FINAL REPORT

IMPACT OF USE OF SATELLITE FOR DATA

prepared for:

DEPARTMENT OF COMMUNICAT
Journal Tower North
300, Slater Street

CDT
Centre de Développement Technologique

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

CDT
Centre de
Développement
Technologique
École Polytechnique
de Montréal

PROJET CDT P832

FINAL REPORT

IMPACT OF USE OF SATELLITE FOR DATA COMMUNICATIONS

prepared for:

DEPARTMENT OF COMMUNICATIONS
Journal Tower North
300, Slater Street
Ottawa (Ontario) K1A 0C8

by:

Jean Conan, Ph.D.
David Haccoun, Ph.D.
Hai-Hoc Hoang, Ph.D.
Electrical Engineering Department
ECOLE POLYTECHNIQUE DE MONTREAL

submitted by:

LE CENTRE DE DÉVELOPPEMENT TECHNOLOGIQUE
DE L'ÉCOLE POLYTECHNIQUE DE MONTRÉAL
Campus de l'Université de Montréal
Case postale 6079, Succursale A
Montréal (Québec) H3C 3A7

April 1984

Jean Conan

Dr Jean Conan
Project Director

CDT
Centre de
Développement
Technologique
École Polytechnique
de Montréal

TABLE OF CONTENTS

(Cont'd) - TABLE OF CONTENTS

(Cont'd) - TABLE OF CONTENTS

(Cont'd) - TABLE OF CONTENTS

LIST OF FIGURES

(Cont'd) -

LIST OF FIGURES

(Cont'd) -

LIST OF FIGURES

LIST OF TABLES

1. PREFACE

As the communication traffic generated by digital data and numerically encoded analog signals is expected to drastically increase in the coming years, it becomes of prime importance for the designers to analyse and field test all the newly available transmission and networking techniques in order to ensure a prober level of service while, at the same time, maintaining appropriately low initial and operational costs. To this end, the impact of using recently introduced communication channels such as the satellite channel must be clearly ascertained, either by analysis or simulation of the different possible scenarios, so that new potential applications can be promoted in the most cost efficient manner.

In such a new framework, the cost of using terrestrial lines as well as transmission systems and network structures similar to the ones currently operated by the common carriers can become prohibitive in a large scale operation such as the one operated by the Government Telecommunications Agency. For example, in this particular case, it might turn out to be more economical, on a long run basis, to carry out the transmission load on a different type of communication network. In addition, the use of more appropriate transmission schemes can result in added values simultaneously in the classical as well as the new applications that become possible thereafter. In relationship with such a different approach, some important questions related, for example, to the possibility of integration of all services of voice and data simultaneously on the same channel must be investigated. Other alternatives including special purposes networks for certain specific applications should also be considered. The recent emergence of communication satellites, either domestic (i.e., the Canadian Anik series of satellites owned and managed by Telesat) or international (i.e., the

Intelsat satellite network), makes possible the use of star shaped communication nets using as terminal nodes relatively inexpensive roof top earth stations. Furthermore, the wide bandwidth of the resulting communication channel as well as its inherent broadcasting capability makes it suitable for the implementation of non classical (i.e., typically non telephone oriented applications) network structures. With respect to the geographical constraints of the country, it is expected that such new telecommunication structures will provide, taking into account the added values inherent in the use of such a new networking arrangement, a sensible reduction in the overall cost of running the communication needs under the responsability of the Government Telecommunication Agency.

Referring to the above remarks, the main objective of the research whose finding are collected in this report has been to investigate the possibility of using satellite communication links and related network structures to fulfill the data communication requirements which fall under the authority of the Agency. Our main methodology has been to project the present state of the art in communication satellite technology in prospective with the needs of the Agency.

2. INTRODUCTION

Since within the current tariff policy of the common carriers the cost of using conventional terrestrial lines and transmission systems can be prohibitive in the running of a large scale transmission network such as the one operated by the Government Telecommunication Agency; it becomes of prime importance for the decision making authorities to be able to ascertain the impact of both new technological breakthroughs and systems developments into the cost/performance trade-off problem associated with their communication network. In this context, it appears that communication satellites have potentially interesting properties which make them extremely attractive for certain applications.

2.1 An Overview of Communication Satellites

The use of satellite based communication systems as practical means of transmission of both continental and intercontinental voice and video information bearing signals is certainly not a new idea when we refer back to the first experimental communication satellite (Telstar) which was launched in July 1962. Following this early experiment, the first commercial international satellite network (Intelsat I) was put in operation in 1965. The different lessons which have been drawn from these early pioneering experiments as well as the ones that followed thereafter have led to a better general understanding of the specific advantages to be expected from the use of satellites as well as the different constraints under which these systems operate. Furthermore a new highly sophisticated electronic technology has emerged from the difficult environment in which satellites operate. From a practical standpoint, a communication satellite might be thought of as a big repeater in the sky. It carries on board one or more repeaters,

each of which listens to some portion of the spectrum, amplifies the incoming signals and rebroadcasts them at a different frequency so as to avoid interference with the incoming signals. The downward electromagnetic beams can be either broad, covering a substantial fraction of the earth's surface, or narrow, concentrating the beam over a terrestrial area only a few hundreds of kilometers in diameter. Following the recent considerable reduction in the price, size as well as power requirements of microelectronics, a new and sophisticated broadcasting strategy which goes by the name of satellite switching has also emerged in the past decade. In such a scheme, each satellite is equipped with multiple antennas and transponders systems so that each downward beam can be focused on a small geographical area and multiple upward and downward transmissions can take place simultaneously. These typically elliptically shaped spot beams can be so small so as to leave an imprint of only a few hundreds kilometers on the earth's surface. Furthermore, as the incoming upward messages reach the satellite, they can be switched (Satellite switching) to the appropriate transponder/spot-beam pair for rebroadcasting downwards. Within the framework of such an organization, it then becomes feasible to provide all the services of a fully switched mesh network within the physical infrastructure of a star shaped network. Equivalently, it follows that by using such a scheme a single satellite system can replace several more conventional network structures.

Yet, another major difference between satellite channels and terrestrial ones is the cost. As an example used only for illustrative purposes, modern satellite spacecrafts carry in their payload at least six transponders which allow each for a traffic of roughly 50 Mbps (Based on a value of 60 Mhz of allowable bandwidth per transponder). Converted back into the equivalent of voice grade circuits (32 Kbits/s) this amounts to roughly 5000 bidirectional voice grade circuits. A

satellite costs about 20-30 million dollars while a launching rocket amounts probably to the same amount. Furthermore, once launched a satellite system can be expected to have a life time of ten year. Within these ultimate parameters constraints, the cost of a voice-grade circuit should not be expected to go down under 500 dollars per year. Of course we did not take into account the cost of the earth stations which probably will go down in the tens of thousands of dollars each, and hence should be quite negligible on a first approximation basis.

A more careful cost breakdown must however be carried out on a short term basis taking into account the cost of purchasing and maintaining the ground equipment, of borrowing money and, most of all of leasing a transponder or part of a transponder according to the "lease cost" published as a tariff submission to the CRTC. The following two cases have been considered as typical of the most likely scenarios to apply in the coming years.

-- CASE 1: FDM/FM system comprising two earth stations carrying three Master-groups and operating under a rigid fixed assignment point to point arrangement.

- Assumptions:

(1) Input signals: Mastergroup (600 voice channels) SSB/FDM occupying (60 KHz - 2.8 MHz) bandwidth.

(2) Bandwidth occupied by frequency modulated mastergroup is 4 times highest modulating frequency. This yields 4x2.8 = 11.2 MHz. As a consequence a 60 MHz transponder is required to handle tree Master groups (i.e., 1800 voice channels/transponder).

(3) Transponder leasing charge 2.2×10^6 $/year/transponder.

(4) All costs are amortized over a 5 years period at 12%.
(5) A mark-up factor of 2.5 is used for maintenance and services.

- Cost calculation:

(a) Space segment cost $2.2 \times 10^6/(1800 \times 12)$ = 102 $/circuit/month
(b) Echo-cancellers (600 $/channel-end) = 67 $/circuit/month
(c) IF and RF equipments for two earth stations (100 K$/station) =12200 $/month
(d) Two antennas installed (90 K$/antenna) = 4000 $/month

TOTAL COST = ((12200+400)/1800)+67+102 = 180 $/circuit/month

-- CASE 2: TDMA system comprising 8 earth stations with Pre/Demand-assigned multiple access.

- Assumptions:

(1) Voice channels digitally encoded at 32 Kb/s.
(2) Satellite bandwidth efficiency 1.5 bit/Hz.
(3) 10% system overhead in bit transmission.
(4) Items 3,4 and 5 of Case 1 above apply.

- Cost calculation:

(a) Space segment cost $2.2 \times 10^6/(1260 \times 12)$ = 146 $/circuit/month
(1260 is the number of circuits per transponder; i.e., $0.9 \times 60 \times 10^6 \times 1.5/(2 \times 32 \times 10^3)$)
(b) Echo-cancellers (600 $/channel-end) = 67 $/circuit/month

(c) IF + RF cost for one earth station = 6100 $/month
(110 K$/station)

(d) One antenna installed = 2000 $/month
(90 K$/antenna)

(e) One TDMA (180 K$/station) = 4000 $/month

TOTAL COST (8x(6100+2000+4000)/1260)+67+146 = 290 $/circuit/month

Nevertheless and following the foregoing discussion, the key point to be made is that depending upon whether one is a user or a carrier with a huge interest in terrestrial lines, communication satellites can be perceived either as a great bonanza or as a highly disruptive threat for future business.

2.2 The Government Telecommunication Agency Data Communication needs

The satellite communication network to be managed by the GTA on behalf of the Federal Government is primarily intended to provide the transmission infrastructure necessary for the integration of a wide range of communications requirements from the various governmental departments and agencies. Since no pre-existing infrastructure of this kind must be taken into account, the future network can be based on the state-of-the-art in satellite technology and provide for the appropriate communication access techniques necessary to complement the existing government shared intercity voice network.

Although the network should be eventually fully integrated, it is expected that the capacity will be initially used on a Pre-Assigned Multiple Access mode (PAMA) and, mainly for digital voice transmission. The access technique will almost certainly be based on the use

of TDMA (Time Division Multiple Access). A second step in the evolution of the network will be the integration of data transmission which is the main area of concern in this study. In this context, it is expected taking into account the actual cost of satellite data circuits versus the corresponding terrestrial circuits as well as the distance between source and destination ant the actual transmission rate that cost effectiveness can be obtained by the use of satellite circuits. Furthermore, since most of the data traffic is bursty (i.e., non continuously dense in time), Pre-Assignment of channels to data traffic can be inherently wasteful of bandwidth and, as such, the use of both demand and random assignment could mean a substantial cost reduction over the more classical fixed assignment techniques. As a consequence, the network must eventually be able to evolve to the Demand Assigned Multiple Access mode (DAMA) so as to allow for the allocation, on demand, of various bandwidths between the different earth stations as well as the efficient sharing of the so formed individual channels between the corresponding local users.

Since the traditional telephone-type voice traffic is not the subject of this study, we now concentrate on the potential use of the satellite to the transmission of data related information. In this context, the rapid explosion of new services in the "Telematic" area makes it necessary to quickly review the associated new potential services which can be envisioned in the very near future. These new services will be essentially based on the digital representation of information associated with the following three basic representation media: written, audio and video documents. Typically the digital equivalent will require the following representation speeds.

(1) Written documents:

The digital equivalent obtained by slow optical scanners requires of the order of 300 bits/character, while the classical numerical codification using teletypes and other similar devices typically yields 8 bits/character.

(2) Audio documents:

The range of speeds required varies typically from 32 Kbits/sec (ADPCM techniques) for voice-grade quality speech reproduction to 500 Kbits/sec for HI-FI quality.

(3) Video documents:

Black and white picture with limited motion reproduction uses typically 2 Mbits/sec while the requirements for high-grade animated pictures (i.e., broadcast quality color TV) can be as high as 100 Mbits/sec.

At the present time, it is safe to say that "Telematic" services will be based almost uniquely on the use of written documents, audio and motionless video as well as high resolution graphics. Further improvements in data compression techniques should be expected to allow for higher resolution as well as a larger range of animation speeds. For example, the use of LPC (Linear Predictive Coding) vocoders should reduce the speed of audio messages down to the 1-2 Kbits/sec range while in the field of video applications the latest TV video compression techniques are expected to yield transmission speeds of the order of 35 Mbits/sec for broadcast quality TV signals. Similarly compression techniques do exist to encode teleconferencing monochrome video signals into 56 Kbits/s channels.

We can summarize the potential applications of these new services as follows:

(1) Videotex: broadcast of alphanumerical and graphical informaton on video screens.

- Over the counter inquiry services.
- Messenger services.
- Electronic directories.
- etc...

(2) Audio-conferencing: audio and graphical information exchange between serveral parties.

- Can also be supplemented by different information exchange tools such as facsimile.

(3) Teletex: quasi instantaneous transfer of written documents with quality reproduction. Teletex and facsimile combines very effectively to provide for the services of a true electronic mail.

(4) Facsimile: allows for the reproduction, irrespective of distances of any paper based document.

- Electronic mail with much decreased delivery time and quality.

(5) Visio/Video-conferencing: a microphone, video camera and terminal in one single "telephone" set.

- TV quality video document projection.
- Interactive data, voice and video communication.

One of the main characteristics of data related transmissions is the burstiness associated with the overall traffic. Data sources have a tendency to emit bursts of information (Usually referred to as packets) interspersed with idle time slots. More specifically, when a large number of data sources are combined independently, the overall traffic can be well approximated by a Poisson point process whose parameter (A measure of the traffic intensity) is expressed in some specific units (For example, packets) per second.

2.3 The Satellite Channel Environment

When communication satellites were first launched, they were used exactly as would be terrestrial facilities; i.e., for point to point communication. This mode of utilization had obvious advantages at that time. First, the cost of a satellite ground station was in the range of the tens of millions of dollars making it very unlikely that many people or any company would buy one for their own private use. Second, because the satellite was used only to interconnect common carriers switching office, the use of the satellite link was more or less transparent (Except for the propagation delay) to the users. Third, and by no means least, circuit switching using point-to-point lines was the most popular and only used technique at this particular time. As such, using a new technology to emulate an old one was the most natural option. In a mere two decades the situation has changed radically. The cost of professional ground stations has gone down by at least three orders of magnitude so that the privately owned corporate earth station becomes feasible. Each company plant, university campus, government building, etc..., could have its own rooftop antenna communicating directly with the satellite. As a consequence, the confluence of these developments and the need to transmit data suggest a radically different use of the satellite communication facility. The

key question to be addressed is the sharing, in an efficient and fair fashion, between uncoordinated and widely dispersed users of a single large bandwidth telecommunication channel with an intrinsic propagation delay whose nominal value, for a single hop transmission, is roughly 0.26 second. Most of our analysis will revolve around this basic one issue. Obviously, it should be expected that if everyone just begins transmitting whatever he wants to whenever he wants to, without regard to what other users are doing chaos will result, and there will be no possible communication at all.

To emphasize this important point, we now state the fundamental assumptions which are made whenever satellite packet broadcasting or Packet Satellite Switching (PSN) techniques are used. If two users broadcast simultaneously, the satellite will rebroadcast the "sum" of the two incoming signals, resulting in garbage. Equivalently, a collision between packets is said to have occured. We therefore assume that each packet carries a strong enough checksum capable of detecting all collisions, so garbled packets can be discarded. Another important property of PSN is that the sender can listen for his own packet, one round-trip time (i.e., one hop propagation time or roughly 0.26 second) later. Since the sender can tell from this whether or not a collision has occured, there is no need in such schemes for an explicit destination to source acknowledgment. If the packet was garbled, the sender learns of the problem simultaneously with the receiver and can take appropriate action to correct the situation without being told explicitly to do so.

Some of the basic advantages of PSN over other conventional networks can be summarized as follows:

(1) Protocols are much simpler to implement mostly because no acknowledgments are required.

(2) The routing problem does not exist.

(3) Congestion problems are avoided because of the sharing of the entire capacity between the different users.

(4) Increasing the scale of the network is just a question of adjusting one single parameter, namely the channel bandwidth rather than performing a complicated heuristic topology optimization.

(5) Mobile users can be easily accomodated.

From a general standpoint, the principal disadvantages of satellite PSN are the longer variable transmission delays and the need for a large number of still relatively expensive antennas. Furthermore, since no acknowledgments are transmitted, effective forward error correction schemes must be in use in order to maintain the required high level of reliability with respect to the transmitted data.

2.4 Outline of Report Content

In the writing of this report, the approach taken by the authors has been to concentrate on the main question which we believe to be the key issue in the use of a satellite channel, namely the sharing between different application users of a commonly available communication channel bandwidth and power.

In this respect, a detailed account of the classical sharing techniques has been carried out in chapter 3. Among the features most likely to be of interest to the Agency are the now currently available SCPC (Single Channel Per Carrier) and MCPC (Multiple Channel Per Carrier) Access Techniques as well as the recently announced Light Route Time Division Multiple Access technique with Demand Access (LRTDMA/DA).

Another fundamental area of concern for the users of satellite based multiple access methods such as PSN is the level of reliability and confidentiality that must be attached to the data during their transmission through the network. The important issue of reliability requires the use of error control procedures. However, in the context of the satellite channel where the propagation delay is quite large, classical terrestrial error control schemes such as error detection and retransmission fail to perform efficiently so that forward error correction is often required either with or without error detection and repetition. The important alternatives are discussed in Appendix A together with a qualitative appraisal of their respective performance. Moreover, whenever one uses a satellite channel, it becomes in fact possible for anybody who cares to go through the trouble of setting up his own receiving antenna to listen to the incoming packets and use this information for his own benefits (Eaves-dropping) and even masquerade as a legitimate user by sending his own fraudulent packets. In such an environment it then becomes of prime importance to protect the privacy of the individual network users by the use of cryptographic schemes. A complete account of all presently available cryptographic techniques including the DES (Data Encryption Standard) and PKS (Public Key crypto Systems) is available in Appendix B.

Finally, the report concludes in chapter 5 by a qualitative analysis of the different access methods taking into account the particular requirements of the Agency. Several possible scenarios are then proposed which could serve as the basis for subsequent and more qualitative studies.

3. CHANNEL ORIENTED MULTIPLE ACCESS TECHNIQUES

3.1 Introduction

Satellite Communication systems are characterized by some combinations of baseband processing, modulation scheme, multiplexing and multiple access techniques.

Multiple access is the shared use of the capacity of the satellite channel, that is the sharing of the bandwidth and power. Multiple access techniques refer to the techniques that allow a number of earth station to simultaneaously interconnect their respective transmission links through a common satellite transponder. Strictly speaking multiple access is an independent concept of baseband processing, multiplexing and modulation. However, all these elements are closely related in a practical system, and very often a particular multiple access technique assumes a given combination of baseband processing modulation and multiplexing procedures. For example in SCPC/PCM/PSK/FDMA, the baseband processing method is Pulse Code Modulation, the RF modulation method is Phase Shift Keying, the multiplexing scheme is single channel and finally the multiple access scheme is Frequency Division Multiple Access.

Multiple access techniques can be classified according to the channel allocation or the type of transponder sharing as follows.

(a) Fixed or Pre-assigned multiple access, in which the channels are permanently assigned to particular users.

(b) Demand assignment or Random access multiple access, in which the channel allocation is changed according to the demand from the users.

In such a system a particular channel is selected and assigned to a pair of users only for the duration of their active communication. In comparison to fixed-assignment multiple-access, such a system remarkably increases the efficiency of the satellite transponder.

Finally, according to the type of transponder sharing, multiple access techniques may be classified as:

(i) Frequency Division Multiple Access (FDMA) in which each earth station has its own carrier frequency and bandwidth to use all the time.

(ii) Time Division Multiple Access (TDMA), in which each earth station uses all the available satellite or transponder bandwidth but only during predetermined and disjoint time-slots.

