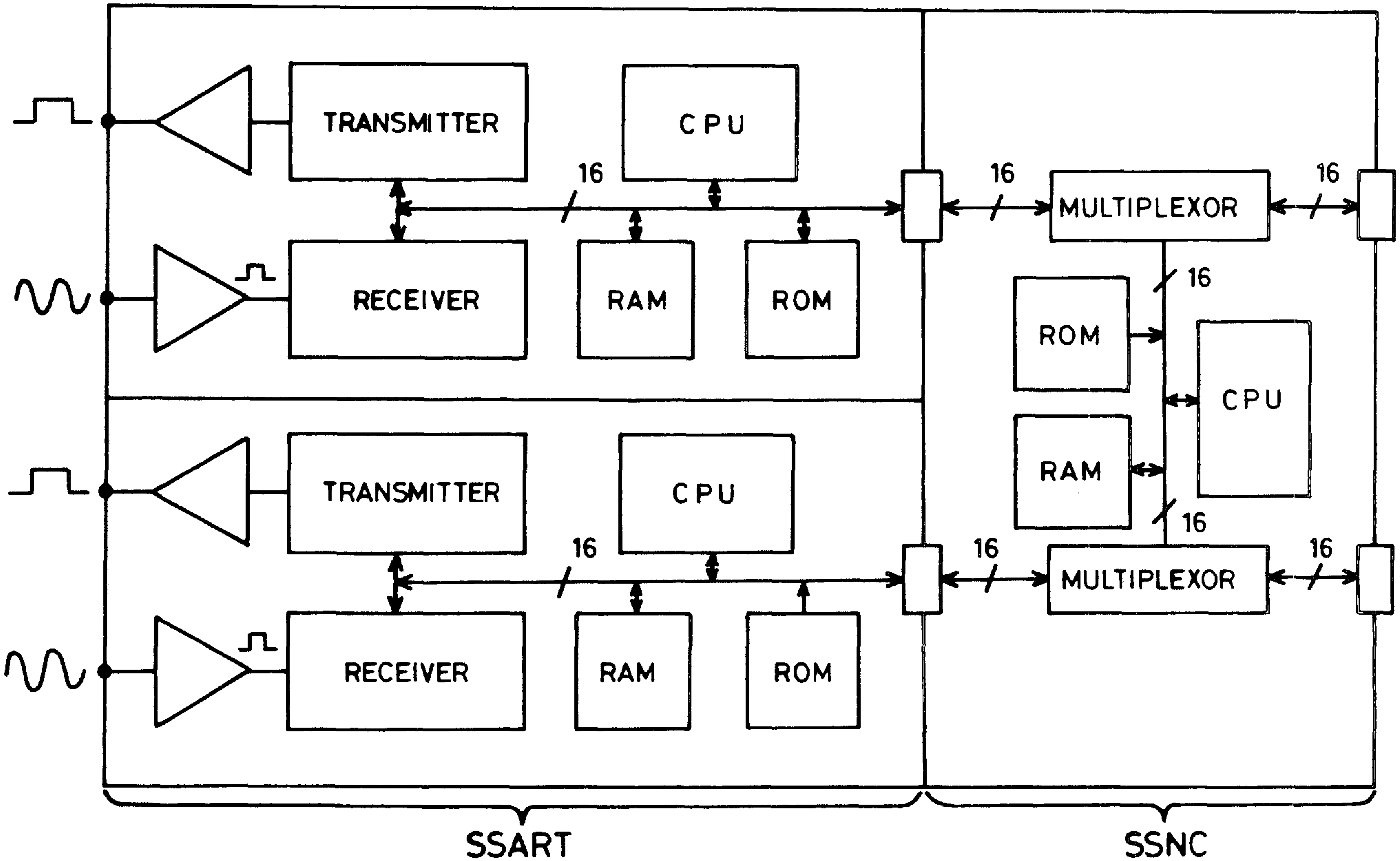


FIGURE (5.7): Station Node Logic (SNL) Block Diagram

Both the receiver and transmitter will be capable of DMA into the SSART-RAM; to maximise the data throughput when necessary. The SSART-CPU will then permit DMA via the SIL to the user device when all the data has been received and the low level protocol removed. This arrangement will require complex bus sharing between the CPU, receiver, transmitter and SIL but it will provide high data throughput.

5.4.3 Operation of the Workstation

The operation of the SNL is split into five parts: SNL initialisation, SNL termination, data transmission, data reception and testing. The operation of the SNL initialisation is:-

(a) The SSNC-CPU disables all SIL parts and determines the number of SSARTs available for data transmission.
(b) The code sequence mapping tables are installed in SSNC-RAM provided by the SSNC-ROM.
(c) Each code sequence mapping is used in turn to send a "request for acknowledgment" message to any node listening on that code.The node will sense the channel for the presence of each code before attempting the request - if a code is detected then it is marked as on-line. If no acknowledgement is received then it is marked as off-line in the SSNC-RAM and will be periodically interrogated to see if it becomes active.
(d) All the SSARTs are initialised and report any component failure to the SSNC-CPU.
(e) Once all SSARTs are initialised and the mapping tables are updated the SSNC-CPU enables all SIL ports and returns the appropriate status to the SIL.

The termination of the SNL in a logical fashion is performed as:-

(a) All further requests for data transmission are disabled by the SSNC-CPU.
(b) The SSNC-CPU transmits a close-down message on all of the code sequences within its memory to all listening receivers.
(c) The receivers are disabled once their data streams end. A timeout will also be employed to overide data which persists.
(d) Operational statistics are reported to the SIL and the SNL is

disabled until an initialisation instruction is received.

The concept of a session and the "session layer" in ISO-OSI has an affect on the amount of information which must be exchanged between two nodes in the course of message transmission. The following description will assume that the messages are of a long duration and therefore a session is equivalent to the message life time. As a result of this a link must be established for each message transaction. When a user device requires the use of an SSART, a request for communications is submitted to the SIL and thus the SSNC. The SSNC will allocate a free SSART (if, n, are allocated then n-1 are available at most as one is dedicated to the SSNC-CPU for network management and high priority messages) to the user. The SSNC will take no further part in the communication protocol until the user device frees the SSART or an SSART timeout occurs due to a device failure, at which time the SSART will reappear on the SSART avaliable list. A typical data transmission will involve:-

(a) A request for transmission will be made via the SIL to the SSART, which will require the code mapping parameter from the request originator.
(b) The SSART-CPU will request the code sequence generation instructions from the SSNC-CPU which will validate the code mapping parameter and return the instructions to the SSART-RAM.
(c) If the mapping parameter is invalid the SSART-CPU will reject the transmission request. If valid it will generate the code sequence and store it in the SSART-RAM. A "request transmission" packet will then be generated and stored in the SSART-RAM. The transmitter will be informed of the address of the start of the data and the code sequence and it will send the message using its DSSS modulation technique.
(d) If the receiver has been previously configured correctly then the answer to the request will arrive and be passed to the SSART-CPU. If not, the timeout algorithm will be implemented with several retries after which if no reply is received it will invoke a failure response. If a refusal is received then a "busy" status will be returned. Should an "acceptance" be received then the transmitter and the SIL port will transfer the data directly with the data rate dominated by the SIL transfer rate - permitting variable data loading.

(e) At the end of data the transmitter will stop data modulation and will return to its idle state awaiting the next transmission request.

The reception of data is more complex and in the case of all communication systems it is the receiver's performance which dictates the protocol and error correction requirements.

(a) The SSART-CPU will receive a request for the inclusion of a code sequence in its search algorithm - each receiver will be capable of "listening" for several codes sequentially.
(b) The SSART-CPU will ask the SSNC-CPU to validate the mapping parameter and to supply the corresponding generation instructions.
(c) A valid request will result in the SSART-CPU generating the code sequence and storing it in the SSART-RAM. It will then join the search list.
(d) The receiver will cycle across the codes until it starts to receivce data on one code. It will then dedicate itself to that code until the data stream is complete after which it will return to the search sequence. It is possible that the SSNC could co-ordinate the search strategy amongst "idle" SSARTs to prevent the loss of data, however this will necessitate a remapping between the user device and the SSARTs should a second, parallel, message be received.
(e) When a "request for transmission" is received the SSART-CPU sends the appropriate acknowledgement packet. This will be a refusal if a second request arrives which has either the same code as the first or if it is from a second source (the source and request numbers will differentiate between the two) otherwise it will be an acceptance.
(f) The receiver will then pass the data directly to the SIL once it is demodulated and this will only cease when either the data ends or a timeout occurs due to a transmitter failure at the data source. The end of data does not denote the end of the session. The termination of a link must be invoked independently of the data activity.

The testing of the stations and their operation is covered in the following section. One final point to make is that this sequence of operations is only one version - its main aim has been to permit both the receiver and transmitter to operate at data rates which are fixed by the user. Consequently the data may not always occur periodically, however the continual coarse acquistion compensates for this fact and only provides data when it is received.

5.4.4 Testing

The test philosophy for the SNLs is to provide a separate test interface which will either replace or be part of the SIL. This test hardware will configure the SIL/SNL interface ports in their test modes and will display all test information received via this port on an accompanying test system. This approach removes the necessity for trouble shooting in the detailed circuitry at an early stage.

At initialisation the SNL will perform self check diagnostics which will include the checking of the RAM areas. Loopback capability will also be incorporated at two levels. The first is at the lowest level of the SSART ie logic levls will be transferred from the transmitter to the receiver. These loopback actions permit isolation testing of the individual workstations without effecting bus activity. Further loopback is provided by self addressing where the receiver "listens" to codes transmitted by its own twin transmitter - this tests the operation of the bus drivers and the ADCs in the receiver.

At various times the SSNC-CPU could be programmed to perform loopback testing on its SSARTs to determine their state and whose errors could be reported to a system log. This can be further developed if the self testing detects a fault on an allocated SSART which is currently idle. The SSNC-CPU and the SIL would then have to allocate a new SSART to the device while maintaining the transparency of operation to the device.

5.5 Design Validation

The validation of the design is a comparison between the developed design and the requirements from which the design is produced. Frequently, a design does not include all the requirements which means that it must be amended: this is preferable to locating design faults on a fabricated design during testing. Current validation techniques rely upon the specification of the requirements in a rigorous and formal format. In this instance the requirements have not been specified formally - indeed the requirements must, by the very nature of this novel system, be flexible until some experience in the expected performance has been gained - and consequently rigorous validation techniques cannot be applied. This validation will be made in two sections: the certification of the system design and the certification of the workstation design.

5.5.1 Certification of the System Design

The system shown in figure (5.6) contains no controller or tuning with only the communications channel linking the SSARTs. The design also facilitates the inclusion of bridges and gateways which would be allocated their own code sequences on either side. Both the bridge and gateway would be responsible for mapping the code sequences to the destination multiple access environment and this would require amendments to the information carried in the header and trailer of the physical layer messages.

The inclusion of the SIL/SNL interface provides a generalised SS-LAN node interface which can be tailored to specific host devices using an appropriate SIL. The use of a passive bus and spur arrangement minimises the effect of a node failure on the SS-LAN and the inclusion of multiple SSARTs provides an alternative virtual link between devices should one or more SSARTs fail. Logical reconfiguration is provided by the SSNC-CPU which performs all network management and code sequence mapping. The SSARTs are capable of generating any of the SS-LAN codes hence the operational mode can be altered by applying different mapping rules at the higher protocol levels.

The comparison of the requirements and design show that the two reflect each other at the system level. This comparison must now be made for the detailed design of the SNLs.

5.5.2 Certification of the SNL Design

The segmentation of the SNL into a single SSNC linked with several SSARTs permits parallel processing should a single user be connected to several SSARTs. Each SSART provides immediate access to the channel with a "request packet" being transmitted at the start of each new transmission. This is used to verify the existence of the destination node and to save time and bandwidth which would otherwise be wasted. The request algorithm includes both a timeout protection and a retry mechanism to compensate for nodes which may be implementing a serial search on several codes.

The data flow is bidirectional within each SSART and the data rate is determined by the SIL and the user; this permits variable data loading (the methods by which this may be achieved are discussed in the next chapter), an important property of the SS-LAN. The SSNC maintains a constant configuration map of the SS-LAN and uses this to detect when a particular code sequence is off or on-line. The SSNC also performs regular fault monitoring and is capable of changing the flow path of data should an SSART be found to be malfunctioning. The testing is also enhanced by loopback facilities and test data which is provided via the SNL/SIL interface when the I/O ports are correctly configured. The comparison of the requirements and the detailed design and operation of the SNL show that the two reflect each other except for the inclusion of the error correction facilities. These have been omitted throughout the design but would be necessary in a final design version - a more accurate figure for the expected raw BER must be calculated before the modified BER can be accomplished in the final design.

5.6 Conclusion

An example of the structure of the SS-LANs has been produced to provide the basis for a theoretical analysis, a computer simulation and a hardware demonstration system. The prime properties of the SS-LAN have been reflected in a manner which provides a high degree of flexibility in the tailoring of an SS-LAN application. The bus and its nodes have been isolated from the intended host devices and provide a reliable, unrestricted and powerful data communications link which possesses both fault tolerance and an ability to detect system errors.

The detailed design of the transmitters and receivers is now required coupled with the provision of the software to drive the microprocessors held in the SSARTs and SSNC. The testing of this system is simplified by the modular design which has provided separate entities which can be tested in their own right. On-line testing is also available and this can be driven either by software or hardware via the SIL/SNL interface. A final point to note is that in the present design the receiver converts the signal using a simple threshold detector (a hardlimiter) and then performs digital correlation using a sliding correlator method based on the polarity coincidence correlator (PCC).

CHAPTER 6

Performance Analysis of the new LAN

6.1 Introduction

As described in an earlier chapter the performance criteria by which a system is measured are the the data throughput, the data delay, the output SNR and the output BER (or more usually the probability of error). This chapter will provide a theoretical analysis of a general, but simple, model of the SS-LAN which will then be used to give quantitative results for these performance criteria. It is not the aim of this analysis to provide answers of great accuracy and precision but it is expected to show the relationship between system parameters such as code sequence length, the number of simultaneous users etc and the resulting SNRs and BERs. These results will then be used to predict the expected performance of the simulator (described in a later section) and provide an empirical means by which the simulator may be tuned. It must be stressed that the model of the SS-LAN will only be simple as inclusion of parameters such as mixed code sequence types, varied code sequence rates and sequence/data ratios lead to highly complex equations which can only be solved using large amounts of computer power while still requiring some important simplifications.

Each of the following sections within this chapter will analyse a separate performance parameter and will slowly build upon the basic SS-LAN model. Consequently there are sections on data throughput, data delay time, node SNRs and node BERs. The final section will discuss the comparison of the SS-LAN parameters with similar parameters for existing LANs based upon TDMA and CSMA techniques and will show in what circumstances each system should be applied in preference to the others.

6.2 Data Throughput

Traditionally, the data characteristics are displayed on throughput versus delay graphs which in the case of the SS-LAN are misleading as will be shown later. For the data throughput it will be assumed that the data which arrives at the transmitter is transmitted immediately and that the destination is ready to receive the data. This situation is close to the steady state of the SS-LAN once the virtual links have been established and can therefore be used as an accurate model for the operation of the LAN. The base equation relates the point-to-point data rate, R_d, with the code sequence frequency, R_c, to give:-

$$R_d = \frac{R_c}{mL} \tag{6.1}$$

where m and L are as defined in equations (2.3) and (2.4). If it is assumed that there is no control or error data included within the information stream then equation (6.1) can be used to plot the data throughput for the point-to-point transmissions (not to be confused with the point-to-point operation mode) versus the code sequence frequency, as shown in figure (6.1).

The next stage is to develop a similar equation for the network wide data throughput or information rate. If there are ,k, simultaneous users transmitting data, from a system designed to handle a maximum of, N, simultaneous users, then the data throughput, R_{tot}, is given by equation (6.2):-

$$R_{tot} = \frac{kR_c}{mL} \tag{6.2}$$

which assumes that each point to point virtual link is maintained by a standard code sequence length and code sequence/data ratio.

The form of equation (6.2) is dependent on the processing gain, mL, which is specified for point to point transmission. This must be extended to take into account the simultaneous data transmission properties of the SS-LAN. Equation (6.2) can be extended to form equation (6.3), where:-

$$R_{tot} = \left[\frac{k}{N}\right] \left[\frac{N}{L}\right] \frac{R_c}{m} \tag{6.3}$$

The term (k/N) is a measure of the operating efficiency of the network, and as such it is defined by, μ, where $0 < \mu \leqslant 1$ for each particular system. Similarly, the term (N/L) is a measure of the orthogonality of the code sequence family and will be represented by, δ, where $0 < \delta \leqslant 1$. If $\delta = 1$ then the code sequence length is equal to the number of nodes but in reality there is mutual interference hence the length must be greater than the number of nodes; this assumes that there is effectively one code per node on the network. The use of and now give rise to the following equation:-

$$R_{tot} = \left[\frac{\mu \delta}{m}\right] R_c \tag{6.4}$$

where the factor ($\mu\delta$/m) is defined as the Data Processing Loss (DPL). The DPL constitutes the factor by which the network data rate is degraded with respect to the maximum frequency of R_c. Figure (6.2) shows the variation of R_{tot} with the DPL for a few values of R_c which are typical for present day LANs and fibre optic LANs.

Figure (6.1) shows the linear relationship between R_d and R_c. The gradient is given by mL with a large value due either to a long code sequence or a large sequence/data ratio. In general only one of these two variables needs to be large as they both perform the same function of increasing the spectral spreading and, m, need only be used if the code sequence lengths are limited by some physical consideration of their generation. The network wide equivalent of the point to point throughputs is shown in figure (6.2) where the abscissa is the DPL. Again a linear relationship is displayed with the dramatic lowering of the throughput shown with low DPL values. The DPL consists of the network efficiency, μ, and the code orthogonality factor, δ, both of which are lower than one and consequently cause a shifting to the left along the x-axis. The conclusions drawn from figures (6.1) and (6.2) are that:-

(a) The network efficiency should be as close to unity as possible or else the bandwidth is wasted: each node should therefore be constantly transmitting data.

(b) The code sequence length should be as short as possible to provide greater throughput at both point to point and network levels.

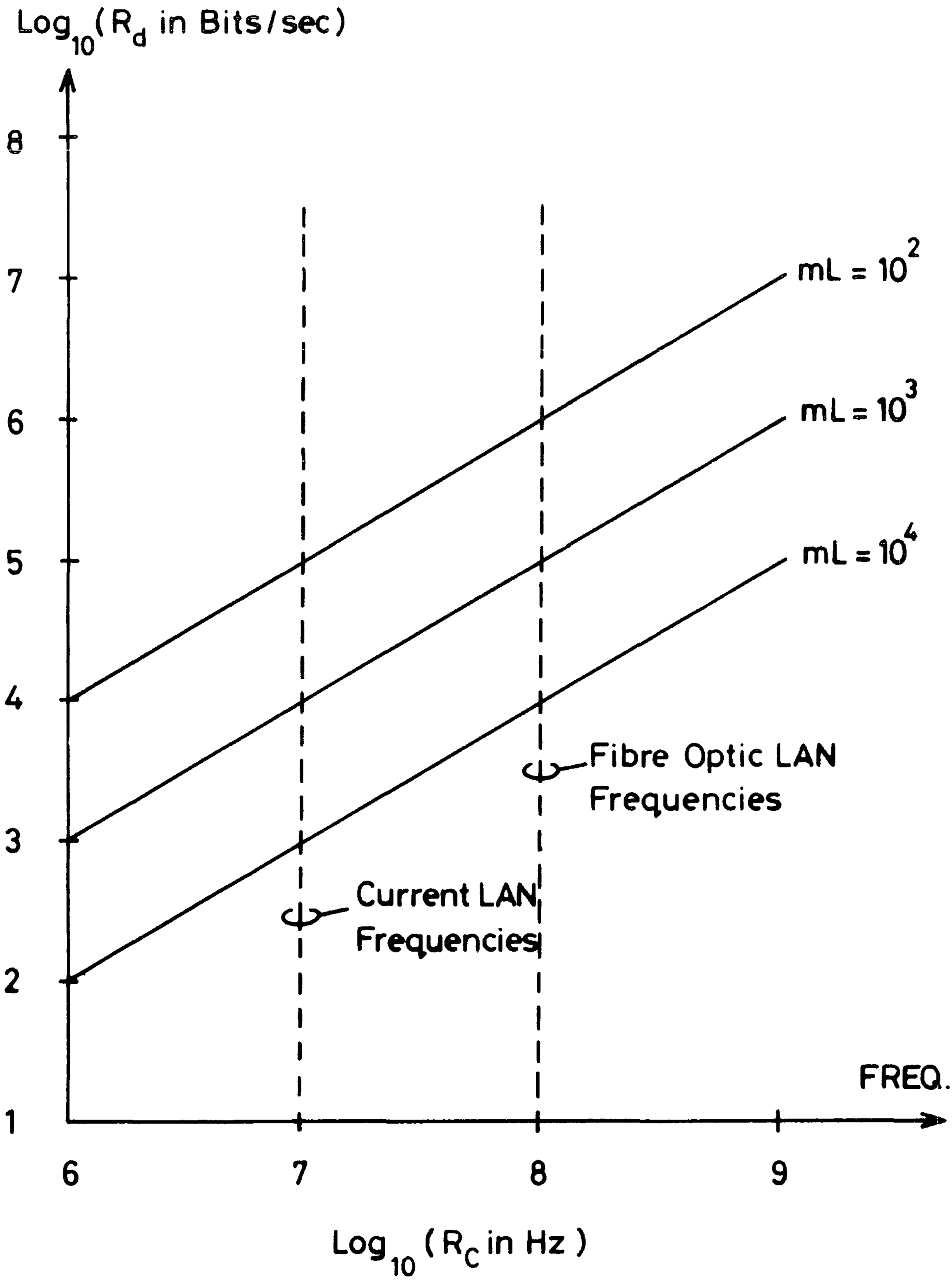


FIGURE (6.1): Point-to-Point Data Throughput vs Transmission Frequency

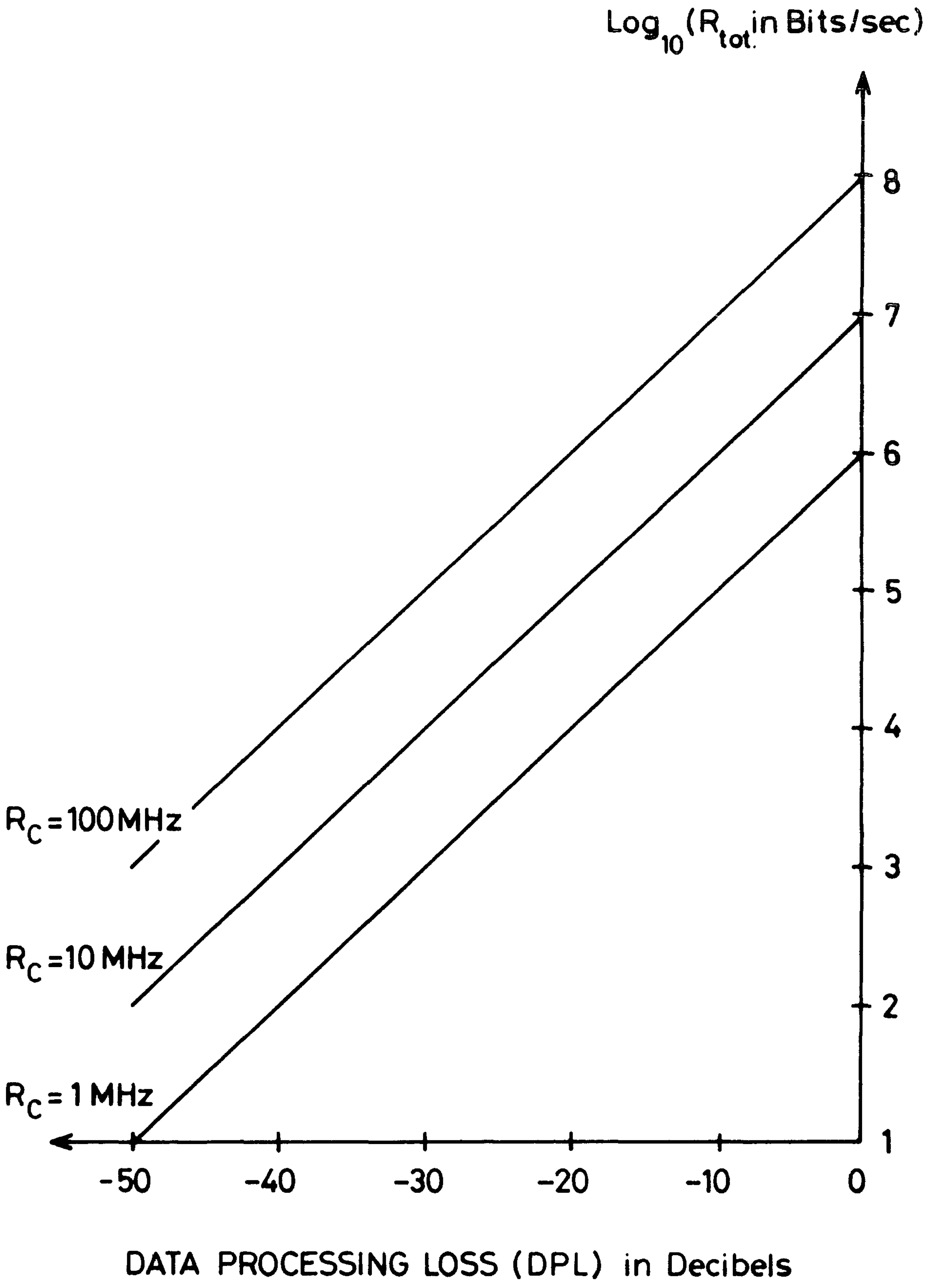


FIGURE (6.2): Network Throughput vs Data Processing Loss

(c) Code families with low mutual interference are required with as many codes as nodes produced in such a manner that the code sequence length is equivalent to the number of orthogonal codes.

At the present these conclusions are made with respect to high data throughput requirements and they will be amended as the other system parameters are considered.

6.3 Delay Characteristics

The delay characteristic is the time taken for the transmission of an information packet and will be calculated from the time at which the data enters the transmitter modulation to the time at which it is completely demodulated by the correlator. This gives rise to the equation:-

Data Delay	=	Overhead at txr	+	Overhead at rxr	+	Access Delay	+	Message Transmission	+	Propagation Delay

In the steady state SS-LAN operation the access delay is zero and the transmitter and receiver overheads will be small when compared with the message transmission time. The propagation delay will be defined as, τ, typically 5 microsecs per kilometre, and is included as a reference for later comparisons with other LANs.

If the message length is defined as, B, bits then the data delay is given by:-

$$D_t = \frac{BmL}{R_c} + \tau \qquad (6.5)$$

This defines the absolute data delay for a point to point data transmission for a message of length B bits. A plot of the variation of D_t with B is given in figure (6.3).

An attempt at the network equivalent to the point to point delay will now be made. The assumptions given in the throughput analysis are assumed true in this analysis also. If the data supplied, S, for transmission across the whole network is transmitted immediately (as should be the case in most instances) by, k, nodes, all of which send an equal amount of data and use the same code sequence characteristics, then the average data delay is given by equation (6.6).

$$D_t = \left[\frac{S}{k}\right] \frac{mL}{R_c} + \tau \qquad (6.6)$$

Inclusion of the definition of the DPL will then provide the final average delay equation:-

$$D_t = \left[\frac{m}{\mu\,\delta}\right] \frac{S}{R_c} + \tau \qquad (6.7)$$

The ratio (S/R_c) is a measure of the data throughput for the network normalised by the transmission frequency of the code sequence. This ratio is used to provide figure (6.4) which is the throughput-delay characteristic for a simplified model of the SS-LAN.

The lines A,B and C are derived from equation (6.2) and show the maximum throughput to code sequence ratio which can be maintained by the delay model. This ratio is defined as:-

$$\frac{R_{tot}}{R_c} = \left.\frac{S}{R_c}\right|_{max} = \left[\frac{\mu\,\delta}{m}\right] \qquad (6.8)$$

thus the maximum must be less than or equal to the DPL of the system. The delay model can be amended by accounting for internal buffering of data at the transmitter to permit data requests which are greater than the data rate supported by the system.

Figure (6.3) shows the message delay for point to point interactions as the message length varies. The message transmission time dominates the delay characteristic (making the propagation delay inconsequential) producing substantial delay times when the point to point rate is only a few hundred bits per second. The network wide equivalent is shown in figure (6.4) where the delay for the transfer for

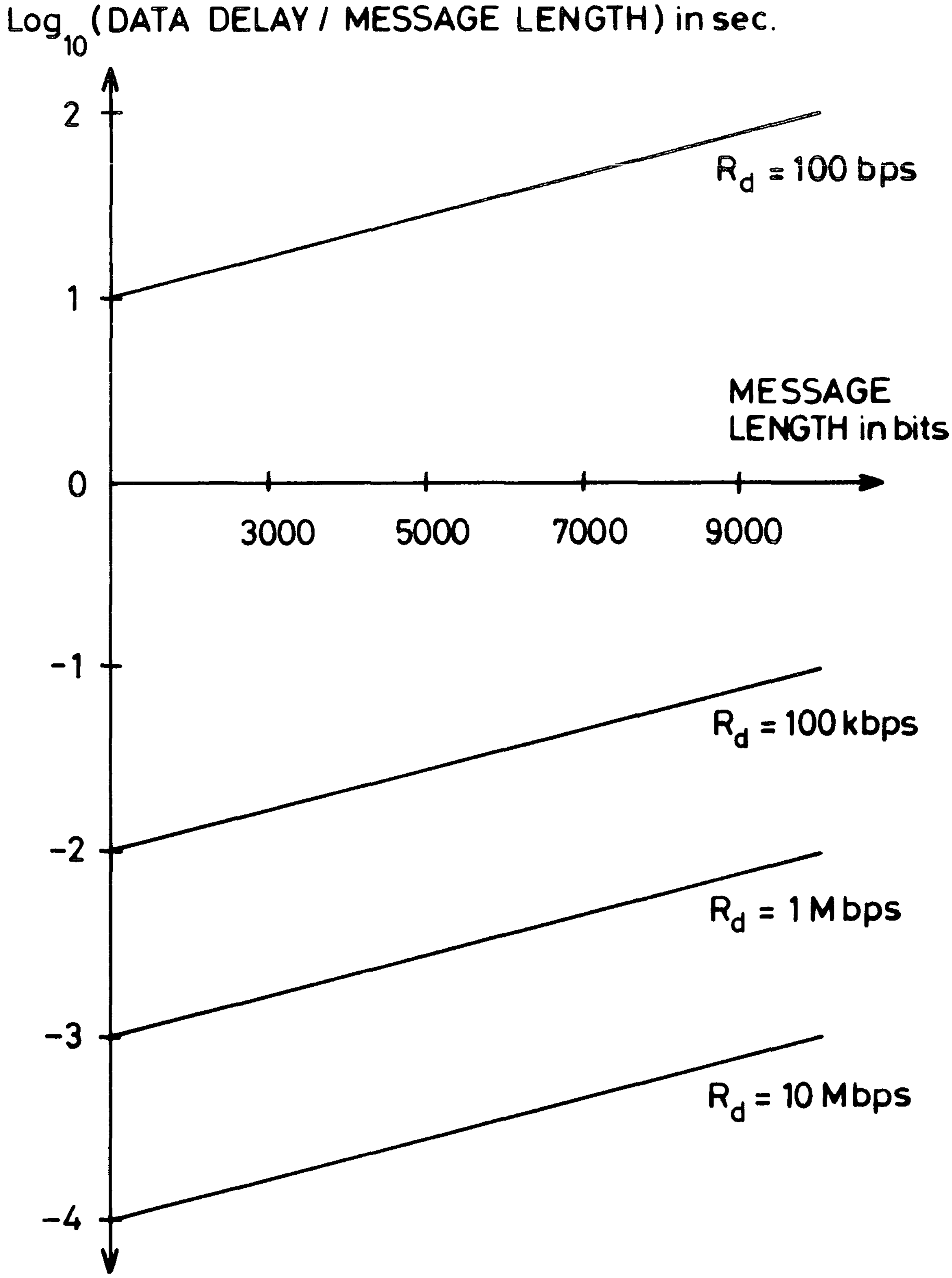


FIGURE (6.3): Point-to-Point Data Delay vs Message Length

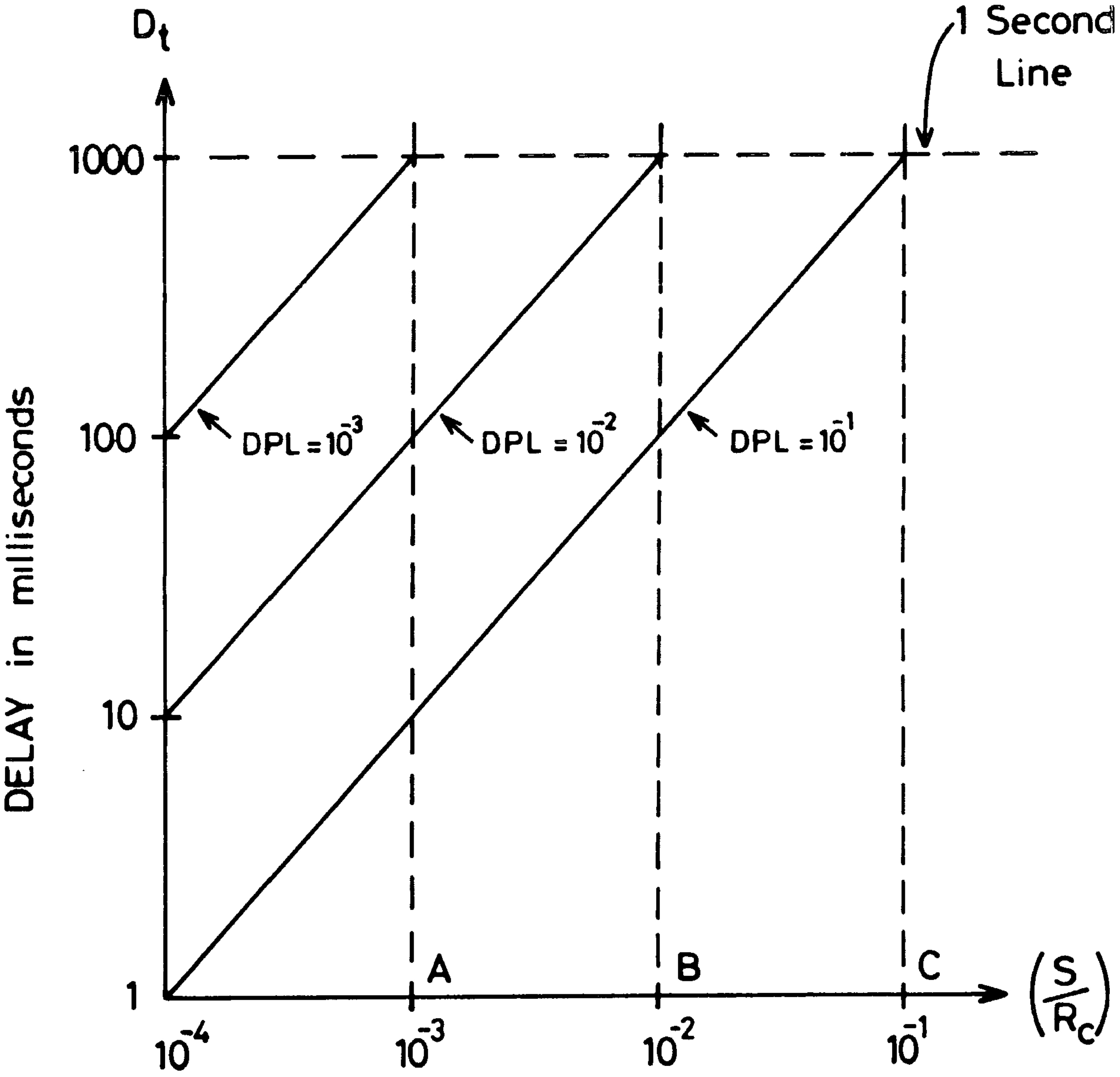


FIGURE (6.4): Delay for Total Network Data vs Network Loading a fixed amount of data across the entire network is shown for variation with the network loading ratio (using the code sequence frequency as the reference data rate). The effects of different DPLs are also shown with the result that a decreasing DPL increases the throughput delay and limits the network loading ratio. The delay limit is set as one second which defines the throughput as, S, bits per second for the network in this delay model of the system. The conclusions which can be drawn from these two figures are:-

(a) The messages should be as short as possible thereby reducing the delay.
(b) The DPL should be as large as possible to reduce the network wide delay and to increase the network loading ratio.
(c) If there are long messages on the system then for a given DPL the delay time for the message is deterministic and non statistical even when the data throughput requests become higher than the data rate supported by the system.