Table 3.1 shows various types of transmission systems combining base band processing, modulation and multiplexing methods together with their adaptability to the different classes of multiple-access.

We now examine in more details FDMA and TDMA as they apply to data communication.

3.2 Frequency Division Multiple Access (FDMA) and Single Channel Per Carrier (SCPC)

Frequency Division Multiple Access is the simplest and most widely used multiple access in the first generations of communication satellites, where each earth station in the network transmits one or more distinct carriers. Each RF carrier occupies its own frequency band and is assigned a specific location within the entire repeater

TABLE 3.1: Adaptability of baseband processing, multiplexing and modulation methods to multiple access modes

Systems	Baseband Processing	Multiplexing	RF Modulation	Multiple access			
				FDMA	TDMA	Pre-Assigned	Demand assigned
FDM-FM/FDM-FMC	SSB/SSBC	FDM	FM	X		X	
SCPC-FM	Companding	Single channel	FM	X		X	X
SCPC-PCM-PSE	PCM	Single channel	PSK	X		X	X
PCM-TDM-PSK	PCM	TDM	PSK	X	X	X	
TDM-QPSK		TDM	QPSK		X	X	X

FDMA: Frequency Division Multiple Access
TDMA: Time Division Multiple Access
FDM : Frequency Division Multiplex
TDM : Time Division Multiplex
PCM : Pulse Code Mululation
FM : Frequency Modulation
FMC : Frequency modulation with syllabic companding
PSK : Phase Shift Keying
QPSK: Qaudrature Phase-Shift Keying
SCPC: Single Channel Per Carrier
SSB : Single side band modulation
SSBC: Single side band modulation with syllabic companding

bandwidth. Guard bands between adjacent carriers are provided to prevent over-lap-ping and to simplify filters and oscillators designs. The satellite transponder receives all carriers allocated in its bandwidth, amplifies them and retransmits them back to earth. Receiving earth station select their desired carriers that contain messages destined to them (See Figs. 3.1 and 3.2).

In elementary forms of FDMA there is no coordination and no clocking control between accessing stations. Each carrier can use either analog modulation such as Frequency Modulation, or digital modulation such as Phase Shift Keying.

One of the main problems of FDMA is the presence of intermodulation products within the carrier bandwidth. These intermodulation products are due to the amplification of multiple carriers by a commun traveling wave tube amplifier that is characterized by both amplitude and phase nonlinearities. As the number of carrier accesses increase, the TWT amplifier is operated near its saturation point in order to supply the required power per carrier, but in the process, the level of intermodulation products is also increased, thus affecting the overall performance. An analysis shows that the intermodulation products produced by the amplitude nonlinearity of the TWT amplifier operating near its saturation point are dominated by third order types. As for their amplitude values, they can be evaluated from the single-carrier amplitude or power transfer characteristics (BHAR 81, SPIL 77).

3.2.1 FDMA Utilization

As mentionned earlier, FDMA is one of the most common and simplest form of multiple access technique. FDMA is extensively used in television distribution and broadcasting by satellite where an entire transponder bandwidth (36 MHz) is often dedicated to a single T.V.

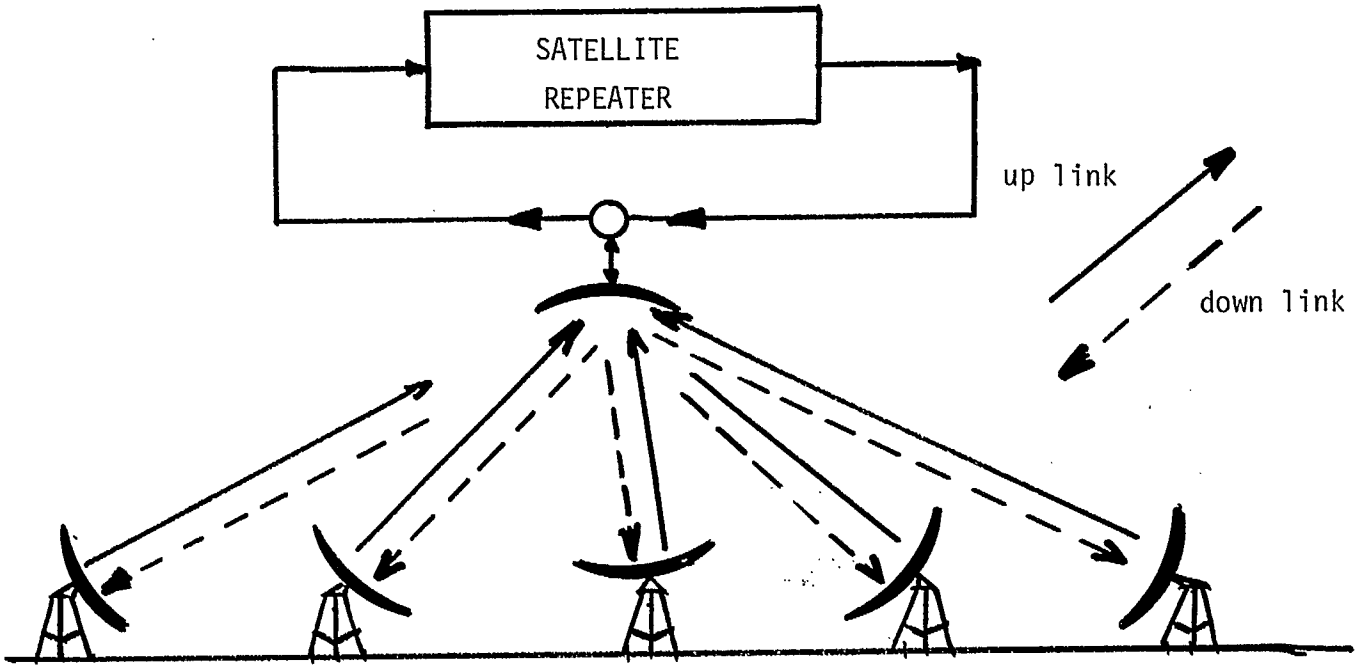


Figure 3.1 Bilateral links connecting earth stations in FDMA.

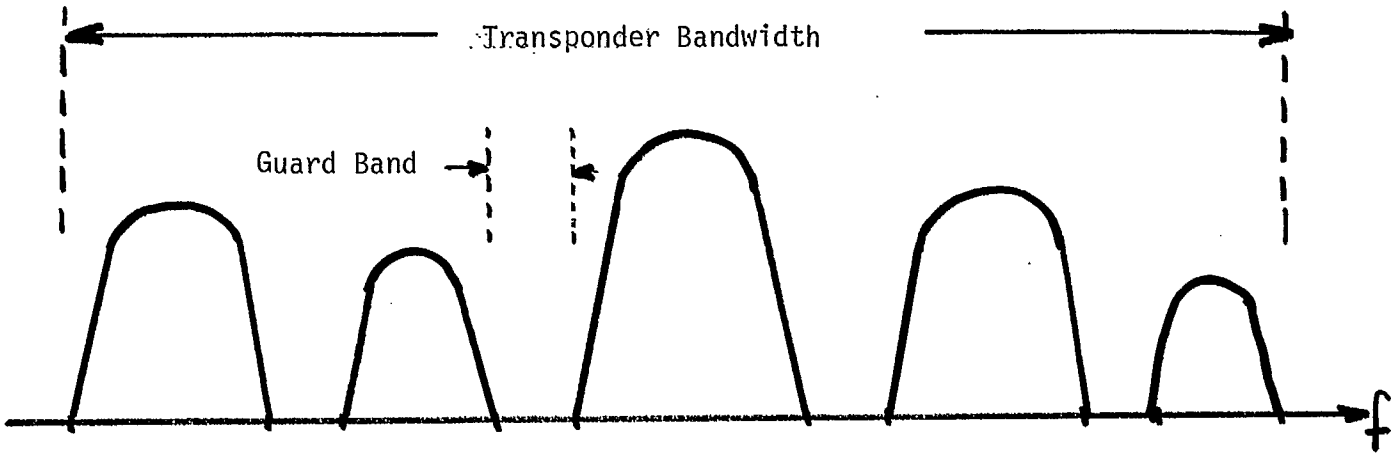


Figure 3.2 Non overlapping FDMA signals in a satellite channel.

signal. For voice as well as for data multi-channel per carrier (MCPC) is also used with FDMA. In such a system, the different baseband signals are first FDM multiplexed at or before they reach the earth station, using some bandwidth efficient modulation scheme (for example Single Side Band-Suppressed Carrier-SSB/SC). The composite signal then frequency modulates a carrier for transmission to the satellite, resulting in a FDM/FM/FDMA scheme. Typical systems have up to 60 voice channels or multiplexed data for each access.

Another practical approach in the use of FDMA consists of providing a separate carrier for each voice or data channel. This Single Channel Per Carrier (SCPC) system has the advantage that it can be used in conjunction with a demand-assignment procedure and hence provides a more efficient system utilization. The modulation can either be analog or digital. One can mention that for voice channels, efficiency of the system can be further improved by voice-activating the SCPC carriers so that carrier power is only turned on during those time intervals where there is actual voice-activity. These systems are presently used in many communication satellite systems, both domestic and international, and both commercial and military.

3.2.2 Single Channel Per Carrier (SCPC)

Single Channel Per Carrier systems are more suitable than FDM/FM/FDMA systems for these applications that require only a few channels per link. They are cost effective for networks consisting of a large number of earth stations, having each a small number of channels.

We recall than in an SCPC system each carrier is modulated by only one voice or data channel. Either analog modulation (e.g. companded frequency modulation, CFM) or digital modulation (e.g. Delta or

PCM modulation with PSK) can be used. For telephony systems, the carrier is usually voice activated causing some 60% of power savings in the transponder since the carrier power is turned off during gaps and pauses in the conversations. For example, SCPC/FM Systems have RF transmission bandwidth varying between 22.5 KHz to 45 KHz per carrier; and SCPC/PCM/PSK systems use 64 Kbits/s in an RF bandwidth of 45 KHz. Consequently a typical 36MHz transponder can accomodate 800 to 1600 simultaneous SCPC voice channels. A typical frequency plan for SCPC system is given in Figure 3.3.

In an SCPC system the assignment of the transponder channels may be either fixed or variable. In the fixed or pre-assigned system, each channel slot of the transponder is dedicated to a particular earth station. In the other case, the channel slots of the transponder are assigned to different earth stations according to their instantaneous needs. This important variation of the FDMA is called Single Channel Per Carrier-Demand Assignment Multiple Access (SCPC-DAMA) (BHAR 81, SPIL 77, PUEN 72, FEHE 83). Most of the system in operations use fixed-assigned SCPC because of its simplicity. The best known SCPC-DAMA is the SPADE system operated by Intelsat (WERT 69, CACC 71).

For small users varying requirements, SPADE provides voice and data services. The system utilizes a demand assignment scheme to allocate an unused frequency pair chosen from a pool of 800 half circuits, in order to establish connectios between two communicating earth stations. The SPADE-DAMA is a digital SCPC system using QPSK modulation and allowing all circuits to be selected by any earth station on demand. A given terminal is never permanently connected to a specific channel and all the channels are paired within the demand assignment pool. The SPADE system is not centrally controlled, and each terminal uses a demand-assignment signaling and switching unit (DASS) for the self-assignment of the channels. A common signaling channel (CSC) is

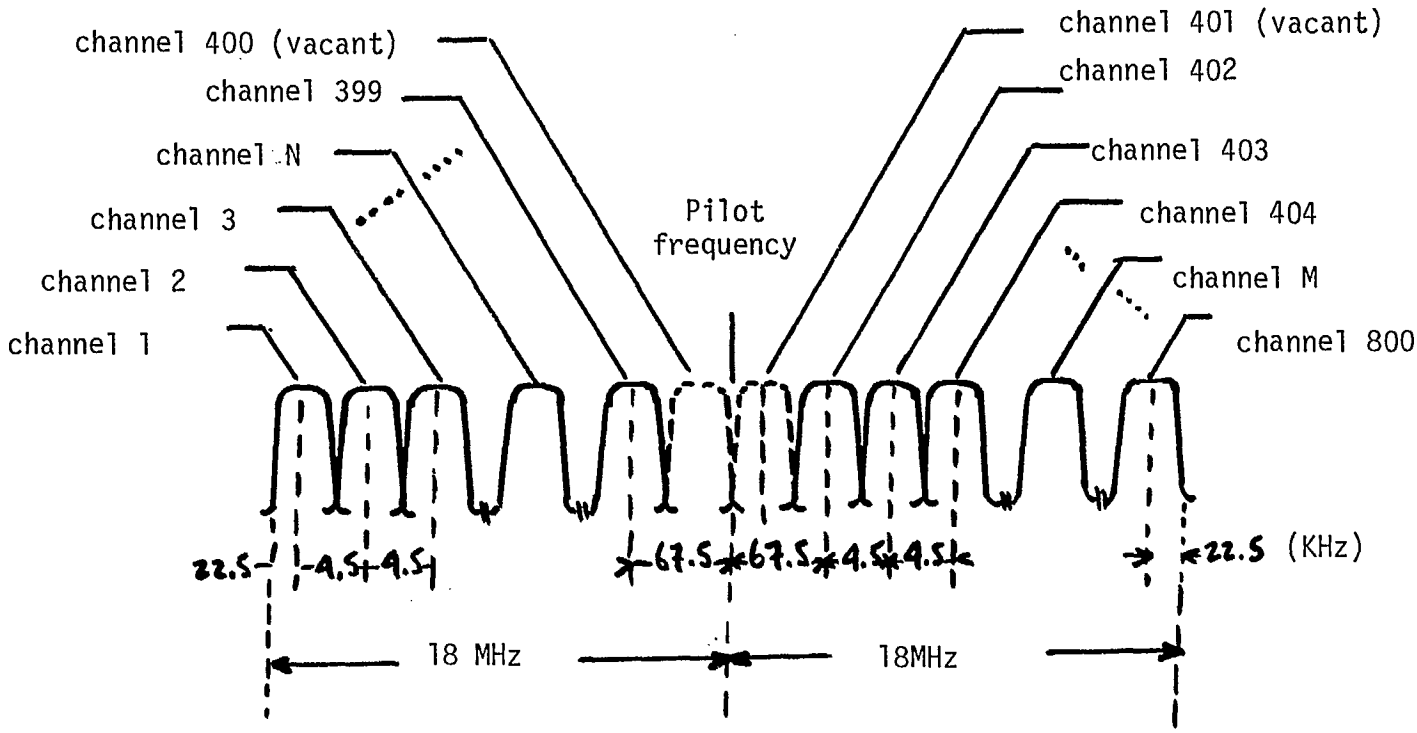


Figure 3.3 Frequency plan for SPADE SCPC with full 36 MHz transponder operation (800 voice channels).

time-shared among all earth stations, and is used to self-assign frequencies, to request channel usage and to update the information on the availability of pool channels.

A connection originating from earth station A to earth station B is established as fellow:

Earth station A first chooses at random (self-assignment) and unused frequency pair from its table of available frequencies in the pool, say f_A. Station A then transmit via the common signaling channel its request for f_A. If the requested channels are available, earth station B sends a message to station A to transmit its message using the frequency pair f_A. The pair of frequencies f_A is thus removed from the pool of available frequencies for all earth stations and is dedicated only to stations A and B until termination of their call.

Thereafter the pair of frequencies f_A is returned to the pool of available frequencies. Unless very few frequencies remain unassigned it is very unlikely that two terminals will simultaneously request the same frequency pairs. On the other hand, because of the long propagation and equipment delay ($\geq$0.25 second), before a requested frequency reaches its destination, that frequency may be assigned to another earth station. Such a situation is readily detected by the originating station which then simply requests another channel.

The SPADE system has proved to be highly successfull in the Intelsat global network, providing connections to over 30 countries. In addition to digital voice using PCM at 64bits/s SPADE also provides

digital services which incorporate forward error control in the form of rate 3/4 or rate 7/8 convolutional encoding with threshold decoding. These data services are:

- Digital data at 48 Kbit/s with rate 3/4 convolutional encoding, resulting in 64 Kbit/s. channel data rate.

- Digital data at 56 Kbits/s with rate 7/8 convolutional encoding, resulting again in 64 Kbit/s. channel data rate.

A list of the voice and data channel unit specifications for an Intelsat SPADE PCM/QPSK/FDMA terminal may be found in reference (FEHE 83).

SCPC systems are receiving wide world attention and are the subject of further developments and applications to provide access to isolated and dispersed users: isolated small comunities, exploration and mining camps in Northern Canada and Alaska, off-shore oil platforms etc... In Canada, new systems are being implemented (SEWE 83) and new equipment and low cost earth stations are being designed for the 14/12 GHz frequency bands (PLEM 83, CORL 83).

Finally, it may be mentionned that the first operational domestic digital SCPC system was the so-called "Thin-Route Network" of Telesat Canada using the ANIK A and ANIK B Satellites. Preassigned channels are used and two hops are necessary to connect the users stations through a central station. Description of that system, and of some of the other systems that followed it are given in reference (BHAR 81) together with an example of an SCPC system design.

3.3 Time Division Multiple Access

3.3.1 Introduction

Time Divison Multiple Access (TDMA) is a technique which allows the sharing of satellite transponder or en entire repeater by several earth stations that transmit in non overlapping bursts. It is a digital multiple access technique that permits individual earth station transmission to be received and repeatered by the satellite in separate non overlapping time slots. The partioning of the time-bandwidth space into distinct time slots avoids the generation of intermodulation products in the non linear transponder. TDMA is hence closely related to Time Division Multiplexing (TMD) that is so widely used in terrestrial communication systems. However here, the interleaving must be achieved at a remote repeater (the satellite) which may be in relative motion to all the users. Therefore each earth station must determine satellite system time and range so that the transmitted signals are timed to arrive at the satellite in their proper time slots. The earth stations are synchronized so that, at a given time only one earth station is transmitting to the satellite, and may thus occupy the entire repeater bandwidth. Figure 3.4 illustrates a TDMA network configuration. Each station transmits in turn a high speed burst of signals, typically PSK or QPSK modulated which arrives at the satellite in its assigned time slot. The bit rates of the transmitted bursts are generally much higher than that of the continuous input bit streams to the ground terminal, thus necessitating elastic buffer storage. Now since only one station signal enters the satellite at a time, there is no intermodulation caused by non linearities and hence, unlike FDMA, TDMA does not suffer from intermodulation products which reduce the useful signal. Furthermore, unlike FDMA it offers total connectivity between all the earth stations of a TDMA network. Compared with FDMA systems, TDMA systems present the following advantages:

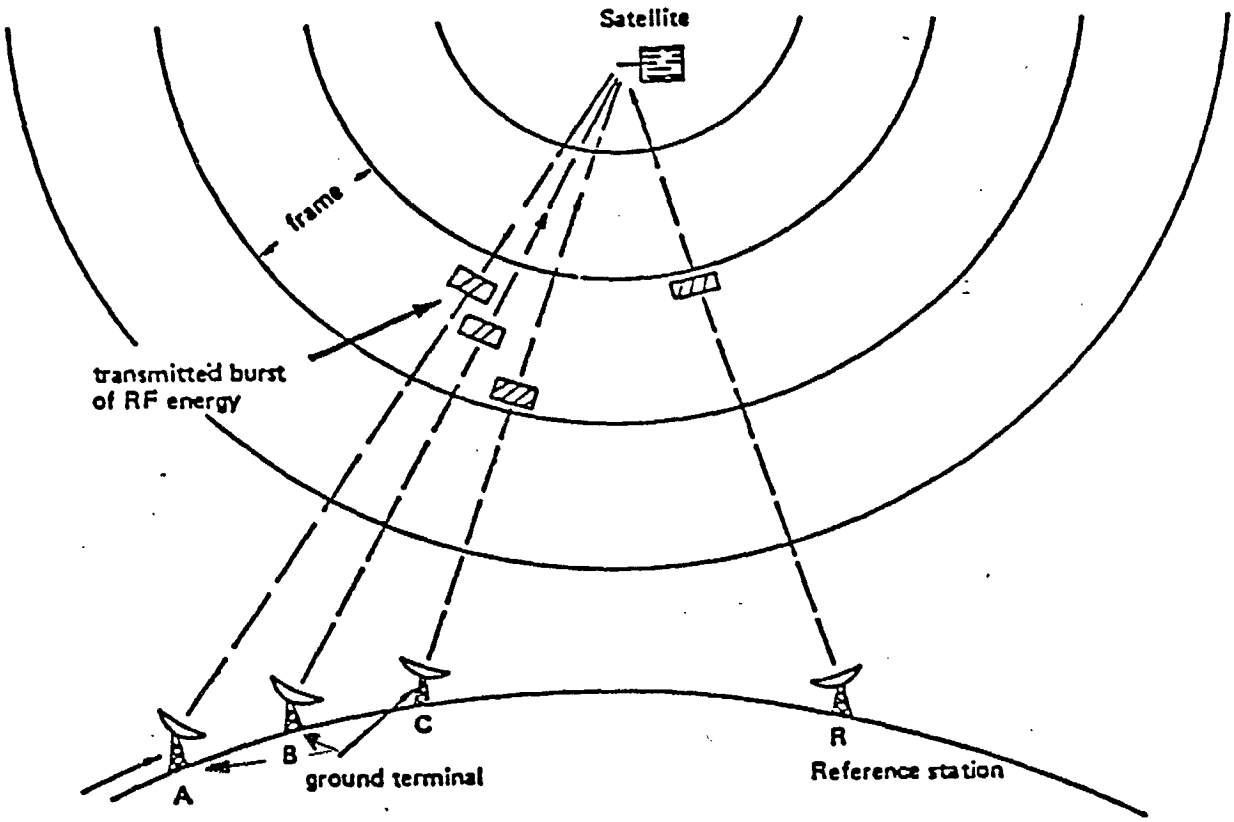


Figure 3.4 TDMA Network configuration.