These conclusions are presented with the aim of maximising the delay characteristics and hence the data throughput, and should therefore be considered with these conclusions derived from the previous data throughput analysis.

6.4 Node SNRs

The schematic diagram for the operation of the SS-LAN is shown in figure (6.5). As specified in the hardware design, the receiver is composed of a threshold detector (or hard limiter), set with the threshold defined as zero for a bipolar signalling technique and a sliding polarity coincidence correlator. The received signal r(t) is constructed from the analysis in appendix one and possesses a receiver input SNR termed as SNR_c. This analysis is concerned with deriving the SNR at the output of the receiver ie $\overline{SNR}_i$.

Shannon's equation for the channel capacity states that the output SNR can be derived from a low input SNR and a large bandwidth, thus the defining SS equation becomes:-

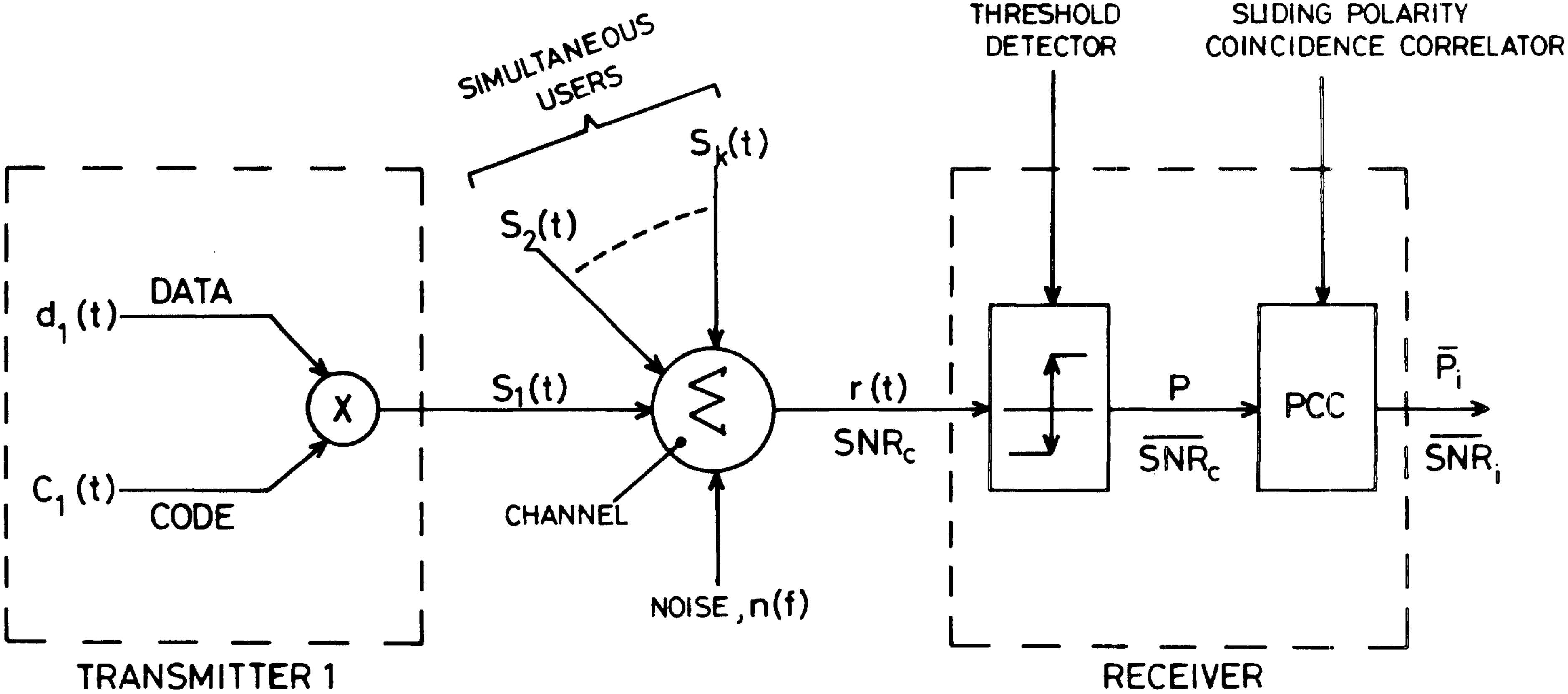


FIGURE (6.5): Schematic of an SS-LAN Communicating Pair

$$\overline{SNR}_i = mL\ \overline{SNR}_c \qquad (6.9)$$

The SNR degradation due to the inclusion of a hardlimiter has been shown by Ekre (68) to be a factor of $2/\pi$, consequently the relevant equation (6.10) is produced as:-

$$\overline{SNR}_c = \frac{2}{\pi}\ SNR_c \qquad (6.10)$$

the SNR_c is $\ll 1$. Appendix three provides a derivation for SNR_c where equation (A3.8) becomes equation (6.11) for a large number of users and therefore, in general,

$$\underset{\text{large } k}{SNR_c} \longrightarrow \frac{1}{(k-1)\ f(P)} \qquad (6.11)$$

provides the criteria of $SNR_c \ll 1$. Manipulation of equations (6.9), (6.10) and (A3.8) provides a definition for $\overline{SNR}_i$:-

$$\overline{SNR}_i = \frac{2mL}{\pi}\left[(k-1)f(P) + \frac{1}{SNR_1}\right]^{-1} \qquad (6.12)$$

From equation (6.9) the single user equivalent can be defined by equation (6.13), and with the inclusion of equation (6.12) and the

$$SNR_d = mL\ SNR_1 \qquad (6.13)$$

definition of the DPL from equation (6.4), equation (6.12) now becomes

$$\overline{SNR}_i = \frac{2}{\pi}\left[\left\{\frac{\mu\delta}{m}\right\} f(P) + \frac{1}{SNR_d}\right]^{-1} \qquad (6.14)$$

where $\overline{SNR}_i$ (the SNR at the output of the i^{th} receiver) is related to the single user input SNR at the receiver, the DPL and the loss due to the propagation of the signal and noise along the channel.

Equation (6.14) is similar to those derived by Pursley (70) for a matched filter receiver and by Weber, Huth and Batson (71) for a general CDMA system. Weber et al derive equation (6.15),

$$\left(\frac{E_b}{N_o}\right)_n = \left[\left(\frac{E_b}{N_o}\right)_R^{-1} + \eta^{-1} \sum_{i=1}^{n-1} \alpha_i \right]^{-1} \qquad (6.15)$$

where (E_b/N_o) is the equivalent of $\overline{SNR}_i$, n=k, η=mL and the function of α_i is used to model the variation in transmitter power - the equivalent of the SS-LANs, f(P). The entity $(E_b/N_o)_R$ is the required single user SNR in order to achieve a given $(E_b/N_o)_n$ value. The rest of their paper is based on the relationship between these two SNRs and their effects on system realisation as shown by the definitions of the "degradation factor" and the "multiple access capability factor". In fact equations (6.14) and (6.15) are equivalent to each other - the factor $2/\pi$ necessary due to the hardlimiting. Pursley provides a more rigorous approach to the derivation of the resulting SNR and relates this to the aperiodic ACFs of the codes. The approximation to his equation is given by (6.16) where N=mL in equation (6.14). The main difference is the

$$\overline{SNR}_{pursley} = \left[\frac{k-1}{3N} + \frac{N_o}{2E}\right]^{-1/2} \qquad (6.16)$$

inclusion of the square root and this is due to Pursley's definition of SNR. The Pursley SNR is equivalent to:-

$$\overline{SNR}_{Pursley} = (\overline{SNR}_i)^{1/2} \quad \text{and} \quad \frac{2E}{N_o} = (SNR_d)^{1/2}$$

and this reduces equation (6.16) to the same form as equation (6.14). The final difference is the factor, 3, in equation (6.16) as highlighted by Weber et al (71). This is a function of the code sequences and a truly random sequence provides the factor 1/3 ie an increase in the resulting $\overline{SNR}_i$. Garber and Pursley have further analysed this factor (72) and have found it to be related to the aperiodic ACF and the actual chip waveform. In equation (6.14) the effects of the code sequence randomness is not incorporated as an effect of the code sequence length (mL) consequently the SNR will be lower than that for the random cases. To compensate for this factor equation (6.14) will be expanded to its final form of equation (6.17) where f(R) is the

$$\overline{SNR}_i = \frac{2}{\pi}\left[\left\{\frac{\mu\delta}{m}\right\} f(R)\ f(P) + \frac{1}{SNR_d}\right]^{-1} \qquad (6.17)$$

compensation factor for the code sequence entropy, as defined by Pursley, Sarwate and Stark (73):-

$$f(R) = \frac{1}{T_c{}^3} \int_0^{T_c} R^2(\tau)\ d\tau \qquad (6.18)$$

T_c is the chip period and R is the aperiodic autocorrelation function for the chip waveform.

Equation (6.17) is the required relationship between the receiver's output SNR and the system parameters. The analysis of this equation is shown in figures (6.6),(6.7),(6.8) and (6.9) where f(R) has been set as 1/3 to obtain the maximum SNR possible and the data processing resolution is defined as the inverse of the data processing loss in decibels, ie DPR=1/DPL.

In figure (6.6) the resulting $\overline{SNR}_i$ is plotted against the receiver input SNR_d for a fixed f(P), where signal and noise are equidistant from the receiver and for a wide range of DPL values. Simple analysis of equation (6.17) shows that for this type of graph the limiting states are:-

$$\overline{SNR}_i \rightarrow \frac{2SNR_d}{\pi} \qquad \text{when} \qquad DPL.SNR_d \leqslant 3$$

$$\overline{SNR}_i \rightarrow \frac{6DPR}{\pi} \qquad \text{when} \qquad DPL.SNR_d \geqslant 3$$

This graph shows the SNR limit for a particular configuration. If an $\overline{SNR}_i$ of at least 30dBs is required then the DPL must be $\leq 10^{-3}$ or else no input SNR_d can produce the desired output SNR. The DPL is therefore a measure of the resolving power of the system hence once this threshold is achieved no further action can improve the performance except an increase in the resolution ie a decrease in DPL. Figure (6.7) is similar to (6.6) except that it demonstrates the effect of f(P) on the system performance. As the noise source approaches the receiver then its effect becomes increasingly significant and demonstrates the near-far effect. The resolution of the DPL is now more easily saturated and all the $\overline{SNR}_i$ are lowered when the noise source is nearer to the receiver than the signal source. Conversley, the resolution is improved when the signal source is nearer than the noise source and is shown by a negative f(P) value on the plot. The explanation is that as SNR_d is increased then ALL of the simultaneous users increase their signal power. The result of this is to reduce the effect of the environmental noise but not that of the other users (only an increase in the PG can achieve this).

Figure (6.8) plots the variation of $\overline{SNR}_i$ against the DPL for an f(P) of zero decibels and for a range of input SNR_d. The salient feature of this plot is that it shows the resolving power of the DPL for a particular input SNR_d. If SNR_d = 30dBs then the maximum output $\overline{SNR}_i$ can only be achieved by using a DPL of 10^{-4} or less. Similarly, in figure (6.9) the effects of signal and noise displacement is shown for an input SNR_d of 30dBs. Once again the near-far effect is duplicated with a serious degradation of the resolution by the excessive noise power and with the converse situation being equally true.

Between them, these four plots can be used to predict the SNR performance of any SS-LAN by modelling its efficiency, node distribution, transmission power, number of simultaneous users and code sequence length and data ratios. Several conclusions can now be drawn from these to define the ideal performance parameters:-

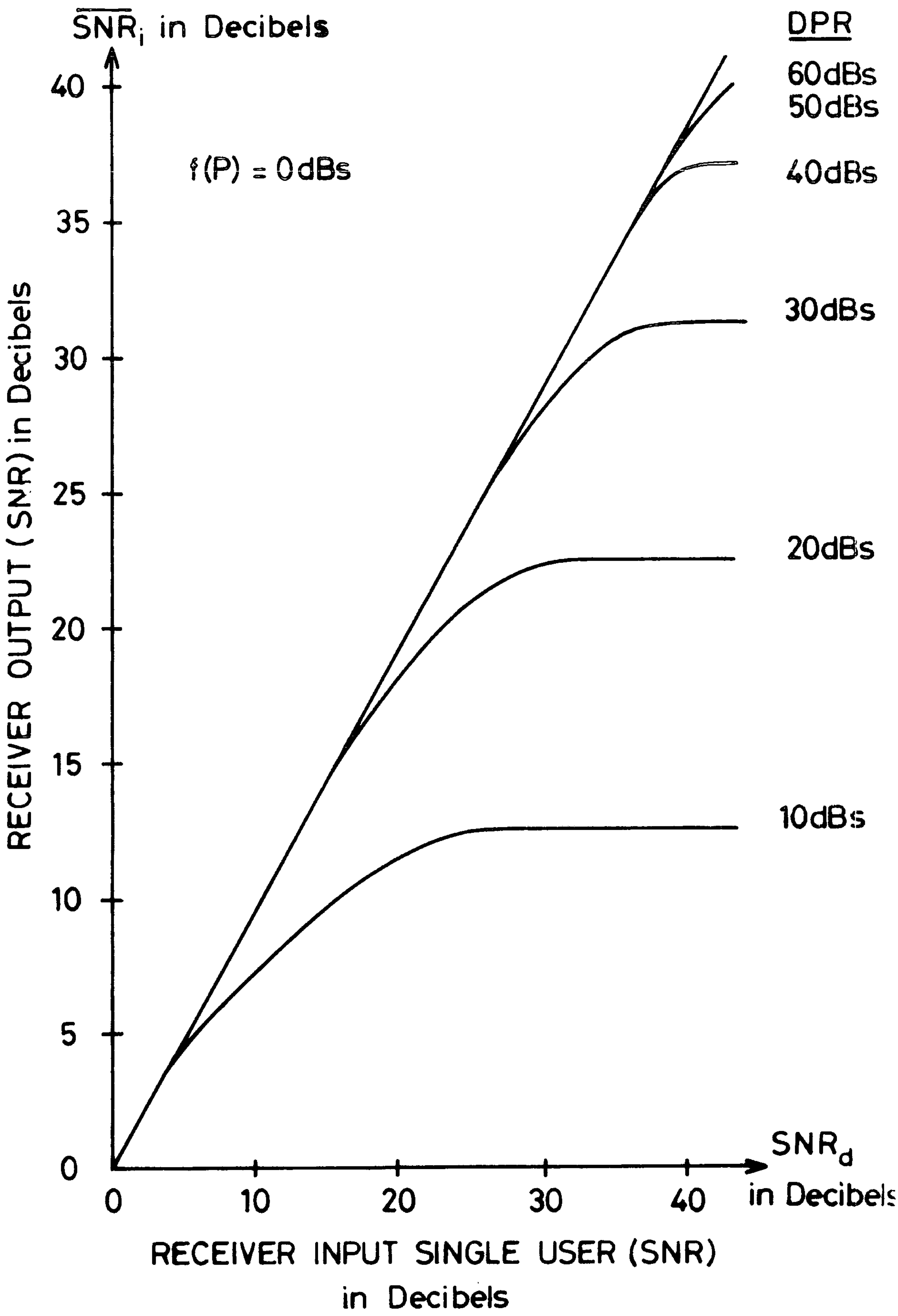


FIGURE (6.6): $\overline{SNR}_i$ vs SNR_d for an f(P) of Ø Decibels

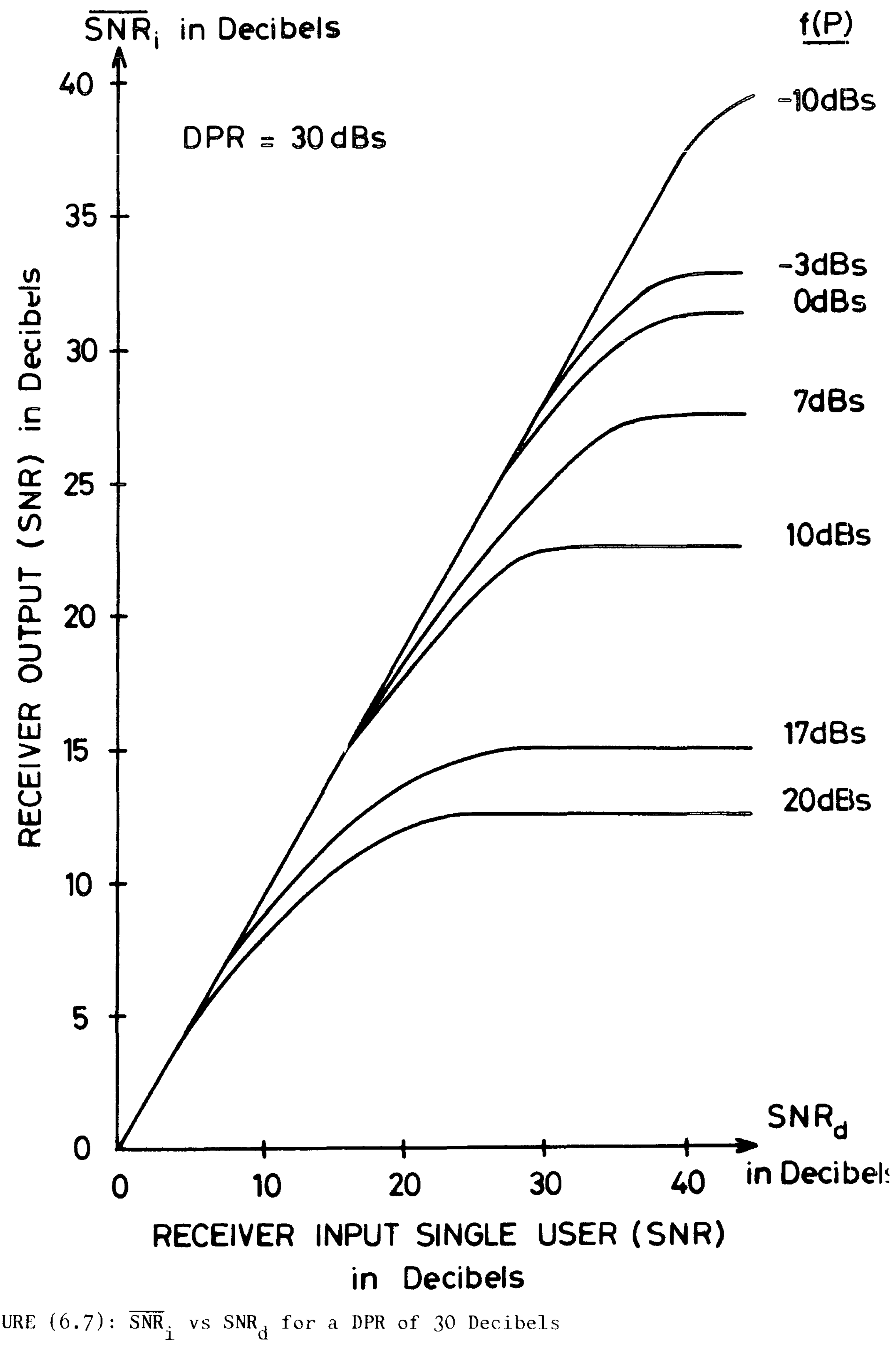


FIGURE (6.7): $\overline{SNR}_i$ vs SNR_d for a DPR of 30 Decibels

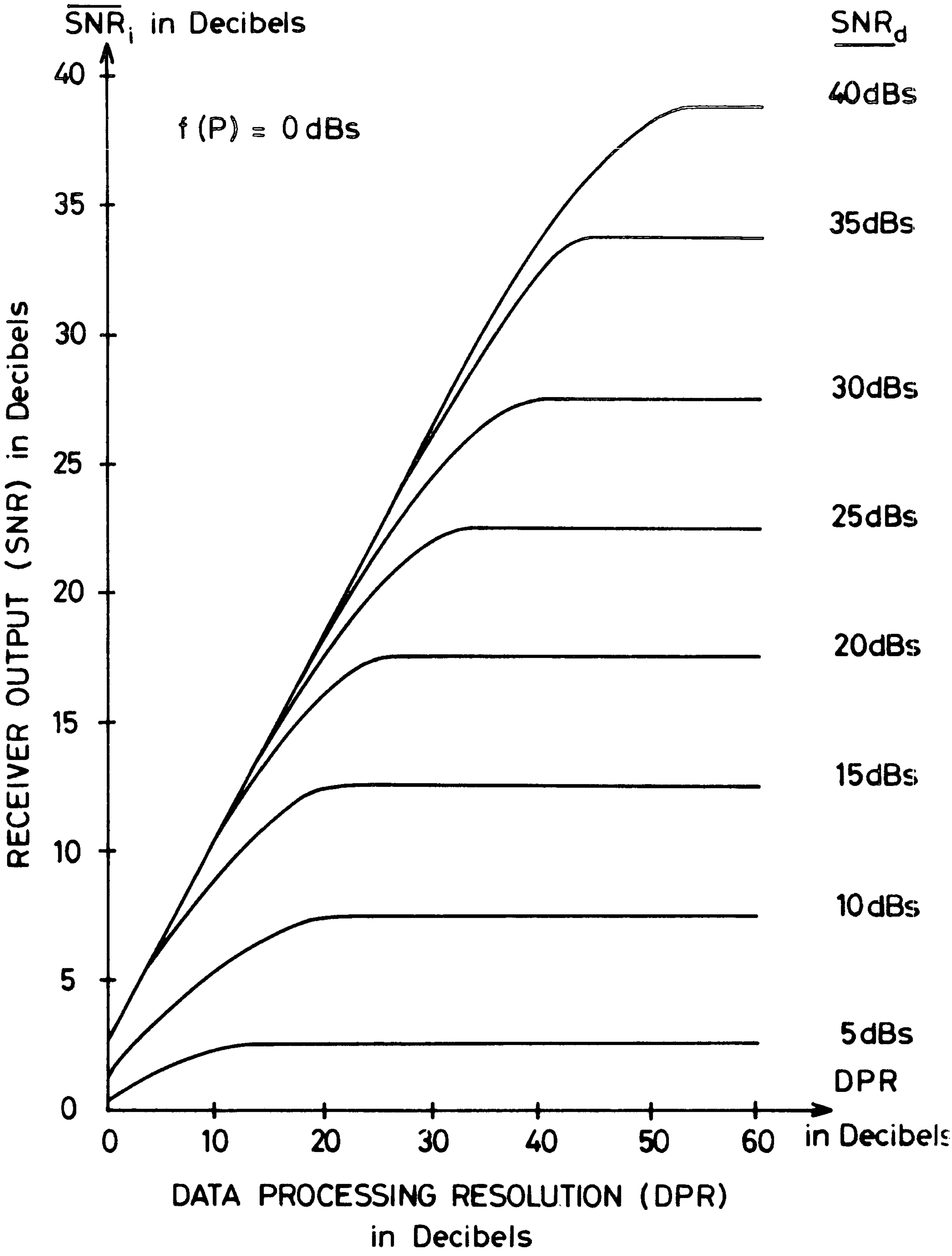


FIGURE (6.8): $\overline{SNR}_i$ vs DPR for an f(P) of Ø Decibels

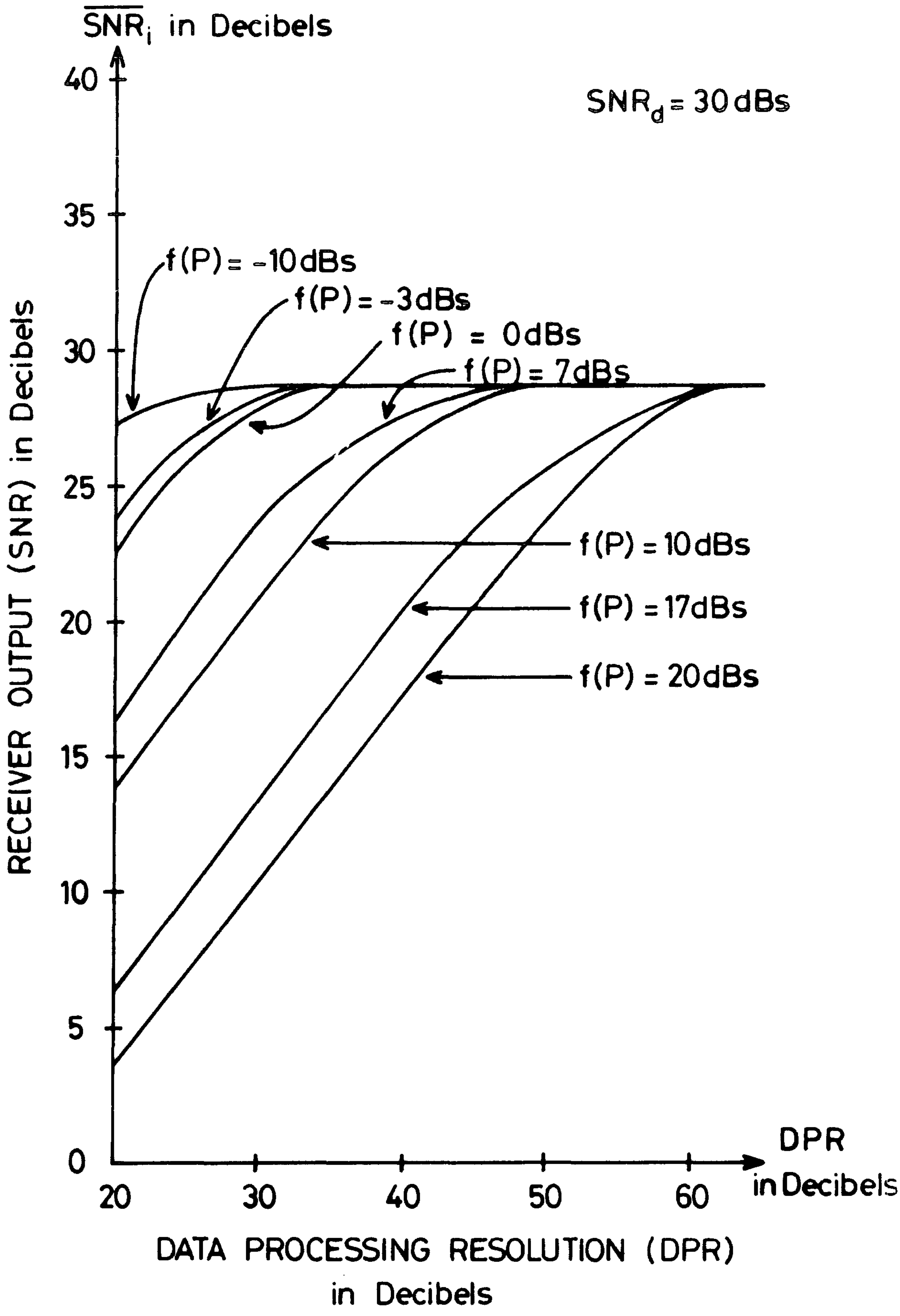


FIGURE (6.9): $\overline{SNR}_i$ vs DPR for an SNR_d of 30 Decibels from these to define the ideal performance parameters:-

(a) The differences between the DPRs of 40dBs, 50dBs and 60dBs is minimal therefore the DPR of 40dBs should be the highest employed as this will give the largest point to point data rate.
(b) The transmission power should only be set at the level for which the equivalent BER is required. An increase in the SNR will have no further effect once the system's threshold resolution has been achieved.
(c) The use of a poor quality channel and large relative distances between the transmitters and receivers will lead to a wide variation in performance due to the near-far effect - this may be overcome by tuning the powers of the transmitter according to their distribution or by the use of relay circuits on the channel.
(d) The DPR should be at least 20dBs or else much of the input power is being wasted.
(e) The transmission SNR should be of at least 15dBs or else it will not, in general, utilise to the full extent the DPR and the simultaneous multiple user environment.

6.5 Node BERs

Chapter two showed that the probability of error was dependent on a function of the SNR ie $\overline{SNR}_i$. Appendix two showed that the actual data content affects the correlation coefficients hence it will be assumed that the data would normally be transmitted as a baseband bipolar signal of $\mp V_d$. The standard probability of error, $\overline{P}_i$ as shown in figure (6.5), is usually defined in this instance as:-

$$\overline{P}_i = \mathrm{Erfc}(\sqrt{SNR}) = Q(\sqrt{SNR}) \qquad (6.19)$$

At the output of the receiver the SS modulation has been effectively removed and so the final probability of error is a result of any further modulation scheme utilising the $\overline{SNR}_i$ value as the source SNR. The probability of error can therefore be defined as:-

$$\overline{P}_i = \mathrm{Erfc}\left[(\overline{SNR}_i)^{1/2}\right] \qquad (6.20)$$

where $\overline{SNR}_i$ is as shown in equation (6.17). Whilst this is not a precise measure of the error, due to the average $\overline{SNR}_i$ source, it is a useful approximation. $\overline{P}_i$ can therefore be defined as the average probability of error of the system.

It is the $\overline{P}_i$ which is shown in the following plots and not the BER because the latter is dependent on the data frequency and whether or not errors are common to different data streams. Figures (6.10), (6.11), (6.12) and (6.13) are the probability of error plots corresponding to the $\overline{SNR}_i$ plots shown in figure (6.6) to (6.9) respectively.

For a communications system which will be operating at data rates of approximately 1MHz then the natural $\overline{P}_i$ must be of the order of at least 10^{-4} or 10^{-5} or else the error correction protocols will severly degrade the data bandwidth. The inclusion of error correcting codes can also be interpretted as spectral spreading ie an increase in the code sequence length. Care must be taken in reaching this conclusion because this new code sequence contains data whose relative position within the code is of critical importance. The new PG is <u>not</u> therefore linearly increased but is defined by a more complex function. The graphs show the $\overline{P}_i$ for the point to point transfer and <u>not</u> for the network wide total. Figure (6.10) shows that once the DPR 20dBs then the $\overline{P}_i$ is particularly small when the SNR_d is greater than 15dBs. The affect of f(P) shown in figure (6.11) indicates that in extremely adverse near-far environments the $\overline{P}_i$ drops to ranges comparable with low DPRs in the previous plot. Figure (6.12) and (6.13) plot the variation of $\overline{P}_i$ with the DPRs and show the thresholding effect at which point it is inefficient to implement a DPR of greater than 40dBs because the lost data bandwidth is not compensated for by a significant increase in system performance. The variation of f(P) is again displayed in figure (6.13) with a $\overline{P}_i$ threshold of 10^{-15}.

The conclusions drawn from these four plots are similar to those given in the previous section. The $\overline{P}_i$ is another method of representing the $\overline{SNR}_i$ with the advantage that the $\overline{P}_i$ can be physically measured by comparison between the transmitted and received data. Consequently for a $\overline{P}_i$ of greater than 10^{-5}, the system needs:-

(a) A DPR bounded by, $20dBs \lesssim DPR \lesssim 40dBs$

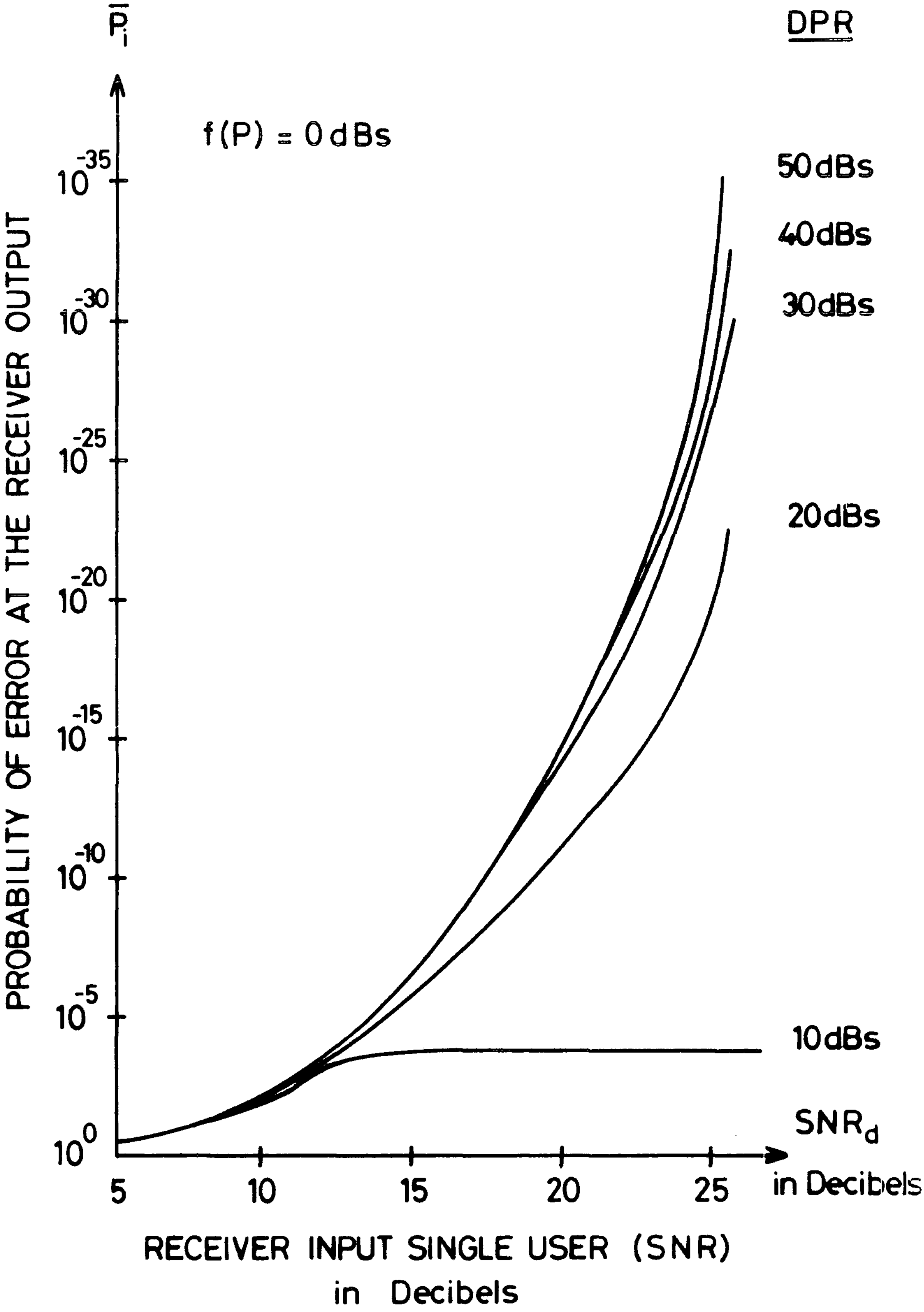


FIGURE (6.10): $\bar{P}_i$ vs SNR_d for an f(P) of Ø Decibels

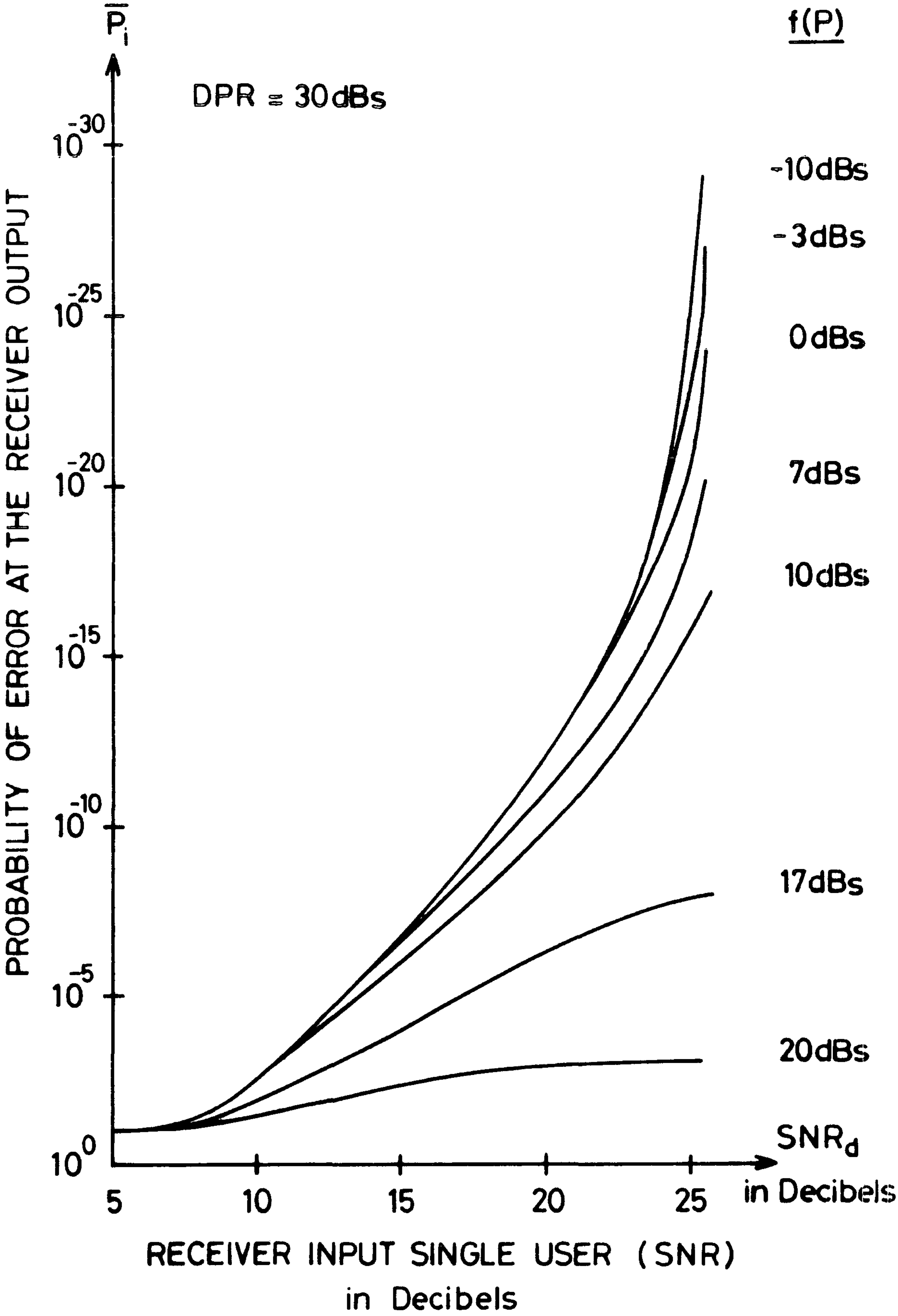


FIGURE (6.11): $\bar{P}_i$ vs SNR_d for a DPR of 30 dBs

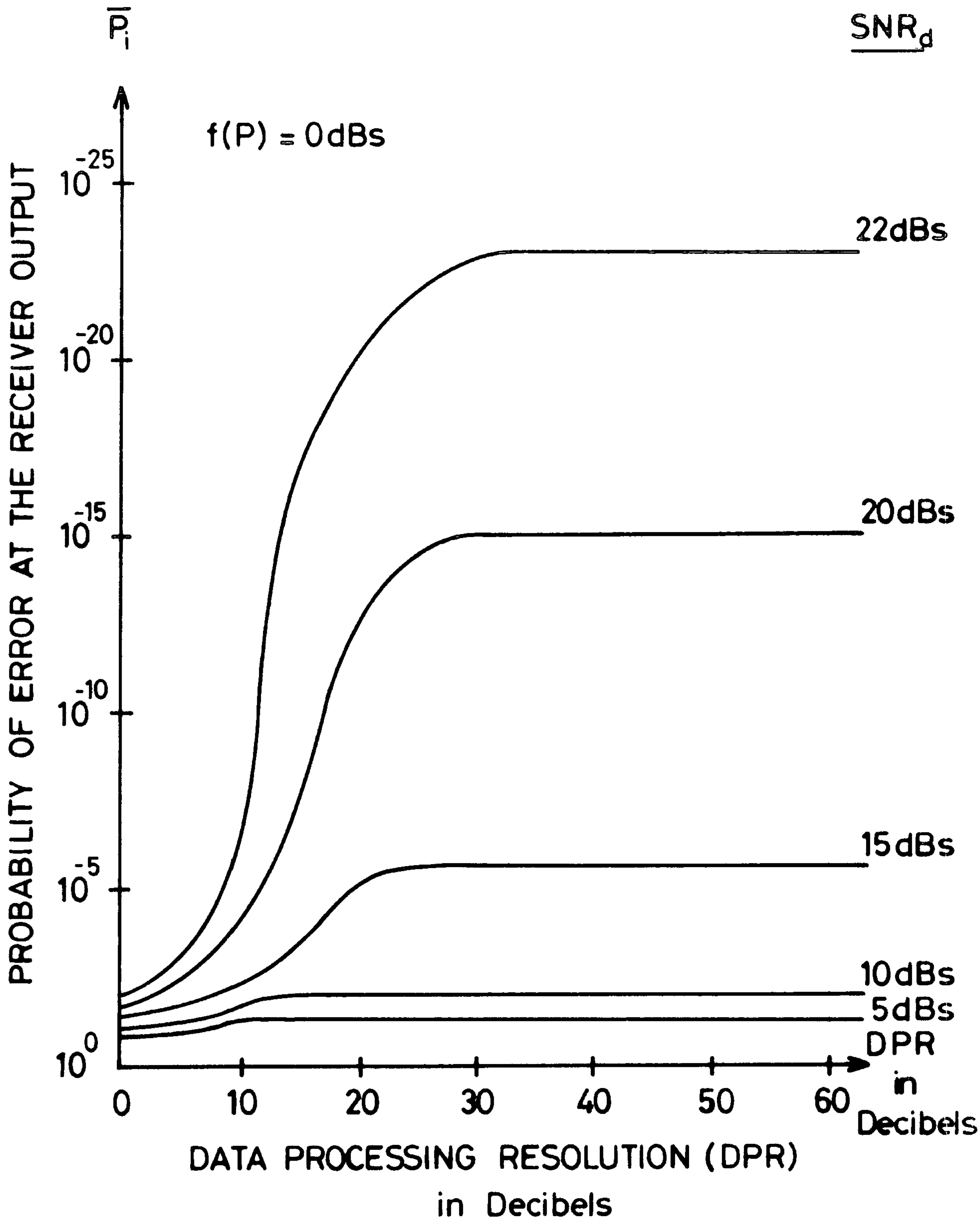


FIGURE (6.12): $\bar{P}_i$ vs DPR for an f(P) of Ø Decibels

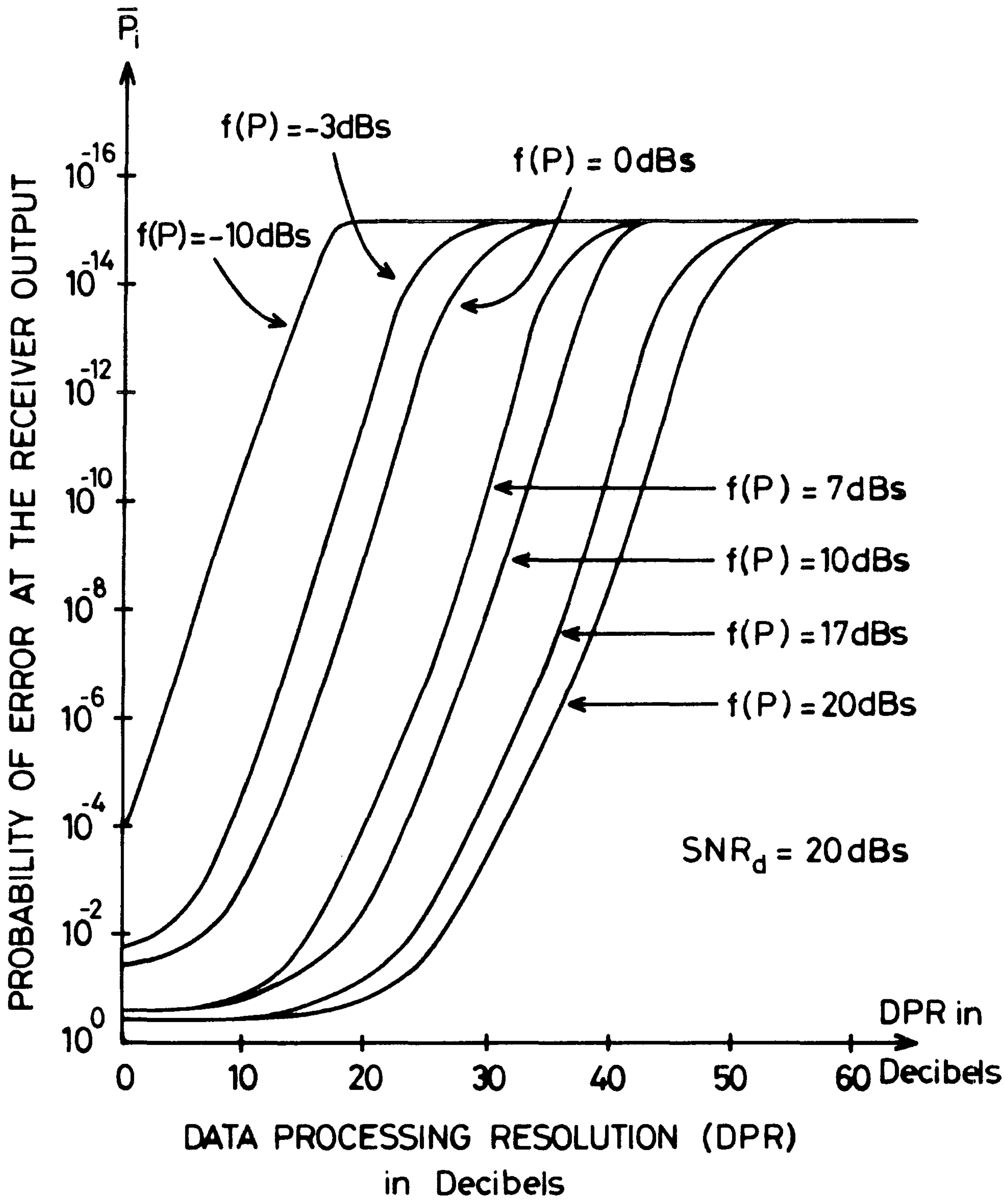


FIGURE (6.13): $\bar{P}_i$ vs DPR for an SNR_d of 20 Decibels

(b) An SNR_d bounded by 15dBs $\lesssim SNR_d \lesssim$ 20dBs
(c) An f(P) bounded by f(P) $\lesssim$ 10dBs.

6.6 Performance Comparison with Other LANs

Chapter two introduced the delay/throughput characteristic as a means by which the performance of TDMA LANs could be compared; including those defined as CSMA LANs. Kleinrock and Tobagi (74) introduced this analysis for the comparison of the new CSMA protocol to the existing ALOHA access mode in an attempt to improve the throughput of the ALOHA system. The analysis was based upon the factor, a, which was defined as the ratio of the propagation delay to the packet transmission time.

By the very nature of CSMA systems there is a probabilistic delay between receipt of the request for message transmission and the actual message transmission. This was reflected by the definition of, S, as the data throughput (or channel utilisation) and, G, as the mean offered data rate. The relation between G and S is that if G bits/sec of data are presented to the network for transmission to other nodes then the amount of successfully transmitted data is S bits/sec. The difference between G and S is a measure of the number of contentions and corresponding loss of bandwidth. The equation for the delay/throughput characteristic developed by Kleinrock and Tobagi was:-

$$\text{Average Delay} = \left(\frac{G}{S} - 1\right)R + a + 1 + r \qquad (6.21)$$

where R is the average retransmission time for each data packet and r is the average pretransmission delay produced by the occurrence of a contention. Equation (6.21) was derived from several important assumptions:-

(a) The average retransmission delay is large compared with the packet transmission time.
(b) The probability of transmission at each node is low for each frame slot and is exponentially distributed across the network.
(c) The packet lengths have a defined average length with an actual

length set as an exponential function.

This form of analysis is further developed by Bux (75) and is applied to three common forms of networks ie Ethernet (CSMA/CD), Token Rings and slotted rings (Cambridge Ring): the analysis of random access communication systems ie CSMA systems, has been thoroughly investigated in a recent publication of the IEEE Transactions on Information Theory (76). Figure (6.14) plots their delay/throughput characteristics and these will be compared with the SS-LAN equivalent at a later stage. The most important feature is that in Ethernet it could be nearly 100 data packets periods before a particular node received its data whereas in the SS-LAN the data is being received almost immediately, but at a slower rate. If the probability of transmission is increased so that it approaches unity then the CSMA/CD system performance becomes dominated by the collision retry algorithms and consequently the delay is greatly increased. This does not occur in the case of the slotted ring and token ring, however they will both limit the amount of information sent by each node. In contrast the SS-LAN operates at its most efficient when its utilisation is high and every node is transmitting data. Figure (6.4) plots the variation of data delay with (S/R_c), however this must be amended before it can be compared with the results from figure (6.14). The amendment is concerned with the y-axis which should represent a normalised data delay; this is normalised with respect to the mean packet transmission time. When this normalisation is applied to figure (6.4) it is found that the mean packet transfer time and mean packet transmission time are equal hence figure (6.4) becomes a horizontal line at y=1. This is clearly superior to the plots shown in figure (6.14) where the SS-LAN curve <u>is</u> the x-axis.

Other differences between the SS-LAN and systems such as Ethernet are that the SNR is effectively constant and is arranged to produce a probability of error of approximately 10^{-9}, defined by the ambient electrical environment. There is no noise immunity in the standard systems except for a cyclic redundancy check employed to determine packet integrity.

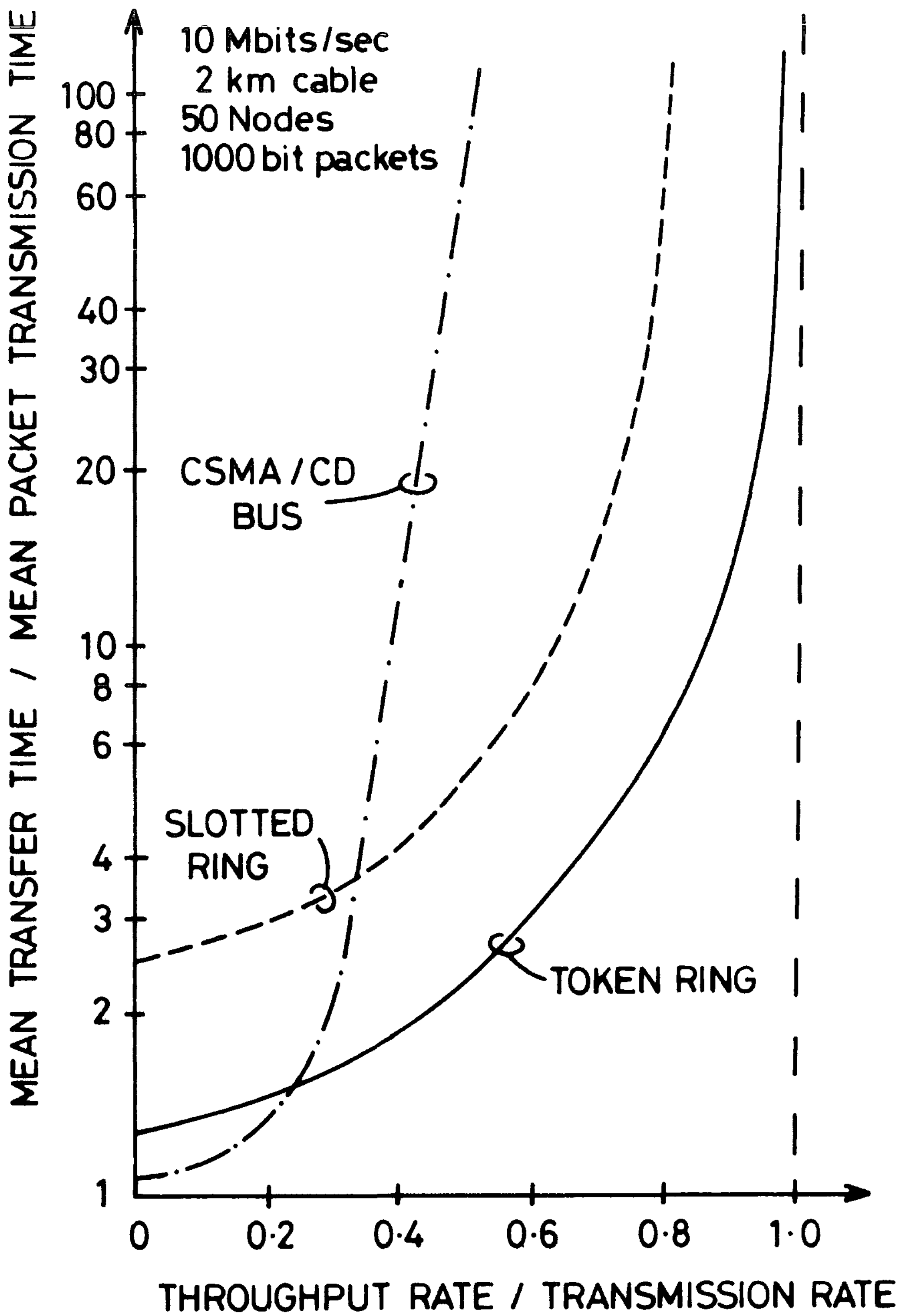


FIGURE (6.14): Transfer Delay Throughput Characteristics

6.7 Conclusion

A simplified analysis of a general SS-LAN has been made to determine typical probabilities of error, signal to noise ratios, data throughputs and data throughput delays. The values of these parameters have been compared with equivalent ones for standard LANs and it has been shown that comparison on equal terms demonstrates the power of the SS-LAN. It does however, also show that the SS-LAN cannot compete with the standard networks on the criteria of data throughput and average delay for medium size packets. It has shown that the most efficient SS-LAN performance can be obtained when:-

(a) the network efficiency is as close to unity as possible - all nodes transmitting simultaneously
(b) the code sequence lengths are as short as possible - maximise point to point data rates
(c) the number of available code sequences should be as close to the length of the code sequence, greater if possible, with as low a mutual interference as possible
(d) the messages should be as short as possible to reduce the message transfer delay - important in real time environments
(e) the DPL should be as large as possible to reduce the network wide delay and to increase the network loading ratio
(f) the DPL should be limited by, $10^{-4} \lesssim \mathrm{DPL} \lesssim 10^{-3}$, so that a high system resolution is available without undue loss of throughput.
(g) the transmission SNR should be limited by, $15\mathrm{dBs} \lesssim \mathrm{SNR}_d \lesssim 20\mathrm{dBs}$, so that a probability of error of at least 10^{-5} is maintained for the data without waste of power
(h) good quality cable is required for the channel so that the near-far effect is minimised

The SS-LAN should therefore be applied in two types of applications:-

(a) Where the messages are short and regular from all nodes and where this information must be transmitted immediately. Such a situation will arise in a real time control environment where the amount of data for transmission is small but where it must reach its destination within a guaranteed time.

(b) Where the system is in general not time dependent but where many nodes will require access to the channel for long uninterrupted periods. The SS-LAN provides the ideal link with the facility to transfer important data immediately without reference to the channel activity.

(c) In systems where there is a significant difference between the average data rate requirements of the nodes. An example is in MMI systems where the human driven nodes will, in general, have a considerably lower data load than those of the machine driven nodes.

CHAPTER 7

Software Simulation of the DS SS-LAN

7.1 Introduction

In an era of rapidly decreasing hardware costs there is a developing tendency to build simple demonstration systems in preference to modelling them with a computer simulation. The usefulness of simulation should never be underestimated especially when a system can exist in a potentially large number of configurations. This is precisely the situation for the SS-LAN and it was therefore appropriate to design a computer simulation which could investigate the performance of the LAN for any physical configuration of the network. At the present time a small scale demonstration system is being built by a separate group and it is hoped that the simulator can be tuned to produce similar results to those of a real system for identical configurations. The simulator can then be used to simulate large and extravagant configurations with some degree of confidence.

The software has been designed, coded and tested according to professional standards (77,78,79) and it is from these that the rest of this chapter gains its structure. The next section will describe currently accepted methods of LAN simulation and will then describe modifications to these for use in the SS-LAN simulation. The following sections will present the requirement specification, the functional specification and design specification (as laid down by the relevant standards) with the next section discussing the validation of the design with respect to the requirement specification. For the sake of brevity not all of the design specification has been included and only its top most levels are presented. The final section describes the contents of a set of example files which contain the configuration instructions and the corresponding simulator readings.

7.2 Simulation Techniques

Most simulators rely upon a set of assumptions which are used to provide a simplified model of the real system while still maintaining its general tenet. In the case of LANs and the simulation of their physical layer the most general assumption applied is that of specifying that the transmitter of every node and the receiver of every node are at opposite ends of the physical channel: every message must propagate the entire length of the network. This assumption has several affects on the software model:-

(a) No physical distribution of the nodes across the LAN is necessary.
(b) The propagation delay of each transmission is fixed depending upon the length of the LAN.

In fact the modelling of the physical layer is simplified even further when considering TDMA and CSMA systems because the former imposes a single transmitter at a time and the latter only requires knowledge of either a successful transmission or a contention declaration.

Similar assumptions are made in some SS simulators (80) however the SS-LAN simulator must model the detailed interaction of the code sequences as they propagate along the channel and consequently the node distribution must also be modelled. Each node is allocated a channel of its own, defined as a pipe. Each pipe is allocated a store in core memory and its length is specified by the length of the SS-LAN and the resolution to which the channel will be modelled. This structure can be interpretted as a displacement diagram on which the relative positions of the signal and nodes can be identified. It is in this form that the position of the nodes can be distributed as they would in the real system. Similarly, noise sources are allocated their own pipes because they also function as a transmitting source.

At a receiving node the received signal is the sum of all the signals represented by the pipes (assuming a linearly additive channel) at that physical position and during that time slot. The fact that the signal and noise are present as separate components is irrelevant as far as the operation of the receiver is concerned but it is essential if the channel effects are to be added to the model - thus this channel representation permits the analysis of the simultaneous user environment whilst maintaining, for observation, the integrity of the signals within the system. It is the simulation of the receivers which consumes the bulk of the processing time and so a distributed transmitter arrangement is used to produce the relevant background noise environment with only a few nodes having active receivers; a similar system is implemented by O'Reilly et al (81) where the other nodes are "lumped" together in the form of a noise generator and only a few nodes have active receivers.

The channel can be represented by a distributed LCR arrangement which will affect the signal amplitude and phase, especially in the case of pulses where the component frequencies will become dispersed. Within the simulator this frequency response is modelled as an amplitude effect only, and is implemented as a signal subtraction at each pulse using the transmission frequency as the reference. Another channel effect is that of reflected signals from a poorly terminated link. In the present system it is assumed that the channel is correctly terminated but inclusion of poor termination would be by adding extra pipes to the channel. A similar solution would provide modelling of multipath effects whereby the original signal is duplicated producing a time delayed version. This effect is not common in LANs due to the limited physical environment of the cable and the modelling of it would require an even greater amount of physical memory. In conclusion it can be seen that the detailed modelling of the code sequence transmission provides a flexible method for the investigation of the effects of the channel on the composite signal.

7.3 Requirement Specification

The software requirement is the software equivalent of the requirement specification for the hardware design. It is split into three sections; the first defines the range and properties of the SS-LANs which the simulation must model, the second is concerned with the functions the software itself must provide and the final section describes the results which the simulation must provide.

7.3.1 System Model Requirements

The physical parameters and their ranges which must be modeled by the simulator are:-

(a) the number of nodes on the SS-LAN - between 2 and 100 nodes
(b) the physical length of the bus channel - between 1m and 2kms
(c) the channel bandwidth - between 10Mhz and 300Mhz
(d) the physical distribution of the nodes along the bus
(e) the physical distribution of environmental noise sources along the bus
(f) different types of receivers and transmitters
(g) different types of noise sources - continuous wave, random, pulse interference

The hardware parameters which will be modeled are:-

(a) transmission energies for logics one and zero - different signal shape and voltage levels
(b) different code sequence generators - linear, nonlinear and punctured PN codes, Bent sequences, Gold codes etc
(c) ADC thresholds and techniques at the "front end" of the receiver
(d) correlation thresholds and techniques - sliding PCC and multilevel correlators
(e) sampling rates for the ADC

The system wide parameters which will be modeled are :-

(a) Transmission frequency of the code sequence - 1Mhz to 150Mhz
(b) Code sequence lengths - 2^3 to 2^{16} bits
(c) Code sequence to data bit ratio - between 1 and 10
(d) Bus memory per metre resolution - between 10 and 1000
(e) Variable data bit patterns - random, all zeroes or all ones

The simulator will only analyse the physical layer transactions and will not model the low level protocols. The mapping between the code sequence and the destination will be fixed for a particular SS-LAN configuration and will not be modified during simulation.

7.3.2 Software Requirements

The simulator will provide a non-interactive environment hence all configuration instructions should be presented via a set of data files and all results should also be stored on a set of data files. A list of all the SS-LAN configurations must be presented along with the details of each configuration such that once the simulation system is started it can be left to run to completion without further user information.

Each configuration of the SS-LAN should be capable of employing any of the model requirements and particularly the ability to maintain nodes which employ different types of code sequence generators - essential for the investigation of mixed code family effects. The software must also protect against incorrect SS-LAN configuration information which may take the form of inconsistent information or data outside of the simulators specification. Incorrect data must be reported to a system log but should not cause the termination of the simulation system.

The software will always model a SS-LAN configuration for a specified period of time and with a well defined bus activity - random operation of the network will not be used. Finally, a termination facility will be supplied which will permit controlled interrupt termination of the simulation system. This is necessary because the run time for the system will be several hours and should an undetected system error occur then it will result in wasted time and perhaps the loss of important information regarding the cause of the error.

7.3.3 The Simulator Results

High speed systems transfer a large amount of information in a short time consequently any simulation will also produce a large amount of information concerning the system parameters. The results from the simulator must therefore be a compressed collection of the total information produced or else the software will require large amounts of storage space and spend much of its time transferring the data. The results from the simulation fall into two categories; those which reflect the theoretical analysis of the configuration and those which reflect the accuracy of the simulation to a real system. The results required for the former are:-

(a) The BER of the individual nodes for a variation in the transmitter SNR, the distribution of the nodes across the network (the function

f(P)) and for the effects of the DPL (or DPR)
(b) The BER of individual nodes with a variety of correlation methods, ADCs and thresholds for both the correlators and the converters
(c) The BER for individual nodes when the bus activity is formed by a variety of different code families and augmented by external noise sources
(d) Point to point data throughputs for different code sequence lengths and code sequence frequencies

The results which will reflect the accuracy of the modeling of a real system are:-

(a) The maximum, average and minimum SNR at the input of the receiver for a range of transmission SNRs, distribution functions, code families and number of simultaneous users
(b) The maximum, average and minimum correlation agreement and disagreements for a range of correlation thresholds, correlation and ADC methods, receiver input SNR and number of simultaneous users.

The results will be compiled from the analysis of several simulation runs for similar SS-LAN configurations with only the parameter of interest changed eg the transmitter SNR or the ADC threshold. The simulation system will be capable of producing any required sets of results in a single incarnation but several separate simulations will be used to provide the different result sets. This method is used because the amount of data stored per set of simulations is reduced as is the number of simulations necessary to produce a set of results.

7.4 The Functional Specification

The functional specification describes the basic algorithm for the simulation and defines the format and contents of the input and output data. It is split into four sections which describe the readings from the actual simulator, the I/P and O/P, the algorithm for the simulator and the software high level construction to implement the algorithm.

7.4.1 Readings from the Simulator

The results from the simulation system have been defined earlier and include values for the BER and data throughput. The tables of the results will be stored in separate files depending on their type and these will be produced from the analysis of a set of files which will contain readings taken from individual SS-LAN configuration simulators. The readings which will be taken from the simulations are:-

(a) signal levels and noise levels at the input to the receiver. These will be in the form of a maximum, minimum and average for each message and will be used to generate the SNR at the front end of the receiver
(b) the correlator output which is the demodulated data. This will be compared with the original data to provide the BER.
(c) message start and end times between data transmission and data reception. This will provide data throughput and delay characteristics.
(d) correlator agreements and disagreements in the form of maximum, minimum and average for each message. This will define the performance of the correlation scheme employed.