(1) Because of the absence of intermodulation products, there is no need for backoff, the satellite transponder can be driven nearly to its saturation, yielding more efficient use of satellite power and increased capacity. In fact, this increased capacity is the main reason for developing TDMA. For example for INTELSAT IV repeaters in a network of 10 stations, FM/FDMA has a typical capacity of 450 one-way voice channels, whereas PCM/PSK/TDMA provides 900 voice channels (BHAR 81). For an INTELSAT V 80 MHz transponder 3200 channels can be supported with TDMA (with Digital Speed Interpolations) whereas only 1100 channels can be supported with FDMA.

(2) Establishment and change of traffic is easily accomodated by changing burst length and burst position. This flexibility is of paramount importance to small systems and may result in significant benefit for large systems. Since time-slot assignments are easy to adjust, non uniform accesses do not pose problems in TDMA and hence change and growth are permitted as networks evolve.

(3) Since all traffic is digital, channel coding for error protection is easily integrated in the system configuration, thus providing useful coding gain. This coding gain can be translated in either a reduction of the transmission power or an increase in transmission speed.

3.3.2 System Configuration

A simplified block design of a TDMA system is given in Figure 3.5. Each earth station has parallel input digital bit streams, or analog streams that are digitized. This data arrives at the sending station asynchronously from all the different users and enters a buffer where discrepancies between users clocks and divided version of the

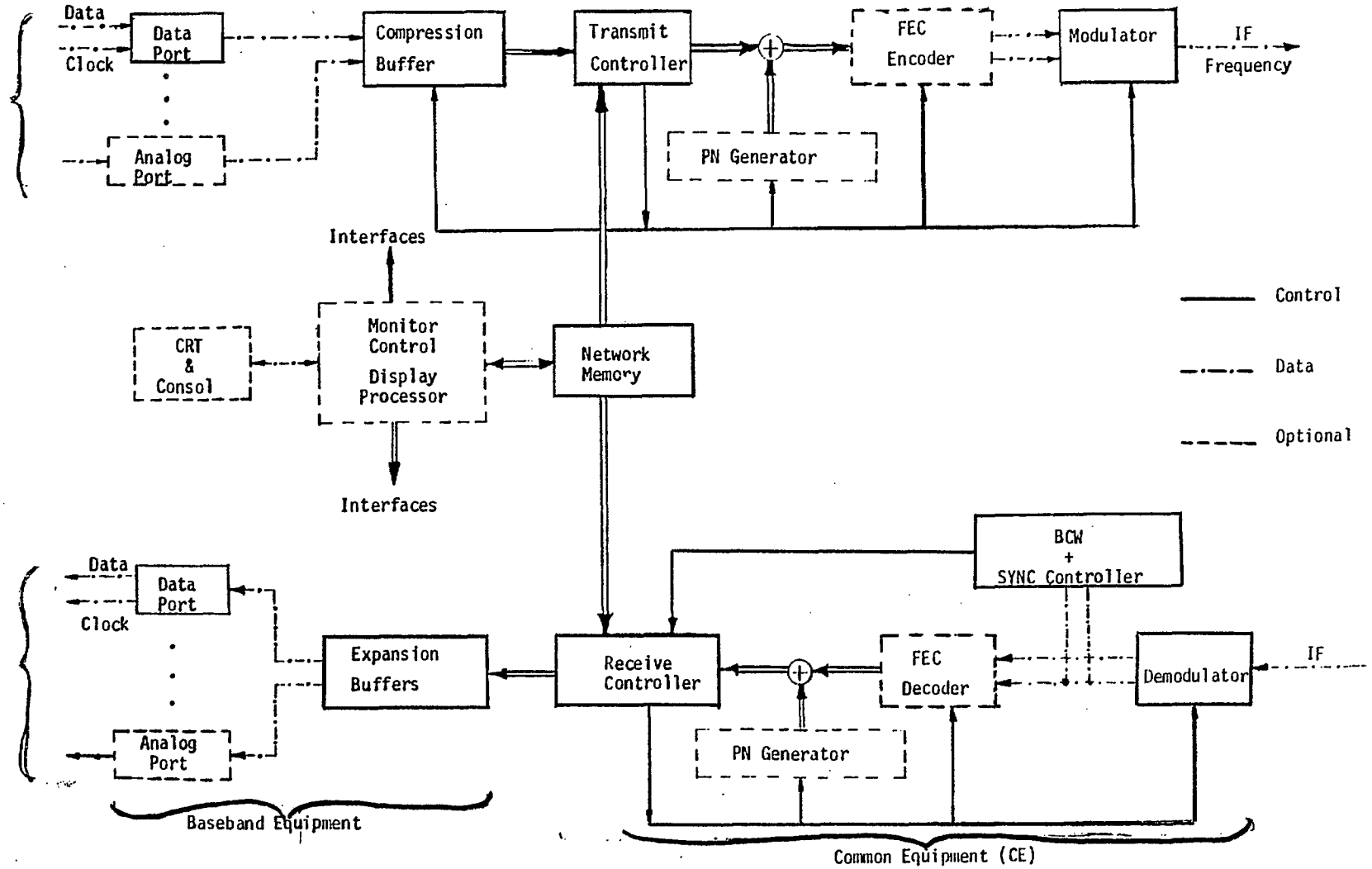


Figure 3.5 Block Diagram of a TDMA system. (From [BHAR 81]) (Dashed boxes are optional).

TDMA clock are resolved. Following this interface a burst compression buffer stores enough data for one sub-burst. The process is repeated for all sub-bursts. The TDMA multiplexer (Transmit Controller) reads on the burst of data, combines it with the preamble to form a multiplexed station burst. The burst so formed may be coded for error protection and then given to the burst modulator which transforms the data stream into a modulated carrier.

At the receiver the same process is repeated in reverse. Demultiplexing is followed by a burst expansion buffer so that sub-bursts are converted in a continous data stream at lower rates. Demodulation includes carrier recovery, symbol timing recovery and detection to recover the message symbols. If FEC encoding was included at the transmitting station, then suitable FEC decoding is performed at the receiving station. Synchronization plays a crucial role in TDMA and will be presented in the next section.

3.3.2.1 Timing hierarchy

Timing hierarchy is an ordered set of time intervals which is essential to TDMA.

The input signal to a transponder carrying TDMA consists of a set of bursts orginating from a number of earth stations that compose the network (See Figure 3.6). This set of bursts is referred to as the TDMA frame. The frame time (often called simply frame) is then the time interval over which the signal format is established and then repeated anew. A frame is subdivided into slots, and a burst time consists of an interger number of slotstimes. A burst is essentially an accessing signal to the transponder and occupies one or more slots in the frame. Because of imperfect timing of bursts small time gaps called

guard times are left between adjacents bursts to make sure that bursts do no overlap with one another.

The first burst in the frame carries no traffic and is used for synchronization and network control purposes. This burst is called the reference burst. Each burst is periodically transmitted with interval of TDMA frame which is usually taken as a multiple of 125 microsecond, corresponding to a 8KHz sampling of voice.

As shown in Figure 3.6 each suburst contains information originating from an earth station and destined for a corresponding earth station. A burst typically consists of a preamble, the message portion itself and sometimes, in coded systems of a postamble. The group of bits forming the preamble are used to synchronize the burst and assist in controlling the network. The reference burst contains only the preamble. In general, the preamble contains the following parts:

- carrier and bit time recovery sequence for coherent demodulation systems.

- Unique Word (UW) codeword which is a special bit pattern used to identify the starting position of the burst in the frame and the position of the bits in the burst.

- House keeping symbols such as order wire carrying telephone and teletype information for inter-stations communications, control, signalling and error-monitoring symbols carrying network management information.

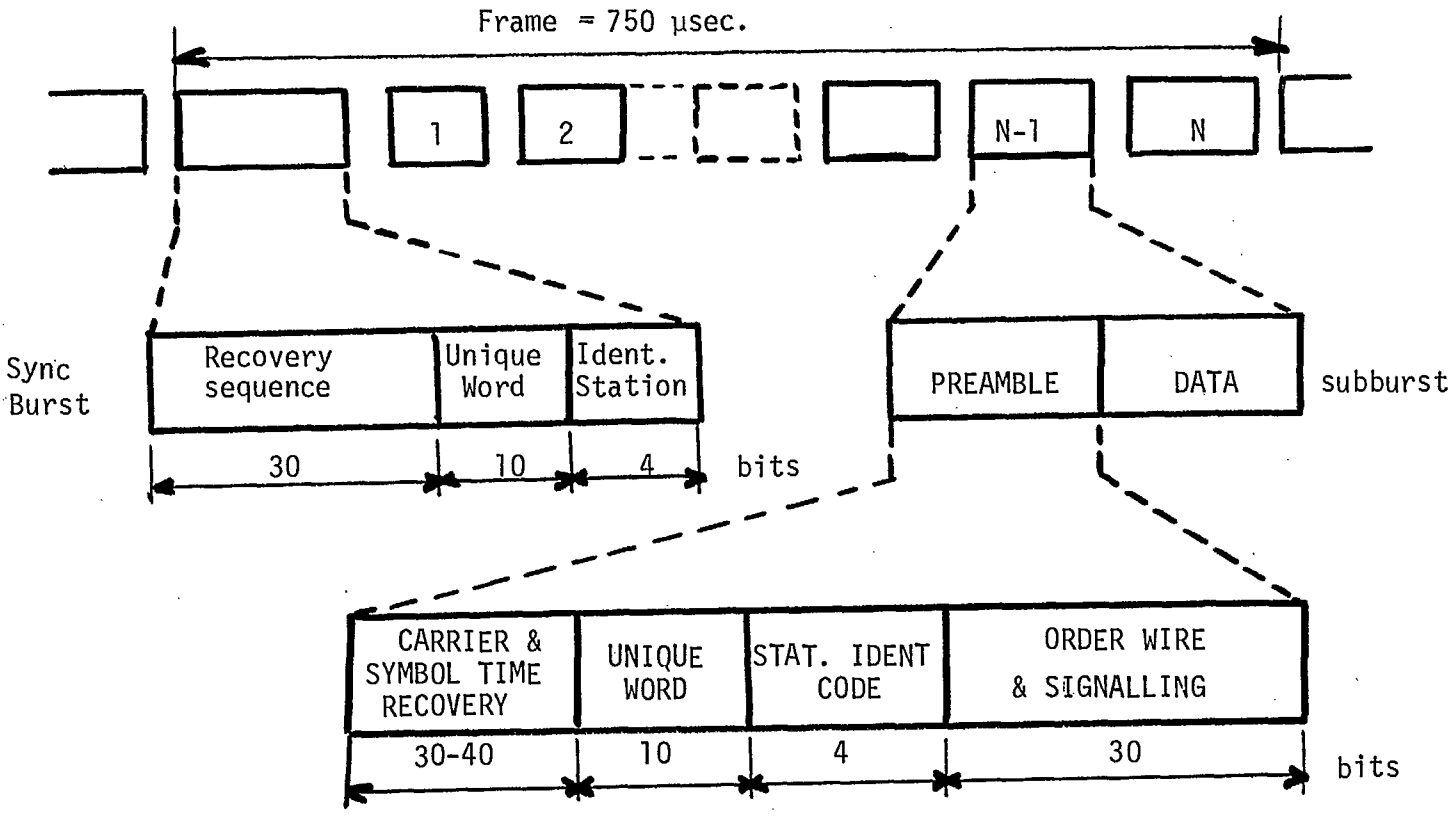


Figure 3.6 Frame Format and Timing Hierarchy of a typical TDMA.

Following the preamble, the message portion of a burst contains the desired time-multiplexed data destined to all corresponding earth stations. This message may be encoded for purpose of error protection. When channel coding is employed then a postamble is used for decoder initialization and quenching for the next burst.

Since a TDMA frame contains an overhead part in addition to a useful message part, then the efficiency of the system must be considered. There an several types of eficiencies (BHAR 81):

- Frame efficiency: Ration of the portion of the frame available for the messages to the total frame length. This definition takes into account all guard times.

- Transmission efficiency of a burst: It is the ratio of the information bits to the total bits transmitted in a burst. Clearly here the only overhead are the preamble and postamble. The transmission efficiency of a frame extends the definition to the entire frame.

- System efficiency: It is defined as the ratio of the paying traffic capacity (in bits/s) to the total available capacity in bits/s. The system efficiency depends on several system parameters such as modulation, coding, networking etc...

3.3.3 Synchronization in TDMA systems

Several differents synchronization problems are involved in operating a TDMA system. Basically there are two types of synchronization that must be performed:

(1) Burst synchronization so that each burst maintain it predetermined timing, referenced to the position of the reference burst and;

(2) bit synchronization within the recovery sequence in order to demodulate the PSK carriers.

The synchronization problems arise in part by signifiant time-delay variations due to pertubations and imperfections of geotationary orbits. Finaly the problem is further compounded by the introduction of multiple-beam satellites where the transmitting earth stations may not receive their own bursts. Synchronization techniques are described at length in (BHAR 81, FEHE 83) Examples of practical systems and monitoning equipment abound for examples in (MOTT 83, PONT 83, PSAR 83)

In this section we only introduce the two principal network acquisition and synchronization methods. These methods can be classified as Open Loop Methods, and Closed-Loop Methods. Other methods such as sync window methods for Satellite-Switched TDMA and some hybrid schemes are described in (BHAR 81) and in (FEHE 83).

Open-Loop Methods: The name of the method is derived from the fact that a station's transmitted burst is not received by that station, hence the synchronization loop is open. In such a method the transmit timing is determined by knowing the range from each station to the satellite. It therefore requires accurate and precise knowledge of earth stations locations and satellite position. Clearly such a method is suitable for multibeam satellite since self visibility is not required from the earth station bursts.

A first approach to this type of synchronization consists of using orbit parameters of the satellite and free-running clocks to attain approximate synchronization bursts. This approach is called coarse-sync, and although not an optimum choice for efficient TDMA systems, it may become attractive for simple and economical systems as well as for nonsynchronous satellites. (DINN 75, HUST 78). The technique of coarse-sync has been improved through improvements of the clock itself and through time transfers between the stations of the network.

A second approach to open-loop synchronization makes use of a reference burst transmitted from a reference station. This reference burst is a special preamble only and is used to mark the start of frame. The reference bursts are received by all the stations of the network, and all transmissions are locked to the time base of the reference station.

Closed-Loop Methods: These methods which are also called loop-back (FEHE 83) provide high precision and accuracy by returning the transmitted signals, through the repeater back to the transmitting station which then determines its transmit timing error. This type of control is thus based on the ability of any station to observe both the reference and its own retransmitted burst and hence to adjust it own timing burst accordingly. This synchronization method is suitable for global and regional beams but, clearly cannot be applied to multi-beam systems since a direct loop-back is not possible in such systems. The closed-loop method was first used in TELESAT TDMA (KWAN 75) and is operational in the INTELSAT TDMA (INTE 83). For example a TDMA/DSI System with INTELSAT V and following satellites. The TDMA system makes use of reference stations to achieve asquisition and burst synchronization. Each reference station is equiped with sufficient redundancy to

provide the required high reliability, and furthermore each reference station will also have a secondary reference station.

TELESAT Canada was the first to use commercial TDMA in 1976 (KWAN 75). The burst transmission rate is 61.248 Mbits/s and the synchronization used the closed-loop method. Since then TDMA has been used or is being planned for use in a number of systems such as INTELSAT V and INTELSAT VI Satellite systems, Satellite Business Systems network, TELECOM I system from France etc... The burst rates vary from 25 Mbs for the french TELECOM I to 120 Mbs for INTELSAT. As higher frequency bands open up for satellite usage (20/30GHz and higher), as a multiple access method TDMA will gain further importance.

3.3.4 Light Route TDMA

In many applications there is no need for a full transponder capacity. Satellite networks requiring only a fraction of a transponder capacity are called "light route TDMA" (LRTDMA). LRTDMA is a very efficient and flexible technique for the transmission of a mixture of voice and data services to users having relatively modest traffic requirements, using one integrated satellite communications facility. LRTDMA operate at data rates below 20 tO 30 Mbps. The advantage of LRTDMA lies in its possibility to reallocate capacity among the users and provide easily point to multipoint connectivity. Furthermore a demand assignment procedure can be applied to LRTDMA, thence adding flexibility to the mixture of different traffic, and making that type of systems eventually compatible with integrated-service digital networks ISDN of the future.

3.3.4.1 LRTDMA Concept and Advantages

LRTDMA may be viewed as a Time and Frequency Division Multiple Access technique in which several earth stations operating in a narrow-band TDMA mode share the transponder bandwidth and capacity using the FDMA mode. Therefore, unlike pure TDMA which operates at a high data rate (60 or 120 Mbps) and occupy the full bandwidth of a transponder LRTDMA or TDMA/FDMA uses considerably lower bit rates, which, depending on the application being served may vary from 256 Kbps to 10 or 20 Mbps.

Because of the partial utilization of the Satellite channel bandwidth by any one of the TDMA network, this type of multiple access method requires that the transponder be backed-off from saturation. Consequently the efficiency in use of the satellite transponder should be lower than with pure TDMA. However the advantage of the technique resides in the flexibility it provides for handling changes in traffic pattern.

From the user point of view LRTDMA offers many advantages, such as [SMAL 83]:

- Full network connectivity using software control.
- Easy addition of earth station to network.
- Rapid variation of the mix and quantity of digital services between network modes.
- Multiplexing of a number of different data rates, whether or not standard, along with voice and teleconferencing services.
- Broadcasting ability.
- Possible use of small earth stations.
- Low bit error rate by using Forward Error Correction.
- Further flexibility by using Demand Assignment.

LRTDMA ia a relatively new concept, but because of the recent increase in the requirements for communications services by small and medium capacity users, and because of the additional traffic generated by new services (electronic mail, communicating word processors, slow scan video etc...), LRTDMA has generated a lot of activity [KAMA 83]. In Canada, Telesat and several canadian telephone companies have carried out LRTDMA field trials in mid-1982 [HANS 83]. The trials were conducted over the 6/4 GHz ANIK D Satellite and using the following traffic.