7.4.2 Data Input and Output

The input for the simulation system will be stored in two types of files: the simulation command files (SCFs) and the simulation scenario files (SSFs). The SCFs will contain the names of all the SSFs which are to be submitted for the complete simulation suite. The SSFs will contain the following:-

(a) System definition - readings required, readings and results filenames, duration of simulation.
(b) Physical parameters - as defined in the requirement specification
(c) Hardware parameters - as defined in the requirement specification
(d) Node parameters - position on the SS-LAN, code generator types and specific node data
(e) Interference parameters - description of all noise sources
(f) Addressing Tables - mapping tables for the code sequence generation information
(g) Node activity - transmission time, length, source and destination

for all messages

This data will be validated in two fashions: the first will verify that the data is consistent with the simulations internal limits and the second will check that all the data is self consistent eg a node can only transmit one message at a time. An example of the structure and contents of the an SSF is shown in appendix five and an explanation of its contents will be given in a later section.

The output data is provided in three distinct sections: the system commentary, the readings file and the results file. The system commentary will display all the significant events within the simulation system's history and will also display any error conditions which may have occurred. It will also provide a schematic diagram of the SS-LAN configuration and will display all internally calculated parameters. An example of the structure and contents of the commentary files is shown in appendices six and seven (for tasks A and B respectively) and an explanation of their contents will be given in a later section. The readings files will be of a binary nature and accessed using block I/O routines in an attempt to minimise the number of disc accesses during the actual simulation. These files will be arranged according to their data types. The results files will be standard for one type of evaluation parameter eg the resultant SNR values for all the nodes and messages in one SSF simulation. The results files will then be analysed by hand until further decisions are made concerning the standard format for the results and their presentation.

7.4.3 The Simulation Algorithm

The algorithm for the operation of the simulation system will be given below in the form of a pseudocode flowchart which is based upon the concepts of a structured language.

"Simulation System Program"

<u>begin</u>
Open the "simulation command file"
<u>while</u> <u>not</u> eof on the SCF <u>do</u>

```
begin
Open the "simulation scenario file"
if open = success
then
  begin
  read and validate the "system parameters", store valid data
  read and validate the "physical parameters", store valid data
  read and validate the "hardware parameters", store valid data
  read and validate the "node parameters", store valid data
  read and validate the "noise parameters", store valid data
  read and validate the "addressing parameters", store valid data
  read and validate the "node activity", store valid data
  if data was valid
  then
    begin
    check the consistency of the data
    if error
    then flag error on system log
    else
      begin the simulation
      for i=1 to total simulation time do
        begin
        find new node activity and prepare the nodes
        propagate all node signals along their pipes
        for j= all noise sources do
          begin
          place noise signal on the pipe
          end
        for j= all previously transmitting nodes do
          begin
          if still data
          then put signal on the nodes pipe
          else terminate nodes transmission activity
          end
        for j= all new transmitters do
          begin
          initialise transmitter
          load in destination code sequence
```

```
            start sending the SS modulated signal
            end
          for j= all receivers do
            begin
            if active and at sample time
            then
              begin
              take sample of total signal at receivers position
              store, if required, signal and noise data
              perform ADC conversion as appropriate
              perform correlation method as appropriate
              store, if required, the correlation data
              store, if required, the demodulation data
              if end of message
              then
                begin
                reset the receiver
                store the readings in the data type readings file
                end
              end
            end
          end of single time pulse
        if no simulator error
        then
          begin
          Open readings file
          manipulate readings to give results
          Open results file
          Store results data
          Close all files
          end
        end of the simulation
      end
    end of this SSF
  endwhile for eof check
close the SCF
terminate the system commentary
terminate the simulation system
```

end of program.

7.4.4 Software Construction

The software will be constructed such that:-

(a) maximal use will be made of core memory and disc accesses will be reduced to a minimum
(b) the node software will be as compact as possible to provide as much space as possible for the bus representation and internal storage of the data readings
(c) all data areas will be tunable at compilation time thus their sizes must be transparent to the general operation of the software
(d) the simulation will run with respect to a simulation time pulse whose duration will be defined at configuratiion time. All operations will be relative to this pulse
(e) termination of the software will be due either to a natural end (or failure) or to an abort request which will cause controlled termination at the current SSF
(f) within a maximum configuration size (set at compilation time) the software will operate in sizes appropriate to the scenario instruction without recourse to task relinking
(g) all node and interference types will be included at task build time and will be available for any scenario without recourse to task relinking
(h) invalid or inconsistent configuration data (from the SSFs) will produce an error report and force the simulation to the next SSF.
(i) access problems to the data files will force the simulation to the next SSF.
(j) runtime simulation errors will be reported on the display commentary and only operating system related errors should cause the software to "crash".

The software environment has been clearly defined and this will be reflected in the following design specification.

7.5 Design Specification

The design specification is only concerned with the higher levels of the design and will not include procedure specifications. It will describe the composition of the software showing how the modularised components perform the functions required and will also show how the design was implemented to run under the host computer operating system.

7.5.1 The System Configuration

The software is broken down into a set of tasks and global data areas (GDAs) as defined by JSP 188 (77) and this is shown in figure (7.1). The functions which will be performed by the tasks are:-

(a) SCENARIO (A) - Access the SCFs and SSFs
Read and validate the data in the SSFs
Perform consistency checks on the data from the SSFs
Display the modelled parameters and the SS-LAN schematic diagram.
Store the data in the relevant GDAs.
Display task activity commentary and error reporting.
Control the simulation system as a whole.

(b) SIMULATOR (B) - Simulate the SS-LAN configuration defined by the scenario.
Display a commentary on the task activity.
Control the task "Data Store" (C).
Store the simulator readings in GDA B%.
Provide all the required readings from the actual simulation.

(c) DATA STORE (C)- Store all simulator readings (B%) on disc files (D%).
Display a commentary on task activity.

(d) ANALYSER (D) - Access all readings and results disc files.
Convert readings information into the required results format.

(e) TERMINATOR (E)- Initiate the controlled interrupt abortion of the

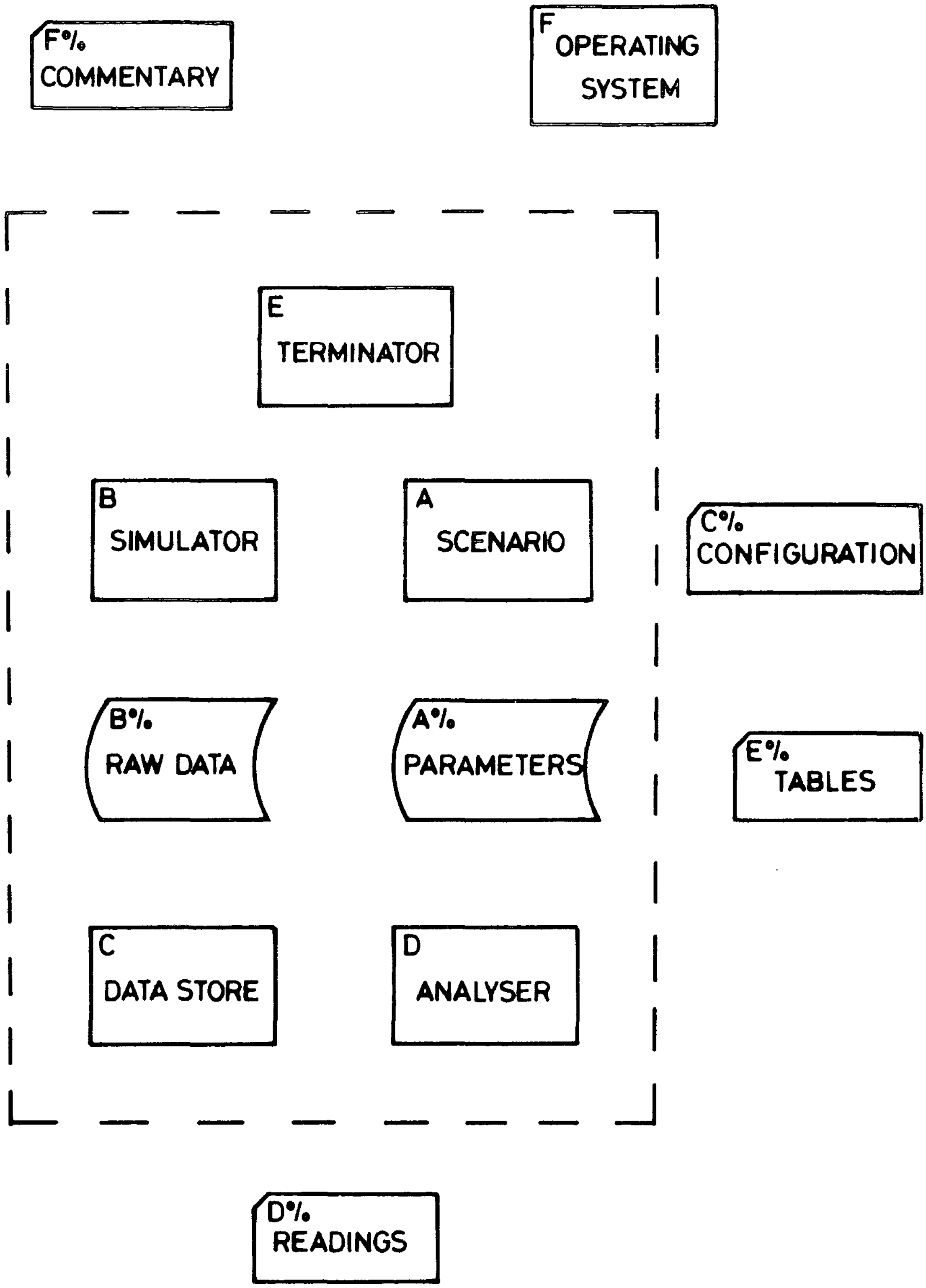


FIGURE (7.1): Task and Global Data Area Block Diagram

simulator system.
Display a commentary on task activity.

The contents and formats of the GDAs are:-

(a)PARAMETERS(A%) - A series of one and two dimensional arrays which contain:-
the system definition parameters eg filenames
the physical parameters eg number of nodes
the hardware parameters eg correlation scheme
the node definition data eg code sequence type
the system parameters eg code frequency
the interference parameters eg noise type
the network addressing tables
the internally calculated parameters

(b) RAW DATA (B%) - Two one dimensional arrays which will contain:-
intertask messages between tasks B and C
Signal and noise levels eg minimum,maximum and average
Received data stream
Correlator agreement and disagreement figures
Message transmission and reception times

(c) CONFIGURATION (C%) - These are Ascii files using a record format and contain:-
The filenames for the SSFs, held in the SCF
All parameters for A% as specified in the requirement and functional specification.

(d) READINGS (D%) - These are files held on disc and are structured in a block data format and contain:-
Readings allocated to these files according to their type.

(e) TABLES (E%) - These are Ascii files and are structured in record format and contain:-
The results allocated to these files according to their type.

(f) COMMENTARY (F%) - These are files (one per task) which are record structured and contain:-
The histories of task activities

Time markings for significant events
Error conditions and messages
Schematic LAN diagram and internally calculated parameters.

7.5.2 The Facility Definitions

The simulation system is split into four facilities (as defined in JSP 188) each of whose constituents are shown in figure (7.2). The four facilities are:-

(a) Scenario Configuration: All the SS-LAN simulation instructions are controlled by this facility. The SSFs are accessed and all data validated before being stored in the A% GDA.
(b) Simulator: The actual simulation of the SS-LAN configurations is controlled by this facility. The production of the readings and their eventual storage in the D% GDA files is also provided.
(c) Data Analysis: the analysis of the readings held in D% is perfromed with the subsequent storage of the results in the E% GDA files.
(d) System Termination: interrupt termination in a controlled manner is provided by this facility. It will terminate all tasks and ensure the closure of all data files, to prevent loss of data.

7.5.3 Description of the Facility Operations

The operation of the four facilities will now be described. A diagrammatic representation of the operation of the scenario configuartion facility is shown in figure (7.3) where:-
(a) The operating system (F) initialises the simulation software and passes control to task A.
(b) Task A creates its commentary file (F%) and requests the name of the SCF, creating A.F%.
(c) This name is provided by the operating system and task A attempts to open the file, creating A.C%.
(d) The first SSF filename is read via A.C% and this file is accessed.
(e) If the file cannot be opened then an error is logged, via A.F%, and the next SSF is approached.
(f) When a valid SSF is located the data stored within it is read in the

	TASK REFERENCE & NAME						GLOBAL DATA AREA					
	A	B	C	D	E	F	A%	B%	C%	D%	E%	F%
FACILITY	SCENARIO	SIMULATOR	DATA STORE	ANALYSER	TERMINATOR	OPERATING SYSTEM	PARAMETERS	RAW DATA	CONFIGURATION	READINGS	TABLES	COMMENTARY
SCENARIO CONFIG.N	*					*	*		*			*
SIMULATOR	*	*	*			*	*	*		*		*
DATA ANALYSIS	*			*		*	*			*	*	*
SYSTEM TERMINATION	*	*	*	*	*	*		*				*

FIGURE (7.2): Facility/Task/Global Data Area Matrix

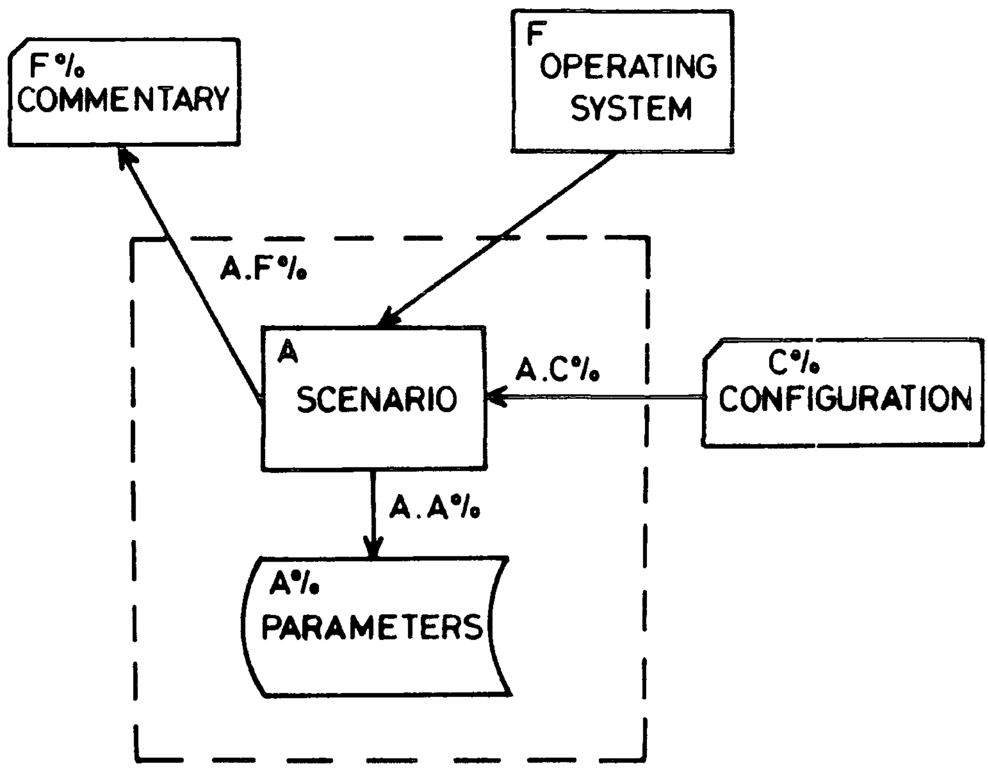


FIGURE (7.3): Scenario Configuration Facility Diagram predetermined order defined by the initialisation of task A.

(g) If the data is found to be invalid then an error message is displayed via A.F%, and the corresponding SSF is closed and the next opened.

(h) Once all the data has been validated it is checked for self consistency. If an error is found an error message is displayed, via A.F%.

(i) The SSF is closed and the consistent scenario is now stored in A% which is configured according to the number of nodes, the number of receivers and the number of noise sources, in A.A%.

(j) The internal parameters are calculated and displayed on the log along with the schematic diagram of the SS-LAN configuration.

(k) Once a successful configuration has been achieved control is passed to other facilities.

(l) Completion of the other facilities results in task A repeating actions (f) to (k) for all the valid SSFs contained in the SCF.

(m) Once the SCF is empty and all SSFs have been analysed then the scenario configuration terminates closing both the SCF and the commentary.

The equivalent operation of the simulator facility is shown in figure (7.4) where:-

(a) The operating system (F) starts task B which then creates its own system log (F%), via B.F%, and then waits for a message to proceed (or terminate) from task A.

(b) The operating system also starts task C which creates its own log, via C.F%, and then it too waits for a message to proceed or terminate. This message will come from task B.

(c) When task B receives the "proceed" message it initialises its own internal data areas and configures the bus representation map according to the data in A%.

(d) Task B then initialises task C which prepares itself for data transfers by opening the reading files whose names are stored in A%. Should an error occur then this is displayed via C.F% and the fail status ripples back to tasks A and B causing the next SSF to be analysed.

(e) After initialisation task B starts the simulation and produces

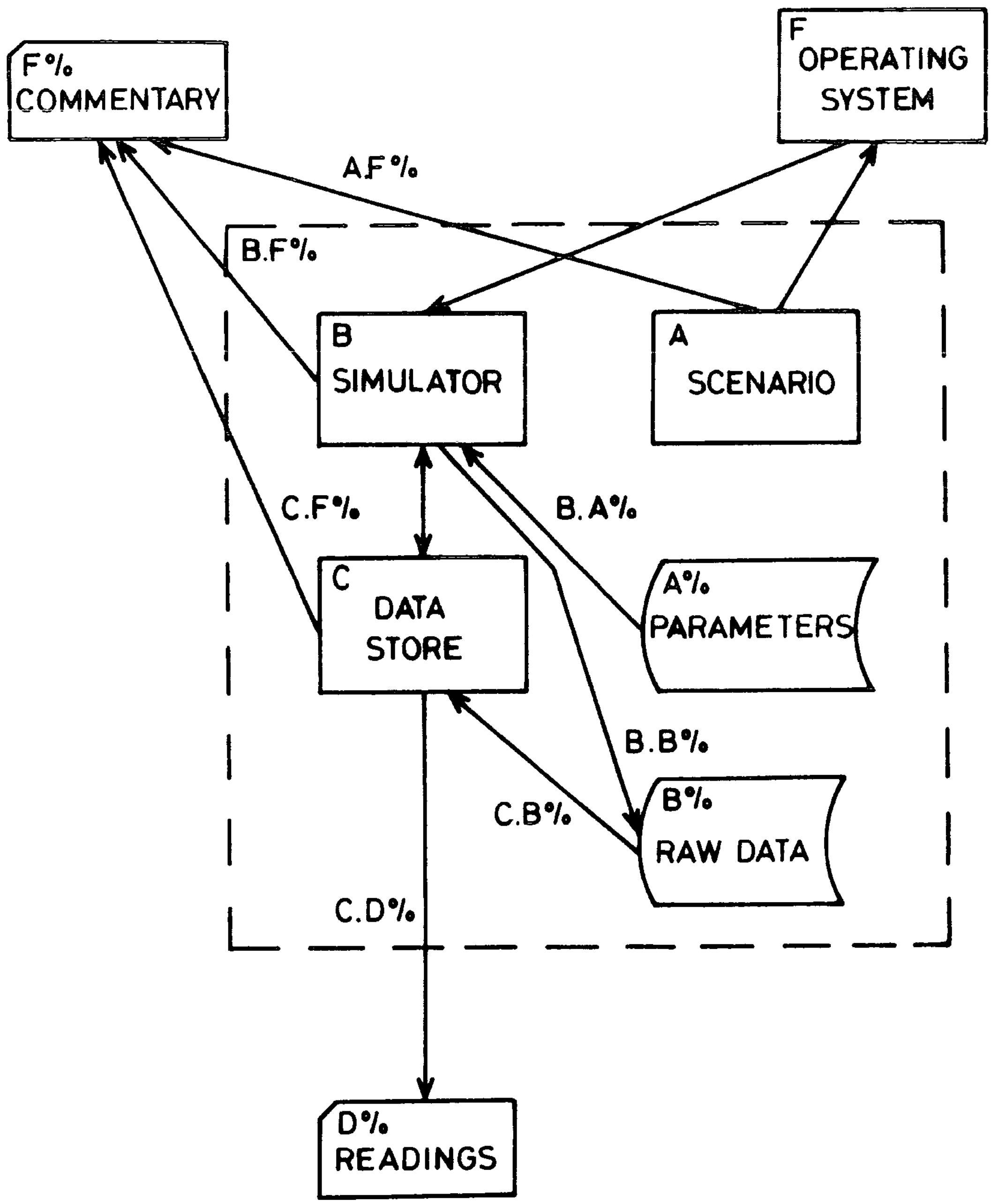


FIGURE (7.4): Simulation Facility Diagram

readings which are stored in B%.
(f) All significant events are recorded in the log, using B.F%.
(g) Once the readings data area in B% is filled task B informs task C using a secondary area in B%.
(h) Task B suspends its activity until task C has emptied the B% GDA.
(i) Task C reads the data from B% and stores it in internal data buffers according to its type. When a buffer is filled this is transferred via a block write to the appropriate readings file via C.D%.
(j) When B% is empty task C returns control to task B and then suspends until further instructions arrive.
(k) Steps (e) to (j) are repeated until the simulation activity is complete - defined at configuration.
(l) Task B causes task C to flush its internal data areas and to put the final data into the D% GDA files. Task C then suspends itself.
(m) Task B returns control to task A.
(n) Steps (c) to (m) are repeated until all the SSFs have been analysed after which tsk B causes task C to terminate. Task C closes its logging file, informs task B of its pending termination and then stops.
(o) Task B closes the logging file and then terminates its own operation.

The operation of the data analysis facility is shown in figure (7.5) where:-

(a) The operating system starts task D which, in turn, creates its own system log, via D.F%, and then waits for a message, to either proceed or to terminate, from task A.
(b) Task A passes the proceed message to task D.
(c) Task D opens the readings (D%) files and results (E%) files as named by the A% GDA and where appropriate.
(d) The readings are read from D%, using D.D% in block format, and stored internally in task D.
(e) The readings are converted into their equivalent results format.
(f) The results are stored in E% as records and are accompanied with descriptive comments to aid readability.
(g) All the files are closed. Task A is informed of the completion of the analysis and task D suspends activity until receipt of the next message.

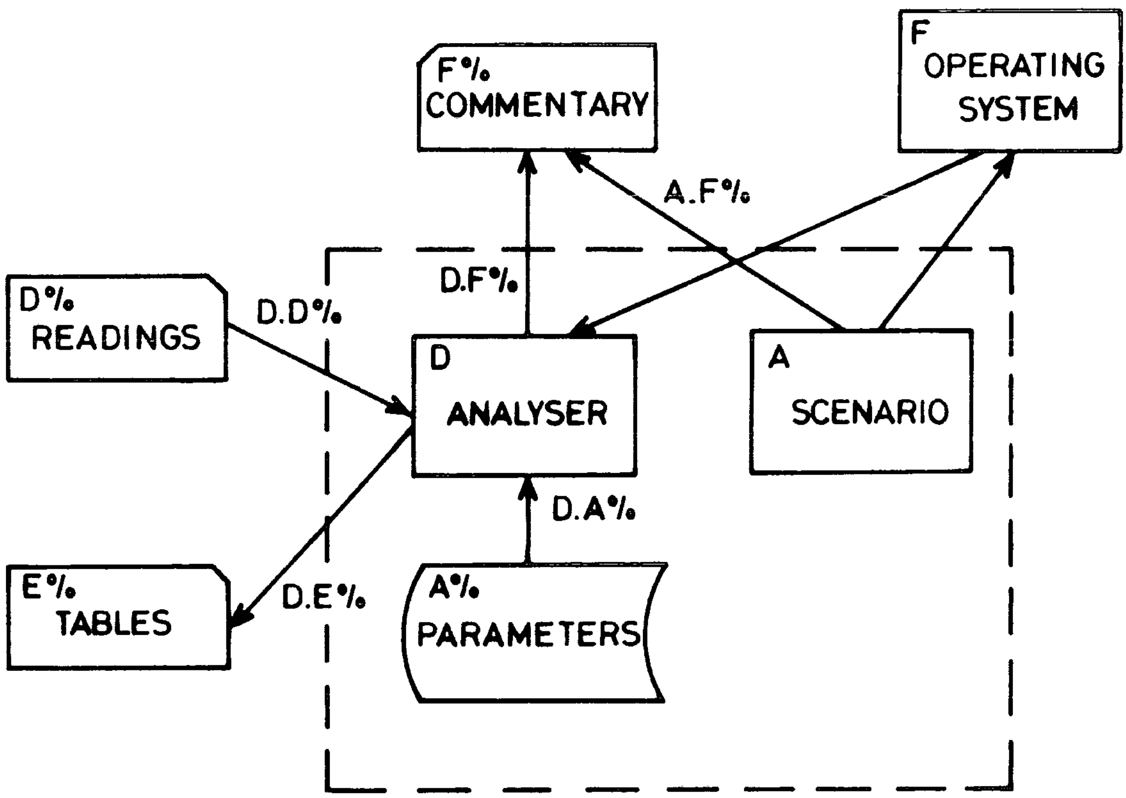


FIGURE (7.5): Data Analysis Facility Diagram

(h) The above actions from (b) to (g) are repeated until the termination instruction is sent.
(i) Task D closes its system log, informs task A of its pending termination and terminates its own operation.

The operation of the system termination facility is shown in figure (7.6) where:-

(a) The operating system activates task E, the termination controller.
(b) Task E establishes its own system log using E.F%.
(c) Task E then informs task A of the termination request and waits for the completion reply from task A.
(d) Task A waits until task B has completed the current configuration simulation and then instructs task B to perform self termination.
(e) Task B informs task C of the termination and they both close down their logging files. Task B informs task A of the pending termination and then terminates its own operation.
(f) Task A closes its logging file and informs task E of the termination completion. It then terminates its own operation.
(g) Task E closes its own system log and terminates its operation. The simulation system has now been interrupt aborted.

7.5.4 Implementation of the Design

The design was coded in Coral 66 (82), the British Government standard language for real time time applications and was run under the DEC operating system, VMS (83), housed in a VAX 11/750 computer providing four megabytes of core memory and full fixed disc support. Each of the five tasks, as defined under JSP 188, were produced as separate subprocesses, under the user "logon" process, and were submitted for control to VMS. The non volatile GDAs C%, D%, E% and F% were VMS files accessed via the VMS file control service linked with the relevant Coral I/O support routines. The two GDAs A% and B% were defined as shared global data areas (SGDAs) which under VMS permits the sharing of global data between processes which are mapped onto the data at run time.

Interprocess communication was provided by the use of group event

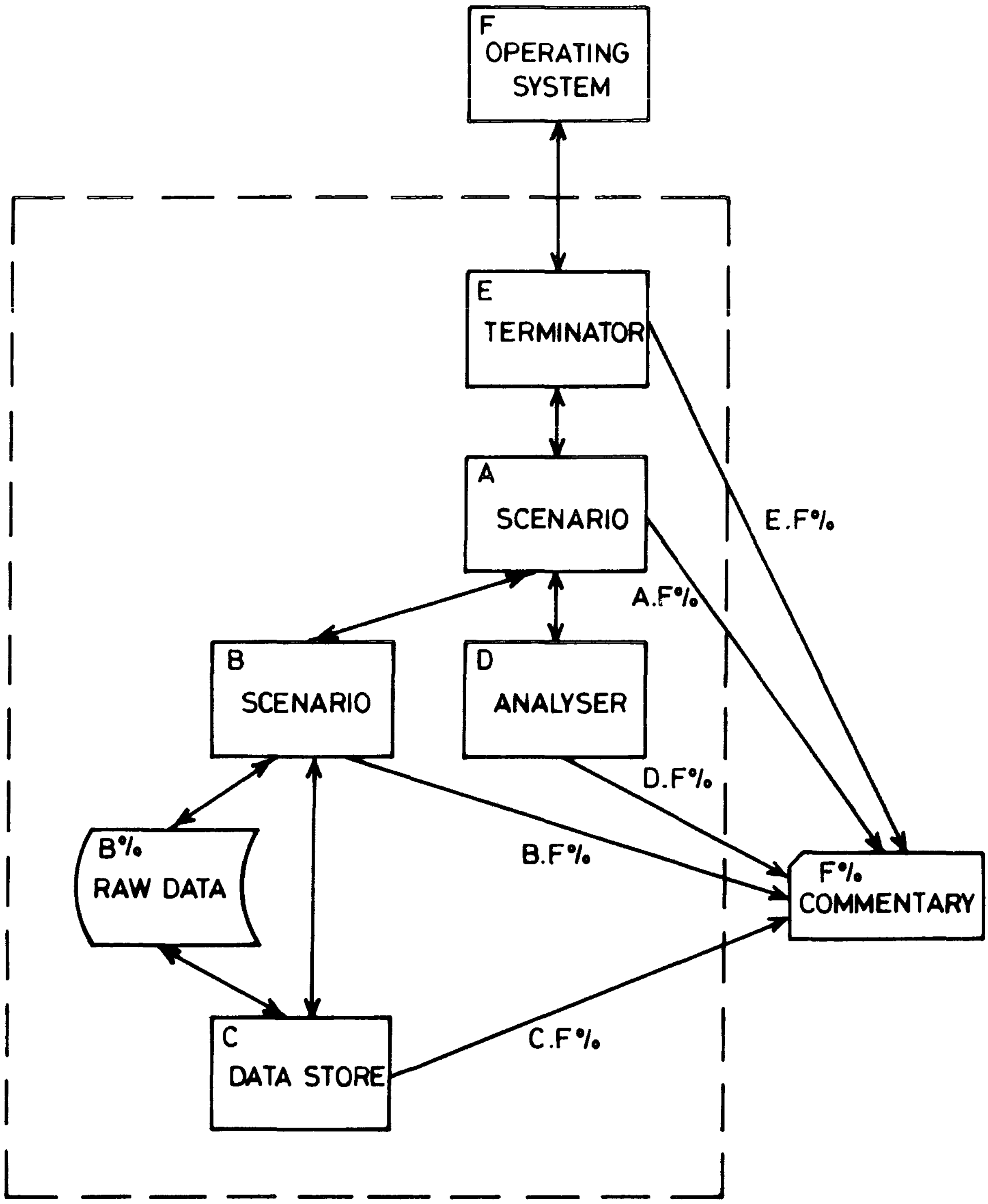


FIGURE (7.6): System Termination Facility Diagram flags. Two event flags were allocated to each process; one for the task proceed instruction and one for the terminate instruction. This communication method was enhanced by the use of SGDAs for passing parameters between processes B and C. Access to the non volatile GDAs was via a set of specialised interface routines which converted the files into a structure useful to the corresponding process and access to the SGDAs was via a set of access procedures, each of which had access to a limited amount of data. These techniques provided the design with some degree of security against access and corruption of global data without severely degrading the system performance.

The complete design has not yet been fully coded and tested however the current version of the software is operating correctly and is producing accurate results; this will be discussed in the following chapter. The software is stored in 20,000 blocks of disc file space and is constructed from 98 source files which contain 157 Coral procedures. Tasks A, B, and C are operational with GDAs A%, B%, C%, D%, and F% fully implemented. Task A requires 40k bytes of core memory, task B 66k bytes of core memory, task C 15k bytes of core memory and the SGDA for A% and B% requires 20k bytes of core memory; the latter must be permanently installed within the VMS core memory.

7.5.5 Testing

The software was tested in two stages: build testing of the modules for each process and integration testing of the simulation system. The build testing included the inspection of each flow path through the programs coupled with a display of the contents of accessed data areas and a log of the entry and exit from each procedure.

The integration tests were presented in the form of controlled SSFs. The results for these configurations were determined, by hand, before application to the software and the simulator's results were then compared with those expected. This was repeated for several SS-LAN test configurations with the conclusion that the software appeared to perform correctly; it is expected that continual use of the simulation will find software errors which can be corrected - it would be impossible to exhaustively test the software beforehand.

The test software was not removed from the simulation system's software. A facility in the Coral 66 permits the inclusion or exclusion of certain parts of the language at compilation time. This provides two important effects: the first is that once complete the software need not be changed (except for small library redefinitions) when being tested or simulating and, secondly, that the run time does not deteriorate due to the inclusion of the test software because it is effectively commented out at compile time.

7.6 Design Validation

The validation of the software design is similar in its purposes to the validation of the hardware design. The requirement specification will be compared with the design specification and while the latter is not presented in its entirety it is still possible to compare its high level function with those of the requirement. The validation will be given in two sections: the certification of the software design and the certification of the software implementation.

7.6.1 Certification of the Software Design

A wide range of parameters are specified for the physical, hardware and system entities in any single SS-LAN configuration. This flexibility is reflected in the design by the use of SSFs which tailored a general purpose SS-LAN simulator to the needs of a particular SS-LAN. The limits of the individual parameters define the total size of the simulation software and the configuration parameters are compared to their minima and maxima.

This approach provides each configuration with the complete range of SS-LAN facilities hence a configuration could have all of the different code sequence generators etc which were required for the analysis of mixed code systems. The validation of the data before storage and the consistency evaluation before simulation mean that incorrect data is detected as early as possible without affecting the run time operation of the simulation. The design also implements the system log for error reporting with the recovery from error detection causing the simulation of the next configuration. The interrupt facility is also provided but termination of the system is not achieved until the current configuration simulation has completed - this ensures consistent data files and a logical shut down.

The simulation analyses bus activity which is predetermined and presented at configuration time. The results produced in this situation have a clearly defined environment and consequently this optimises the time taken for the production of useful results. The results are stored in data files according to their type and are produced from the relevant readings of the simulator. The readings provide the received messages - the BER can therefore be calculated, and the transmission and reception times - to calculate the data throughput and delay characteristics. Similarly the SNRs and correlator values are also provided to define the accuracy of the modelling technique. In conclusion the design provides all the relevant readings to produce the required results as given in the requirement specification.

7.6.2 Certification of the Software Implementation

The software resides in the core memory of the VMS operating system only when the processes are active. The SGDAs are automatically installed in memory whenever a mapped process is active. The actual simulator process is configured to provide a bus representation which is as large as possible - the total size of all the SGDAs, process B and the access procedures for F% must be smaller than the user memory provided by VMS. As the processes communicate via the event flags in VMS (these are maintained by VMS) it is permissible for the processes to be loaded to and from disc when inactive. Disc swapping is relatively slow and therefore it is performed as infrequently as possible, but it is necessary due to the nature of the large data areas required for the SS-LAN configuration instructions, the readings and the bus representation.

The provision of data files for the GDAs C%, D%, E% and F% secures the data in non volatile memory and the occurence of a system crash during simulation will not cause the loss of all information. The system log F% could in fact be printed on a hardcopy terminal which would provide an instant reference to the current simulation activity.

7.7 Realisation of the Simulation I/O Files

Appendices five, six and seven (these will be refered to in the future as A5, A6 and A7 respectively) contain examples of the simulation I/O files for one individual SS-LAN configuration simulation. A5 shows the contents of the SSF for the SS-LAN configuration while A6 is the system log produced by task A during the reading and validation of that SSF. A7 is the task B system log which is created during the actual simulation of the network and its activity as instructed by the SSF. The detailed operation of the software will now be explained with respect to the SSF and system logs to demonstrate the interaction between these files and to show how their contents are related to a real system and the simulation system.

Both tasks A and B are installed as memory resident within VMS. Once processing has begun then tasks A and B create their system logs (A6 and A7 respectively). This is shown in A6 and A7 by the logging of the "Start up time:" and the announcement of the start of the tasks, "TASK B STARTED". Task B now suspends awaiting the instruction to proceed from task A. Meanwhile task A performs its self initialisation and opens the SCF, "Command file open", which was specified at run time. The first named SSF is then opened, "Simulation run file open", and the relevant data storage is initialised. Task A is now ready to read and validate the contents of the SSF (A5). The statements, in A6, from "Set param reading" to "Noise param complete" log the sequence of the data retrieval and the state of the primary validation of the data.

The first data elements are responsible for the configuration of the data structures within A% and this is reflected in the contents of A5 which state that the network willbe configured with 8 nodes, of which only 3 will have active receivers and that there will be one external noise source (not implemented in the present version of the software). The statements on the right hand side of the exclamation marks are comments on the type and range of the accompanying data. The "MODE" is not used in the present version of software. The type of readings, "READINGS", specifies which readings must be generated by the simulator and therefore they also denote which files must be opened for the storage of this information. In this example all of the readings files are required but none of the tables files are necessary; each file stores the data pertaining to one type of reading.

The "TOTAL TIME" denotes the number of pulses which must be simulated (the pulse duration is determined later) and the "HIGHWAY LENGTH" specifies the total length, in metres, of the channel. The "RESOLUTION" specifies the number of core memory words which will represent each metre of the channel (ie the precision by which the channel can be represented). The "CHANNEL" statistics are not currently implemented however in the future these will be used to determine the channel dispersion and decay effects. The "SEQUENCE RATE" denotes the code sequence frequency, R_C, and the "SEQ/DATA RATIO" is the number of code sequences per data bit, m.

The "ADDRESSING TABLES" define the network addressing scheme (this is in the form of the unique receiver mode) and so there must be one entry for each node, eight in this case. The address generation instructions assume that the code sequences are m-length PN codes and they therefore define the sequence length, shift register length, shift register start state and the feedback taps mask (defined for the modular PN generator configuration). The "NODE DEFINITION" contents describe the physical structure of each node; again there must be 8 entries. The data entries are the position of the node on the highway, the type of transmitter (code family type), the voltage levels for the transmission of logic one and zero and the receiver state - only nodes 0, 2 and 4 show active receivers. For each active receiver the data specifies the receiver type (code family type), the ADC threshold (it is assumed that a single bit hardlimiter is employed) and the correlation threshold (an absolute value which must be equal to or less than the code sequence length).

The final set of information describes the transmission activity on the network. This data is presented under the heading "NODE ACTIVITY" and includes the start pulse number for the commensuration of data transmission, the number of nodes which start transmitting new data during that pulse, the source node number, the destination node number and the number of data bits to be transmitted; the last three items are supplied for each transmitting node. This information is specified for each pulse whenever a new transmission is required and the termination of this list is denoted by the presence of -1 in the "Time of Tx" slot.

Once task A has read and validated the data then it generates all the internal parameters and displays (A6) a selection of useful parameters which define the code sequence chip period, the propagation delay for the total length of the highway, the pulse period and the total simulated time. It also shows the total code sequence period for sequences of length 127 and 1024 bits and lastly, the number of simulation pulses which will represent the propagation of one chip. Task A now validates the self consistency of the total data structure and announces the completion of the data parsing, "Data Parse Complete", at which time it reawakens task B for the simulation of the defined scenario.

Task B initialises its internal data areas and logs the start time of this specific simulation. The simulation is now started and the completion of each 10,000 simulation pulses is logged. In this example A7 shows that the results are produced between pulses 10,000 and 20,000. The reading types displayed are dependent on those specified in A5 (in this case all of them) and here the SNR, correlator, timing and data readings are shown for nodes 4,0 and 2 in that order (their receivers were specified as active) - the actual readings produced will be explained in the following chapter.

Once the period of 20,000 pulses has been simulated then the completion time is logged, "Simulator finish time:" and control is returned to task A - task B suspends processing. The SCF for task A contains only one SSF. Task A will therefore detect the EOF of the SCF and this will drive it into its termination algorithm. Task B will be instructed to terminate and this will result in the closure of its system log as denoted by the time stamp of its final operation (A7). Finally, task A closes its own system log (A6) and the simulation of the scenario defined in the SSF(A5) is complete.

7.8 Conclusion

A general SS-LAN simulation system has been designed, coded and tested. This simulation will analyse the performance of particular SS-LAN configurations, which are defined at run time via a set of files, and provides the results in a comprehensive set of data files. The software was designed using a structured top-down approach and was implemented using Coral 66 running under VMS. The simulation design has been validated with respect to the requirement specification and has been thoroughly tested at both individual module and integrated system levels.

The full implementation of the design is not complete however the current software version is operating and producing accurate results. The total software core memory size is 141k bytes to which must be added the system core memory size of 140k bytes. The software has run continuously for a maximum of 57 hours and its performance during these periods has been as expected with no software failures or system "crashes".

At the present time the design has been implemented only on a VAX 11/750, however the top most levels could be implemented on any system with sufficient memory and an adequate operating system. Certain software changes would be necessary but these would be limited to the operating system interface. Similarly a different language could also be used but this would have a more significant effect on the detailed design. At a later stage the software will be extended to include the analysis of standard LAN configurations eg Ethernet and Cambridge Ring, and this will provide further insight into the accuracy of the models implemented by the simulation.

CHAPTER 8

Simulation Results and Discussion

8.1 Introduction

The computer simulation of the SS-LAN was designed to simulate a set of related but different LAN configurations. The individual simulation runs produce a separate set of readings which give the SNR, number of errors, data delay times etc for a particluar number of users, with the same code length and with a certain DPL etc. It is this set of readings which are collated to give the results from the simulator. These results are in a form similar to the graphs produced in chapter 6 which discussed the theoretical analysis of a simple SS-LAN configuration. The results from the simulation should closely ressemble those from the theory as regards their general shape and the relative displacement of the curves on each graph however they should not be precisely equivalent as there are different types of assumptions made in each case.

This chapter will compare the results obtained from the computer simulation with those of the theoretical analysis. The first section will describe how the readings from the simulator are collated to form the results analysed in the following sections. The next section presents the results as a set of graphs whose forms are similar to those derived in chapter six and the following section compares and contrasts these with that theoretical analysis. It then proposes an explanation and solution for the discrepancies between the two. The final section discusses the repercussions on the SS-LAN system design and the computer simulation prompted by the discrepancies between the simulation results and the expected system performance.

8.2 Simulation Readings to Results

The analysis in chapter six derived a relationship between the SNR at the output of the receiver, the number of users transmitting simultaneously and the code sequence for the system. The output SNR_i cannot be measured directly (it can be determined by its effect on the probability of error) and therefore it would provide insight into the performance of the SS-LAN and the accuracy of the simulator if this could be related to some physically measureable property of the receiver. Equations (6.9) and (6.10) define the $\overline{SNR}_i$ as:-

$$\overline{SNR}_i = \frac{2mL}{\pi} SNR_c \tag{8.1}$$

The analysis in appendix four shows the derivation of the SNR_i as a function of the correlator threshold and the probability of successful conversion by the hardlimiter. In the case where this probability tends to 1/2, as would be expected in the steady state for the simultaneous users, then equation (A4.10) is true.

$$\overline{SNR}_i = \frac{C_t^2}{L} \tag{8.2}$$

Equation (8.2) is normalised for a single code sequence to data ratio. If the factor, m, was included then C_t would have to be amended to incorporate an increased sequence length, thus C_t would be limited as $-mL \leq C_t \leq mL$. If equations (8.1) and (8.2) are equated then:-

$$\frac{2L}{\pi} SNR_c = \frac{C_t^2}{L} \tag{8.3}$$

Incorporation of the definition of C_t from appendix four means that if $C_t = AG - DG$ then:-

$$C_t^2 = (2AG - L)^2 \quad \text{or} \quad C_t^2 = (L - 2DG)^2 \tag{8.4}$$

Substitution of equations (8.4) into (8.3) mean that:-

$$AG = L\left(\frac{SNR_c}{2\pi}\right)^{1/2} + \frac{L}{2}$$

$$DG = -L\left(\frac{SNR_c}{2\pi}\right)^{1/2} + \frac{L}{2} \tag{8.5}$$

Within the PCC the data is declared as zero if the, AG, value is greater than some predefined threshold and is defined as a one if the, DG, value is less than some predefined threshold. These two thresholds are usually defined symmetrically about the L/2 position of the correlation curve.

A graph of the equations of (8.5) is shown in figure (8.1) where the input SNR_c is plotted against the normalised agreement and disagreement count. The salient point on the x-axis is when the maximum threshold is achieved at $SNR_c=\pi/2$. In reality this will not occur because it breaks an earlier assumption that the probability of successful decoding is P=1/2, which is itself only true if $SNR_c \ll 1$. The significance of equations (8.5) are that the values AG, DG, L and SNR_c can be provided directly by the simulation and so the simulation performance can be compared with that of a theoretical analysis. An extension of this is possible when equation (A3.8), from appendix three, is incorporated. The correlation thresholds may then be analysed for their relationships with the number of simultaneous users, the near-far effect and the single user SNR.

The simulation system cannot measure the value of $\overline{SNR}_i$ but it can calculate the bit error rate and consequently the probability of error. The $\overline{SNR}_i$ can then be calculated using the inverse relationship of equation (6.20). The graphs shown for the $\overline{SNR}_i$ and $\overline{P}_i$ in chapter six can now be recreated using readings from the simulator and whilst the simulator is not expected to produce identical plots there should be considerable similarity in their shapes with only the absolute values being different. In fact the theoretical analysis provides the best case results and therefore the more realistic simulation scenario should

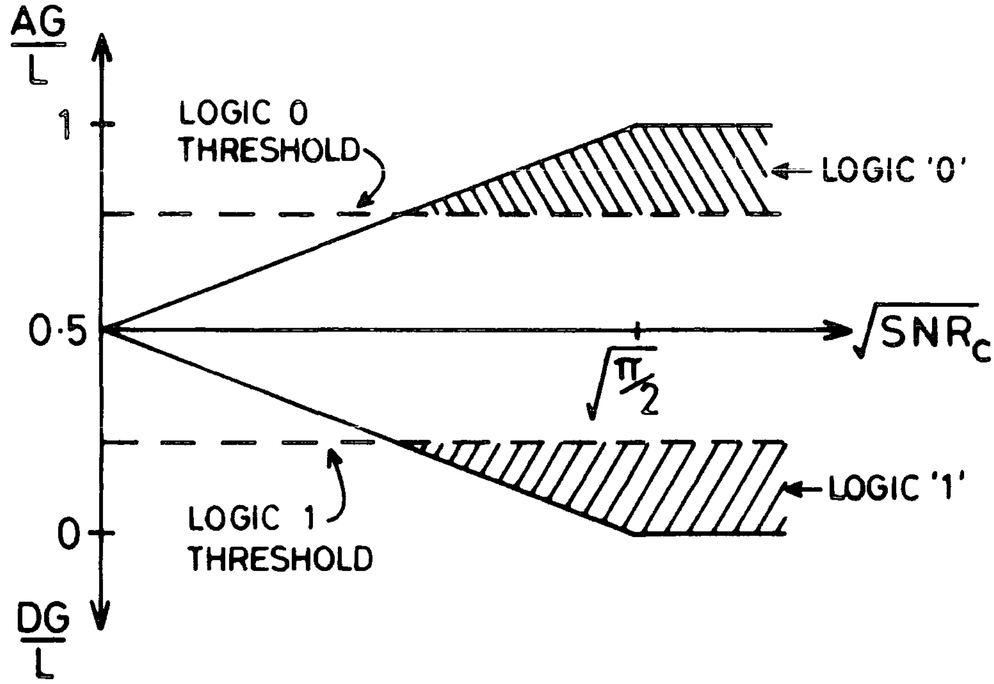


FIGURE (8.1): Correlator Threshold Relationships produce smaller SNR_c and larger $\overline{P}_i$ values.

Throughout the analysis of the receiver it has been assumed that the ADC is achieved using a simple hardlimiter: the consequence of this is a reduction in the output SNR by 3dBs. The receiver performance can be improved by employing a multibit ADC converter eg four bits giving 16 levels of comparison. The correlation then involves the weighting of the correlation function depending on the most significant values. The corresponding correlation circuit is four parallel correlators each comparing the local sequence with the waveform shown by one of the bits from the ADC and the final correlation value is the weighted sum of all of these subcorrelations. The disadvantage with this method is the significant increase in the complexity of the receiver for a performance improvement of between 10% and 20%.

Another method is to introduce a dither signal to the received signal. Freeman (85) has analysed several types of dither signal for their effects on a sliding PCC system. The aim of this method is to overcome the problem of regulated noise eg a high amplitude rectangular wave, which can overwhelm the simple hardlimiter. The inclusion of both these techniques in a single system has been analysed by Chang and Moore (86) and they have found that the use of a dither signal is equivalent to an increase in the signal quantisation thus it is possible to use a hybrid system which will improve system performance without requiring significantly more complex hardware.

8.3 Results

The results presented in this section originate from the computer simulation of SS-LAN configurations only; they are the result of twenty five different SSFs which were processed over the period of one month - some of the individual SSFs required upto sixteen hours of continuous CPU time. The actual results are produced from the hand collation of the simulation readings and fall into one of two categories: the first is concerned with those which describe the performance of a real SS-LAN system and the second with those which describe the performance of the computer program. This section will describe the limitations of the present version of the simulator, present the graphs produced by the results and will explain the collation method by which the readings were converted to results. The results presented are the variation of the correlation agreements with the channel chip SNR, the variation of the correlator SNR with the DPR, the variation of the probability of error with the correlation threshold, the variation of the relative chip count with the correlation threshold, the variation of the CPU processing time with the data.resolution product and the variation of the normalised CPU processing time with the data.resolution product.

8.3.1 Limitations of the Simulation Software

Whilst the design of the software has been validated with respect to its requirements, the implementation in Coral 66 is not complete and certain aspects of the design have not yet been coded. The absence of these has an effect on the type of results which can be obtained from the current version of the simulator software. Those aspects of the design which have not yet been implemented are:-

(a) There is no modelling of environmental noise, not even that of thermal noise. Consequently the single user SNRs are infinite.
(b) There is no modelling of channel decay and so the transmitted signal will propagate without the loss of power and without distortion of its envelope. This is equivalent to no near-far effect.
(c) Only m-length PN sequences can be generated however there is <u>no</u> restriction on the length of sequence or the feedback taps mask (except as defined by the design specification).
(d) The system operates in the "unique receiver mode" only with <u>no</u> addressing restriction on the nodes.
(e) The data transmitted is always logic one and so the successful reception of an entire word will be represented by the numerical value of minus one (-1).
(f) A compiler restriction has limited the highway representation to a maximum core memory size of 32k bytes. This means for example that the system can model a maximum of ten nodes coupled with a resolved distance of 800 words, ie 100 metres with a resolution of 8 words per metre - the highway is represented by an integer array hence each word requires 4 bytes of core memory under VAX VMS.

The modifications to the software will be made in a later version but it should be noted that the present software version will still produce worthwhile readings which can be related to an actual SS-LAN configuration.

8.3.2 Correlation results

Figure (8.2) shows the variation of the average number of correlation agreements/disagreements normalised with respect to the code sequence length (AG/L and DG/L) with the root mean of the channel chip SNR ($\sqrt{SNR_c}$) - the latter is the channel SNR derivative before hardlimiting has been applied. Figure (8.2) is therefore the empirical equivalent of figure (8.1). The circled points represent plots where the received data was in error and the asterisk denotes a plot which was produced by several coincident results. The dotted curve represents the "ideal" correlation threshold for a given SNR_c, as derived in an earlier section, but it should be noted that the noise characteristics of the system are theoretically modelled using the variance of a Gaussian distribution (appendices three and four) and not an absolute value. The vertice at the point SNR_c=0.25 will be referred to in the section (8.3.4) (the results for the probability of error) and the vertice at SNR_c=1 is the plane at which the signal and noise average powers are equal.

The values of SNR_c are calculated from the individual SNR_cs of each chip but averaged over the entire message length. Consider, for example, the transmission of ten bits of data using a code sequence length of 127 bits and modulating each data bit with a single code sequence period. The number of chips transmitted is 1270 and consequently the reciver must detect 1270 chips - each chip being detected with an appropriate SNR. The average SNR_c is therefore the sum of all of the individual SNR_cs divided by the number of chips. The maximum SNR_c and minimum SNR_c are also calculated however these have not been presented and the reason for this will be discussed in the following section. The normalised correlation values are calculated in a fashion similar to that for the SNR_c. Each chip reception produces a corresponding correlation value which if it surpasses the threshold represents the detection of data. It is the correlation values which

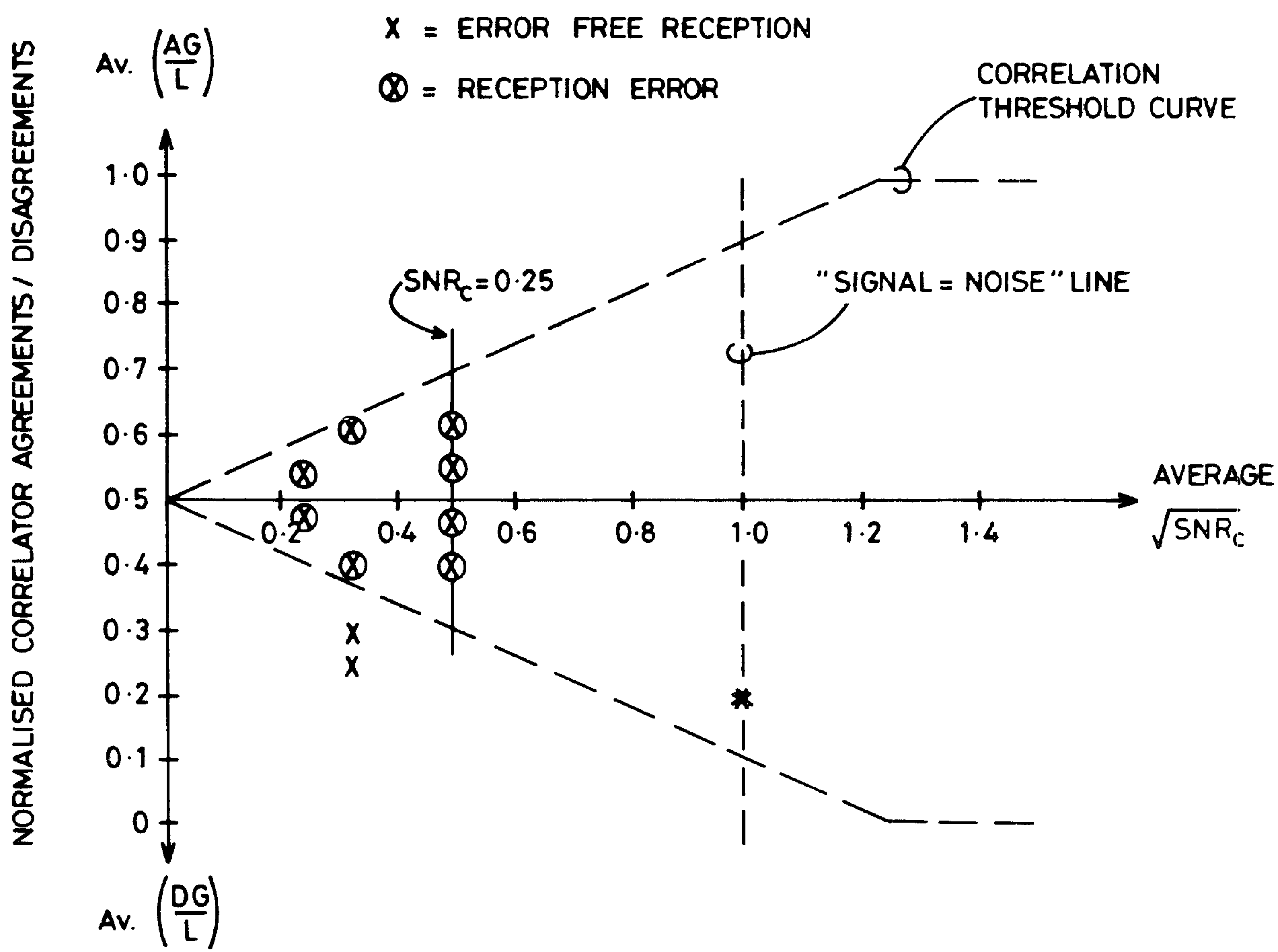


FIGURE (8.2): Correlation Agreements vs Channel SNR surpass this threshold that are forwarded for the average correlation results - the average being determined over the entire message reception. The correlation value is the difference between the number of agreements and disagreements and so each value can be devolved into its two constituent elements. It is the symmetry between the number of agreements and disagreements which is used to represent the data contents of logic one and zero and gives rise to the symmetric form of the correlation threshold curve. Likewise, it produces the symmetric plots for the reception of false data and the asymmetric plots for the correct data (only logic ones are transmitted by the simulator) as shown in figure (8.2).

The average, maximum and minimum SNR values are shown in appendix seven (under the heading for each active receiver) and are incorrect due to a software error and therefore the SNR_c results have been calculated from the average signal and noise values (these have be validated manually) which accompany the invalid SNR values. The AG and DG values are given in appendix seven as the "Average No. of 0s agrees" and the "Average No. of 1s agrees" respectively and the code sequence lengths are defined in appendix five under the heading of the "ADDRESSING TABLES".

8.3.3 SNR Results

Figure (8.3) shows the variation of the $\overline{SNR}_i$ at the receiver output (C_t^2/L as derived in appendix four) with the DPR. Figure (8.3) is therefore the empirical equivalent of figure (6.8) in which there is <u>no</u> near-far effect. The asterisks denote points at which several plots were coincident and it can be seen that a "good" straight line fit can be made between the points.

The DPR is the inverse of the DPL in decibels and is defined by equation (6.4), DPR=mL/k with the usual definitions applying. Each of the derivatives for the DPR is available from the relevant SSF as can be seen in appendix five. In this SSF the 'm' is defined as 1 by the "SEQ/DATA RATIO", L is defined by the "ADDRESSING TABLES" and is 127 for each node and 'k' is defined as 3 under the "NODE ACTIVITY". The parameter C_t^2/L is the square of the average difference between the

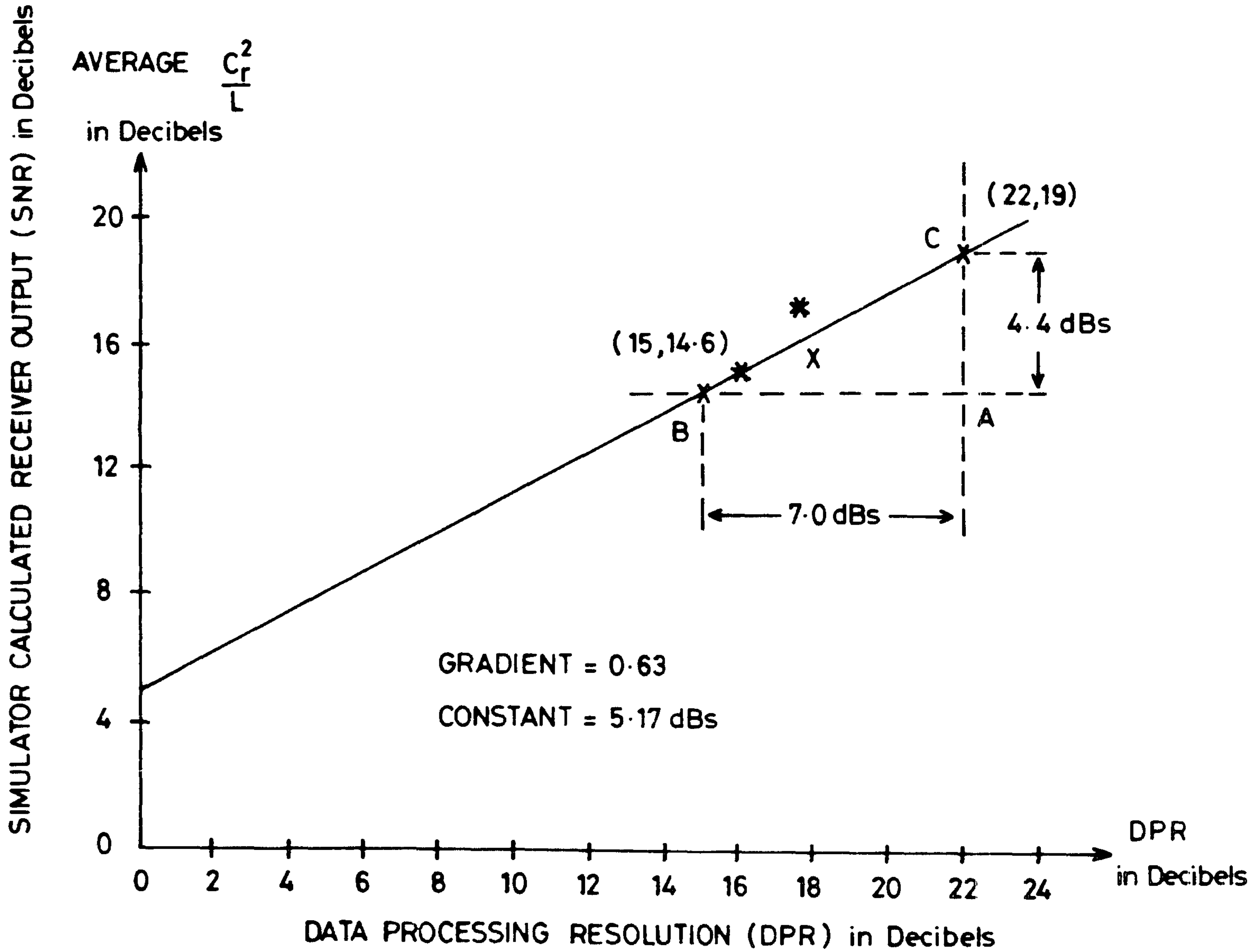


FIGURE (8.3): Average Cr^2/L vs DPR with NO Near-Far number of agreements and disagreements in the correlation sum, normalised with respect to the code sequence length. In the case of figure (8.3) the correlation threshold is defined as the "ideal" for the average SNR_c (using figure (8.1)) and all of the points shown were cases where there was a zero bit error rate. The value of C_t^2 was derived from the reading provided in the system log of task B. Consider, for example, appendix seven. In this instance, node 0 has a BER of zero (all "Received DATA" bits are logic one) and the corresponding C_t is 65 as shown by "Average No. of 1s difference". L is again retrieved from the relevant SSF and consequently C_t^2/L can be calculated.

The gradient and constant for the straight line can be calculated using the construction of the triangle ABC and it is found that the gradient is 0.63 and the constant is 5.17dBs.

8.3.4 Probability of Error Results

Figures (8.4) and (8.5) are the probability of error plots. There are no equivalent figures in chapter six for two reasons: firstly, there is no environmental noise ie SNR_d is infinite and secondly that for the "ideal" correlation thresholds the probability of error was zero. Figure (8.4) is, instead, a plot of the variation in P_e with the correlation threshold - this can be obtained by inspecting the variation of P_e for only a few data bits thereby dominating the effects of the modified Pursley BER equations. In the case of figure (8.5) this is the first plot which is not directly concerned with the physical performance of the SS-LAN configurations but is an attempt to find a new criteria by which the P_e may be determined ie one in which the data contents of the transmitted and received data do not need to be compared. Figure (8.5) is the variation in the chip error with the correlation threshold.

Both figures (8.4) and (8.5) are plotted from the same data sets. The correlation threshold is specified within the SSF and an example can be found in appendix five where it is defined as 90, for node zero, under the heading of "NODE DEFINITION". The DPR is fixed as 16dBs, the code sequence length (L) as 127 and the channel SNR_c is found to be 0.25 (see figure (8.2)). The probability of error is calculated by comparing the number of ones transmitted (see appendix five under the heading of

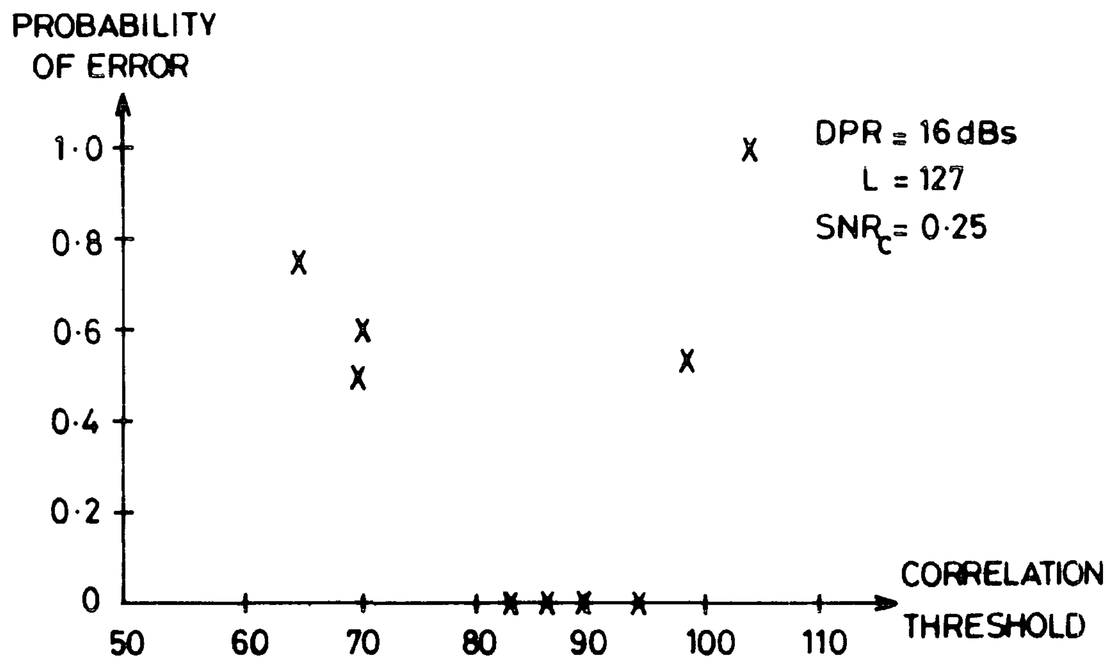


FIGURE (8.4): Probability of Error vs Correlation Threshold

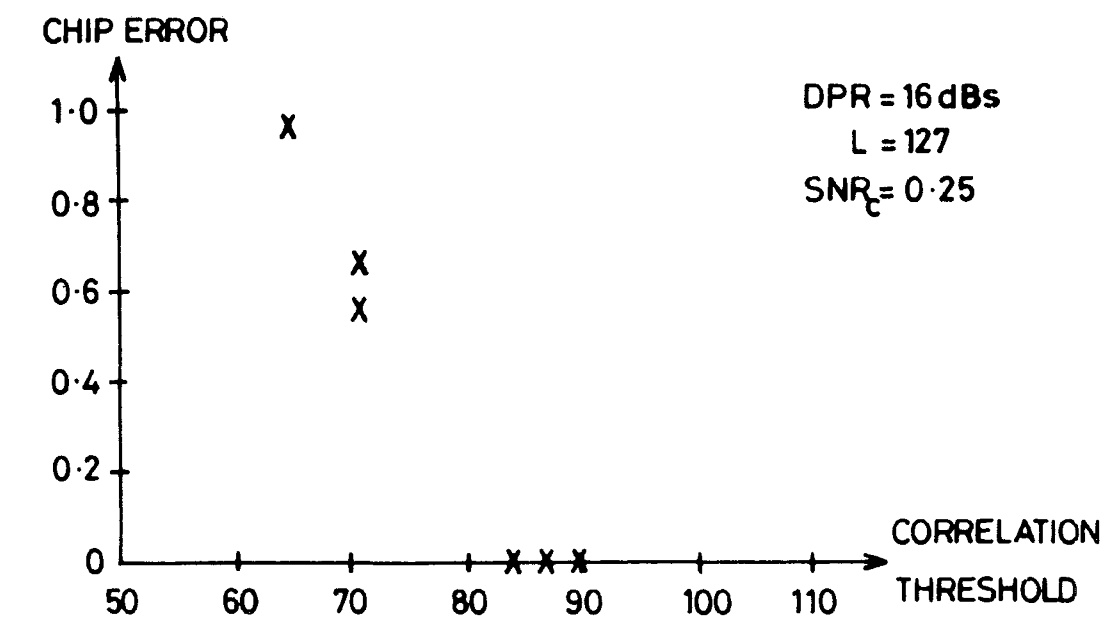


FIGURE (8.5): Chip Error vs Correlation Threshold

"NODE ACTIVITY") against the number received (see appendix seven alongside the phrase "Received DATA" for each node). The corresponding chip error is calculated as the difference between the expected number of chips received and actual number of chips received divided by the expected number of chips. The expected number of chips is equal to the number of chips transmitted (an example of this calculation was shown in section (8.3.2)) and the actual number of chips is the modified difference between the "Start time of data" and the "Finish time of data" as shown for each receiver in appendix seven.