- Full Duplex 1.544 Mbps for data or digitally-encoded Video between any two stations
- One 64Kbps Delta modulated audio for conference system
- Three 64Kbps full-duplex synchronous data
- Three full duplex 32Kbps delta modulated voice signals

All of this traffic resulted in a 4.032 Mbps information bit rate for the LRTDMA. Furthermore with the addition of the rate 3/4 convolutional coding, the transmitted data rate was increased to 5.376 Mbps. Further details on this systems may be found in [Hans 83].

To summarize LRTDMA can provide flexibility, relability and security at reasonable cost for the transmission of a mixture of voice, data and new services through one integrate satellite communications facility for users having small to medium traffic requirements.

3.4 Code division multiple-access

The classical problem of multiple-access is to allow multiple users to access simultaneously the satellite channel without causing too large a performance degradation for any individual user. From a system point of view, the access scheme should also be efficient in the

sense that the useful capacity of the link itself should not be severely reduced by the multiple-acces method. In the previous two sections we have presented the two most common techniques of FDMA and TDMA which atempt to solve the problem by separating the signals in the frequency domain and in the time domain respectively. Although widely used each of these techniques has certain drawbacks and limitations associated with it. These inherent difficulties are further compounded if the system were to operate in a hostile environment, for example if interference such as intentional jamming or multipath were presents; For both FDMA and TDMA systems large degradations in system performance could then result.

For these reasons, in certains applications Code Division Multiple-Access (CDMA) may be a very attractive and competitive multiple access scheme.

IN CDMA the transmission is digital, and both the time and frequency are dynamically shared by all the users; the users operate at the same nominal frequency and simultaneously use the entire repeater bandwidth. Unlike either FDMA or TDMA, CDMA usually operates in an asynchronous manner so that problems of network timing do not exist, and hence minimal coordination in either time or frequency is needed between the various transmitters in the system. CDMA techniques can provide antijam capacity to reject intentional interference, and furthermore have a low probability of intercept, that is, provide a reduction of the probability of reception by unauthorized users. Finally, from the system point of view, as the number of simultaneous users increases the performance degradation is only gradual,and when the number of users decreases and the system becomes underused, the resulting excess capacity becomes increased margin. In practice it is the capacity of rejecting external interference which is the crucial factor in deciding to utilize CDMA [BHAR 81].

The two principal CDMA techniques are:

1- Direct Sequence (DS) or Pseudo-Noise (PN) modulation

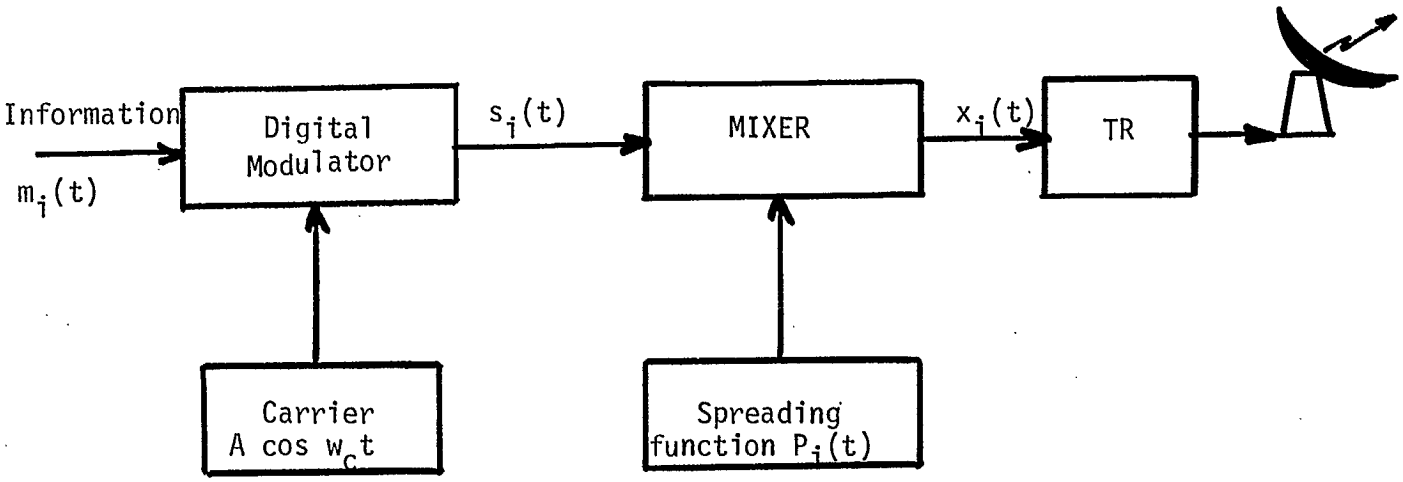
In these systems the waveform used to represent a 1 or a 0 in the PN sequence is called a chip, and its time duration is called chip time.

2- Non Coherent Frequency Hopping (FH) modulation

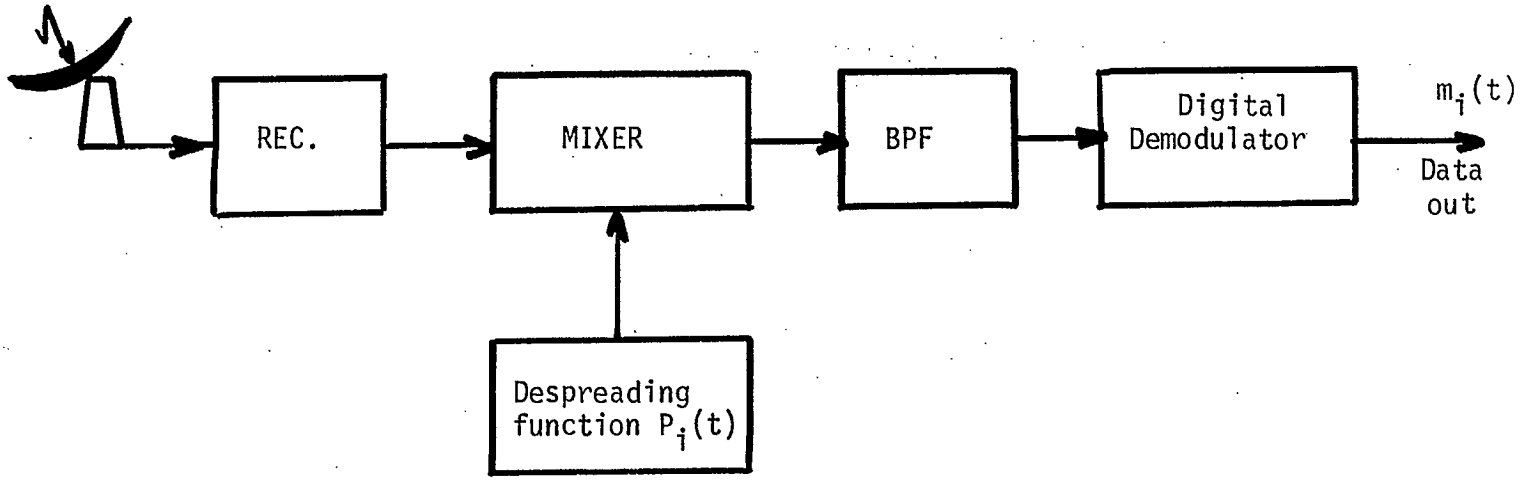
In both of these techniques, the basic idea is to spread the spectrum of the modulated signal so that the modulation bandwidth is much larger (typically 10^3 to 10^6) than the information bandwidth. Hence the basic concept of CDMA is that of spread spectrum modulation technique, an essential ingredient of which is the PN sequence or PN code used to spread the bandwidth. After introducing the fundamentals of spread spectrum, we will briefly present the techniques of Direct Sequence and Frequency Hopping. A more complete presentation of the subject of spread sprectrum and its application to CDMA is beyond the scope of this report. However, a very rich and extensive literature exists on these subjects, and the reader may consult for example [BHAR 81], [DIXO 75], [DIXO 76], [HOLM 82], [VANT 79], [COOK 83].

3.4.1 Basic Concepts of Spread Spectrum

We will assume that the information to be transmitted is digital. A basic spread-spectrum system is shown in Fig. 3.7. A digital source of rate R_s bits/s, $R_s = 1/T_s$ delivers a signal $m_i(t)$ which enter a digital modulator, typically a PSK modulator, where Ts is the symbol duration. The carrier frequency is ω_c and occupies the same RF



a) Transmitter



b) Receiver

Figure 3.7 Basic Block Diagrams of a Spread Spectrum System: a) Transmitter; b) Receiver.

bandwidth. The output of the PSK modulator, $s_i(t)=A_i \text{Cos}\,[\omega_c t + \phi_i(t)]$, is multiplied by a spreading function $p_i(t)$ with chip rate $R_c = 1/T_c$, resulting in the waveform $x_i(t) = p_i(t)\, s_i(t)$. The spreading function $p_i(t)$ is independent of the data. The function $x_i(t)$ is combined with the other signals $x_j(t)$, $j = 1,2,\ldots$, each using a different spreading function, $p_j(t)$, and transmitted in the channel. Assuming M signals to be transmitted, then the transmitted signal is:

$$X(t) = \sum_{i=1}^{M} x_i(t) = \sum_{i=1}^{M} p_i(t)\, s_i(t)$$

and the received signal is:

$$r(t) = \sum_{i=1}^{M} x_i(t) + I(t) + n(t)$$

Where I(t) is an interfering signal, whether or not intentional, and where n(t) is the additive noise.

At the receiver the intended $i^{\underline{th}}$ user will use as the despreading function, the same function $p_i(t)$ as the one used to spread $s_i(t)$ at the transmitter. Following the despreading multiplier, a conventional PSK demodulation will provide the original baseband digital signal $m_i(t)$. It should be clear that other waveforms will not be despread and thus will only appear as noise to the demodulator if the set of spreading functions $p_i(t)$ are properly chosen, that is with a very low cross correlation. Figure 3.8 shows the various sprectra in a spread-spectrum system.

In particular it shows that unless both spreading and despreading functions are identical, at the receiver the operation of

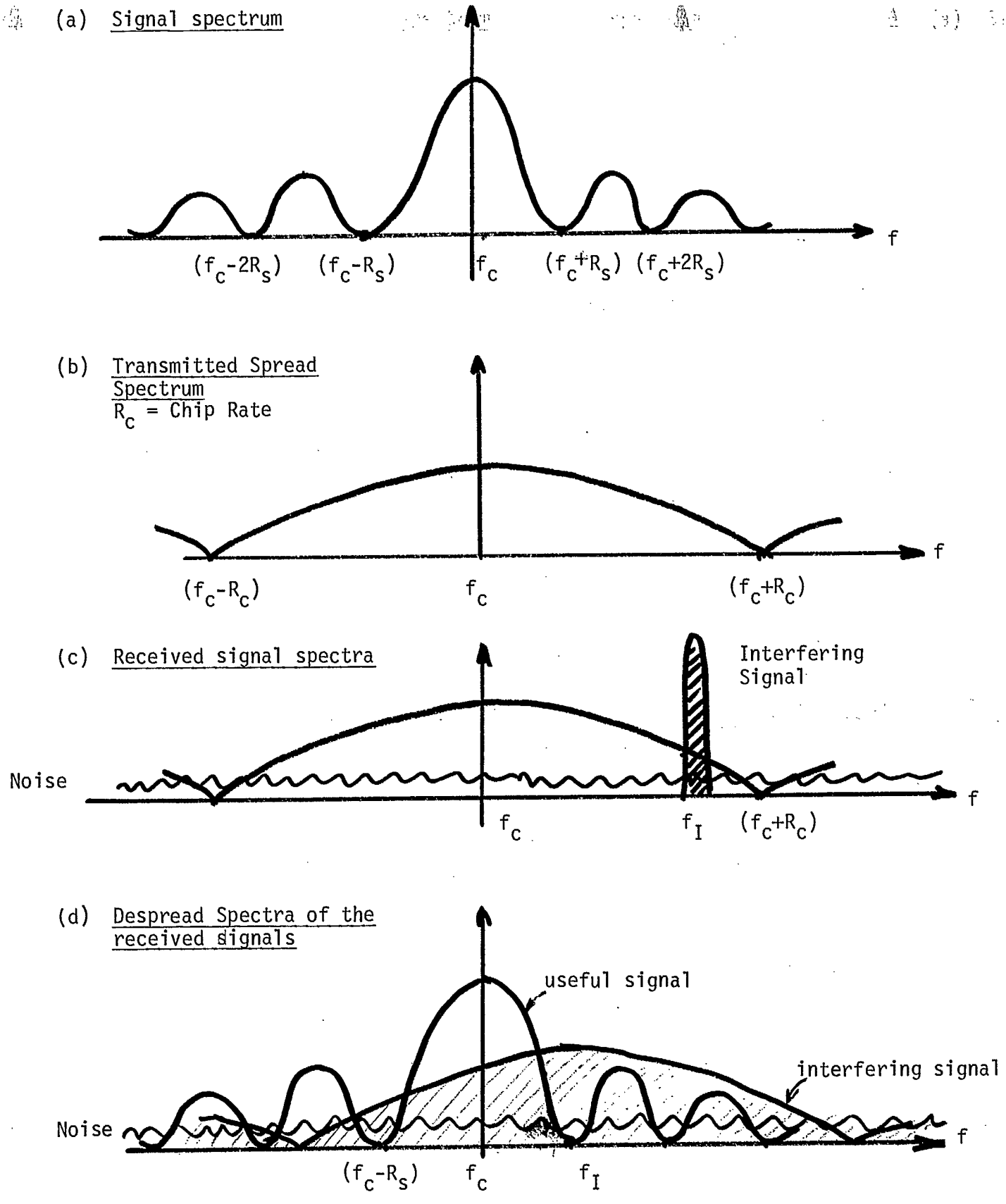


Figure 3.8 Illustration of the spreading and despreading effects on useful signal and interference.

multiplying the received signal by $p_i(t)$ always results in a spreading. Hence an interfering signal will be spread at the receiver, and therefore to be effective, a jamming or interferring signal must increase its power by the same amount as the bandwidth expansion, i.e. the spreading, of the useful signal.

For CDMA systems the amount of interference rejection is especially important. A measure of this rejection is called the Processing Gain of the system, G_p, and is essentially equal to the total bandwidth expansion, that is the ratio of the RF bandwidth to the information bit rate.

$$G_p = \frac{BW_{RF}}{R_s}$$

From Fig 3.8, $G_p = 2R_c/R_i$, and typical processing gains for spread sprectrum systems vary from 20 to 60 dB. The processing gain is not unlike a Figure of Merit of conventional systems, and the input and output signals to noise ratios are related by:

$$(S/N)_{out} = G_p\ (S/N)_{in}$$

Because a system has always some internal system losses, a processor cannot perform on an interfering signal by an amount equal to the processing gain. The Jamming Margin M_j of a system is a measure by which a spread-sprectrum system can withstand an interfering signal while delivering some minimum signal to noise ratio at its output: It is related to the processing gain by the relation

$$M_j = G_p - L_{sys} - \left(\frac{S}{N}\right)_{out} \text{(dB)}$$

where L_{sys} is the internal system implementation loss. For example, a system with a processing gain of 33 dB, a minimum $(S/N)_{out}$ of 10 dB and a system implementation loss of 2 dB would have a jamming margin of 33-2-10 = 21 dB. Such a system could not be expected to operate satisfactionaly with an interference larger than 21 dB above the desired signal. Equivalently this result also indicates that an interfering transmitter can have $10^{2.1}$ = 126 times more power than the desired signal's transmitter before it can affect adversely the operations of the receiver.

We have mentionned earlier that the two most widely used spread-spectrum techniques are direct sequence (DS) and frequency hopping (FH). There exist also the two techniques of time-hopping (TH) and pulse-FM or "chirp" method, and some hybrid schemes. These spread-sprectrum systems are related to one another and are differentiated only by their modulation formats.

Spread-spectrum advantages and disavantages

The effect of spreading and despreading the signals present, in addition to code division multiplexing the following advantages:

1- Selective Addressing:

This is possible through the use of the modulating code sequence to recognize a particular signal among many others. Hence, by assignining a particular code to each receiver in a network, a transmitter can select any given receiver for communication by transmitting the particular code of that receiver. The codes being all distinct, then only the intended receiver will receive the message.

2- Message privacy:

This advantage is inherent in spread-spectrum signals. Naturally the degree of privacy is dependent upon the codes used for spreading the signals and can vary from the relatively simple to the truly secure and encryption types.

3- Low density output signals:

Low density transmitted signals are used in preventing them from interfering with other systems, as well as in providing a low probability of intercept. Of course this low density signals are a direct consequence of the bandwidth expansion of the spreaded signal. Hence a given transmitted power is spreaded over a large bandwidth, resulting in a very low power density. An unintended receiver with a relatively narrow bandwidth will thus receive only a very small amount of intercepted RF power. For example a digital signal spreaded at rate of say 20 Mbps will occupy a larger RF bandwidth than 24 MHz. Suppose the transmitted power is 10 Watts. The RF power spectrum density is then 0.416 μW/Hz, and a 50 KH/z bandwidth receiver will collect only 208 mWatts of that power, that is, will hardly appear as a hostile interference.

4- High resolution Ranging:

High resolution range measurements can be obtained by Direct Sequence types of spread spectrum signals due to the high speed codes used. This property derives directly from the synchronization requirements which in practical systems may reach a fraction of one bit. Hence the inherent resolution capability is the range which corresponds to that fraction of one bit period. For example for a 20 Mbps code, a

bit period is 200 ns. Considering a resolution of 0.1 bit then the range between receiver and transmitter can be measured to within 20ns., that is within 6 meters.

5- Interference Rejection

As mentionned earlier spread spectrum systems provides an interference rejection capability, which is measured by the jamming margin of the receiver. This jamming margin is a function of the code sequence rate (for DS systems) and the number of available frequencies in a FH systems.

Spread spectrum systems present, of course, some desadvantages which are essentially:

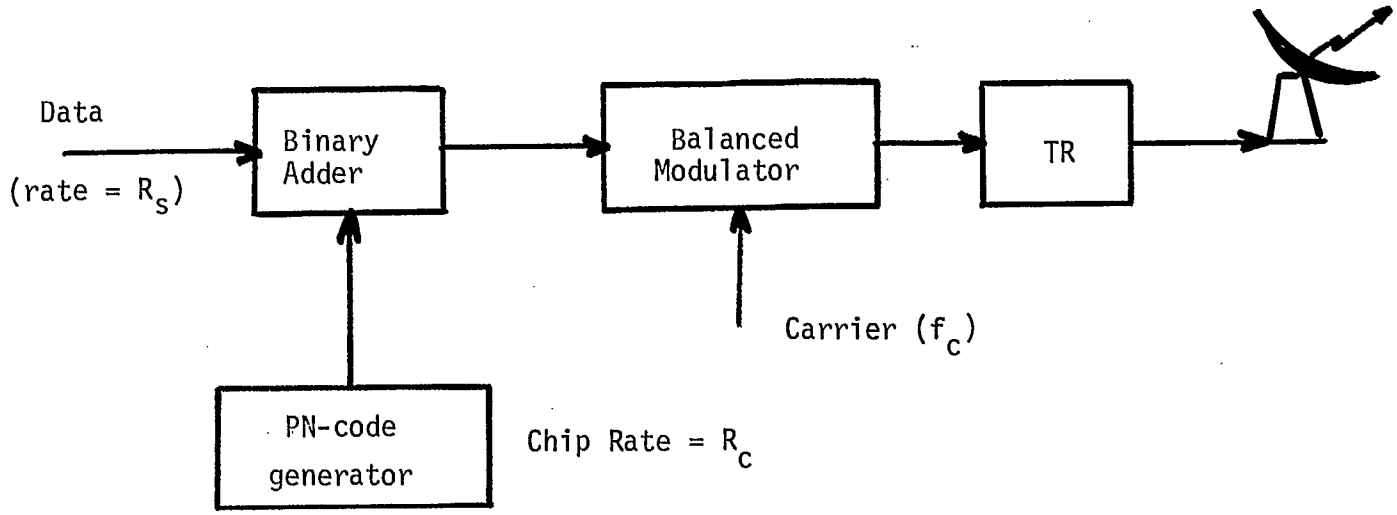
1- Large bandwidth requirements and ensuing difficult frequency allocation.