8.3.5 Timing Results

The timing diagrams shown in figures (8.6) and (8.7) are concerned with the performance of the simulator only. They describe the relationship between the physical processing time consumed and the increase in complexity and accuracy of the simulation. Figure (8.6) plots the variation in CPU time (in thousands of seconds) with the data.resolution product and figure (8.7) plots the variation of the normalised CPU time (the normalisation is with respect to the time simulated for a real system) against the data.resolution product.

In each figure the plots are shown according to their DPR and these range from 16dBs to 22dBs. The data.resolution product is the product of the number of data bits transmitted in a single transmission (this is available from the SSF and is shown in appendix five under the heading "NODE ACTIVITY") and the highway resolution (this is also defined in the SSF and is shown in appendix five by the header "RESOLUTION"). The CPU time is calculated from the times shown in the system log for task B. In appendix seven the CPU time is the difference (in seconds) between the "Simulator finish time:" and the "Simulator start time" and the normalised CPU time (or the CPU time/Real time parameter) is the CPU time divided by the real time. The real time is calculated and displayed in the system log for task A under the heading "Modelled simulation parameters" as the "Total simulated run time in micro" and an example of this is shown in appendix six.

In figure (8.6) the DPR=18dBs plots fall on a "good" straight line whose gradient and constant can be calculated using the construction of triangle ABC. The gradient is calculated to be 54.1 and the constant as 399 seconds. Similarly for figure (8.7) the gradient is calculated to be $3.17*10^6$ and the constant as $25.8*10^6$. It should be noted that the points which lie on the line differ only in their data.resolution product and their highway length and that the variation with data.resolution was the only plot, of several attempted, which displayed any correlation whatsoever.

8.4 Discussion

The results presented in the previous section will be compared and contrasted with those produced by the theoretical anaysis in chapter six. The discrepancies between the two will arise for two major reasons: the first is that the theoretical analysis is for a generalised model in which the simultaneous user noise is assumed to be Gaussian in nature (this will not be the case for small numbers of simultaneous user) and secondly that the generation of the readings by the simulation software is not "ideal" and the actual readings may not accurately reflect the simulation.

The discussion is presented in four sections each dedicated to one type of result: the correlation agreements, the SNR, the probability of error and the timing considerations.

8.4.1 Correlation Comparisons

In figure (8.1) the "ideal" correlator threshold is determined for each channel chip SNR, SNR_c. This plot assumes that the SNR_c is constant for each threshold and so as the SNR_c decreases then the correlation threshold must decrease or else it will not be possible to receive the data reliably. If the correlation threshold was set higher than the "ideal" then the receiver will not detect any data because the threshold will never be surpassed and conversley if the threshold is set lower than the "ideal" then the receiver will detect false data because the threshold will be prematurely surpassed. These are analogous to the probability of miss and probability of false reception in other communication systems. The threshold for the channel chip SNR is set (due to the hardlimiting) as $\pi/2$ however this cannot be achieved in the

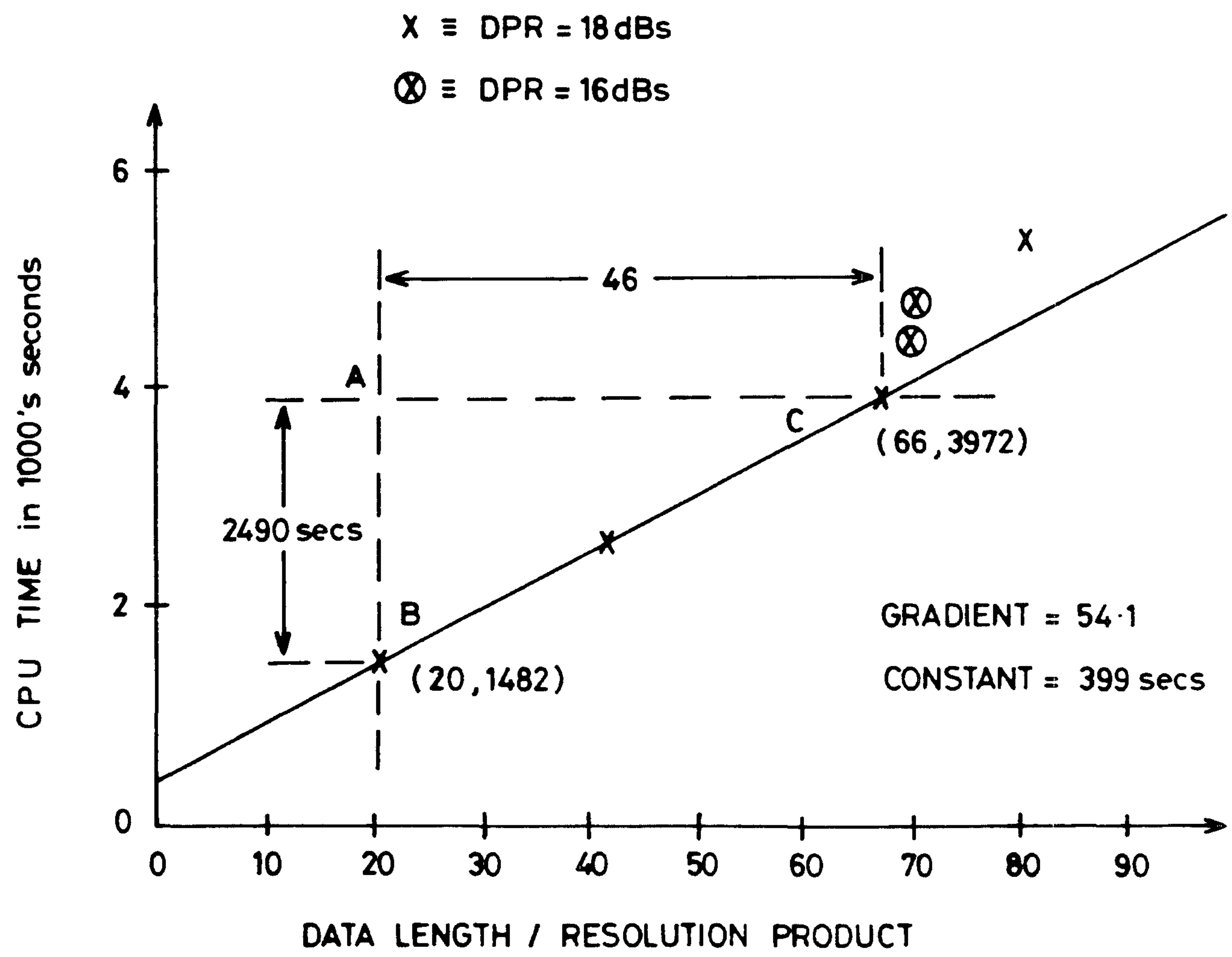


FIGURE (8.6): Absolute CPU Time vs Data/Resolution Product

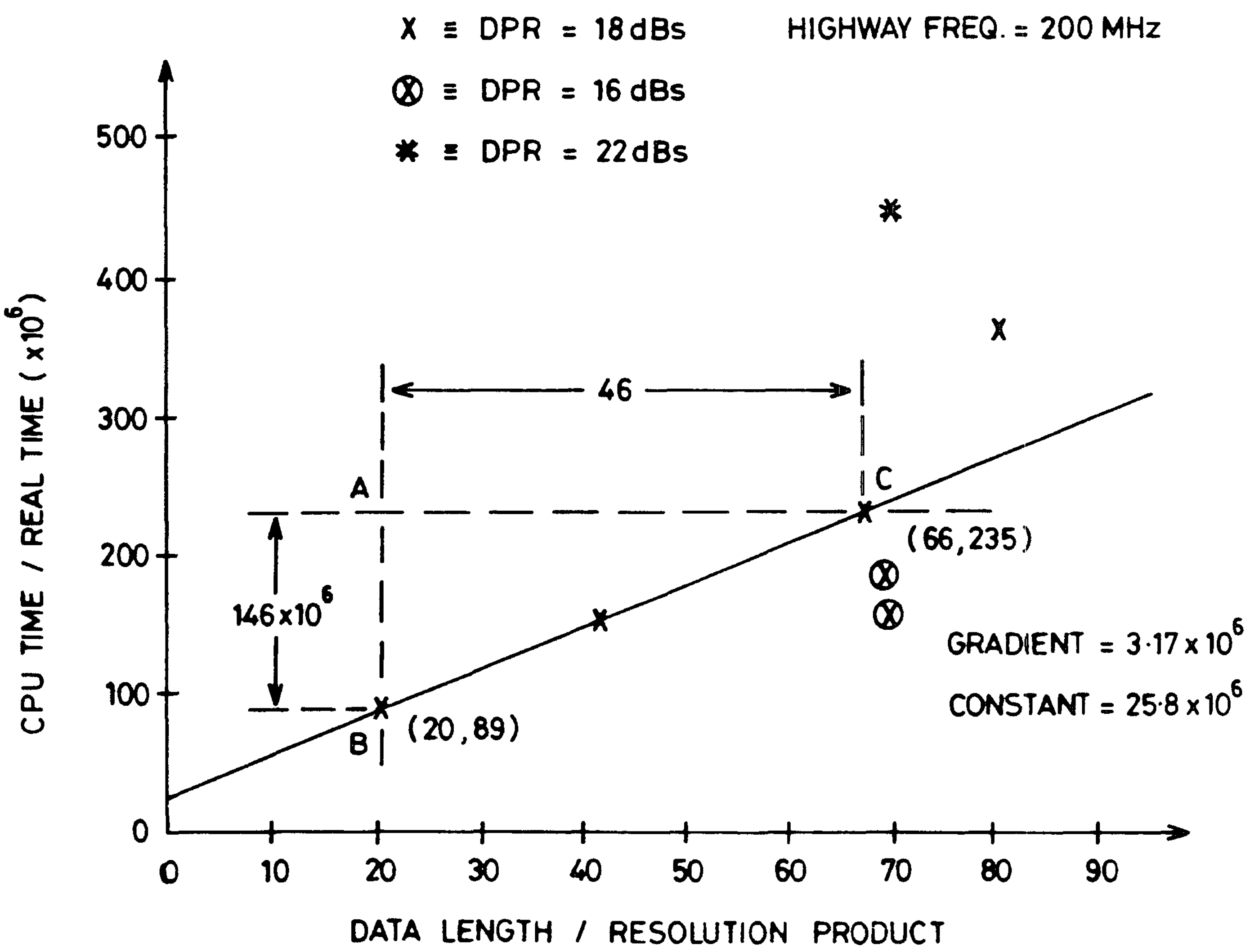


FIGURE (8.7): CPU Time/Real Time vs Data/Resolution Product simultaneous multiple environment - it also violates the basic premise of the previous analysis that $SNR_c \ll 1$ (the effect on $\sqrt{SNR_c}$ is less stringent however $\sqrt{SNR_c} < 1$ must still be true).

The empirical form of this diagram is shown in figure(8.2) and it can be seen that the general consequences of the correlation threshold are upheld. If the number of agreements/disagreements is less than expected for a particular SNR_c then the data is received in error otherwise it is faithfully received. At the point of signal and noise equality it can be seen that this concept is violated however the premise for $SNR_c \ll 1$ is also violated hence the theory will not accurately predict the system's performance. The average correlation agreements/disagreements count must tend towards the correlation threshold specified for that receiver and it can be seen how this varies by inspecting the points on the line, $SNR_c=0.25$ as shown in figure (8.2).

In figure (8.2) the SNR_c is an average hence the true SNR_c will be a band (+ve and -ve) about this point. The variation with this banded region will produce the errors predicted by the modified Pursley equations even when the correlation threshold is defined as its "ideal" value. The SNR_c values are not considered accurate because the number of simultaneous users is always less than 6 (hence a non Gaussian distribution) and this is supported by the fact that the maximum and minimum SNR_c values produce a large banding effect which confuses the plot; it is for this reason that they were omitted.

8.4.2 SNR Comparisons

Figure (6.8) in chapter six is a plot of the modified Pursley equation (6.17) for a near-far effect of 0dBs. In the computer simulation the individual user SNR_d is infinite and so the boundary of $DPL.SNR_d \gg 3$ must be applied changing equation (6.17) into the form :-

$$\overline{SNR}_i \longrightarrow \frac{6}{\pi} DPR \qquad (8.6)$$

when converted to decibels this equation (8.6) becomes equation (8.7) :-

$$\overline{SNR}_i \text{ in dBs} = \text{DPR in dBs} + 10 \log_{10} \frac{6}{\pi} \qquad (8.7)$$

which is a straight line with a gradient of unity and a constant of 3.3dBs.

Similarly, appendix four showed that if $SNR_c \ll 1$ then $\overline{SNR}_i \rightarrow C_t^2/L$ (equation A4.10). The substitution of this term into equation (8.6) is presented in an empirical fashion by figure (8.3) where the constant is calculated as 5.2dBs and the gradient as 0.63. Both figures (8.3) and (6.8) are consistent and this suggests that the simulator is producing results which can be quoted with a good degree of confidence.

When the SNR_d is large then the $\overline{SNR}_i$ is dominated by the mutual interference between the simutaneous users. The Pursley equations are derived for an <u>average</u> noise environment (as will the corresponding receiver and its correlation threshold) and assume that synchronisation has been achieved. Consequently if the noise power was increased above this average some data would be missed ie the error probability would be defined by the corresponding $\overline{SNR}_i$. If the noise was lower then there would be no difference because this would be reflected in a large difference between the defined correlation threshold and actual correlator value ie a relative increase in the sustainable correlation threshold.

The constant value of 3.3dBs in the theoretical analysis is dependent on the reciprocal of the aperiodic autocorrelation function f(R) and the simulation value implies that this is lower than its theoretical value of 1/3 for random codes. The accuracy of this figure is dependent on the values of C_t^2/L which should be higher if the gradient is to approach unity. The values of the correlation threshold define the average C_t and a small error here will become more significant. The value of C_t is set by the SNR_c on the channel and consequently the ideal threshold setting may not have been stipulated.

A similar plot to that in figure (8.3) was drawn for the variation of the term $2mL\ SNR_c/\pi$ with DPR, however no correlation between the two was possible which suggests that there are significant errors in the SNR_c readings and that the accurate generation of these should be investigated. It is the accuracy of figure (8.3) which suggests that the SNR_c reading is at fault and not the analysis. The tolerance of figure (8.2) is attributed to the dependence on the root of the SNR_c reading and thereby reducing the margin of error.

8.4.3 Probability of Error Comparisons

The plots for the probability of error are as shown in figures (6.10) to (6.13) inclusive and these could not be replicated from the simulation plots for two reasons :-

(a) the amount of data transmitted was insufficient to demonstrate a probability of error of 10^{-6} and lower, as is predicted for a large SNR_d and a DPR of approximately 20dBs.
(b) there was no near-far effect or environmental noise to dominate the probability of error and thereby reduce it to the region of 10^{-2} or higher.

The probability of error has therefore been investigated for its dependence on the correlation threshold. It should be noted that the modified Pursley figures of (6.10) to (6.11) do not show this variation as they assume synchronisation has been achieved. Figure (8.4) shows that the P_e curve assumes a V-shape centred about its ideal correlation threshold as defined by the SNR_c and code sequence length. The shape of this curve supports the explanation of the false acquisition and miss rate for the receiver as the correlation threshold moves from a low threshold, through the ideal value and onto high threshold values. With a high threshold a position is reached where no data is received whereas with a low threshold it is found that each correlation shift produces a correlation value which surpasses the threshold. It is predicted that as the noise tends towards a Gaussian distribution then the curve will become more pointed because the variation of the SNR_c with respect to the noise will produce a smaller band (as explained for figures (8.1) and (8.2)) and so a smaller displacement from the ideal threshold will increase the probability of error.

The attempt in figure (8.5) to derive the probability of error curve from a different set of data shows that the slope of the curve is similar but that the absolute values differ. For low thresholds the chip error has increased and the discrepancies between the absolute values may be attributed to the manner in which the chip error is modified. This modification assumes that the first data bit is received correctly, however this is particularly invalidated if the P_e is large ie why should the first bit be received correctly? The solution to this is either to calculate the chip error without taking the first data bit into account or to incorporate the discrepancy between the predicted start chip and the actual start chip. The analysis must also be amended to include the chip error when a larger than expected number of chips are required to receive the same amount of data. It may also have been expected that the chip error would not predict the P_e if the correct number of chips were received but that the data content was reversed ie data one to zero or vice versa. This will not be the case until the "ideal" threshold value is close to the half code sequence length at which point a relatively few number of chip errors will produce this data inversion.

8.4.4 Simulation System Timing

Figure (8.6) shows that for a DPR of 18dBs it requires 54 seconds of CPU time to simulate the propagation of one data bit across one metre of the resolved channel and that the overheads for the program software require approximately 400 seconds.These figures are approximately true for the DPR of 16dBs however the difference is due only to the number of simultaneous transmitters; it is predicted that a significant change in the code sequence length will produce a significant increase in the CPU processing time, however insufficient results are available to support this at the present time. It is also expected that an increase in the number of active receivers will increase the CPU processing time because it is the receivers which perform the repetitive and time consuming correlation calculation.

Figure (8.7) relates the physical CPU processing time to the period simulated for a real system and shows that it requires $3x10^6$ seconds to simulate one second of a real system. Several seconds of real time must be simulated for systems with low probabilities of error and so the CPU time must be reduced before this aspect of the simulation becomes realisable - it should be remembered that the age of the universe has been calculated to be 10^{18} seconds thus a real time simulation of 1000 seconds will require a long simulation time. Once again the relative processing time would appear to be related to the data.resolution product which implies that the processing time consumed by the two algorithms which are responsible for:-

(a) the pulsing of the signal propagation along the channel - this is repeated hundreds of times during each simulation
(b) the data length affects the total number of chips which must be processed and it is this number which dominates the processing time spent on the correlation calculations.

These modifications must be implemented or else it will be impossible to investigate the bit error rate performance of the SS-LAN configurations unless the probability of error is high and the frequency of transmission is high which is an unlikely combination in the design philosophy of a useful implementation.

8.5 System Repercussions

The aim of the simulation system is to analyse the performance of extravagant systems eg ones with hybrid code sequence types, and to suggest modifications to the system to rectify poor performance or system design errors. The previous discussions have shown that in general the results from the simulator are in agreement with those of the theoretical analysis and therefore the operation of the simulation can be accepted with a high degree of confidence. Consequently the repercussions of the results analysis must now be considered for their effects on the design of the SS-LAN nodes and the simulator; the latter will improve the performance and accuracy of the simulator in such aspects as the generation of the channel chip SNR readings. The node alterations will also be reflected in some simulation system changes but these will not be discussed separately.

8.5.1 SS-LAN Node Design Amendments

Two design errors were detected in the design of the receivers for SS-LAN nodes by a set of results which have not been discussed in the previous sections; this is because their readings were invalidated by these errors. The two problems are:-

(a) once the receiver has been initialised then the correlator will contain erroneous data which could cause an incorrect data demodulation
(b) when the SS modulation involves the use of multiple code sequences for each data bit (ie $m > 1$) then if a continuous acquisition algorithm is performed (as opposed to a single acquisition and a demodulation technique) the receiver will falsely lock onto the code sequences.

An example of this first problem can be explained if the correlation of a 127 bit long code sequence is analysed. At initialisation the correlator will be reset to some state - let this be the all zero condition. If the system is to operate in a low SNR_i environment then the correlator threshold will also be low, for example 66 bits. If the code sequences are m-length PN codes then they contain 63 zeroes and 64 ones. Once chip reception and correlation begins the first few chips will displace some of the preset chips and it is possible that with a low threshold this initial, small number of chips will be enough to exceed the threshold. This is because the correlation score is already 63 for zeroes hence it only requires the agreement of another three different chips to exceed the threshold. The solution to this problem is to disable the output of the correlator for the first (L-1) chips thereby removing the possibility of a premature acquistion.

The second problem is more subtle and can be demonstrated using figures (8.8a),(8.8b) and (8.8c). Figure (8.8a) shows how two consecutive data bits are SS modulated using a single code sequence of 31 bits length and with four sequences per data bit. If the two data bits are equal then all the code sequences are of the same phase and alignment. The desired ACF is shown in figure (8.4b) where the data will be acquired after the reception of the fourth and eighth code sequences. The actual ACF is shown in figure (8.4c) where the correlation threshold is surpassed at the reception of the fourth, fifth, sixth, seventh and

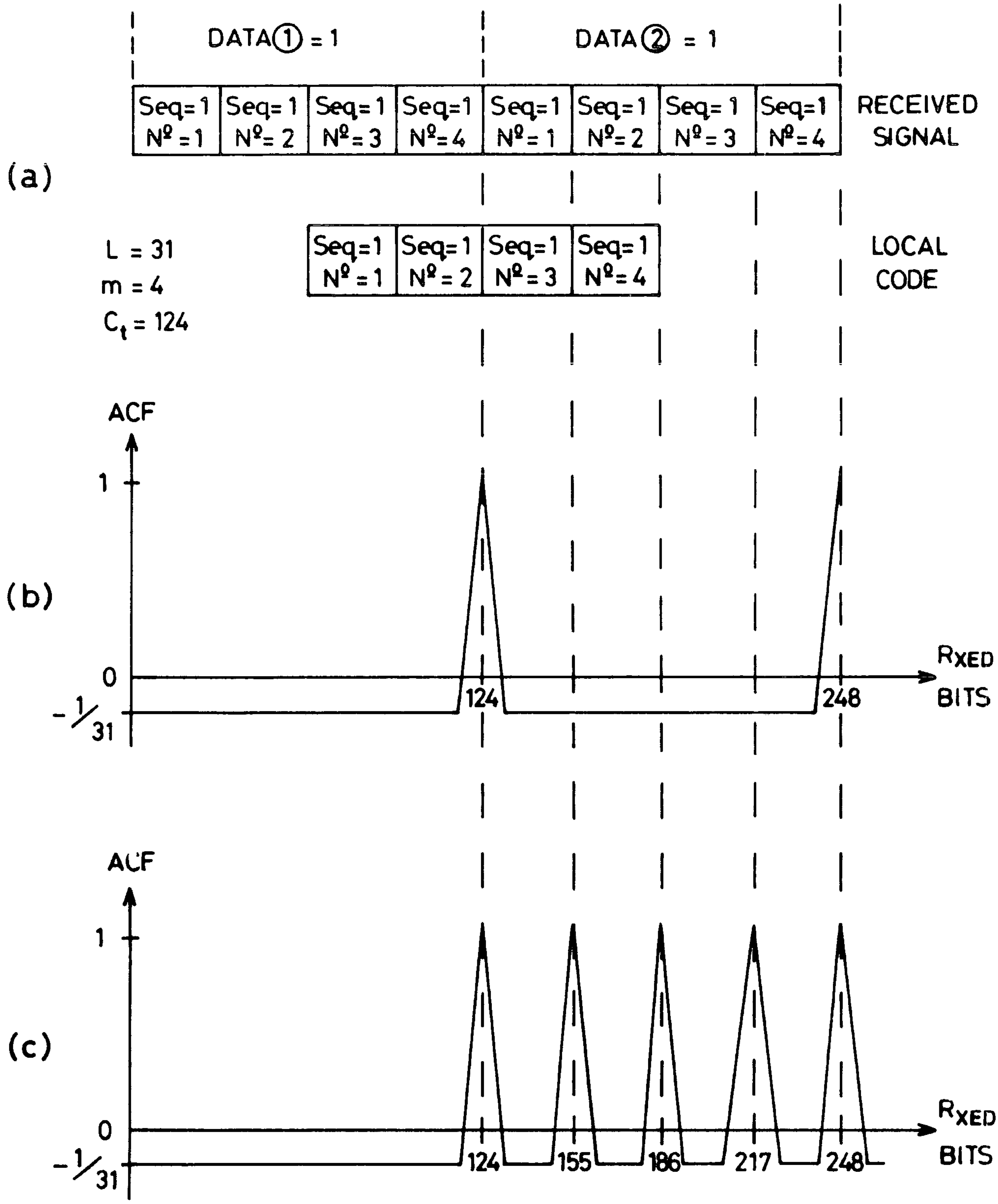


FIGURE (8.8a): SS Modulation Structure

FIGURE (8.8b): The Desired ACF

FIGURE (8.8c): The Actual ACF

eighth code sequences; the incorrect reception of three extra data bits. The explanation for this is that under the present system the receiver operates using a continuous acquisition algorithm where each chip shift is analysed for correlation in its own right. Each of the code sequences are identical and so once acquisition has occured it will be repeated every code sequence period as opposed to every four code sequence periods. The solution to this problem is to either perform a single acquisition followed by demodulation ie count each shift until multiples of the code sequence occur, at which point correlation is performed, or to disable the acquisition for periods proportional to the effective code sequence length. The difference between these two methods is that the former will miss data if a chip is lost or duplicated (unless tracking is also implemented) whereas the second can be designed to perform coarse acquisition over a predefined range of bits.

This second problem will also occur in a system which employs the unique receiver mode. Here a contention may arise which, if the reciever is performing continuous acquisition, will cause the receiver to acquire two different data sets as one data stream. Such a problem can not be remedied by the application of higher protocols because the error rate could become as large as 0.5 thereby rendering all error correction techniques useless. Once again the solution is to use a modified acquisition technique which performs the correlation over a discrete set of bits as opposed to all of them.

8.5.2 Simulator Software Amendments

The most important amendments to be made to the simulator are:-

(a) the implementation of task D (Data Analyser) to present the results in a form ready for the production of the performance graphs.
(b) the determination of the SNR_c must be refined once it has been established that the present algorithm still produces questionable values in a more Gaussian distribution.
(c) the propagation of the signal along the highway should be accomplished by the use of pointers as opposed to physically moving the data and the highway should be represented by a one dimensional array which will be configured to the highway specification at run time.

(d) the correlation algorithm should be implemented on a word basis as opposed to the present bit basis (this is direct modelling of the hardware implementation). The shifting of the chips through the correlator should also be implemented via a set of pointers as opposed to the present physical shifting of the data.
(e) the results for a simulation should be produced at the termination of a simulation run even when all of the expected data bits have not been received.

The most significant fact is that the simulator is functioning and is producing worthwhile results. Now that the basic development of the software is complete the modifications can be made to improve its performance range and the accuracy of its readings.

Conclusion

The theoretiocal analysis of the SS-LAN has been adapted such that its results can be compared with those generated from the computer simulation. It has been shown, for example, that the receiver $\overline{SNR}_i$ can be represented by C_t^2/L, where C_t is the difference between the number of agreements and disagreements when data is successfully acquired and where both C_t and L can be provided directly by the simulator. The results from the simulation software are displayed in four plots: the normalised correlation agreements/disagreements vs channnel chip SNR, the receiver output SNR vs the DPR, the probability of error vs the correlation threshold and the probability of chip error vs correlation threshold. A second set of results have also been produced and these provide insight into the performance of the simulation software: CPU time vs the data.resoultion product and the normalised CPU time vs the data.resolution product.

It has been shown that for a specific channel chip SNR there exists an "ideal" correlation threshold which will produce error free data reception. Any displacement from this value will produce acquisition errors whose severity will lie between the two extremes of total data loss or total data fabrication. The simulator has shown results which are in general agreement with these predictions but it has been found that the Gaussian distribution of the channel chip noise causes a banding around this "ideal" threshold. This is represented by an average SNR_c, and therefore $\overline{SNR}_i$,which results in data reception errors. This effect is independent of any environmental noise and occurs because the threshold must be aligned to the average SNR_c with the resulting loss of data for lower SNR values. The simulator has also shown the variation in the probability of error as the correlation threshold is incremented through the ideal threshold and this is also in good agreement with the theory.

The simulator results have produced an $\overline{SNR}_i$ vs DPR constant of 5.2 dBs (produced by the hardlimitng and aperiodic ACF) and a gradient of 0.6 and these compare favourably with the theoretical values of 3.3 dBs and 1.0, respectively. It has also been shown that the probability of chip error is indicative of the probability of error and this will remove the necessity of comparing the transmitted and received data for the determination of P_e. The CPU requires 54 seconds of processing time to simulate unit data.resolution and it requires $3*10^6$ seconds to simulate each real second of system activity.

The results from the simulator are in good agreement with those from the theoretical analysis and consequently the simulator performance can be quoted with a good degree of confidence. Two system design modification requirements have been highlighted by the simulator and these require the disabling of the receiver for (L-1) chips at initialisation and the limitation of the continuous acquisition scheme when using a code sequence/data ratio (m) of greater than one. Several simulation software changes are also required to lower the processing time (essential for the analysis of low P_e values) and to improve the accuracy of the readings. In conclusion the simulation can now be applied to the analysis of more extravagant SS-LAN configurations with the knowledge that it will provide information which can be considered reliable.

CHAPTER 9

Conclusion

9.1 Summary of the Present System

A new LAN has been designed which employs a SS multiple access scheme at the network's physical layer. Established LANs utilise a time sharing method of one type or another to produce a multiple access system however the major inherent limitation is that only one transmission may occur on the channel at any one time. The new SS-LAN removes this limit and therefore permits each and every node to have unrestricted access to, and unlimited use of, the channel independent of the current activity supported by the channel. The SS modulation is achieved by the distribution of a code sequence family (ie DS SS modulation) and this provides a wide range of system properties: unrestricted medium access, contentionless channel access, total decentralisation, variable data loading, privacy and security of data, parallel processing and priority mechanism, noise immunity and dynamic reconfigurability. It is not impossible for TDM based LANs to provide many of these properties however they would not be an intrinsic part of the modulation technique as is the case for the SS-LAN.

The logical to physical mapping of the code sequences can be achieved in several ways, each of which can be applied to a different system implementation. The present six methods are a point to point mode, a broadcast mode, a node group zone mode, a function mode, a message type mode and a protocol mode. Each mode emphasises a particular property of the SS technique but this may be at the cost of some of the other properties. This does not limit the system because a single physical realisation of the network can be run time altered to fit a particular configuration.

Naturally there are some disadvantages in the use of the SS-LAN. The first is that the individual node to node data rate is considerably lower than that of TDMA networks. Secondly, the individual transmitter and receiver clocks must be as stable as possible or else their relative drift will degrade the system performance. The third disadvantage is concerned with the actual distribution modes of the codes. In some cases their operation will violate the accepted definition of the ISO-OSI reference model thus their implementation must be considered carefully. It has also been established that a performance comparison between the SS-LAN and other LANs is difficult due to significant differences in their operation but that a true reflection of their comparative performance is related to the individual node access capability.

An analysis of code sequences for use in SSMA systems concluded that Bent sequences were best employed in secure networks where the code sequence distribution was not dependent upon a large number of sequences. Non secure implementations could employ Gold codes however their ideal application would be for a system in which a large number of codes were necessary but where the number of simultaneous users was not critical. M-length PN codes have no long term use within the SS-LAN but they are useful for the calibration of the system performance. It is possible that specialised code sequences can be created for particular systems and the set of criteria for their properties was found to be: high in-phase ACF coefficients, low out-of-phase ACF coefficients, low CCF coefficients (in particular the aperiodic CCFs), a linear spreading function and as large a family as possible coupled with ease of generation of the individual members.

The system hardware is constructed from three blocks: the SIL which acts as a specialised interface between the node logic and the host device, the SSNC which is the node logic controller and the SSARTs which are the individual SS transmitters and receivers. This construction provides flexibility in the use of the Spread Spectrum Local Area Network (SS-LAN) nodes thus one host may control several parallel SSARTs, several hosts may be linked to several SSARTs and any intermediate distribution. The receivers are designed upon a digital correlation method which performs hardlimiting on the channel signal and then employs a sliding polarity coincidence correlation technique to acquire and demodulate the received signal. An analysis of this system has been derived, expanding upon an equation derived by Pursley, for the output SNR, probability of error, data throughput and message delivery times. Graphs of each of these characteristics have been provided and the conclusions derived from them concerning the parameter values for a defined system performance are: network efficiency should be as high as possible, the code sequences as short as possible, the single user SNR should be limited to between 15dBs and 20dBs, the Data Processing Resolution (DPR) should be limited to between 20dBs and 40dBs and the channel should be of the highest quality so that the near-far effect is minimised.

It was determined that the SS-LAN should be employed in one of two environments: where the messages are short but regularly transmitted from all nodes and where immediate tranmission is imperative or where the time transfer delay is non critical but where many nodes will want to transmit long messages continuously.

A large number of different SS-LAN configurations are possible due to the application of:- different code sequences families, the different operational modes, the different physical system properties etc. Consequently a detailed theoretical analysis is difficult especially if the system is of a non random nature ie small and with dominant nodes. The simplest method for analysis is to use a simulator. The design of such a simulator has been presented and this permits the simulation of any configuration (within size limits of the host computer) of an SS-LAN by defining it within a specified file format. The configuration data is validated, simulated and produces results which are stored in another set of files. The readings which can be taken from the simulation system are collated to perform the necessary result and consequently several simulations are necessary for each collated result. The present version of the software occupies 141k bytes of core memory and does not implement the full design specification and consequently the full range of the simulation analyses can not be performed; it should however be noted that even with these restrictions the simulator still produces worthwhile results for realistic systems. The relationship between the results from the simulator and those from the theoretical analysis has also been determined and it was found that it was dependent upon the operation of the correlator and the definition of the correlation threshold.