2- Greater system complexity.

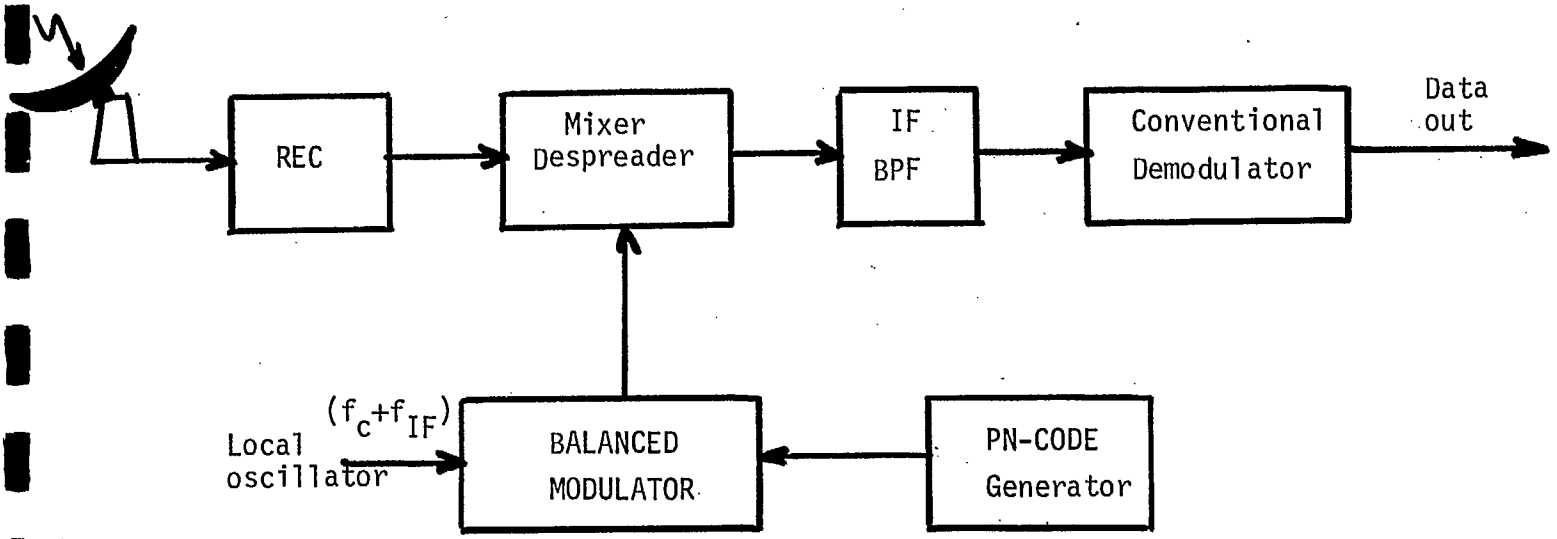
Having introduced some basic concepts and important properties of spread spectrum systems, we now present the two principal techniques presently used in CDMA: Direct Sequence and Frequency Hopping systems.

3.4.2 Direct Sequence Systems

A block diagram of a simplified direct sequence system is shown in Fig. 3.9. The RF carrier is modulated by the sum of the digitized information and the spreading sequence provided by the pseudo noise or PN code generator. The PN code generator generates a pseudo-



(a) Transmitter



(b) Receiver

Figure 3.9 Block Diagram of a DS system: (a) Transmitter; (b) Receiver.

random sequence or code stream at a chip rate R_c, where R_c is much larger than the information rate R_s. The combined sequence phase modulates the carrier using either biphase or quadriphase PSK. Since $R_c \gg R_s$, then the spreading, or equivalently the bandwidth expansion and processing gain are under the control of R_c and the PN sequence.

The DS receiver operates in reverse of the transmitter. The received wideband signal is multiplied by a locally generated exact replica of the PN sequence or code in order to perform the despreading of the intended signal and the spreading of any interfering signal. The despreading has the effect of compressing the wideband signal into a bandwidth which corresponds to the digitized information stream alone. The callapsed signal passes through the narrow band IF band pass filter and is demodulated by the PSK (or QPSK) demodulator as if the PN sequence had never existed on the carrier. Provided the synchronization of the PN generators can be maintained the system is transparent to its information channel.

One of the main problems in DS systems as well as in other spread spectrum systems, is to synchronize the code sequences in the transmitters and receivers and to keep them synchronized [DIXO 76], [HOLM 82], [COOK 83]. A treatement of pseudo-random sequences is beyond the scope of this report and will not be presented. A discussion of these sequences and their correlation properties will be found for example in [BHAR 81], [HOLM 82] and [DIXO 76].

3.4.3 Frequency Hopping Systems

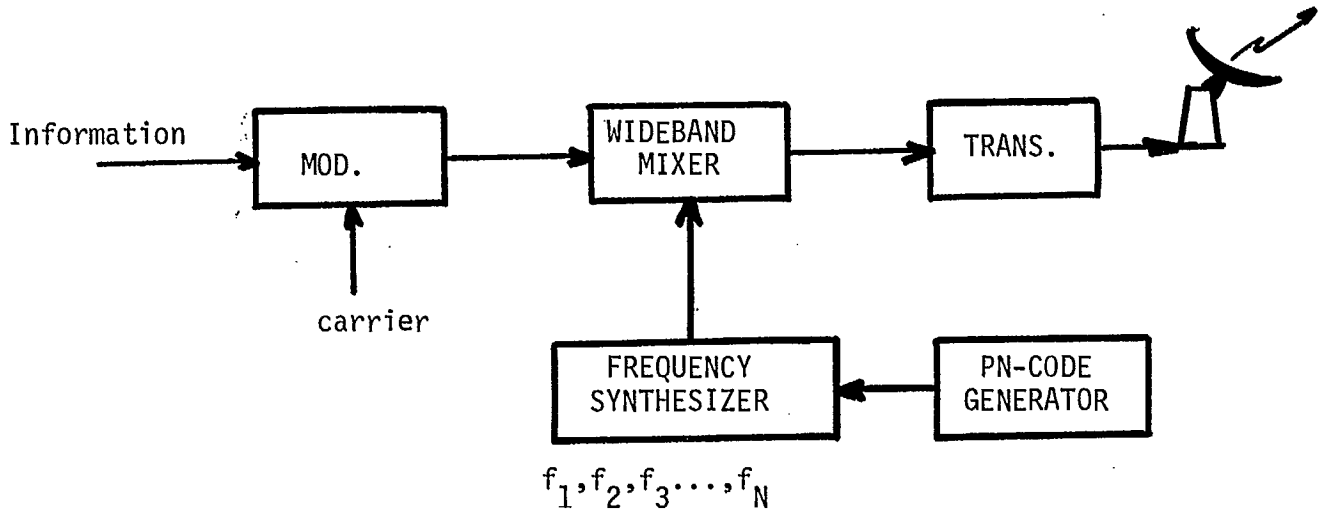
Frequency hopping systems are quite similar to DS systems, but instead of using one carrier it uses a carrier randomly-hopped by a PN sequence. It is therefore an extension of the basic idea of spreading the information signal bandwidth, but here, an unintended receiver is prevented from receiving a message (or an unwanted signal is prevented from interfering with a desired signal) by moving around in the frequency domain. Of course, this moving around is done under the control of a PN sequence in such a way that undesirable receivers cannot find the operating frequencies.

As shown in Fig. 3.10 the PN code generator hops the different frequencies of a frequency synthesizer, producing a spreading by pseudo-randomly hopping the final carrier frequency over a wide range of available and prescribed frequencies. The code sequences are of the same types as those used in DS systems except that their clock rate is usually smaller, usually not exceeding a few hundred kilobits per second.

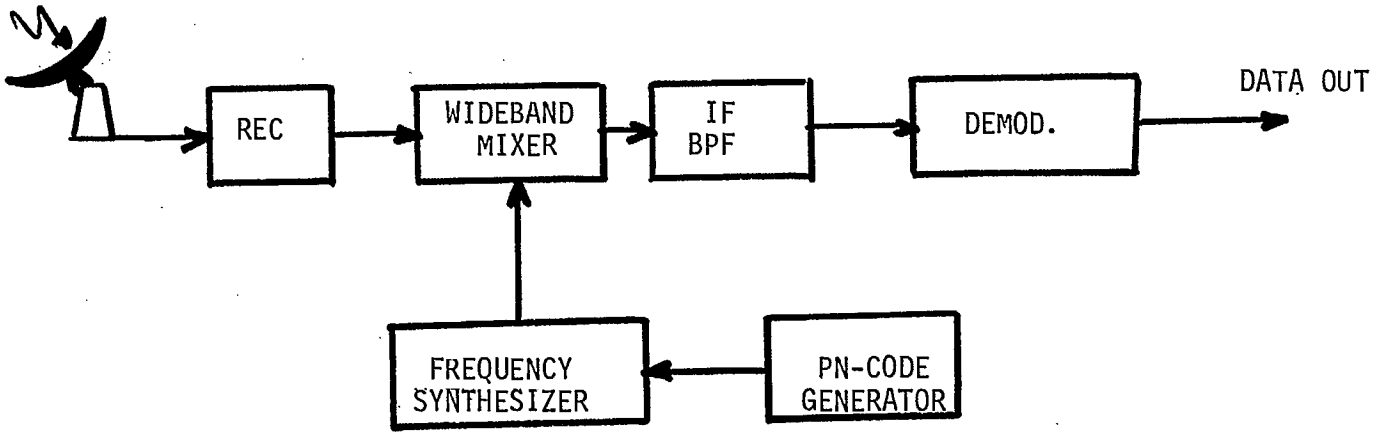
In order to properly demodulate each encoded chip or message bit, the rate of frequency hopping is related to the information rate R_s. In fact, because of the difficulty in maintaining phase coherence between sucessive frequency hops, coherent demodulation is not used for FH systems; incoherent processing is used instead.

If the frequency separation between discrete frequencies is equal to the message bandwidth, then the processing gain of a FH system is:

$$G_p = \frac{BW_{RF}}{BW_{mess}} = N$$



(a) Transmitter



(b) Receiver

Figure 3.10 Block Diagrams of a FH System: (a) Transmitter; (b) Receiver.

Where N is the number of available frequency choices or channels.

Although using the same basic principles DS and FH systems present some differences which are important for their implementations.

The chip rates R_c are slower in FH system then in DS systems. As a consequence the initial acquisition times is also smaller in FH systems. Typical initial acquisition times are of the order of the millisecond for FH systems and 2 or 3 seconds for DS systems. In order to achieve a large processing gain a DS system must have a large chip rate R_c, whereas a FH system must have a large number of available channels. Furthermore, in order to be protected by partial band jamming, FH system usually incorporates some form of foward error correction, using either block or convolutional coding.

Finally a hybrid FH/DS may sometimes offer a simpler implementation in attaining a given processing gain since the overall processing gain of the hybrid scheme is equal to the sum of each processing gain of the DS and FH systems considered alone.

REFERENCES

[BHAR 81] BHARGAVA, V., HACCOUN, D., MATYAS, R., NUSLP, P., "Digital Communication by Satellite, Modulation, Multiple access and Coding", John Wiley, 1981.

[CACC 71] CACCIAMANI, E.R., "The SPADE System as Applied to Data Communications and Small Earth Station Operation", COMSAT Tech. Review, Vol 1., 1971. ICDSC-1, London 1969.

[COOK 83] COOK, C., ELLERSICK, F., MILSTEIN, L., SCHILLING, D., "Spread Spectrum Communications", IEEE Press, N.Y., 1983.

[CORL 83] CORLESS, W.S., "Design Highlights of a low Cost Earth Station", Proc. 1sr Satellite Communication Conference, Ottawa, June 1983, pp. 14.4.1-14.4.4.

[DINW 75] DINWIDDY, S.E., "A simple TDMA system for Satellite Data Communications Networks", Proc. ICDSC-3, Kyoto 1975, pp. 328-384.

[DIXO 75] DIXON, R.C., "Spread Spectrum Systems", John Wiley 1975, N.Y.

[DIXO 76] DIXON, R.C., "Spread Spectrum Techniques", IEEE Press, N.Y. 1976.

[FEHE 83] FEHER, K., "Digital communications, Satellite/earth station engineering", Prentice Hall, 1983.

[HANS 83] B. HANSON, A. SMALLEY, M. ZULIANI, "Implementaion of a Light-Route TDMA Communications Satellite System for Advanced Business Networks".

[HOLM 82] HOLMES, J.K., "Coherent Spread Spectrum Systems", John Wiley, N.Y., 1982.

[HUST 78] HUSTED, J., and DINWIDDY, S., "Low Cost Satellite Data Transmission Networks Using Demand Assigned TDMA", Proc. ICDSC-4, Montreal 1978, pp. 8-15.

[INTE 83] INTELSAT, TDMA/DSI System specifications-Document BG-42-65E-B/6/80.

[KAMA 83] S.S. KAMAL, R. MATYAS, R.G. LYONS. "Demand assignment for Light-Route TDMA", Intern Journal of Satellite Communications, Vol. 1 No.1, July 1983, pp. 51-56.

[KWAN 75] KWAN, R., "The TELESAT TDMA Field Trial", Proc. ICDSC-3, Kyoto 1975, pp. 135-143.

[MOTT 83] MOTT, R., ASSAL, F., KROLL, R., "Guard Space and burst duration monitor for TDMA", Proc. 1sr Satellite Communications Conference, Ottawa, June 1983, pp. 8.5.1-8.5.5.

[PLEM 83] PLEMEL, R.A., ROTHERY, R., JAMES, L., "Design of a 14/12 GHz SCPC Earth Station Network for Nation-Wide Remote Telephony", Proc. 1sr Satellite Communications Conference, Ottawa, June 1983, pp. 14.2.1-14.2.4.

[PONT 83] PONTANO, B., COLBY, R., "The INTELSAT TDMA control and monitor system", Proc. 1sr Satellite Communications Conference, Ottawa, June 1983, pp. 12.8.1-12.8.4.

[PSAR 83] PSARRAS, E.T., KRAWCZYK, L., AYLES, D., WONG, P., "Network Management and Control System of the TCTS Satellite TDMA Trial", Proc. 1sr Satellite Communications Conference, Ottawa, June 1983, pp.19.5.1-19.5.4.

[PUEN 72] PUENTE, J.G., SCHMIDT, W.G., and WERTH A.M., "Multiple Access Techniques for Commercial Satellites", Proceeding of the IEEE, February 1972.

[SEWE 83] SEWERINSON, A., "A Remote Subscriber Satellite System", Proc. 1sr Satellite Communications Conference, Ottawa, June 1983, pp. 14.1.1-14.1.4.

[SMAL 83] A.R., SMALLEY, "Light Route TDMA for Business Communications".

[SPIL 77] SPILKER, J., "Digital Communications by Satellite", Prentice Hall, 1977.

[WERT 69] WERTH, A.M., "SPADE: A PCM FDMA Demand-assignment System for Satellite Communications", Proc. ICDSC-1, London 1969.

[VANT 79] VAN TREES, H.L., "Satellite Communications", IEEE Press, N.Y., 1979.

4. PACKET ORIENTED MULTIPLE ACCESS TECHNIQUES

4.1 Introduction

The problem of multiple access in the design of satellite systems has been solved in the past with voice communications in mind. The common multiple access techniques are: frequency division multiple access (FDMA), time division multiple access (TDMA), and code division muliple access (CDMA). In those multiple access protocols the satellite resource available is subdivided into separate voice grade channels, and the basic unit for allocation is a channel. Channels can be either (i) fixed assigned, or (ii) demand assigned to users.

The channel-oriented multiple access protocols are suitable for voice traffic and may also be suitable for some data traffic. Data communications in general, however, have very diverse requirements ranging from inquiry-response systems with intermittent traffic to file transfers with large volumes of data. In this environment, an appropriate measure of traffic carrying capacity is no longer the number of voice grade channels, but instead the aggregate throughput rate in number of messages (or packets or bits) that can be transported per unit of time while satisfying user-specified delay constraints.

The problem of interest in this chapter begins where the traditional satellite system designers leave off. A satellite channel of C bits/s is assumed to be available which may have been derived from an FDMA, TDMA or CDMA system at a higher level of satellite resource allocation. The satellite channel is to be shared by a population of users, each with her/his own earth station, for communication among themselves. The users have random traffic demands and delay constraints. The problems of modulation, clock synchronization, coding,

random noise etc. are assumed to have been solved already. The emphasis is on packet-oriented multiple access protocols for sharing the channel.

The key measure of performance of a multiple access protocol is its channel throughput versus average delay trade-off characteristic. The channel throughput is defined to be the ratio of the rate of successfully transmitted data blocks to the rate CP, where C is the channel transmission rate in bits per second and P the (average) length in bits of transmitted data blocks. Overheads that are directly attributable to the implementation of a multiple access protocol are accounted for in the calculation of channel throughput. Hence the maximum channel throughput is less than one and is of interest as a gross measure of performance.

The three major classes of packet-oriented multiple access protocols are reviewed: contention protocols, reservation protocols, and adaptive mixed mode protocols. Specific protocols from each classes are described and their performance characteristics examined.

4.2 Random Access

In packet-switching data communications using satellite channels the propagation delay between any source-destination pair is so considerable compared to the packet transmission time that the essential environmental requirement of the carrier sense multiple access (CSMA) is not met. In such an environment, random access techniques essentially reduce to different ALOHA schemes as well as improved contention resolution algorithms.

4.2.1 ALOHA Schemes

In the unslotted or pure ALOHA [ABRA 70] each terminal which has a packet to send simply transmits it. If it overlaps in time at the transponder with one or more transmissions from other stations, a conflict occurs, and the packets are not received correctly in the downlink. If within some appropriate time-out period following its transmission, a user receives an acknowledgment from the destination, then it knows that neither conflict nor transmission error occured. Otherwise it must retransmit the packet. The broadcast property of the channel allows each sending station to monitor the success or failure of its transmissions (with respect to conflict) and to quickly queue unsuccessful packets for retransmission after the maximum station-to-station propagation-plus-transmission time has elapsed. To avoid continuously repeated conflicts, the retransmission delay is randomized across the transmitting devices, thus spreading the retry packets over time.

It is possible to increase the maximum channel efficiency for constant-length packets by slotting. In the resulted slotted ALOHA, time is divided into slots of duration equal to the transmission time of a single packet (of constant length). Each user is required to synchronize the start of transmission of its packets to coincide with the slot boundary. When two packets conflict, they will overlap completely rather than partially, providing an increase in channel efficiency over pure ALOHA.

To evaluate the performances of ALOHA protocols from the view point of throughput-delay trade-off, analyses were performed for several models of traffic under various assumptions [ABRA 70, ABRA 73, KLEI 73]. For illustration purposes, let us consider the model of traffic from many small users sharing a slotted ALOHA channel. It is constructed with the following assumptions:

- there are an infinite number of users who collectively form an independent traffic source,
- the traffic entering the channel is an independent process,
- the channel traffic is Poisson, and
- the channel traffic is independent over any K consecutive slots.

The critical system parameters are:

S_o = average number of packets per slot generated by the source,
S = throughput rate in packets/slot,
G = channel traffic including new transmissions and retransmissions in packets/slot,
K = number of slots over which random retransmission delays are uniformly distributed, and
R = number of slots which can fit into a roundtrip propagation time.