The comparison of the results from the simulation system and the theoretical analysis has also been performed and it has been found that the two are in good agreement. The theoretical analysis predicts an $\overline{SNR}_i$ vs DPR constant of 3.3dBs (due to the use of hardlimiting and random codes) and a gradient of 1 which compares favourably with the simulator results of 5.2dBs and 0.6 respectively. The general variation of the correlation threshold with the channel chip SNR (SNR_c) has also been verified by the simulator however it was found that the variance of the SNR_c reading was large and so the corresponding banding around the ideal threshold position was also large. This banding was greater than expected and this is thought to be due to an error in the generation of the SNR_c reading. The probability of error due to the variation in the correlation threshold was also determined and this demonstrated that the variation of the specified threshold with respect to the ideal threshold produces an increased P_e, the greater the relative displacement. This probability of error is different to that predicted by the modified Pursley equation which is only valid when synchronisation has been achieved.

It was also found that the probability of error can be represented by the chip error which is a measure of the number of chips actually received against the number which should have been received. This equivalence means that the determination of the P_e does not require the comparison of the received data with that transmitted and so the amount of data to stored within the simulator is reduced. The simulator requires 54 seconds of CPU time to simulate unity data.resolution and it can simulate 1 second of real time in $3*10^6$ seconds of continuous CPU time for a fixed DPR of 18dBs.

The comparison of the theoretical results and the simulator results have prompted the inclusion of several modifications to both the SS-LAN system design and the computer simulation. The two system modifications are concerned with the receiver initialisation and the use of the continuous acquisition scheme for a code sequence/data ratio of greater than one; the proposed solution should resolve these problems. The simulator modifications are aimed at reducing the processing time so that it can be used for the analysis of systems with low probability of error which would be impossible with the present version of software. In conclusion the simulation system can now be applied to the analysis of more extravagant SS-LAN configurations with the knowledge that it will provide results which can be considered reliable.

This thesis has presented the design of a new LAN which employs DSSS modulation as the means for physical transmission. The logical and physical design of the network has been determined and this has incorporated the selection of different code sequence families for certain types of system implementation. The theoretical analysis for a simplified system has been presented and discussed and this has been compared and contrasted with the results produced by a sophisticated computer simulation program. The two have been found to be in good agreement and so the initial research for the development of an actual system has been completed.

9.2 Development of the SS-LAN

Considerable development work is needed on the SS-LAN. The first stage is to build a simple demonstration system (approximately six nodes operating at a code frequency of 12MHz) whose performance can be compared with both the theory and the simulation system. The results from all three should bear a close ressemblence and therefore the simulation will have a base for confidence from which more complex configurations may be realistically simulated.

These more complex systems are a mix of large configurations and ones in which the code sequences are mixed: this hybrid type of system has not been discussed in the literature due to the computational effort required for the analysis of the ACFs and CCFs. Code mixing is a useful method for enhancing system security while employing specific codes for their application to particular system problems. If the simulator has been proven to produce realistic results then the simulation of such systems will save considerable effort in the construction of demonstration systems.

The second area of development is concerned with the protocols which must be implemented at the different layers of the ISO-OSI reference model. These protocols must reflect the simultaneous multiple user environment coupled with the unrestricted access nature of the communications link. Such a system can provide a virtual link which has an effectively endless message length but on which it may also still be necessary to provide flow control, error control and status information. The fundamentally different nature of the SS-LAN implies that radically different protocols will be required which must efficiently use the SS-LAN while also maintaining the spirit of the OSI reference model.

Throughout the description of the SS-LAN the data rate values have been defined in the range usually described as highspeed ie 100kbits to 10Mbits per second. This limits the technology and techniques which can be presently applied to the construction of the SS-LAN. A lowering of the data rate permits two simplifications to the SS-LAN:-

(a) A purely software based system can be developed, employing the minimum amount of hardware and therefore performing the correlation within the controlling microprocessor.
(b) The construction of a set of VLSI chips to replace the major hardware components of the transmitter and the reciever.

The development of these two implementations will considerably reduce the cost of the individual nodes (which is at present dominated by the number of correlator chips) and improve the reliability of the system. The reduction in the size of the nodes will also increase the areas of application which may one day rival the example where a small Ethernet has been installed inside a photocopying machine.

9.3 Areas of Application

The types of system application for the SS-LAN are:-

(a) in electrically noisy environments, such as a factory or power plant
(b) where the data integrity must remain secure and/or private
(c) as a background monitoring system for some other system
(d) where many conversations are required simultaneously eg voice/data networks

Each of these topics has several subdivisions. Consider first the case of electrically noisy environments. This type of environment is usually attributed to a system which is physically located next to heavy electrical equipment. An extremely hostile environment is present on the domestic mains wiring also and one potential application of the SS-LAN is in production of an "integrated domestic network" (IDN). This would utilise the mains as its communications channel and would be used to link together domestic appliances such as central heating controllers and thermostats etc. In general the data rate requirement for such a system is low (a 100 bits/sec at most) hence a long code sequence could be employed to provide good noise immunity, coupled with a software based node which would reduce cost and increase reliability. The use of orthogonal codes will allow different messages to use the mains as well as isolating adjacent houses from mixed message corruption. Several organisations are at present investigating the mains as a potential method for remote meter reading of the gas and electricity meters but these have incurred problems due to the interference of the mains by television signals.

In the case of industrial applications, the SS-LAN can provide good control flow of data with each node being able to send data immediately it is ready. The natural noise immunity is also favourable and it could also be used as a real time system provided the message lengths were short; less than 100 words.

The second area is in a system where the number of users is large but where their access privilege for data is also varied. A typical system could be on a university campus where both students, staff and administration could use a system of bridged networks on which all information is available to those with the correct access rights. Such a system usually implements the exchange of passwords etc with the corresponding attempts to "break" the system. The SS-LAN would place the data encryption (which would require cryptographically secure codes for a secure system) at a level below that accessible to a user ie the codes sequences would be distributed according to the classification of the node and so only nodes of the same classification could exchange information. Indeed, should the request connection be decoded then the encrypted data must still be decoded separately before it could be understood by an unauthorised user. The implementation of the unique transmitter mode also provides a broadcast facility which will enable nodes of the same classification to communicate efficiently. Such a system could be useful for a mailbox application where a particular node must constantly supply data to a group of other users.

The third type of application is concerned with the use of the SS-LAN as a background service providing simultaneous support to some primary system. Two types of this system can be envisaged: the first is a fault tolerance monitor and the second as a service, such as network management. The advantage of the SS-LAN is that it can perform these tasks concurrently with the primary task of the system. Once again the implementation of the fault monitoring system would require a low rate SS-LAN, thereby permitting a software based version. Such a system could monitor the presence and correct operation of the boards linked to the backplane of a computer and could cause the switching of boards in the situation where an error is detected. In the case of a network manager we have an example of a loosely coupled version of the fault tolerant system. The manager could declare nodes active or inactive and could maintain a consistent view of the system within all the nodes. This could be accomplished by the normal protocol within an Ethernet implementation but this modification would remove the management load from the general activity of the nodes.

The final applications are related to the most important property of the SS-LAN; that of its ability to support simultaneous multiple users without restriction of any kind on their access rights. When this is coupled with the variable data loading the SS-LAN becomes ideal for the support of telephone conversations along a single channel without the need for data buffering within the receiver or transmitter. The transmission of data can be laid on top of this collection of conversations, as can be other conversations, without redress to the channel activity and provided the system is operating within its multiuser specification. An advantage of this system is that should the specification be violated then there will be a soft degradation of the system and not a catastrophic fail.

9.4 Concluding Remarks

The SS-LAN is not the only solution to the problems faced by present day LANs. Many people feel that a single LAN for every system is unnecessary and that different types of applications will require different types of LANs. The SS-LAN is a highly flexible system which is ideal for many types of applications and which utilises a single standard hardware construction tuned by configuration software. This tuning moulds the standard system to each particular application and thus a single SS-LAN can be used for the different system requirements. A great deal of work is still necessary before a commercial system can be produced but this thesis has shown that the general principles of its operation are well founded and the properties of a particular configuration can be predicted with a good degree of confidence.

REFERENCES

(1) Vaughan, H.E., "Research Model for Time-Separation Integrated Communication", Reprint, IEEE Transaction on Communications, Vol COM-27, July 1979, pp.940-947.
(2) Shannon, C.E., "Communication in the Presence of Noise", Reprint, Spread Spectrum Techniques, book, Ed. R.C. Dixon, IEEE Press, 1976, pp. 17-28.
(3) Shannon, C.E. and Weaver, W., "The Mathematical Theory of Communiction", book, University of Illonois Press, 8th Edition, 1980, pp.125.
(4) Costas, J.P., "Poisson, Shannon and the Radio Amateur", Reprint, Spread Spectrum Techniques, book, Ed. R.C. Dixon, IEEE Press, 1976, pp.29-39.
(5) Sloane, N.J.A., "The Packing of Spheres", Scientific American, Vol.250, January 1984, pp.92-101.
(6) Polishuk, P., "Fiber Optics Development in the United States, Progress Since FOC/LAN 83", Proceedings of FOC/LAN 84, Las Vegas, Nevada, pp.3-8.
(7) Scholtz, R.A., "The Origins of Spread Spectrum Communications", IEEE Transactions on Communications, Vol COM-30, May 1982, pp.822-854.
(8) Scholtz, R.A., "Notes on Spread Spectrum History", IEEE Transactions on Communications, Vol COM-31, January 1983, pp.82-84.
(9) Price, R., "Further Notes and Anecdotes on Spread Spectrum Origins", IEEE Transactions on Communications, Vol COM-31, January 1983, pp.85-97.
(10) Bennett, W.R., "Secret Telephony as a Historical Example of Spread Spectrum Communication", IEEE Transactions on Communications, Vol COM-31 pp.98-104.
(11) Alem, A., Huth, G.K., Holmes, J.K. and Udalar, S., "Spread Spectrum Acquisition and Tracking Performance for Shuttle Communication Links", IEEE Transactions on Communications, Vol COM-26, November 1978, pp.1689-1703.
(12) Giordano, A.A., Sunkenberg, H.A., DePedro, H.E., Stynes, D., Brown, D.W. and Lee, S.C., "A Spread Spectrum Simulcast MF Radio Network", IEEE Transactions on Communications, Vol COM-30, May 1982, pp.1057-1073.
(13) Yue, O.C., "Spread Spectrum Mobile Radio, 1977-1982", IEEE Transactions on Vehicular Technology, Vol VT-32, February 1983, pp.98-105.
(14) Masamura, T., "Satellite Communication system using TDM and SSMA",

IEEE Transactions on Aerospace and Electronic Systems, Vol AES-19, November 1983, pp.906-914.
(15) Dixon, R.C., "Spread Spectrum Systems", book, Wiley Interscience, 1976, pp.318.
(16) Cook, C.E. and Marsh, H.S., "An Introduction to Spread Spectrum", IEEE Communications Magazine, March 1983, pp.8-16.
(17) Harris, R.L., "Introduction to Spread Spectrum Techniques", AGARD-NATO Lecture Series, No.58, 1973, pp.(3.1)-(3.21).
(18) Kusaka, H. and Nishida, F., "A Spectral Analysis of M-ary DS SSMA Communication Systems", IEEE Transactions on Communications, Vol COM-31, April 1983, pp.541-545.
(19) Pickholtz, R.L., Schilling, D.L. and Milstein, L.B., "Theory of Spread Spectrum Communications - A Tutorial", IEEE Transactions on Communications, Vol COM-30, May 1982, pp.855-884.
(20) Pickholtz, R.L., Schilling, D.L. and Milstein, L.B., "Revisions to a Theory of Spread Spectrum Communications - A Tutorial", IEEE Transactions on Communications, Vol COM-31, February 1984, pp.211.
(21) Scholtz, R.A., "The Spread Spectrum Concept", IEEE Transactions on Communications, Vol COM-25, August 1977, pp.748-755.
(22) Judge, W.J., "Multiplexing Using Quasiorthogonal Binary Functions", Reprint, Spread Spectrum Techniques, Ed. R.C.Dixon, IEEE Press, 1976, pp.65-67.
(23) Lynn, P.A., "An Introduction to the Analysis and Processing of Signals", book, Macmillian Press, 2nd Edition, pp.256.
(24) Cowan, D., "The Integrity of Serial Data Highway Systems", PhD Thesis, University of Durham, 1983.
(25) Flint, D.C., "The Data Ring Main", book, Wiley Press, 1983, pp.375.
(26) Lee, K.C.E., "Local Area Networks", book, NCC Publications, 1982, pp.375.
(27) Bux, W., Closs, F.H., Kuemmerle, K., Keller, H.J. and Mueller, H.R., "Architecture and Design of a Reliable Token Ring Network", IEEE Journal on Selected Areas of Communications, Vol SAC-1, November 1983, pp.756-766.
(28) Hopper, A. and Williamson, R.C., "Design and Use of an Integrated Cambridge Ring", IEEE Journal on Selected Areas of Communications, Vol SAC-1, November 1983, pp.766-774.
(29) Huber, D.E., Steilin,W. and Wild, P.J., "SILK: An Implementation of a Buffer Insertion Ring", IEEE Journal on Selected Areas of

Communications, Vol SAC-1, November 1983, pp.766-774.
(30) "The Ethernet - Data Link Layer and Physical Layer Specifications", Standard, Version 2.0, Digital, Intel and Xerox, November 1982, pp.103.
(31) Schmidt, R.V., Rawson, E.G., Norton, R.E., Jackson, S.B. and Bailey, M.D., "Fibernet II: A Fiber Optic Ethernet", IEEE Journal on Selected Areas of Communications, Vol SAC-1, November 1983, pp.702-710.
(32) Kirstein, P., "The UNIVERSE Project", Proceedings of ICCC 82, North Holland, September 1982, pp.442-447.
(33) Hensel, P., "LANs and the Impact of Optical Fibres", Electronics and Power, November/December, 1983, pp.187.
(34) Finley, M.R., "Optical Fibres in LANs", IEEE Communications Magazine, August 1984, pp.22.
(35) Ebihara, Y., Ikedu, K., Nakamura, T.N., Ishizaka, M., Shinzawa, M. and Nakayama, K., "GAMMA-NET: A Local Computer Network Coupled by a High Speed Optical Fibre Ring Bus", Computer Networks, December 1983, pp.375-388.
(36) Cohen, A.R., Heller, J.A. and Viterbi, A.J., "A New Coding Technique for Asynchronous Multiple Access Communication", IEEE Transactions on Communications Technology, Vol COM-19, No.5, October 1971, pp.849-855.
(37) Davies, P.A. and Shaar, A.A., "A Synchronous Multiplexing System for an Optical Fibre Local Area Network", Electronics Letters, Vol.19, No.10, May 1982, pp.390-392.
(38) El-Hakeem, A.K., Hafez, H.M. and Mahmoud, S.A., "Spread Spectrum Access to Mixed Voice-Data Local Area Networks", IEEE Journal on Selected Areas in Communications, Vol SAC-1, No.6, December 1983, pp.1054-1069.
(39) Hafez, H.M., El-Hakeem, A.K. and Mahmoud, S.A., "Spread spectrum Access in Two Hop CATV Data Networks", IEEE Journal on Selected Areas in Communications, Vol SAC-3, No.2, March 1985, pp.312-322.
(40) Stahlman, M., "Inside Wang's Local Net Architecture", Data Communications, January 1982, pp.85-90.
(41) Feller, W., "An Introduction to Probability Theory and its Applications", 3rd Edition, Vol.1, Wiley International, pp.509.
(42) Federal Standard 1026, "Telecommunications: Interoperability and Security Requirements for use of the Data Encryption Standard in Data Communication Systems"
(43) Rivest, R.L., Shamir, A. and Adleman, L., "A Method for Obtaining

Digital Signatures and Public Key Cryptosystems", Communications of the ACM, Vol.21, February 1978, pp.120-126.

(44) British Standards Institute, "The ISO-OSI Reference Model", Draft documents, June 1982, pp.78.

(45) Mockapetris, P., Postel, J. and Kirton, P., "Name Server Design for Distributed Systems", Proceedings of the 7th ICCC, Sydney, October 1984, pp.561-570.

(46) White, J.E., "A High Level Framework for Network Based Resource Sharing", AFIPS Conference Proceedings, NCC, Vol.45, 1976, pp.561-570.

(47) Ellis, G., Dillon, S., Stritter, S. and Whitnell, J., "Experiences with a Layered Approach to Local Area Network Design", IEEE Journal on Selected Areas in Communications, Vol SAC-1, November 1983, pp.857-868.

(48) CERN, "The DATABUS Protocol Specification"

(49) Stremler, F.G., "Introduction to Communication Systems", book, 2nd Edition, Addison Wesley, pp.702.

(50) IEEE Project 802, "Local Area Network Standard", Draft Standards D, Nos. 802.2,802.3,802.4,802.5,802.6, November 1982.

(51) Healy, T.J., "Coding and Decoding for Code Division Multiple User Communication Systems", IEEE Transactions on Communications, Vol COM-33, April 1985, pp.310-315.

(52) Bhargava, J.K., Haccoun, D., Matyas, R. and Nuspl, P.P., "Digital communications by Satellite", book, Wiley Interscience, 1981, pp.569.

(53) Welch, L.R., "Lower bounds in the maximum cross correlation of signals", IEEE Transactions on Information Theory, Vol IT-20, May 1984, pp.397-399.

(54) Lin, S. and Costello, D.J., "Error Control Coding: Fundamentals and Applications", book, Prentice-Hall, 1983, pp.603.

(55) Milstein, L.B. and Ragonetti, R.R., "Combination Sequences for Spread Spectrum Communications", IEEE Transactions on Communications, Vol COM-25, No.7, July 1977, pp.691-696.

(56) Beale, M. and Tozer, T.C., "A Class of Composite Sequences for Spread Spectrum Communications", Computer and Digital Techniques, Vol.2, April 1979, pp.87-92.

(57) Maskara, S.L. and Das, J., "Concatenated sequences for Spread Spectrum Systems", IEEE Transactions on Aerospace and Electronic Systems, Vol AES-17, No.3, 1981, pp.342-350.

(58) Shaar, A.A., "A Study of Code Division Multiple Access with Reference to Fibre Optic Local Area Networks", PhD Thesis, University of

Kent, April 1985.
(59) Han, W.K. and Hemmati, F.H., "Punctured PN Codes", Electronic Letters, No.12, May 1984, pp.494-495.
(60) Gold, R., "Optimal Binary Sequences for Spread Spectrum Multiplexing", Reprint, Spread Spectrum Techniques, IEEE Press, Ed. by R.C.Dixon, 1976, pp.119-121.
(61) Stark, W.E. and Sarwate, D.V., "Kronecker Sequences for Spread Spectrum Communication", IEE Proc-F, Vol.128, NO.2, April 1981, pp.104-109.
(62) Olsen, J.D., Scholtz, R.A. and Welch, L.R., "Bent Function Sequences", IEEE Transactions on Information Theory, Vol IT-28, No.6, November 1982, pp.858-864.
(63) Lempel,A. and Cohn,M., "Maximal Families of Bent Sequences", IEEE Transactions on Information Theory, Vol IT-28, No.6, November 1982, pp.865-868.
(64) Shaar, A.A. and Davies, P.A., "A Survey of one coincidence sequences for frequency hopped Spread Spectrum systems" IEE Procs-F, Vol.131, No.7, December 1984, pp.719-724.
(65) Lin, D., "The Generation and Comparison of Linear and Nonlinear Binary Sequences for Spread Spectrum Communication", 3H Project, Department of Applied Physics and Electronics, Durham University, April 1984.
(66) Comer, D.J., "Digital Logic and State Machine Design", book, HRW, 1st Edition, 1984, pp.436.
(67) Special on "OSI Standard Architectures and Protocols", Proceedings of the IEEE, December 1983, pp.1331-1449.
(68) Ekre, H., "Polarity Coincidence Correlation Detection of a Weak Noise Source", IEEE Transactions on Information Theory, January 1963, pp.18-23.
(69) Taylor, A.S., "Characterization of Cable TV Networks as the Transmission Media for Data", IEEE Journal on Selected Areas in Communications, Vol SAC-3, No.2, March 1985, pp.225-265.
(70) Pursley, M.B., "Performance Evaluation for Phase Coded Spread Spectrum Multiple Access Communication - Part I: System Analysis", IEEE Transactions on Communications, Vol COM-25, August 1977, pp.795-799.
(71) Weber, C.L., Huth, G.K. and Batson, B.H., "Performance Considerations of Code Division Multiple Access Systems", IEEE Transactions on Vehicular Technology, Vol VT-30, February 1981, pp.3-9.

(72) Garber, F.D. and Pursley, M.B., "Optimal Phases of Maximal Length Sequences for Asynchronous Spread Spectrum Multiplexing", Electronics Letters, Vol.16, No.19, pp.756-757.

(73) Pursley, M.B., Sarwate, D.V. and Stark, W.E., "Error Probability for Direct Sequence Spread Spectrum Multiple Access Communications - Part I: Upper and Lower Bounds", IEEE Transactions on Communications, Vol COM-30, May 1982, pp.975-984.

(74) Kleinrock, L. and Tobagi, F.A., "Packet Switching on Radio Channels - Part I: Carrier Sense Multiple Access Modes and their Throughput Delay Characteristics", IEEE Transactions on Communications, Vol COM-23, December 1975, pp.1400-1416.

(75) Bux, W., "Local Area Subnetworks: A Performance Comparison", IEEE Transactions on Communications, Vol COM-29, October 1981, pp.1465-1473.

(76) Special on "Random Access Communications", IEEE Transactions on Information Theory, Vol IT-31, March 1985, pp.117-302.

(77) MoD(PE), "JSP188 - Real Time Software Documentation Specification", Draft, September 1979, pp.12.

(78) Logica Ltd, Logica Standards for the Production of Software, No.LS.047A.

(79) Jackson, M.A., "Principles of Program Design", book, Academic Press

(80) Drossman, M.M., Ziegler, C., Cacteer, B., Hardley, P. and Schilling, D.L., "Digital Computer Simulation of Spread Spectrum Communication Systems", IEEE 1981 International Conference on Communication, Denver, CO, Vol.3, pp.45.4.1-45.4.5

(81) O'Reilly, P.J.P. and Hammond, J.L., "An Efficient Simulation Technique for Performance Studies of CSMA/CD LANs", IEEE Journal on Selected Areas in Communications, Vol SAC-1, No.2, January 1984, pp.235-238.

(82) Webb, J.T, "Coral 66 Programming - Official Definition", book, NCC, 1978, pp110.

(83) DEC, DEC VAX/VMS User Manuals for the VAX 11/750.

(84) Wolff, S.S., Thomas, J.B. and Williams, T.R., "The Polarity Coincidence Correlator: A Nonparametric Detection Device" IRE Transactions of Information Theory, January 1962, pp.5-9.

(85) Freeman, J.J., "The Action of Dither in a Polarity Coincidence Correlator", Reprint, Spread Spectrum Techniques, book, IEEE Press, Ed. by R.C.Dixon, 1976, pp56-61.

(86) Chang, K-Y, and Moore, A.D., "Modified Digital Correlator and its

Estimation Errors", IEEE Transactions on Information Theory, Vol IT-16, No.6, November 1970, pp.699-706.

APPENDICES

APPENDIX 1

An Analysis of a DS SS Communications System

Consider the schematic representation of a DSSS communications system as shown in figure (A1.1) where the carrier modulation has been excluded - producing a baseband system. This analysis is similar to that provided by Judge (22) but is for a system where there are, k, simultaneous users. Let the data signals of the k^{th} user take the form $d_k(t-\tau_k)$ where τ_k is the relative time displacement between it and the reference code and let the code sequence of the k^{th} user take the form $C_k(t-\tau_k)$. The resulting signal transmitted from the k^{th} user is given by equation (A1.1):-

$$s_k(t-\tau_k) = C_k(t-\tau_k)\ d_k(t-\tau_k) \tag{A1.1}$$

Assuming that the channel is linear and additive, that the noise which does not originate from other users is defined by n(t) and that, k, users are transmitting simultaneously, then the signal on the channel is given by equation (A1.2):-

$$r(t) = n(t) + \sum_{i=1}^{k} s_i(t-\tau_i) \tag{A1.2}$$

If the reference signal is defined by the receiver using code sequence one then the signal at the receiver is given by equation (A1.3):-

$$r_1(t) = s_1(t) + n(t) + \sum_{i=2}^{k} s_i(t-\tau_i) \tag{A1.3}$$

At the receiver the incoming signal is multiplied by the local code sequence. It will be assumed that the two sequences have undergone synchronistaion and that the data demodulation is represented by an integration and dump method - this is equivalent to synchronised digital decorrelation. The output from the receiver is now defined by equation (A1.4):-

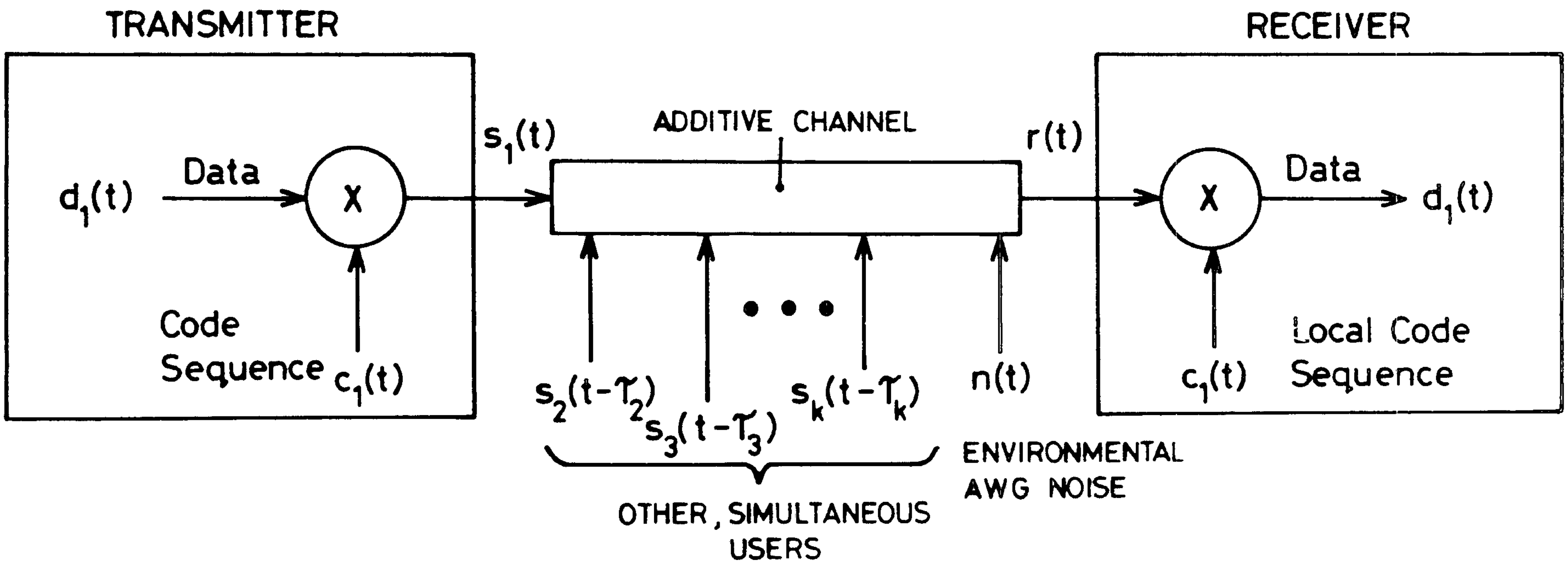


FIGURE (A1.1): A Schematic Diagram of a DS SS System

$$r_1(t) = \frac{1}{T_d}\int_0^{T_d} s_1(t)C_1(t)dt + \frac{1}{T_d}\int_0^{T_d} n(t)C_1(t)dt$$

$$+ \frac{1}{T_d}\sum_{i=2}^{k}\int_0^{T_d} s_i(t-\tau_i)C_1(t)dt \qquad (A1.4)$$

Substituting equation (A1.1) into (A1.4) will give:-

$$r_1(t) = \frac{1}{T}\int_0^{T_d} C_1^2(t)d_1(t)dt + \frac{1}{T}\int_0^{T_d} n(t)C_1(t)d_i(t-\tau_i)dt$$

$$+ \frac{1}{T_d}\sum_{i=2}^{k}\int_0^{T_d} C_i(t-\tau_i)C_1(t)d_i(t-\tau_i)dt \quad (A1.5)$$

If the signal $C_1^2(t)$ is formed to become unity and the signals $C_1(t)C_i(t-\tau_i)$ are arranged to give zero then the signal collapses to give,

$$r(t) = d_1(t) + \frac{1}{T_d}\int_0^{T_d} n(t)C_1(t)dt \qquad (A1.6)$$

where the noise has first undergone spectral spreading (cf equation (A1.1)) before it has been integrated over the data bit period. This will give rise to a small signal whose energy is spread over a wide bandwidth hence filtering about the data bandwidth will remove most of this noise. This will result in the receiver output being the desired data signal, ie

$$r_1(t) = d_1(t) \qquad (A1.7)$$

APPENDIX 2

Correlation Coefficients for a DS SS System

The ACF of two binary signals was given in equation (2.1) with its importance demonstrated in equation (A1.5) - the ACF should be as near to unity as possible to enable a receiver to pick out the desired signal from the received signal. In the SS-LAN the received signal is converted from its natural analogue form to a more useful digital equivalent. The situation can then be represented by figure (A2.1a) where a local sequence is correlated with the received signal in an attempt to re-acquire the local sequence in that signal. The figure shows the local sequence and the incoming binary sequence, with data modulated onto the code sequence using SS and a general reference position in the code sequence defined as, l. the variation in the ACF will now be determined. Using the definition of the aperiodic autocorrelation function (52), where:-

$$C(l) = \begin{cases} \sum_{j=0}^{L-1-l} x^i{}_j x^i{}_{j+l} & 0 \leq l \leq L-1 \\ \sum_{j=0}^{L-1+l} x^i{}_{j-l} x^i{}_j & 1-N \leq l \leq 0 \end{cases}$$

then from figure (A2.1a) the ACF is given by:-

$$ACF = d_0 \frac{1}{L} C_i(l) + d_{-1} \frac{1}{L} C_i(l-L) \qquad (A2.1)$$

Comparing this with the relevant ACF component in equation (A1.5) produces:-

$$\frac{1}{T_d} \int_0^{T_d} C_1{}^2(t)\, d_1(t)\, dt = \frac{1}{L} \left[d_{1,0}\, C_i(l) + d_{1,-1}\, C_i(l-L) \right] \qquad (A2.2)$$

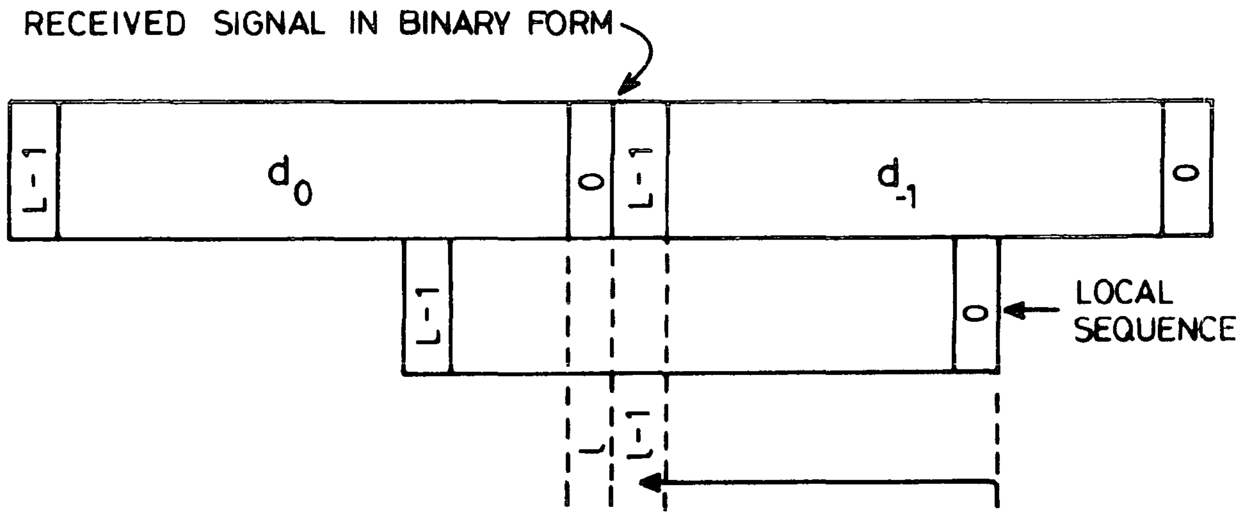


FIGURE (A2.1a): Autocorrelation of Binary Sequences

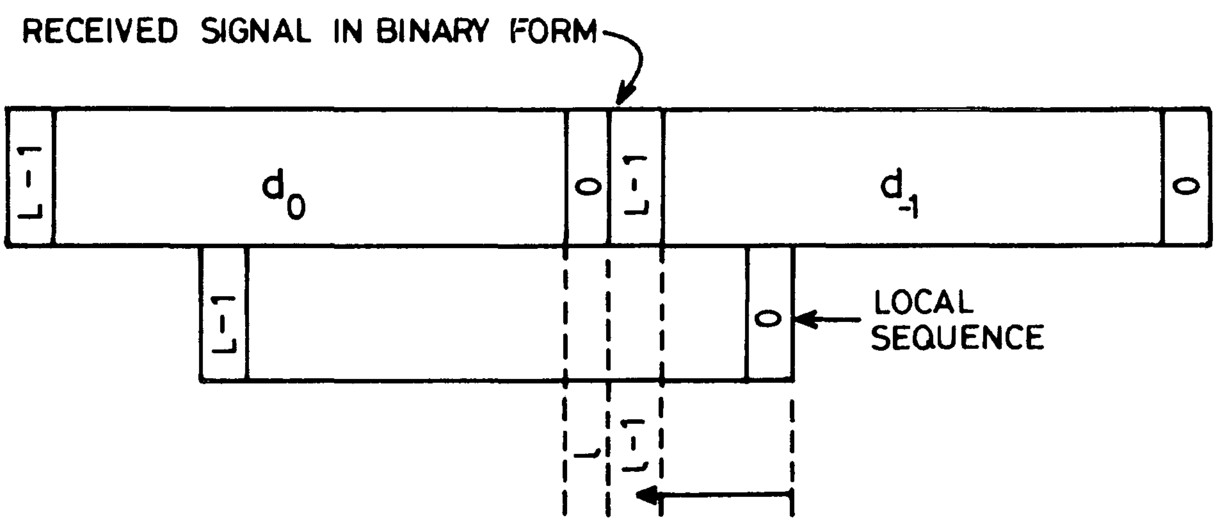


FIGURE (A2.1b): Cross Correlation of Binary Sequences

The data components $d_{1,0}$ and $d_{1,-1}$ are the data values for the modulated signal with respect to the offset local sequence. Should these two values both be logic one or zero then we have that:-

$$ACF_e = \frac{d_{1,0}}{L}\left[C_i(l) + C_i(l-L)\right] = \frac{d_{1,0}}{L}\theta_i(l) \qquad (A2.3)$$

where the term inside the brackets is defined as the periodic autocorrelation function or even autocorrelation function. Similarly should the two data values be different then we have that, if $d_{1,0} = d_{1,-1}$ then:-

$$ACF_o = \frac{d_{1,0}}{L}\left[C_i(l) - C_i(l-L)\right] = \frac{d_{1,0}}{L}\hat{\theta}_i(l) \qquad (A2.4)$$

where the term inside the brackets is defined as the odd autocorrelation function.