Then the following results are obtained:

- the probability that a newly generated packet is successfully transmitted

$$q = \left[e^{-G/K} + \frac{G}{K} e^{-G}\right]^K e^{-S}$$

- the probability that a previously blocked packet is successfully transmitted

$$q' = \frac{1}{1-e^{-G}} \left(e^{-G/K} - e^{-G}\right) \left[e^{-G/K} + \frac{G}{K} e^{-G}\right]^{K-1} e^{-S}$$

- the throughput rate

  $S = Gq'/(q'+1-q)$

- the average time in slots until a packet is successfully received

  $D = R+1+[R+1+(K-1)/2](1-q)/q'$

- as long as the maximum value of the system throughput rate for the K chosen is not exceeded, the system will follow the input, i.e., $S = S_0$. Otherwise, the system will go unstable.

Although the maximum achievable channel utilization is low (1/2e for pure ALOHA, and 1/e for slotted ALOHA), the ALOHA schemes are superior to fixed assignment when there is a large population of bursty users [TOBA 76]. Moreover, these bounds can be exceeded if some stations have a significantly higher traffic rate than others. This "excess capacity" effect can result in an efficiency approaching one for certain limiting cases at the expense of many retransmissions per packet (and thus large delays) for stations with small rate [KLEI 73]. Note also that the performance results are obtained by assuming that acknowledgments are instantaneous, always received correctly and for free. Concerning the degradation in channel capacity due to the overhead created by the error control traffic, it has been shown that, for a common-channel configuration with nonpriority acknowledgment traffic, the channel capacity of slotted ALOHA drops to 14 percent of the channel bandwidth in the case of a large population of small users [TOBA 78].

The most ennoying characteristic of ALOHA schemes is their inherent instability. It is then necessary to find the control functions which provide the best system performance. Markov decision theory has been successfully been applied to the design and analysis of control procedures suitable to slotted ALOHA [LAM 75].

4.2.2 Tree Retransmission Schemes [CAPE 79]

The tree algorithms can be used in conflict resolution instead of retransmitting randomly in the future in order to offer stable multiaccess protocols. The data are assumed to be transmitted in fixed length packet form. The channel is assumed to be slotted and such that the sources can determine after transmission whether there have been zero, one, or multiple packets in the slot. If a packet does not collide with other packets, then it is assumed to be successfully transmitted. Tree algorithms are based on the observation that a contention among several active sources is completely resolved if and only if all the sources are somehow subdivided into groups such that each group contains at most one active source. In its simplest form, the tree algorithm consists of the following.

Each source corresponds to a leaf on a binary tree. This represents a binary addressing scheme. The channel time axis is slotted and the slots are grouped into pairs. Each slot in a pair corresponds to one of the two subtrees of the node being visited. Two consecutive slot pairs are separated by the round-trip propagation delay. Let T(x) and T(y) be two rooted subtrees and assume that no collisions have occurred up to the beginning of the present pair of slots. Within an epoch, which is defined as the interval beginning with the pair of slots in which a conflict first occurs and ending with the pair of slots where the original conflict is finally resolved, the binary tree algorithm can be stated as follows.

(1) Set k = 0, T(x) = T(k,0), T(y) = T(k,1).

(2) Transmit all the packets from sources in T(x) in the first slot of the present pair of slots, and transmit all the packets from sources in T(y) in the second slot.

(3) If any collision occur in the preceding step, then
   (a) until the collisions are resolved, no new packets are transmitted;
   (b) resolve the first collision (if any) before resolving the second (if any).

   A collision in T(x) or T(y) is resolved by dividing T(x) or T(y) into two halves (say T(k+1,i) and T(k+1,i+1) where i = 0,2,4,6,..., or 2**(k+1)-2), setting T(x) = T(k+1,i) as well as T(y) = T(k+1,i+1), and k = k+1; and repeating steps (2) and (3).

Note that the number of slots used in an epoch is twice the number of nodes visited. In degenerate cases, where there is no conflict in the first pair of slots, the epoch is simply this pair.

For Poisson source model this scheme provides a maximum throughput of 0.347 packets/slot. Upper and lower bounds to average packet delay as functions of average packet arrival rate are given in [CAPE 79]. They are all expressed in terms of algorithmic steps. An algorithmic step equals one roundtrip delay plus two slots.

If each time returning to the root node we allow the tree to be reconfigured according to the current traffic conditions, then the optimal tree for Poisson source model is binary everywhere except for the root node whose optimal degree depends on traffic conditions. Then the optimum dynamic tree algorithm can be stated as follows:

(1) Observe the number h of slots in the preceding epoch.

(2) Calculate m = rh where r is the average packet arrival rate.

(3) Determine the optimum tree where the degree g(i) of an ith level node is given by

$g(0) = 1$, if $m \leq 1.70$

n, if $1.70 + 1.15(n-2) < m \leq 1.70 + 1.15(n-1)$

$g(i) = 2$, for $i \geq 1$, all m.

(4) Execute the tree search in the following epoch.

A specific class of such dynamic schemes, where all the nodes are restricted to be binary except for the root node whose degree can take on even values, achieves a throughput of 0.430 packets/slot.

4.2.3 URN Scheme [KLEI 78]

The URN scheme may be considered as a probabilistic group reservation scheme. It adapts smoothly to network load fluctuations, and is based on the URN model described below. For purposes of adaptation the goal is to achieve the value of 1 packet/slot for channel traffic, i.e., the sum of new transmissions and retransmissions in the channel. If the number n of ready users at any time is known to individual users instantaneously, then the URN model can be used to determine the number k of users getting access right. In the URN model each user is represented by a colored ball in a urn: black color for ready, white for not ready. Let k be the number of balls drawn from the urn. The probability of a successful transmission is that of getting exactly one black ball in the sample of size k. If N is the number of balls (users), and n is the number of black balls (ready users), then the

probability of a successful transmission is maximized when k is set equal to the integer part of N/n. The selection of which k users should get access right may be achieved via identical random number generators at individual users, or via a window mechanism as well as other methods.

A solution for estimating n consists of a binary erasure reservation subchannel. An idle user who becomes ready (n increases by 1) sends a message of a few bits in the subchannel. When a ready user turns idle (n decreases by 1), the condition is detected by other users from examining his (flagged) last packet or its positive acknowlegment. An erasure (collision) in the subchannel means two or more users become ready in the same time slot. In this case, the increase of n is assumed to be two. Hence, the reservation subchannel is provided for users to communicate with each other in a limited fashion.

The adaptability of the URN scheme with respect to channel load fluctuations can be illustrated by considering the two following extreme cases. On one hand, if $n = 1$ then $k = N$; all users get access right, but only one is going to use it. On the other hand, if $n > N/2$ then $k = 1$; most users are ready, the URN scheme becomes effectively a TDMA scheme.

4.3 Demand Assignment With Distributed Control

4.3.1 FIFO Reservation - Slotted ALOHA [ROBE 73]

In this scheme reservations are made explicitly. Time division is used to provide a reservation subchannel. The channel time is slotted according to the maximum data packet size. After every M slots one slot is divided into V small slots each of which can contain a

reservation, acknowledgment, or small data packet. The reservations contend on the V small slots in a slotted ALOHA mode. Besides the reservation slots, all other slots are data slots and are used on a reservation basis, free of conflict. The frequency of occurrence of reservation slots can be made adaptive to the load on the channel and the need to make new reservations. Moreover the reservations may also be sent "piggy-backed" in data packets.

A simple distributed control technique honors each reservation, which could be up to eight packets, on a first-come-first-served (FIFO) basis. There is one common queue for all stations and by broadcasting reservations they can claim space on the queue. A station must remember only the total number of outstanding reservations and the slots at which its own reservations begin. In fact it is not necessary for any station but the originating one to remember which space belongs to whom, since the only requirement is that no one else uses the slots. Upon seeing a successful (not collided) reservation each station adds the number of slots requested to the total reservation count. The originating station has now blocked out a sequence of RESERVED slots to transmit its packets in. To allow a station to detect out-of-sync conditions and to acquire the correct reservation count, each station sends in its data packet transmissions what it believes to be the total reservation count.

There are two channel states, ALOHA and RESERVED. On start up and every time thereafter when the reservation queue goes to zero, the channel is in the ALOHA state. In this state all slots are small and the ALOHA mode of transmission is used. The first successful reservation causes the RESERVED state to begin. In this state one contended reservation slot occurs after every M reserved data slots. To avoid confusion, M is kept constant for the entirety of each RESERVED

state but it is allowed to change each time the state is entered. The initial reservation which starts the state contains a suggested value for M. This value is used until the state terminates. The robustness of this system is achieved by a proper encoding of the reservation packets to increase the probability of their correct reception at all stations. The simplest strategy is to use the standard packet sum check hardware, and send three independently sum checked copies of the reservation data. Furthermore, to limit the effect of errors, a station reacquires synchronization if it detects a collision in one of its reserved slots or an error in a reservation packet.

FIFO-Reservation offers delay improvements over TDMA. As compared to ALOHA, higher system capacity is achieved at the expense of higher delay at low channel throughputs due to higher overhead [LAM '77].

The maximum channel throughput is $(1-\theta)$ where θ is the minimum capacity required for the reservation subchannel. Let L denote the average number of data packets per reservation request, then the ratio of data bits to reservation request bits is $v = V*L$. If the reservation subchannel is used by reservations only, and its maximum throughput is C_{SA} under the slotted ALOHA protocol then

$$\theta = 1/(1+C_{SA}v)$$

$$1-\theta = C_{SA}/(C_{SA}+1/v)$$

If the partition between data and reservation slots in a frame is dynamically varied, and the reservation requests are also piggybacked; then v will be very large when the network is heavily loaded with long messages at individual users. Under these conditions the channel throughput approaches 1 packet/slot.

The average message delay can be approximately computed by taking the sum of the average delay D' incurred by the reservation request of the average delay D" incurred by the message itself after the reservation has been made. To calculate D' and D" the original broadcast channel at C bps is slit up into two separate channels: a data channel at C" = (1-θ)C bps and a reservation channel at C' = θC bps. D' is the computed using the delay formula of slotted ALOHA given in Section 4.2.1 while D" is obtained via classical queueing formulae.

4.3.2 Reservation - TDMA [WEIS 79]

A separate reservation subframe is used with each station permanently assigned a slot for sending reservations. The remaining frame time is divided into data packet slots in the same manner as above; each station is permanently assigned one data slot per data frame, with the idle station slots in each data frame dynamically allocated according to the table of outstanding reservations. Synchronization is acquired and maintained by having each station send its own reservation table entry in its reservation slot.

The procedure for evaluating throughput and delay performances given in Section 4.3.1 can also be applied here. Let M denote the number of data slots in a frame of duration T (in seconds), and N the number of users then

$$\theta = N/(MV)$$

$$D' = (M+2)T/(2M) + rT^2/[2(1-rT)]$$

where r is the average number of reservation requests per second from a Poisson source model.

4.3.3 Reservation - Tree Retransmission [CAPE 79]

Although the tree algorithm may be used to access the reservation channel in a multichannel reservation access system as illustrated in [CAPE'79], its suitability to a time-division reservation subchannel in an unichannel reservation access system remains to be explored.

4.3.4 Multiqueue Reservation - TDMA [BALA 79]

In this scheme a variable length frame consists of a TDMA reservation slot, N preassigned (PA) slots, and Y reservation access (RA) slots where N is the number of nodes and Y is a random variable. The value of Y actually depends on the reservations placed in N mini-slots of the TDMA reservation slot by the nodes. The number of PA slots must be such that their total duration is at least equal to the round trip propagation delay. Each node places a reservation for a number of slots equal to its packet queue size minus one. Hence, by the end of a frame all the nodes will have emptied out all the packets they had in their buffers at the beginning of that frame. The end of a frame is thus a renewal point. To achieve symmetry a superframing technique can be used. It consists in changing the node service order from frame to frame in a cyclic permutation manner.

The scheme can be analyzed as a cyclic multiqueue system attended by a single server under a modified gating service discipline [BALA 79]. An imbedded Markov process is obtained by examining the system at special epochs located at the beginning of each cycle/frame. In cases where the number of arrivals (at any node) during different time slots are independent and identically distributed, the reservation vector are sent in zero time, and under the assumption that the distributions of individual queue sizes are independent of each other arrival

processes at all the nodes, and are independent and identically distributed; a simplified approximate functional equation is obtained for probability generating function of queue length (in number of packets)/ frame size (in number of slots). Assuming steady state solution exists, and Poisson input at each queue the approximate functional equation is numerically solved and the mean and standard deviation are computed for the queue sizes at the nodes at the beginning of a frame as well as the frame length. GPSS simulations are also performed. Numerical results and simulation results are in "excellent" ($\leq$ = 5% difference) agreement over a wide range of N (from 2 to 40) and the system utilization factor (up to 0.70).

4.4 Adaptive Strategies and Mixed Modes

4.4.1 Implicit Reservation - ALOHA [CROW 73]

A frame concept is used to permit implicit reservations with the slotted ALLOHA approach. The fixed frame duration must be at least one round-trip propagation time. A user who has successfully accessed a slot in a frame is guaranteed access to the same slot in the succeeding frame and this until the user stops using it. Unused slots are free to be accessed by all users in a slotted ALOHA contention mode. A unused slot in the current frame is a slot which either was idle or contained a collision in the preceding frame. Users need to simply maintain a history of the usage of each slot for just one frame duration. Reservation-ALOHA is effective only if the users generate stream-type traffic or long multipacket messages.

Performance analyses of Reservation-ALOHA protocols are reported in [LAM'78, LAM'79]. Let us consider the version of Reservation-ALOHA where an End-of-Use flag is included in the header of the

last packet before a user gives up his reserved slot. Under the following assumptions:

- users have independent, identical message arrival statistics, and
- a successful packet transmission occurs in each nonreserved time slot with a constant probability S,

the channel throughput U is given by

$$U = S/[S+(1-S)/\overline{v}]$$

where $\overline{v}$ is the average number of packets that a user transmits before she/he gives up her/his reserved time slot.

Hence, if S remains constant as $\overline{v}$ increases indefinitely then the maximum channel capacity of 1 packet/slot can be attained.

For a specific case, let us consider the following user model. A population of N users of identical behavior and message arrival statistics is considered. Messages arrive to each user according to a stationary Poisson process with rate r messages/second. Each message consists of a group of h packets with the first two moments h1 and h2 and probability generating function H(z). Each user can reserve at most 1 time slot in a frame at a time. Each user has infinite buffering capacity.

Assume that the delay d_A incurred by a user to successfully transmit a packet into a nonreserved time slot has a known probability density function with the Laplace transform $D_A(s)$ and mean value $\overline{d}_A$.

If $\overline{d}_A$ is a known function of S, then $\overline{v}$ and S can be obtained by numerically solving the following system of equations:

$$S/(S+(1-S)/\overline{v}) = N\ r\ h1\ T/M$$

$$\overline{v} = \{1+r[\overline{d}_A+(h1-1)T]/(1-r\ h1\ T)\}h1$$

where M is the number of slots in a frame of T second duration. Moreover, the average message delay is given by

$$d = x10/[1-r(x1-x10)] + r(x20-x2)/2/[1-r(x1-x10)] + r\ x2/2(1-r\ x1)$$

where x20 is the second moment of the service time with mean x10 and Laplace transform $D_A(s).H(\exp(-sT).\exp(sT)$, x2 the second moment of the service time with mean x1 and Laplace transform H(exp(-sT)), and

$$x10 = \overline{d}_A+(h1-1)T$$

$$x1 = h1\ T$$

4.4.2 Round-Robin Reservation - Fixed Assignment [BIND 75]

A fixed assignment frame structure is used in which each station is assigned a slot in the frame permanently. In this slot, a station can both send a data packet and make a (piggyback) reservation using bits in the data packet header. Each station with packet to send always uses its permanently assigned slot but, in addition, can share with other busy stations the slots not in use by idle stations. Equal size frames are of duration greater than the round-trip propagation delay, and the number of slots in a frame is larger than the number of stations.

Distributed control is used to perform the dynamic allocation of idle slots. Each station is required to transmit information regarding its own queue of packets piggybacked in the data packet header (transmitted in the previous frame). A zero count indicates that the corresponding slot is free. Each station also maintains a table of total unserviced reservations heard from each station, with each idle station slot allocated one at a time on a sequential round-robin basis to the next station having a nonzero table entry. When an idle station becomes busy, it always first sends in its permanently assigned slot, deliberately creating a conflict if the slot was dynamically assigned to another station. The conflict is assumed to be detected by all stations, causing them to stop using the slot for dynamic allocations. The station can then begin using its slot to send data and also to make reservations for other packets in its queue, allowing it to participate in the dynamic allocations. To maintain synchronization and to allow initial acquisition, one of the stations broadcasts a copy of its table at infrequent intervals.

4.4.3 Reservation upon Collision (RUC) [BORG 77]

The basic concept in these schemes is to switch back and forth between contention mode and reservation mode. Thus, the channel can operate in two different states (A) S-(Slotted) ALOHA state, and (B) RESERVATION state. The channel starts with and remains in State (A) until a collision. When k ($k \geq 2$) packets collide all stations build up their own "Channel Queue Tables" CQT which assign the next k slots to the k collided stations for the retransmission of their packets and the channel initiates a State (B) period. If R is the number of slots in a round-trip time, collision can be detected and CQT consequently updated during the first (R-1) slots of a State (B) period. When COT's are empty, the channel returns to State(A).

Note that any new packets arrived at stations during a State (B) period are scheduled at random for transmission in a subsequent State (A) slot. The slot is randomly chosen among the next K State (A) slots. The implementation of the RUC technique is based upon the knowledge of the origin of the collided packets at each time slot. Such an information is achieved by attaching an ID to the packet. The channel is divided into time slots of fixed length t. Each slot t is again divided into a data subslot t' and an ID subslot t". The packet ID's are transmitted in such a way to be recognized whatever number of packets collide in the slot. For example, packet ID may simply be the existence of carrier in the window assigned to the station which is actually transmitting a packet. Its detection then reduces to the well-known ON-OFF keyed signal detection problem.

The RUC techniques favorably compare with the slotted ALOHA system: the delay-throughput performances show that an almost double channel utilization is achieved at the same average delay for Poisson source model. For a throughput S not exceeding M/(M+1), where M is the number of users, the average delay (in time slots) is given by the following formula [BORG 78].

$$D = (1-\alpha)\ (R+Q) + \tau/T + 1.5$$

where: $\alpha = (1-S)\ \exp(-s)/(1-S\ \exp(-S))$

$R = [\tau/T] + 1$

$0 \leqslant Q' = S(1-S\ \exp(-S))/[2(1-S)(1-\exp(-S))]$

τ = roundtrip propagation delay

T = time slot duration.

An improved version of the RUC techniques, called Split Reservation upon Collision (SRUC), extends its suitability to other channel environments [BORG 78].

4.4.4 Priority Oriented Demand Assignment (PODA) [JACO 77]

In the context of a satellite channel, PODA has been proposed as the ultimate scheme which attempts to incorporate all the properties and advantages seen in many of the previous schemes. It has provision for both implicit and explicit reservations, thus accomodating both stream and packet-type traffic. It may also integrate the use of both centralized and distributed control techniques thus achieving a high level of robustness.

4.4.4.1 Basic characteristics

- Explicit reservations are used for datagram messages.
- A message stream is set up by an explicit reservation, and then maintained by automatic scheduling at predetermined intervals.
- High reliability is achieved by acknowledgment and retransmission.
- High availability is attained by distributed control.
- Robustness and mixed receiving rate require the integration of both centralized and distributed control techniques.

PODA schemes use TDMA channelization. Channel time is divided into information subframes and control subframes. Information subframes are used for scheduled datagram and stream transmissions with piggybacked reservations and acknowledgments. Control subframes are devoted to timely reservations such as initial stream reservations, urgent reservations. Several PODA versions exist: in FPODA systems the control frames are accessed in fixed TDMA mode, while CPODA systems are characterized by slotted ALOHA control subframes.

4.4.4.2 Reservations and Scheduling

All datagram reservations received successfully in the satellite downlink are entered into a scheduling queue, maintained by all stations performing distributed scheduling. Reservations are first ordered according to urgency, which is a function of potential lateness and priority, and then further ordered within the same urgency according to additional criteria to provide fairness to the stations involved. Channel time in the information subframe is then assigned to messages according to the latest ordering of reservations in the scheduling queue.

A reservation for a stream is made only once, at the beginning of the stream use, and is retained at each station in a separate stream queue when received in the downlink. Each stream reservation contains information defining the stream repetition interval, desired maximum delay relative to this interval, and priority. Whenever the interval starting time is near, a reservation is created for that stream's next message and entered into the scheduling queue; the reservation urgency is calculated according to the stream queue information and is treated the same as datagram reservation urgency.

To maximize channel efficiency while satisfying urgency constraints, each reservation in the scheduling queue may actually be made for a group of several distinct datagrams subject to a maximum allowed burst length.

To achieve the reduced delay of distributed control while also making centralized assignments for stations unable to participate in the scheduling, the information subframe is partitioned into a centralized assignment section and a distributed assignment section.

The centralized assignment section represents assignments made at least one hop in the past, while the distributed assignment section represents assignments being made from the scheduling queue in the current frame. All stations which perform distributed scheduling can also make the centralized assignments. The assignments are sent by a station transmitting a scheduled burst in the distributed assignment section. A separate control burst is thus necessary for this function only when no messages are otherwise scheduled for the distributed assignment section.

4.4.4.3 Scheduling Synchronization

To use of distributed scheduling introduces potential synchronization problems due to local receiving errors at each station performing scheduling. The synchronization acquisition and maintenance process consists of 3 states: the initial acquisition state, the out-of-sync state, and the in-sync state, with the transition into and out of a particular state based upon consistency checks.

When a station is first turned on, it enters the initial acquisition state in which it has no information about the state of the scheduling queue in the other stations. In this state it listens for new reservations and observes the packets being transmitted in the channel but does not itself send any data messages or reservations. It thus builds up a reservation list compatible with other stations. When certain number of correct consistency checks have been made (by comparing its own scheduling decisions against the transmissions actually taking place in the channel), the station enters the out-of-sync state.

In the out-of-sync state, the station schedules the received reservations and monitors for consistency scheduling, but it does not

send any data packets or make reservations. If consecutive consistent schedulings are made within a certain period of time, the station enters the in-sync state.

While in the in-sync state, the station can send reservations and scheduled messages. Whenever it detects a transmission in the channel which it has not scheduled for the same time, it readjusts its scheduling queue using the header information contained in the transmission. If a certain number of consecutive inconsistent checks are detected within a given time period, the station enters the out-of sync state, preventing possible subsequent disturbance to transmissions made by other stations due to its own incorrectly sent data packets.

4.4.4.4 SATNET Experiments [JACO 78, CHU 78]

In the SATNET four earth stations communicate with each other over a shared satellite channel supporting 64 kbps data transmission with a bit error rate on the order of 10^{-6} or 10^{-7}. The data rate for a packet can be 16, 32, or 64 kbps; however, one of the four earth stations has its nominal receiving rate limited to 16 kbps. The network protocols are implemented in the Satellite IMP which itself is interfaced to the PSP terminal. This self-contained terminal consists of redundant burst modems and associated frequency selection equipment, interface equipment supporting a variety of transmission modes, and appropriate equipment to support both local and remote test and monitoring functions. The Satellite IMP implements global timing control, demand access protocols, input and output to the satellite channel PSP terminal, internal network protocol (e.g., control of information flow), the SATNET side of the host access protocol, certain measurement capabilities, software to control and receive data from the PSP terminal test and monitoring functions, and software to allow interaction

with the SATNET monitoring and control center. Gateways are used to interface SATNET to ARPANET. Gateway computers act as host computers on each of the connected networks.

The objectives of the experiments are:

- measuring channel and equipment performance in the broadcast packet satellite environment,
- measuring the performance of the various demand assignment algorithms and the related portion of the SATNET protocol, including the various control strategies,
- measuring SATNET's capability to support certain types (interactive, file transfer, stream) of application using the traffic generation and statistics collection programs located in the Gateways, and
- demonstrating the use of SATNET for providing service to certain applications, e.g., point-to-point and conference packet speech, communication between elements of a set of identical remotely located local networks, etc.

Experimental measurements and simulations have been performed in order to study the behavior of the PODA protocols [CHU 78] such as expected packet delay as a function of traffic load for various minimum control subframe sizes and for different message classes, unfairness at heavy load, effects of noise on packet delays and throughput.

4.4.5 Interleaved Frame Flush-Out (IFFO) [WIES 79]

These schemes combine reservation and contention techniques.

The basic frame has a reservation structure of R slots in which unreserved slots may be used for transmission on a contention

basis. The first slot of each frame, which consists of M minislots (where M is the number of terminals), is known as the status slot. It is used by the terminals for reservations in a contention free manner. The status slot is followed by a sufficient number of reserved slots for all reservations received. Since the frame length must at least equal to the maximum round-trip propagation delay, the additional slots which remain unreserved (if any) are then used for transmission on a contention basis.

Let a packet which arrives at a station in frame k be called k-packet. Reservation minipackets for all k-packets will be transmitted in the first slot of frame (k+1). While A(k) reservations are transmitted for each of the k-packets, some of these reservations may not be needed due to the possibility of successful transmission in contention slots. IFFO schemes require that k-packets selected for contention transmission be transmitted only during the contention slots of frame k, and not during the contention slots of later frames. By observing the channel traffic, the terminals can count the number S(k) of packets successfully transmitted by each terminal in the contention slots of frame k, not later than the end of the status slot of frame (k+2). Then, each terminal will know how many reserved slots are actually needed by every other terminal, and will be able to determine which of the slots of frame (k+2) are assigned to it.

Let the state variable in frame k be R(k), the number of actually needed reserved slots in frame k. The number of contention slots in frame k is then $N(k) = \max(R-1-R(k),0)$. Since $R(k+2) = A(k) - S(k)$, R(k+2) depends only on R(k), and not R(k+1). Under appropriate conditions the process $\{R(k)\}$ consists of two interleaved Markov chains $\{R(2j)\}$ and $\{R(2j+1)\}$ with identical statistics. They may then be analyzed separately.

There exist 3 versions of IFFO:

- Pure Reservation (PR) IFFO where unreserved slots just remain idle.
- Fixed Contention (F) IFFO where the transmission policy in contention slots is to transmit a packet if, and only if, it was generated at the terminal during the previous slot.
- Controlled Contention (C) IFFO where in each contention slot a nonempty terminal will transmit a packet with probability p(m,n,R(k)) when there are m k-packets present at the terminal at the beginning of slot n in a frame of R(k) effectively reserved slots. A construction procedure for p(m,n,R(k)) is given in [WIES 80].

Analyses and simulations have been performed for M terminals with infinite buffer and sharing a common satellite channel [WIES 80]. Packets are assumed to arrive at each terminal according to a Bernoulli process with rate r packets/slot. Exact forms for the transition probability matrices for the PR and F IFFO protocols are obtained. Some independent assumptions are necessary in the derivation of an approximate transition probability matrix for the C IFFO protocol. These transition probability matrices are then used to determine the equilibrium probability distributions for the number of reserved slots per frame. Numerical solutions for the equilibrium probability distributions are generated using finite transition probability matrices. The performance of the protocols is evaluated by computations and simulations, and comparisons to other protocols. It is shown to be stable and superior to that of other protocols over a wide range of throughput values in terms of packet delays.

FINAL REPORT

IMPACT OF USE OF SATELLITE FOR DATA COMMUNICATIONS

prepared for:

DEPARTMENT OF COMMUNICATIONS
Journal Tower North
300, Slater Street
Ottawa (Ontario) K1A 0C8

April 1984 Projet CDT P832

4.4.6 Distributed Reservation Control (DRC) [GREE 81]

In DRC schemes the available bandwidth is partitioned into a data channel and a narrow-band reservation subchannel. The binary signaling rate of the reservation subchannel is made L times ($4 \leq L \leq 8$) larger than the packet signaling rate of the data channel. A surrogate contention process is created at the reservation subchannel level. In a DRC system a transmission frame of M packet slots has a duration longer than any round trip propagation delay. For the reservation purpose each packet slot is divided into L equally spaced segments. The successful transmission of a packet by means of the DRC algorithms involves the following phases.

(i) Packet Acceptance Phase. Each node continually monitors the channel for generating an estimate of the overall traffic rate and establishing the acceptance threshold for the purpose of flow control.

(ii) Reservation signal Transmission Phase (Frame 0). Upon packet acceptance by a node, a uniformly distributed random integer between 1 and ML is generated, say X, and the reservation signal will be transmitted by the node over the reservation subchannel during the next occurrence of the Xth segment.

(iii) Reservation Signal Detection Phase (Frame 1). All nodes which transmitted a reservation signal in any segment of a given frame will monitor the following frame to determine if their reservation request was granted, and, if so, to identify their assigned transmission slot. If a reservation request is rejected, then the reservation process must be reinitiated using a new random integer X.

(iv) Packet Transmission Phase (Frame 2). Each node which obtained a packet slot assignment on a given frame will transmit its packet on its assigned slot during the next frame.

(v) Packet Verification Phase (Frame 3). On the frame following a packet transmission, the transmitting node will monitor the data channel during its assigned slot and verify if the packet was transmitted without error/interference. If so, the packet transmission process is complete; otherwise, the packet reservation process will be reinitiated using a new random integer X.

Two DRC algorithms are proposed. In DRC-I it is assumed that all nodes can make a reliable binary decision as to whether a segment contained no reservation signal or at least one reservation signal. Let N(x) denote the number of segments on which the reservation signal was detected from the beginning of the frame up to and including the x_th segment. If N(x) > M then the reservation request is rejected; otherwise, the N(x)_th packet slot is assigned.

In DRC-II each node is required to make a ternary decision as to whether a segment contained zero, one, or plural reservation signals. Let Ns(x) and Np(x) denote the number of singly and plurally occupied segments detected during the first x segments in frame. Let Nst be the total number of singly occupied segments in frame. Pflag is set to 1 if plural reservation signals were detected on the x_th segment, or to 0 otherwise. Two cases are considered.

Case I (Pflag = 0) If Ns(x) $\leq$ M, transmit packet on the Ns(x) th slot of next frame; otherwise, the reservation request is rejected.

Case II (Pflag = 1) If Nst > M, all reservation requests occurring on plurally occupied segments are rejected. Otherwise, the remaining Na slots (Na = M-Nst) are available to attempt to resolve contention among some or all of the plurally occupied segments.

For datagram type messages, the performances of DRC algorithms with M = 64 and a controlled operation, in terms of attained throughput R and corresponding average delay W (in frame times) are as follows.

L	DRC-I		DRC-II	
	R	W	R	W
2	.65-.68	5.7-6.6	.77-.80	4.9-5.6
4	.80-.84	3.7-3.9	.97-.995	4.0-4.4
8	.88-.91	3.0-3.2	.97-.995	3.3-3.5

4.5 A Qualitative Appraisement

Since the performance of a multiple access protocol is strongly dependent upon the traffic model and network loading, it is necessary to examine the traffic characteristics and transmission requirements of users of a message- (or packet-) oriented communication network.

Let us consider a population of N users sharing a single satellite channel. A user constitues a traffic source that can be modeled as a random point process with instants of message arrivals being the points of interest. A message is a block of data that has a time delay constraint associated with it for delivery to a destination user. Each user is then characterized by

λ_i = the message generation rate,

S_i = the average message delay constraint,

L_i = the average number of packets per message,

T_i = the average inter-arrival time between messages,

where $T_i = 1/\lambda_i$

The bursty factor β_i of the traffic source representing the user i is defined to be

$$\beta_i = S_i/\lambda_i$$

If each user has his/her dedicated subchannel as in fixed assignment protocols FDMA, TDMA, CDMA; then the bursty factor β_i is related to the ratio of peak to average sub-channel data rate PAR_i and the sub-channel throughput S_i as follows

$$PAR_i \geq 1/\beta_i$$

$$S_i \leq \beta_i$$

A traffic source with β much smaller than 1 is said to be bursty [LAM 78]. Hence, it is clear that

> "For bursty users, channel-oriented multiple access protocols such as fixed assignment or demand assignment (over a period of time much longer than the average message delay constraint) is very inefficient (such that the channel throughput is much smaller than 1)".

In order to improve the throughput of a broadcast channel shared by users with random bursty traffic, it is desirable to dynamically allocate transmission capacity on a per message (or packet) basis. Then, a multiple access protocol is simply an algorithm (possibly distributed as well as nondeterministic) for determining the access rights of the users. It should resolve channel access conflicts without excessive overhead.

The average delay-throughput performance of a multiple access protocol is determined primarily by the time overhead required to identify a ready user, when one or more are present, and assign channel access to him/her. The exact identity of the user is unimportant. This can be accomplished by a polling protocol: the central controller polls a set of passive terminals. Polling protocols, however, are not suitable for a satellite channel due to the long channel propagation time. They are also not suitable for large population of very bursty users.

In contention and reservation protocols, ready users actively seeks channel access instead of waiting to be polled. Contention protocols all have distributed control. Each user makes his/her own decision regarding channel access based solely upon observable outcomes in the broadcast channel. The overhead incurred by contention protocols for assigning channel access to ready users is independent upon the level of traffic. Hence, pure contention protocols are suitable for a large population of bursty users.

The objective of reservation protocols is to avoid collisions entirely. Since users are geographically distributed, a reservation subchannel is required for users to communicate with each other. Reservation protocols mainly differ in their manner to solve the two key problems: (1) implementation of the reservation subchannel, and (2) implementation of a distributed global queue.

In summary, the performance of a multiple access protocol is strongly dependent upon the traffic model and network loading. In general, some traffic characteristics favor one class of protocols more than others. Some of them may be listed as follows.

TRAFFIC MODELS	MULTIPLE ACCESS PROTOCOL FAVORED
nonbursty users	TDMA, FDMA, or CDMA
bursty users, short messages	ALOHA, S-ALOHA
bursty users, long messages, large N	FIFO Reservation - S-ALOHA Implicit Reservation - ALOHA
bursty users, long messages, small N	Reservation - TDMA Multiqueue Reservation - TDMA Round-Robin Reservation - Fixed Assignment
mixed traffic with priorities (datagram and stream traffic)	CPODA ou FPODA

REFERENCES

[ABRA 70] Abramson, N., "The ALOHA System - Another Alternative for Computer Communications", AFIPS Conf. Proc., Vol. 37, 1970, pp. 281-285.

[ABRA 73] Abramson, N., "Packet Switching with Satellites", AFIPS Conf. Proc., Vol. 42, 1973, pp. 695-702.

[BALA 79] Balagangadhar, M.M. and Pickholtz, R.L., "Analysis of a Reservation Multiple Access Technique for Data Transmission via Satellites", IEEE Trans. Commun., Vol. COM-27, 1979, pp. 1467-1475.

[BIND 75] Binder, R., "A Dynamic Packet-Switching System for Satellite Broadcast Channels", Proc. ICC, 1975, pp. 41.1-5.

[BORG 77] Borgonovo, F. and Fratta, L., "A New Technique for Satellite Broadcast Channel Communications", Proc. 5th Data Commun. Symp., 1977, pp. 2.1-4.

[BORG 78] Borgonovo, F. and Fratta, L., "SRUC: A Technique for Packet Transmission on Multiple Access Channels", Proc. ICCC, 1978, pp. 3121.(1-7).

[BRID 75] Bridwell, J.D., "Packet Reservations with Constrained Response Time for Satellite Channels", Proc. NTC, 1975, pp. 24.4-7.

[CAPE 79] Capetanakis, J., "Tree Algorithms for Packet Broadcast Channels", IEEE Trans. Inform. Theory, Vol. IT-25, 1979, pp. 505-515.

[CAPE'79] Capetanakis, J., "Generalized TDMA: The Multiaccess Tree Protocol", IEEE Trans. Commun., Vol. COM-27, 1979, pp. 1476-1484.

[CHU 78] Chu, W.W. and Naylor, W.E., "Measurement and Simulation Results of C-PODA Protocol Performance", Proc. NTC, 1978, pp. 4.2.1-7.

[CROW 73] Crowther, W.R. et al., "A System for Broadcast Communication: Reservation-ALOHA", Proc. 6th Hawaii Int. Syst. Sci. Conf., Honolulu, Jan. 1973.

[GREE 81] Greene, E.P. and Ephremides, A., "Distributed Reservation Control Protocols for Random Access Broadcasting Channels" IEEE Trans. Commun., Vol. COM-29, 1980, pp. 726-735.

[HSU 78] Hsu, N.T. and Lee, L.N., "Channel Scheduling Synchronization for the PODA Protocol", Proc. ICC, 1978, pp. 42.3.1-5.

[JACO 77] Jacobs, I.M., Lee, L.N. and Viterbi, A., "CPODA: A Demand Assignment Protocol for SATNET", Proc. 5th Data Commun. Symp., 1977, pp. 2.5-9.

[JACO 78] Jacobs, I.M., Binder, R. and Hoversten, E.V., "General Purpose Packet Satellite Networks", Proc. IEEE, 1978, pp. 1448-1467.

[KLEI 73] Kleinrock, L. and Lam, S.S., "Packet Switching in a Slotted Satellite Channel", AFIPS Conf. Proc., Vol. 42, 1973, pp. 703-710.

[KLEI 75] Kleinrock, L. and Lam, S.S., "Packet Switching in a Multi-access Broadcast Channel: Performance Evaluation", IEEE Trans. Comun., Vol. COM-23, 1975, pp. 410-423.

[KLEI 78] Kleinrock, L. and Yemini, Y., "An Optimal Adaptive Scheme for Multiple Access Broadcast Communication", Proc. ICC, 1978, pp. 7.2.1-5.

[KLEI'78] Kleinrock, L. and Gerla, M., "On the Measured Performance of Packet Satellite Access Schemes", Proc. ICCC, 1978, pp. 3122.(1-8).

[LAM 75] Lam, S.S. and Kleinrock, L., "Packet Switching in a Multiaccess Broadcast Channel: Dynamic Control Procedures", IEEE Trans. Commun., Vol. COM-23, 1975, pp. 891-904.

[LAM 77] Lam, S.S., "Delay Analysis of a Time Division Multiple Access (TDMA) Channel", IEEE Trans. Commun., Vol. COM-25, 1977, pp. 1489-1494.

[LAM'77] Lam, S.S., "Satellite Multiaccess Schemes for Data Traffic", Proc. ICC, 1977, pp. 37.1.19-24.

[LAM 78] Lam, S.S., "A New Measure for Characterizing Data Traffic", IEEE Trans. Commun., Vol. COM-26, 1978, pp.137-140.

[LAM'78] Lam, S.S., "An Analysis of the R-ALOHA Protocol for Satellite Packet Switching", Proc. ICC, 1978, pp. 27.3.1-5.

[LAM 79] Lam, S.S., "Satellite Packet Communication - Multiple Access Protocols and Performance", IEEE Trans. Commun., Vol. COM-27, 1979, pp. 1456-1466.

[LAM'79] Lam, S.S., "On Protocols for Satellite Packet Switching", Proc. ICC, 1979, pp. 58.6.1-6.

[LEE 77] Lee, L.N. and Jacobs, I.M., "A Priority-Oriented Demand Assignment (PODA) Protocol and an Error Recovery Algorithm for Distributively Controlled Packet Satellite Communication Networks", Proc. EASCON, 1977, pp. 14.1.a-f.

[ROBE 73] Roberts, L.G., "Dyanmic Allocation of Satellite Capacity through Packet Reservation", AFIPS Conf. Proc., Vol. 42, 1973, pp. 711-716.