Clearly the ACF is dependent on the relative data components and care is required to ensure that during signal acquisition (by use of a sliding correlator) the out of phase ACFs are as close to zero as possible. Should the ACF become close to 1 or -1 then a false acquisition will occur with a corresponding false data demodulation.

A similar analysis may be performed for the determination of the variation in the cross correlation coefficients. Figure (A2.1b) shows the cross correlation effect between the local sequence and the k^{th} component of the received signal (this component is from the k^{th} simultaneous user). The k^{th} cross correlation function is now given by equation (A2.5) which is compared with its equivalent component in equation (A1.5).

$$\frac{1}{T_d}\int_0^{T_d} C_k(t-\tau_k)\, d_k(t-\tau_k)\, dt = \frac{1}{L}\left[d_{k,0}\, C_{k,1}(l) + d_{k,-1}\, C_{k,1}(l-L)\right] \qquad (A2.5)$$

Once again the CCF depends upon the relative data values of the k^{th} transmitter, hence if they are equal then:-

$$CCF_e = \frac{d_{k,0}}{L}\left[C_{k,1}(l) + C_{k,1}(l-L)\right] = \frac{d_{k,0}}{L}\,\theta_{k,1} \qquad \text{(A2.6)}$$

where the bracketed term is defined as the periodic or even cross correlation function. If the data values are different then:-

$$CCF_o = \frac{d_{k,0}}{L}\left[C_{k,1}(l) - C_{k,1}(l-L)\right] = \frac{d_{k,0}}{L}\,\hat{\theta}_{k,1} \qquad \text{(A2.7)}$$

where the bracketed term is defined as the odd cross correlation function. The total noise component due to the cross correlation effects is given by equation (A2.8):-

$$\text{Noise} = \frac{1}{L}\sum_{i=2}^{k}\left[d_{i,0}\,C_{i,1}(l) + d_{i,-1}\,C_{i,1}(l-L)\right] \qquad \text{(A2.8)}$$

In conclusion it is clear that the ACF and CCF are dependent on the odd and even autocorrelation function and cross correlation functions respectively with the probability of each data value determining the final CCF and ACF values.

APPENDIX 3

Chip SNRs in the SS-LAN

The modulated signals are transmitted from the nodes onto the baseband frequency in a binary format. The simultaneous user environment converts this binary signal into an analogue signal whose value at a particular time is the difference between the number of nodes transmitting a logic one and those transmitting a logic zero. The maxima and minima are set by the maximum number of simultaneous users and the voltage levels, as shown in figure (A3.1a). The SNR at the input to the receivers is shown as SNR_c in figure (6.5) and a theoretical analysis will now be derived for this entity. SNR_c is defined as:-

$$SNR_c = \frac{\text{Average Power of the Signal}}{\text{Average Environment Noise Power} + \text{Average simultaneous User Noise Power}}$$

The SS-LAN is assumed to employ a bipolar signalling technique with a voltage level set at $\mp V$ volts. The average power of the signal is given by equation (A3.1):-

$$\text{Average Signal Power} = \frac{1}{2}(V-\bar{v}l_s)^2 \qquad \text{(A3.1)}$$

where, l_s is the distance (in metres) between the transmitter and the receiver and, $\bar{v}$, is the voltage loss per metre due to the channel.

The average environmental noise power is assumed to be represented by the variance of the additive white noise, σ_e^2, and the average simultaneous user noise will be represented by the variance of the noise power distribution, σ_n^2, and consequently SNR_c is given by equation (A3.2):-

$$SNR_c = \left[\frac{2\sigma_n^2}{(V-\bar{v}l_s)^2} + \frac{2\sigma_e^2}{(V-\bar{v}l_s)^2}\right]^{-1} \qquad \text{(A3.2)}$$

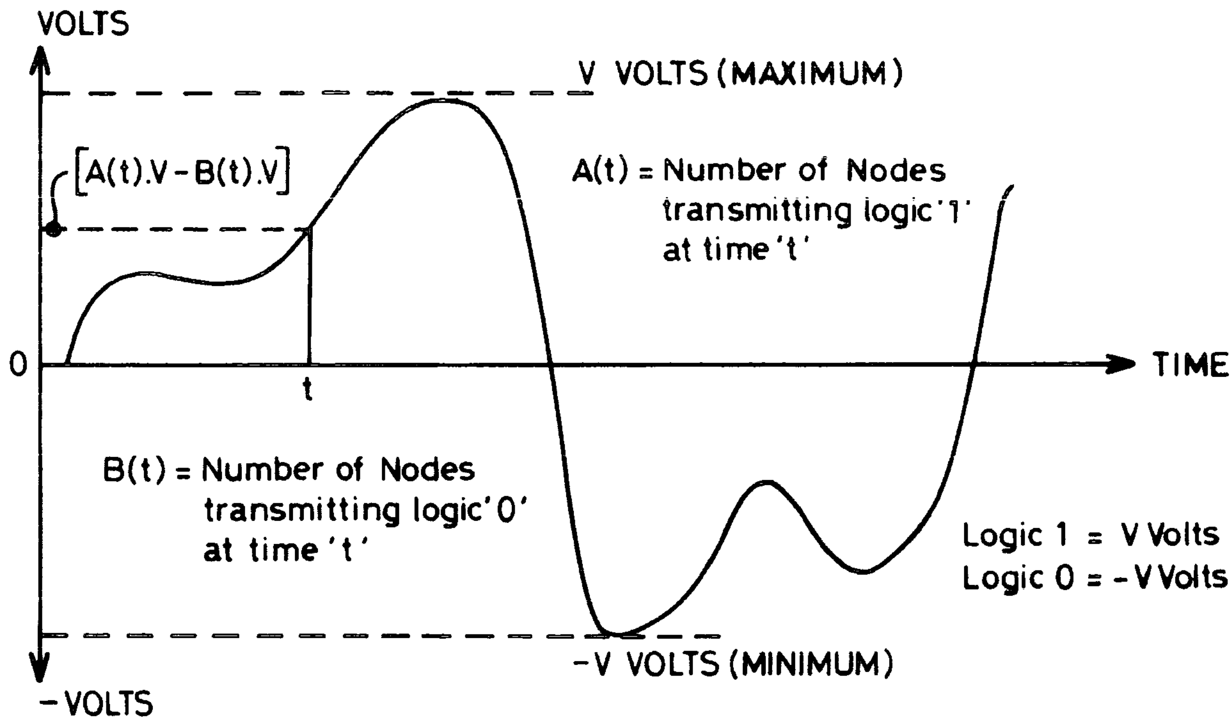


FIGURE (A3.1a): Analogue Signal Construction for the SS-LAN

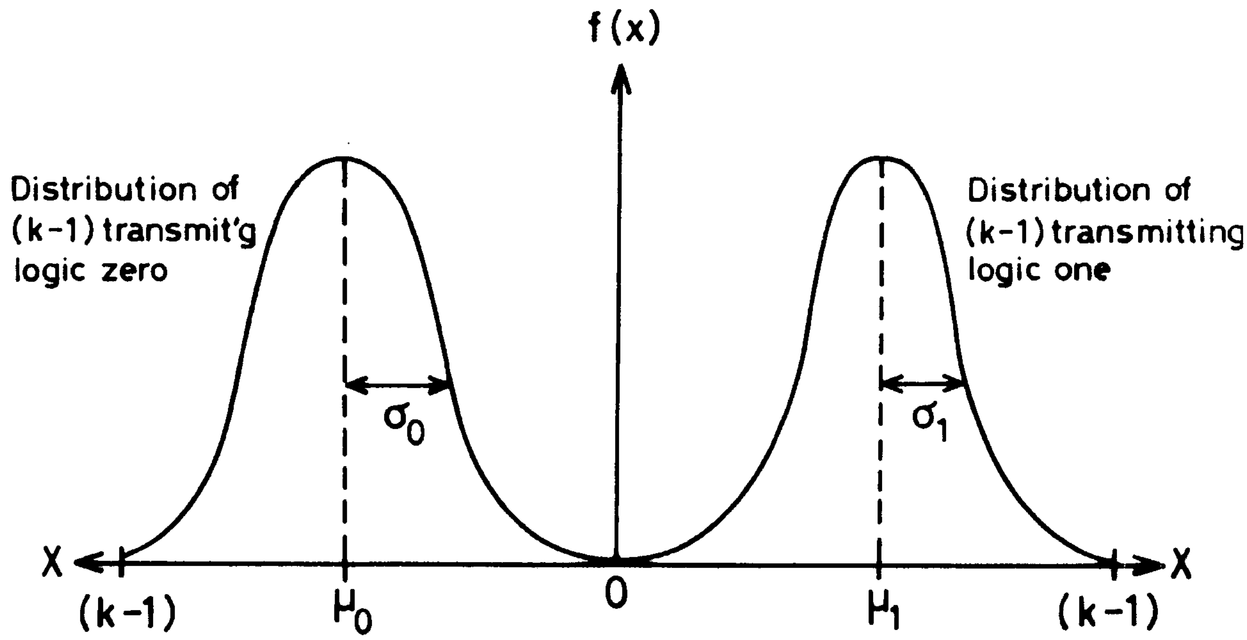


FIGURE (A3.1b): Distribution Curves of Signal Voltage Levels

The function $(V - \overline{V}l_s)^2/2\sigma_n^2$ can now be defined as the single user SNR at the input to the receiver and is denoted as SNR_1.

$$SNR_c = \left[\frac{2\sigma_n^2}{(V - \overline{V}l_s)^2} + \frac{1}{SNR_1} \right]^{-1} \qquad (A3.3)$$

The next step is to derive an equation for σ_n^2 which must be dependent on the number of simultaneous users and on the code sequence employed for transmission. Figure (A3.1b) shows the two distribution curves for (k-1) other simultaneous users whether transmitting logic one or zero. σ_0 and μ_0 are the standard deviation and mean of the distribution of the nodes transmitting logic zero respectively. The corresponding values for logic one are σ_1 and μ_1. The variance and mean for the resulting noise is represented by σ_n^2 and μ_n where:-

$$\sigma_n^2 = (\sigma_0 + \sigma_1)^2 \qquad \text{and} \qquad \mu_n = \mu_0 + \mu_1 \qquad (A3.4)$$

The distribution is represented by a Gaussian curve (all nodes transmit independently of each other) consequently the variance and mean for each distribution may be calculated from the discrete Gaussian curve formulas provided by Feller (41) where:-

$$\sigma_0^2 = (k - 1)P_0(1 - P_0)\,\frac{1}{2}(V - \overline{V}l_n)^2$$

$$\sigma_1^2 = (k - 1)P_1(1 - P_1)\,\frac{1}{2}(V - \overline{V}l_n)^2$$

$$(A3.5)$$

$$\mu_0 = -(k - 1)P_0\,(V - \overline{V}l_n)$$

$$\mu_1 = (k - 1)P_1\,(V - \overline{V}l_n)$$

P_0 = the probability of a node sending logic zero for the chip
P_1 = the probability of a node sending logic one for the chip
l_n = average distance between the noise transmitters and the receivers in metres.

For a cyclically random code let $P_0 = P_1 = 1/2$, then σ_n^2 and μ_n become:-

$$\sigma_n^2 = \frac{1}{2}(k - 1)(V - \bar{V}l_n)^2 \quad \text{and} \quad \mu_n = 0 \qquad \text{(A3.6)}$$

Substituting this variance into equation (A3.3) provides:-

$$SNR_c = \left[(k - 1)\left(\frac{V - \bar{V}l_n}{V - \bar{V}l_s}\right)^2 + \frac{1}{SNR_1}\right]^{-1} \qquad \text{(A3.7)}$$

The term $(V-\bar{V}l_n/V-\bar{V}l_s)^2$ is now represented by f(P), a function of the power ratio between the signal and the noise. Hence the final form is:-

$$SNR_c = \left[(k - 1)\ f(P) + \frac{1}{SNR_1}\right]^{-1} \qquad \text{(A3.8)}$$

The input SNR at the receiver is therefore dependent on the number of simultaneous users, the relative distance between the signal and noise sources and the receiver and the <u>single</u> user SNR. f(P) is a measure of the power loss due to the channel propagation and also of the relative source powers of the signal and the users noise sources.

$$f(P) = \left(\frac{V - \bar{V}l_n}{V - \bar{V}l_s}\right)^2 \qquad \text{(A3.9)}$$

Equation (A3.9) is the defining equation and this can be represented in a decibel form of:-

$$f(P) \text{ in dBs} = 20\ \mathrm{Log}_{10}\left(\frac{V - \bar{V}l_n}{V - \bar{V}l_s}\right) \qquad \text{(A3.10)}$$

Some important points to note are that when the source and the other users are equidistant from the receiver then $l_n = l_s$ and f(P) = 1 or 0dBs. If the signal is considerably nearer than the noise sources then $f(P) \rightarrow (V)^{-2}$, or if V=10 volts then $f(P) \rightarrow -20$dBs ie the SNR is <u>improved</u>

due to the proximity of the signal. Conversely if the signal is significantly further away than the noise then $f(P) \rightarrow V^2$ or $f(P) \rightarrow$ 20dBs ie the SNR is reduced due to the large displacement between the receiver and transmitter.

A more common approach to the definition of the near-far function is made by the consideration of the channel attenuation factor. The propagation of the signal along a coaxial cable can be described by a time independent wave equation of the form:-

$$V(x) = V_0 e^{-x\sqrt{\gamma}} + V_0 e^{x\sqrt{\gamma}} \qquad \text{(A3.11)}$$

where V_0 is the initial signal voltage, γ is the propagation constant and $V(x)$ is the voltage level at a distance, x, from the source. The two components represent the forward and reverse signals as they propagate in each direction along the channel. The propagation constant is comprised of the attenuation constant, α, and the phase constant, β, such that $\sqrt{\gamma} = \alpha + j\beta$.

If the forward direction of the signal is considered then it is represented in its time independent form by equation (A3.12):-

$$V(x) = V_0 e^{-\alpha x} . e^{j\beta x} \qquad \text{(A3.12)}$$

where the signal amplitude at the position, x, along the channel is given by:-

$$V(x) = V_0 e^{-\alpha x} \qquad \text{(A3.13)}$$

The definition of f(P) as given in equation (A3.9) can now be redefined in its exponential form as given in equation (A3.14):-

$$f(P) = \left(\frac{V_0 e^{-\alpha l_n}}{V_0 e^{-\alpha l_s}} \right)^2 \qquad \text{(A3.14)}$$

Simple algebraic manipulation of equation (A3.14) coupled with the definition of f(P) in decibels produces equation (A3.15), where:-

$$f(P) \text{ in dBs} = 8.69\alpha(l_s - l_n) \tag{A3.15}$$

When the frequency of transmission is in the megahertz region then the attenuation constant becomes proportional to the square root of the frequency (69) because the attenuation mechanisms are dominated by the "skin effect". The attenuation constant is now defined as:-

$$\alpha = (\pi\sigma\mu)^{1/2}.(f)^{1/2} \tag{A3.16}$$

where, μ, is the permeability of the conductor and , σ, is its conductivity. At a frequency of 10MHz the attenuation factor is typically 30dBs per kilometre for standard coaxial cable and therefore equation (A3.15) becomes:-

$$f(P)\Big|_{R_c=10MHz} \text{ in dBs} = 0.03\ (l_s - l_n) \tag{A3.17}$$

If the signal source is nearer to the receiver than the noise source then $l_s < l_n$ and so $f(P) < 0$dBs otherwise if the noise source is the closer then $f(P) > 0$dBs. A near-far loss of 10dBs can be produced by a signal source/noise source displacement of approximately 330 metres and one of 20dBs by 660 metres.

This analysis may also be modified to incorporate differences in the transmission power between the signal source and the other users. An average noise power could be incorporated however its effects would be similar to those of having the distance of the noise sources either nearer or further away depending on the actual relative powers.

APPENDIX 4

Correlation Thresholds and SNRs

The correlator in each of the nodes of the SS-LAN define the performance of the system ie a poor correlator will not be capable of resolving the required signal from the simultaneous multiple noise. The receiver is as shown in figure (6.5) where a hardlimiter feeds a sliding PCC which calculates the statistic:-

$$C_{xy} = \sum_{i=1}^{L} \operatorname{sgn}\left[r(t - iT_c)\right] \quad \operatorname{sgn}\left[C(t - iT_c)\right] \tag{A4.1}$$

C_{xy} is the difference between the number of agreements and disagreements obtained from the comparison of the incoming signal and the locally generated code sequence. Data is declared as present if the correlation is greater than some threshold, C_t, ie

$$\left|C_{xy}\right| > C_t$$

only the magnitude is necessary because the SIK must transmit both logic one and zero data bits. The SNR at the output of the correlator ie $\overline{SNR}_i$ can now be derived with respect to the operation of the correlator and its threshold. If P_a is defined as the probability of detecting an agreement then the probability of detecting x aggreements is:-

$$f(x) = \binom{L}{x} P_a^{\,x} \, (1 - P_a)^{L-x} \tag{A4.2}$$

From equation (A4.2) it follows that the probability of at least AG agreements is given by:-

$$Pr(x \rangle AG) = \sum_{x=AG}^{L} \binom{L}{x} P_a^{x} (1 - P_a)^{L-x} \qquad (A4.3)$$

As the agreements threshold AG is lowered then this probability increases and if the code sequence is large then $Pr(x \rangle AG)$ tends towards the Gaussian distribution (from the present Binomial distribution) with a skewness defined by, P_a.

A similar situation is found for the probability of detection of the number of disagreements. If P_d is the probability of the detection of a disagreement by the correlator then the probability of detection of data is given by:-

$$Pr(x \rangle DG) = \sum_{x=DG}^{L} \binom{L}{x} P_d^{x} (1 - P_d)^{L-x} \qquad (A4.4)$$

which again tends towards a Gaussian distribution if $L \to \infty$. The variance and mean of the agreements distribution is denoted by σ_a^2 and μ_a respectively and by σ_d^2 and μ_d for the corresponding entities for the disagreement distribution. Hence:-

$$\sigma_a^2 = L\ (1 - P_a)\ P_a \qquad \mu_a = LP_a$$

$$\sigma_d^2 = L\ (1 - P_d)\ P_d \qquad \mu_d = LP_d \qquad (A4.5)$$

If σ_n^2 is denoted as the resultant variance of the configuration of the two variances above and similarly for μ then the resulting SNR can be defined by:-

$$\overline{SNR}_i = \frac{(C_t - \mu)^2}{\sigma_n^2} \qquad (A4.6)$$

where C_t is the threshold for the difference between the agreements and disagreements and as shown in chapter two. For each correlation comparison the following rule must apply,

$$P_a + P_d = 1 \tag{A4.7}$$

and in figure (6.5), P, is defined as the probability of obtaining a correct bit from the hardlimiter. In the case of the presence of a signal then for greement to be correctly determined $P=P_a$ which means that if the relationships for σ^2 and μ are used as defined in appendix three then:-

$$\sigma^2 = 4L\ (1 - P)\ P \qquad\qquad \mu = L(2P - 1) \tag{A4.8}$$

This mean is produced because the correlation is a subtraction of the signals hence the mean will also be a subtraction. This provides the general SNR equation of (A4.9):-

$$\overline{SNR}_i = \frac{[C_t - L(2P - 1)]^2}{4L(1 - P)P} \tag{A4.9}$$

This equation states that the SNR at the output of the receiver is defined by the threshold set for the correlation. It is also therefore the seed for the probability of detection as shown by Wolff, Thomas and Williams (84). As C_t is lowered then the $\overline{SNR}_i$ is lowered but the probability of detection and error rate are increased because there is an increased likelihood of the correlation threshold being surpassed by the presentation of incorrect data from the hardlimiter. In the SS-LAN, $P \longrightarrow 1/2$ hence the $\overline{SNR}_i$ will tend to:-

$$\left.\overline{SNR}_i\right|_{P \to 1/2} \longrightarrow \frac{C_t^2}{L} \tag{A4.10}$$

where C_t is defined as $C_t = AG - DG$.

APPENDIX 5

A Simulation Scenario File (SSF)

The SS-LAN simulation system is configured by an SSF for the analysis of a particular network format. This configuration information includes the physical composition of the network and the transmission activity supported by the channel as well as defining the internal structure of each of the nodes. In the SSF shown below the network contains eight nodes distributed over a 25 metre long coaxial cable with each node operating in the unique receiver mode and employing pseudorandom sequences of 127 bits in length.

```
"NUMBER OF NODES"        8                  ! Total node number
"NUMBER OF RXRS"         3                  ! Receivers operational
"NUMBER OF SOURCES"      1                  ! Noise sources active

"MODE"              1                       ! Mode of operation
"READINGS"          1,1,1,1                 ! Type of readings
                    0,0,0,0
                    0,0

"FILENAMES"                                 ! Timings table file.
                                            ! Message table file.
                                            ! Synchronisation table file.
                                            ! Correlator table file.
                                            ! Rxed data table file.
                                            ! SNR table file.
                    "TKA004.TST"            ! Message readings file.
                    "TKA003.TST"            ! Timings readings file.
                    "TKA002.TST"            ! Correlator readings file.
                    "TKA001.TST"            ! SNR readings file.

"TOTAL TIME"                  20000         ! Time for simulation.

"HIGHWAY LENGTH"              25            ! Length in metres.

"RESOLUTION"                  2             ! Words/metre.

"CHANNEL"                     1             ! MHz Bandwidth
                              1             ! Quality factor
                              0             ! Decay factor

"SEQUENCE RATE"               2000          ! in 100kHzs increments

"SEQ/DATA RATIO"              1             ! seqs/data bit

"ADDRESSING TABLES"
                    No.=0                   ! Node Number

                              7             ! SR Length
                              127           ! Code sequence length
                              1             ! SR start state
                              16            ! SR feedback taps mask
```

```
                     1               ! SR O/P taps mask
          No.=1                      ! Node Number

                     7               ! SR Length
                     127             ! Code sequence length
                     1               ! SR start state
                     64              ! SR feedback taps mask
                     1               ! SR O/P taps mask
          No.=2                      ! Node Number

                     7               ! SR Length
                     127             ! Code sequence length
                     1               ! SR start state
                     2               ! SR feedback taps mask
                     1               ! SR O/P taps mask
          No.=3                      ! Node Number

                     7               ! SR Length
                     127             ! Code sequence length
                     1               ! SR start state
                     112             ! SR feedback taps mask
                     1               ! SR O/P taps mask
          No.=4                      ! Node Number

                     7               ! SR Length
                     127             ! Code sequence length
                     1               ! SR start state
                     8               ! SR feedback taps mask
                     1               ! SR O/P taps mask
          No.=5                      ! Node Number

                     7               ! SR Length
                     127             ! Code sequence length
                     1               ! SR start state
                     62              ! SR feedback taps mask
                     1               ! SR O/P taps mask
          No.=6                      ! Node Number

                     7               ! SR Length
                     127             ! Code sequence length
                     1               ! SR start state
                     84              ! SR feedback taps mask
                     1               ! SR O/P taps mask
          No.=7                      ! Node Number

                     7               ! SR Length
                     127             ! Code sequence length
                     1               ! SR start state
                     38              ! SR feedback taps mask
                     1               ! SR O/P taps mask

"NODE DEFINITION"
          No.=0                      ! Node number

                     20              ! Position on highway in metres
                     1               ! Type of transmitter
                     -500            ! Output logic 0 in mVs
```

```
                500         ! Output logic 1 in mVs.
                1           ! Receiver state, -1=inactive

                1           ! Receiver type
                0           ! ADC threshold in mVs
                90          ! Correlator threshold
No.=1                       ! Node number

                1           ! Position on highway in metres
                1           ! Type of transmitter
                -500        ! Output logic 0 in mVs
                500         ! Output logic 1 in mVs
                -1          ! Receiver state, -1=inactive
No.=2                       ! Node number

                20          ! Position on highway in metres
                1           ! Type of transmitter
                -500        ! Output logic 0 in mVs
                500         ! Output logic 1 in mVs
                1           ! Receiver state, -1=inactive

                1           ! Receiver type
                0           ! ADC threshold in mVs
                90          ! Correlator threshold
No.=3                       ! Node number

                1           ! Position on highway in metres
                1           ! Type of transmitter
                -500        ! Output logic 0 in mVs
                500         ! Output logic 1 in mVs
                -1          ! Receiver state, -1=inactive
No.=4                       ! Node number

                12          ! Position on highway in metres
                1           ! Type of transmitter
                -500        ! Output logic 0 in mVs
                500         ! Output logic 1 in mVs
                1           ! Receiver state, -1=inactive

                1           ! Receiver type
                0           ! ADC Threshold in mVs
                90          ! Correlator threshold
No.=5                       ! Node number

                1           ! Position on highway in metres
                1           ! Type of transmitter
                -500        ! Output logic 0 in mVs
                500         ! Output logic 1 in mVs
                -1          ! Receiver state, -1=inactive
No.=6                       ! Node number

                1           ! Position on highway in metres
                1           ! Type of transmitter
                -500        ! Output logic 0 in mVs
                500         ! Output logic 1 in mVs
                -1          ! Receiver state, -1=inactive
No.=7                       ! Node number
```

```
                                25         ! Position on highway in metres
                                1          ! Type of transmitter
                                -500       ! Output logic 0 in mVs
                                500        ! Output logic 1 in mVs
                                -1         ! Receiver state, -1=inactive

"NODE ACTIVITY"
                                1          ! Time of Tx.
                                3          ! No. of nodes.

                                1          ! Source node.
                                0          ! Destination node.
                                35         ! Data bits sent.
                                3          ! Source node.
                                2          ! Destination node.
                                35         ! Data bits sent.
                                5          ! Source node.
                                4          ! Destination node.
                                35         ! Data bits sent.

                                -1         ! End of activity marker.
```

APPENDIX 6

Task A System Log (F%)

The task A system log is a commentary on the validity of the data contained within the SSFs (see appendix five). Once the SSF has been validated then the log displays the parameters calculated for the internal operation of the simulator. The log shown below is the validation commentary of the SSF shown in appendix five in which the simulator analyses a period of only 33 microseconds for an actual SS-LAN communications system.

```
Simulator System Log
====================

Start up time: 27-JUN-1985    15:16:06.14


TASK A STARTED
==============

Data validation and parsing

Initialisation at:  27-JUN-1985    15:16:06.19

Command file open

Simulation run file open

Task initialised, start data read: 27-JUN-1985    15:16:06.55

Task A data read process

Task A data read initialised

Set param reading
Set param complete
Physical param reading
Physical param complete
System param readings
System param complete
Address table params
Addressing complete
Node definition params
Node definition complete
Activity info reading
Activity read complete
Hardware param reading
Hardware param complete
Noise source readings
Noise param complete
Data read complete: 27-JUN-1985    15:16:21.59
```

++

```
          Modelled simulation parameters

Code sequence freqs in MHzs          200.00000
Code sequence period in msecs          0.000005
Highway propagation time in micros     0.083333
Highway pulse period in micros         0.001667
Total simulated run time in micro     33.333332
Period of 127 bit SS sig in msecs      0.000635
Period of 1024 bit SS sig in msecs     0.005120
Highway pulses/chip period                3

          Modelling complete
```

++

```
Data Parsing
Initialisation Complete
File validation
File validation complete
Highway validation
Highway validation complete
Node definition validation
Node definition complete
Run time validation
Run validation complete
End of Data Parsing
Data parse complete:     27-JUN-1985    15:16:33.17

Task A termination procedure

Command file is closed

Termination due to EOF of command file

TERMINATION COMPLETE

TASK A STOP
```

APPENDIX 7

Task B System log and Simulation Readings (F%)

The task B system log is a commentary on the actual SS-LAN simulation and also contains, in the present version of the software, the readings obtained from that simulation. The log shown below for task B is based upon the files shown in appendices five and six and the results indicate that the three active receivers, at nodes zero, two and four, have correctly received the 35 bits of data transmitted to them.

```
Simulator System Log
====================

Start up time: 27-JUN-1985    15:14:34.37


TASK B STARTED
==============

SPREADNET simulator

Initialisation complete at:    27-JUN-1985    15:16:33.90

Simulator start time:    27-JUN-1985    15:16:33.93
Pulse No. 1000

NODE No.= 4

Average Signal =    -500
Maximum Signal =    500
Minimum Signal =    -500
Average Noise =     -1000
Maximum Noise =     1000
Minimum Noise =     -1000
Average SNR =       -995
Maximum SNR =       500
Minimum SNR =       -500

Average No. of 1s agrees =     96
Average No. of 1s disagr =     31
Average No. of 0s agrees =     0
Average No. of 0s disagr =     0
Average No. of 1s difference =     65
Average No. of 0s difference =     0
Maximum No. of 1s agrees =     96
Minimum No. of 1s agrees =     0
Maximum No. of 1s disagr =     31
Minimum No. of 1s disagr =     0
Maximum No. of 1s difference =     65
Minimum No. of 1s difference =     0
Maximum No. of 0s agrees =     0
Minimum No. of 0s agrees =     0
Maximum No. of 0s disagr =     0
```

Minimum No. of 0s disagr = 0
Maximum No. of 0s difference = 0
Minimum No. of 0s difference = 0

Start time of data 403
Finish time of data 13357

Received DATA -1
Received DATA 7

NODE No. = 0

Average Signal = 500
Maximum Signal = 500
Minimum Signal = -500
Average Noise = -1000
Maximum Noise = 1000
Minimum Noise = -1000
Average SNR = -995
Maximum SNR = 500
Minimum SNR = -500

Average No. of 1s agrees = 96
Average No. of 1s disagr = 31
Average No. of 0s agrees = 0
Average No. of 0s disagr = 0
Average No. of 1s difference = 65
Average No. of 0s difference = 0
Maximum No. of 1s agrees = 96
Minimum No. of 1s agrees = 0
Maximum No. of 1s disagr = 31
Minimum No. of 1s disagr = 0
Maximum No. of 1s difference = 65
Minimum No. of 1s difference = 0
Maximum No. of 0s agrees = 0
Minimum No. of 0s agrees = 0
Maximum No. of 0s disagr = 0
Minimum No. of 0s disagr = 0
Maximum No. of 0s difference = 0
Minimum No. of 0s difference = 0

Start time of data 418
Finish time of data 13372

Received DATA -1
Received DATA 7

NODE No. = 2

Average Signal = -500
Maximum Signal = 500
Minimum Signal = -500
Average Noise = -1000
Maximum Noise = 1000
Minimum Noise = -1000
Average SNR = -992

```
Maximum SNR =        500
Minimum SNR =        -500

Average No. of 1s agrees =    102
Average No. of 1s disagr =    25
Average No. of 0s agrees =    0
Average No. of 0s disagr =    0
Average No. of 1s difference =     77
Average No. of 0s difference =     0
Maximum No. of 1s agrees =    102
Minimum No. of 1s agrees =     0
Maximum No. of 1s disagr =    25
Minimum No. of 1s disagr =    0
Maximum No. of 1s difference =     77
Minimum No. of 1s difference =     0
Maximum No. of 0s agrees =    0
Minimum No. of 0s agrees =    0
Maximum No. of 0s disagr =    0
Minimum No. of 0s disagr =    0
Maximum No. of 0s difference =     0
Minimum No. of 0s difference =     0

Start time of data  418
Finish time of data 13372

Received DATA  -1
Received DATA  7

Pulse No. 20000
Simulator finish time:   27-JUN-1985    16:29:43.22

Logical termination of task B
Task B termination: 27-JUN-1985    16:29:43.66
```