[TOBA 76] Tobagi, F. and Kleinrock, L., "Packet Switching in Radio Channels: Part III - Polling and (Dynamic) Split Channel Reservation Multiple Access", IEEE Trans. Commun., Vol. COM-24, 1976, pp. 832-845.

[TOBA 78] Tobagi, F. and Kleinrock, L., "The Effect of Acknowledgment Traffic on the Capacity of Packet-Switched Radio Channels", IEEE Trans. Commun., Vol. COM-26, 1978, pp. 815-826.

[WIES 79] Wieselthier, J.E. and Ephremides, A., "A New Multiple Access Protocol for Satellite Communication Networks", Proc. 18th CDC, 1979, pp. 147-150.

[WIES 80] Wieselthier, J.E. and Ephremides, A., "A New Class of Protocols for Multiple Access in Satellite Networks", IEEE Trans. Automat. Contr., Vol. AC-25, 1980, pp. 865-879.

5. CONCLUDING REMARKS AND SUGGESTIONS

In this chapter, we discuss the basic parameters (reliability characteristics, required response time, volumes of traffic) of the various data transmission services to be considered for transmission over the satellite channel. All the pertinent information related to the different data communication services which has been needed in the elaboration of the different scenarios has been drawn from the, unfortunately, rather disparate sources of data available to us and pertaining only to a sample of all the governmental departments whose telecommunications needs fall under the responsability of the Agency. As a consequence several scenarios are suggested with regards to the particular applications. Each one of these suggestions could be the subject of further studies where more quantitative methods of analysis should be applied to rank order the remaining alternatives and eventually suggest a final choice for the ultimate solution.

5.1 Classes of Data Transmission Requirements

5.1.1 Reliability Parameters and their Use on Realistic Links

In any digital communication system whether terrestrial or via satellite errors are unavoidable, and hence error performance becomes a key transmission parameter. With the advent of Integrated Services Digital Network (ISDN) concepts, much work has been undertaken, especially by the CCITT Study Group XVIII, in order to define and specify errors and error performance [CCIT 80], [CCIT 81]. The error performance consists of three basic parts: the method of specification of the error performance, the level of end-to-end error performance, and the allocation of error degradation among the possibly various transmission systems that make up the end-to-end connection.

When specifying error performance, one has to cater to the network user and translate the specifications into measurable performance parameters. CCITT has considered two alternatives in its G.821 Recommandation [CCIT 80], [CCIT 81]. The first is mean bit error rate (BER) while the second is error free time intervals.

BER, defined in a measurement period T, is the ratio of the number of bits in error to the total number of bits transmitted in the time period T. As for error free time intervals, a time period of one second is considered with the ensuing requirements for percentage of error free seconds (EFS). However for data blocks transmission a specification in tenths of seconds or error free deciseconds (EFdS) appears to be preferred since a tenth of a second is closer to the duration of the data blocks used in most data transmission systems; hence EFdS provide a finer characterization in those systems where the noise is essentially white, that is where the errors occur randomly and where the burst lengths are very small: e.g. satellite channels and some radio channels.

Consequently in characterizing and specifying a connection error performance both mean BER and error free time periods are used. When considering end-to-end performance level, the definition of different parameters for voice and data requirements may lead to the assignment of values which would imply different performance levels for different services. In the case of integrated services this could lead to some contradictions and difficulties. Therefore the values for the two parameters described above are chosen to represent approximately equivalent practical performances as they apply to a 64 kbps connection.

The mean bit error rate is specified to be equal to 1×10^{-6} with a time period of 10 minutes in order to be defined as "acceptable". It is also required that this mean BER be achieved for 90% of the time period. In addition an "unacceptable" level of performance is defined as a BER worse than 1×10^{-3} for more than some percentage of seconds. Hence not only the system must meet an acceptable level of 1×10^{-6} during 90% of the time, during the remainder 10% of the time it cannot be degraded below the 1×10^{-3} BER level.

As for the free time intervals, over a one second time period the requirement is 92% EFS, whereas for time periods of deciseconds, the requirement is 92% to 92.2% EFdS.

The notion of error free seconds (EFS) should be a user oriented notion. That is, as a user perceives it, EFS is the percentage of time during which there are no errors. For data transmitted in blocks or packets, a time reference T_0 = 1 s. is meaningful to a user only if the time to transmit a complete block or a packet is also of the order of one second. Therefore, in addition to EFS the percentage of error free blocks (EFB) appears to be a more appropriate service-oriented performance measure. One could also consider error free messages (EFM) [DECI 82] but then a typical message length must be also defined. Since a message may be viewed as a self contained information entity, it may be represented by one or several blocks or packets. Therefore EFB and EFM should be considered separately, in particular EFB values should be specified separately for different block lengths, and EFB calculations can be derived from the error process model of the channel.

In order to allocate the allowable degradation among the various transmission systems that make up an overall connection, CCITT developed the concept of a 27 500 km Hypothetical Reference Connection

(HRX) for digital voice and data application [DECI 82]. An HRX represents a typical "worst-case" end-to-end digital connection formed by tandem point-to-point circuits operating at a specified bit rate, and defined by the intermediate switching points and the distances between them. Clearly, designing to such a reference, for most real connections a better performance is garanteed.

An HRX is divided into Local, National and International portions, as shown in Figure 5.1 [CCIT 80], [DECI 82]. For 64 kbps, the permitted error performance degradation which have to be allocated to the various portions of this HRX are:

- 10% of minutes with BER $\geq 1 \times 10^{-6}$
- X% of seconds with BER $\geq 1 \times 10^{-3}$
- 8% seconds in error.

For 64 kbps international connection the error performance objectives are given in Table 5.1 [DECI 82].

(a) T_o = 1 second or 1 minute		
BER	% Available time	Performance classification
BER $> 10^{-3}$ T_o = 1 second	< X%	Unacceptable
10^{-6} < BER < 10^{-3} T_o = 1 second	< (10-X)%	Degraded
BER < 10^{-6} T_o = 1 second	> 90%	Acceptable

(b) T_o = 1 second	
BER in 1 second	% Available seconds
> 0	8%
= 0	92% EFS

TABLE 5.1: Error performance objectives for 64 kbps:
(a) T_o = 1 second or 1 minute
(b) T_o = 1 second

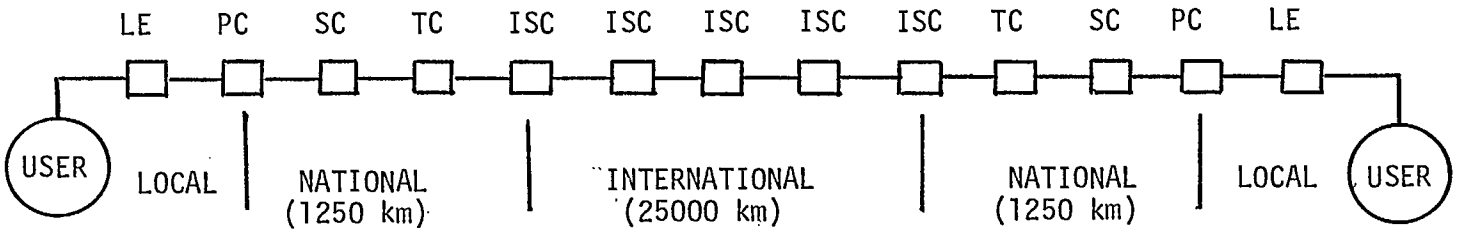


Figure 5.1 Simplified model of the longest hypothetical reference Connection (HRX).

LE = Local Exchange
PC = Primary Center
SC = Secondary Center
TC = Tertiary Center
ISC = International Switching Center

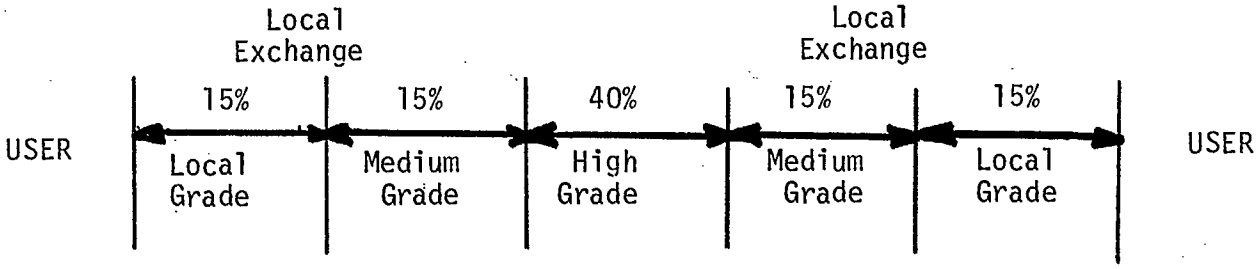


Figure 5.2 Overall performance degradation for the various portion of the HRX.

In the particular case of satellite links the following design requirements apply:

- 99.2% of minutes with BER $\leq 1 \times 10^{-6}$
- 99.4% error free seconds.

Expressed in long term mean BER, these requirements are equivalent to approximately 2×10^{-7} and 1×10^{-8} for 99.4% of the worst month, respectively.

Since a subscriber to subscriber connection involves local, national and international portions, for the special case of satellites the notion of a satellite equivalent distance (SED) which could utilize the "per kilometer" allocation for the degradation was used. As a result it is proposed that satellites be allotted 20% of the overall error performance based on an equivalent distance of 12 500 km. More precisely recommendation G.821 states that based on the understanding that satellite error performance is essentially independent of distance, an allowance of 20% of the permitted 10 minute and error free seconds is allocated to a single satellite Hypothetical Reference Digital Path utilized in the high grade portion of HRX. The HRX error allocation is indicated in Figure 5.2 where 25 000 km was considered for the high grade (satellite) portion. The satellite performance objectives become then:

- 10 minute intervals : 98% at 1×10^{-6}
- Errored seconds : $\leq 1.6\%$
- Severely errored seconds (worse than 1×10^{-3}) : $\leq 0.03\%$

In this section we have presented how the most important error parameters are specified. There are also other error parameters such as jitter, wander, slip and delay that may have to be specified in a given application. These parameters are not treated here but a discussion on them may be found in [DECI 82], [SAST 84], [GRUB 82] or [URIE 82].

5.1.2 Response and Delivery Time Characteristics

From a general standpoint, it appeared sensible to the members of our team to classify the overall data communication needs in two broad categories according to the level of reliability required at the receiving end of the transmission path. In this framework, the most appropriate boundary to be chosen appears to be the one which separates between the implicitly non redundant data used in all the broad sense Teleprocessing applications, Electronic Mail and Office Automation and the rather redundant types of data which are manipulated in applications such as Facsimile Transmission and Video/Audio Interactive communications.

In general, since a tight error control requires the use of a repetition scheme (ARQ), it follows from the inherent propagation delay implied by satellite communication that, at least for high reliability level data transmission, higher response and delivery time characteristics than on terrestrial networks must be expected. However, for highly redundant data such as facsimile, the added propagation delay should be quite negligible so that only a lower service time must be expected when compared to the classical message switching transmission techniques used on present terrestrial networks.

The following subclasses, together with their associated required response times will be considered typical of potential applications.

(1) Real time applications which are essentially characterized by a response time within a few seconds. Typical situations are real time on-line control and alarm monitoring. They imply the use of a powerful low rate forward error correcting scheme in order to guarantee the required level of reliability within the response time constraints.

(2) Time sharing applications, where the response time should probably be within 5-10 seconds. ARQ or FEC schemes will guarantee the required level of reliability depending on the raw channel bit error rate.

(3) Remote job entry, where essentially no on-line processing is requested from the users. Typically, the transactions represent users' jobs submitted to remote computing locations and simply amount to small file transfers in both directions (typically a few thousands characters). Response time should be typically less than a minute (excluding obviously the time required to do the processing).

(4) Bulk file transfer requires highly reliable data transmissions no fast response time is required. These transmissions will be made outside the periods of peak traffic and will make use of ARQ error control schemes.

(5) Facsimile, audio and video conferencing require a lower quality level of transmission and no or very little error control. High rate FEC might be required for satellite stations on the fringe of the transmitter footprint. Response time will be around .25 sec (one hop-propagation delay).

5.1.3 Burstiness and Data Traffic Characterization

Data communications involves the transmission of information originating from many data processing applications (time-sharing, data-base inquiry response, teleprocessing, remote job entry, etc...) which may have large differences in their transmission characteristics and requirements. Most important not only the length of the messages could vary from a single byte (one character) to several thousands of bytes, data traffic has often a low duty cycle, that is, assumes a large peak-to-average ratio in data rates. In other words such data traffic has a "bursty" character which must be taken into consideration when analyzing and designing a digital communication system and data network. In particular for bursty data, packet switching is preferable to circuit switching for the data network.

In order to formalize the notion of burstiness, a measure called "bursty factor" has been defined [LAM 78]. The bursty factor provides bounds on the measures of peak-to-average ratio and duty cycle.

The bursty nature of a data traffic depends not only on the randomness in the generation of the messages, but also on the message lengths and delay constraints that have to be met. For example consider a source that generates 2000 bits messages every 10 seconds. If the message delay constraint is say, 10 seconds, then a 200 bits/s. channel can be used, and clearly over this channel the data will not look bursty at all. On the other hand suppose the message delay constraint were specified to be much shorter, say 0.2 second. Then a 2000/0.2 = 10 000 bits/s. channel must be used and over this channel the data will appear rather bursty with a peak to average ratio equal to 10 000/200 = 50, or equivalently with a 2% duty cycle. Therefore the concepts of peak-to-average ratio, duty cycle and message delay constraint are all related to the burstiness of the data.

As an alternative to the peak to average ratio (PAR) concept the "bursty factor" β of the traffic source is defined as [LAM 78]

$$\beta = D/T$$

where D is the specified average message delay constraint, and where T is the average interarrival time between messages. If the average message length is L then the channel transmission rate R_s must satisfy

$$R_s = L/D$$

Consequently the peak to average ratio PAR becoming

$$PAR = R_s/(L/T) \geq 1/\beta$$

and the duty cycle U satisfies

$$U = 1/PAR \leq \beta.$$

From these definitions we see that the bursty factor is an upper bound on the duty cycle, and the smoother and regular the traffic, the larger the bursty factor will be. From a data network point of view, for data sources with very small β (and hence even smaller duty cycle), message switching may be more attractive than circuit switching. On the other since the duty cycle is upper bounded by the bursty factor, data traffic sources with large a bursty factor will not necessarily use efficiently circuit switching. That is, for message switching the parameter of interest is the bursty factor, whereas for circuit switching it is the duty cycle.

5.2 Typical Traffic Volumes

Based on information which has been made available to us through the Agency, we have computed the data traffic requirements associated with some typical data transmission applications. The results of this investigation are reported in this section.

- APPLICATION 1:

Description: National Centralized Teleprocessing Application which implies a central node with a main frame in Hull connected through 13 I/O ports at 1200 bits/sec and the packet switching network DATA-PAC to 5 regional centres across the country. The application is geared at computer based real-time on-line order entry and retrieval

Hypothesis: Specific queries are assumed to be of 30 characters (on the average). Duty hours are based on 6 hours/day (average use).

-- Data traffic by regional centres:

ONTARIO: 16 collection nodes (300 bits/sec Asynchronous)

6,683 queries/day → 6,683 x 30	=	200,490 char/day
# of characters received in response	=	1,740,000 char/day
T O T A L	:	1,940,490 char/day

Estimated traffic (averaged) 90 char/sec

QUÉBEC: 12 collection nodes (300 bits/sec Asynchronous)

805 queries/day → 805 x 30	=	24,150 char/day
# of characters received in response	=	208,000 char/day
T O T A L	:	232,490 char/day
Estimated traffic (averaged)		90 char/sec

ATLANTIC: 10 collection nodes (300 bits/sec Asynchronous)

185 queries/day → 185 x 30	=	5,550 char/day
# of characters received in response	=	47,200 char/day
T O T A L	:	52,750 char/day
Estimated traffic (averaged)		25 char/sec

PRAIRIE: 8 collection nodes (300 bits/sec Asynchronous)

611 queries/day → 611 x 30	=	18,330 char/day
# of characters received in response	=	155,000 char/day
T O T A L	:	173,330 char/day
Estimated traffic (averaged)		8 char/sec

PACIFIC: 8 collection nodes (300 bits/sec Asynchronous)

1466 queries/day → 1466 x 30	=	43,980 char/day
# of characters received in response	=	379,000 char/day
T O T A L :		422,980 char/day
Estimated traffic (averaged)		20 char/day

Global traffic: the total estimated traffic (averaged) is then 131.5 char/sec. From the number of I/O ports involved and their maximum speed (300 bits/sec), it follows that the peak traffic that this application can handle is 54 (# of I/O ports) x 300 = 16,200 bits/sec whereas the average traffic based on 8 bits/char is only 1,052 bits/sec.

- APPLICATION 2:

Description: National Centralized Teleprocessing Application implying coast to coast on-line Time Shared used of a large scale computer and use of DATAPAC as the transport back-bone.

Hypothesis: The utilization of the system (average) is based on the following assumption; 22 working days (average) per month, 6 hours (average) per day.

-- Data traffic by regional centres:

ONTARIO:	transactions sent	=	17,796 Kchar/month
	transactions received	=	43,986 Kchar/month
	T O T A L :		61,782 Kchar/month

i.e. 130 char/sec in 26 I/O ports at 1200 bits/sec (asynchronous)

QUÉBEC:	transactions sent	=	18,383 Kchar/month
	transactions received	=	41,868 Kchar/month
	T O T A L :		60,252 Kchar/month

i.e. 127 char/sec in 26 I/O ports at 1200 bits/sec and 2 I/O ports at 300 bits/sec (asynchronous)

ATLANTIQUE:	transactions sent	=	12,887 Kchar/month
	transactions received	=	31,985 Kchar/month
	T O T A L :		44,872 Kchar/month

i.e. 94 char/sec in 23 I/O ports at 1200 bits/sec, 1 I/O port at 2400 bits/sec, 1 I/O port at 4800 bits/sec and 6 I/O ports at 300 bits/sec (asynchronous)

PRAIRIE:	transactions sent	=	19,054 Kchar/month
	transactions received	=	44,289 Kchar/month
		T O T A L :	63,343 Kchar/month

i.e. 133 char/sec in 30 I/O ports at 1200 bits/sec,
3 I/O ports at 300 bits/sec (asynchronous)

ONTARIO:	transactions sent	=	11,679 Kchar/month
	transactions received	=	27,241 Kchar/month
		T O T A L :	38,920 Kchar/month

i.e. 82 char/sec in 19 I/O ports at 1200 bits/sec
(asynchronous)

Global traffic: The total estimated traffic (averaged) is then 566 sec. From the number of I/O ports involved and their speed (124 ports at 1200 bits/sec, 11 ports at 300 bits/sec, 1 port at 2400 bits/sec, 1 port at 4800 bits/sec), it follows that the peak traffic that this application can handle is 124x1200+11x300+1x2400+1x4800 = 159,300 bits/sec, whereas the average traffic based on 8 bits/char is only 566 x 8 bits/sec = 4,528 bits/sec. As a concluding remark, it appears that the overall capacity allocated for this application is probably slightly oversized.

From these preliminary data, it seems sensible to set up the following assumptions. Based on five (5) applications of the types considered above, a typical traffic average will be of the order of 30

Kbits/sec. Allowing for a two-fold increase in the required channel capacity, this suggests the value of 60 Kbits/sec. The remaining capacity of 4 Kbits/sec which makes up the difference in a 64 Kbits/sec standard channel could then be used to accomodate the remaining slow speed traffic including Electronic Mail and Facsimile applications.

5.3 Classification of Suggested Types of Access for Typical Applications

As mentionned in the preceding section, it appears that an overall data traffic of 64 Kbits/sec should be sufficient to accomodate the current needs of the Agency. However, in order to take into account the possible future data traffic growth, increases by multiple of 64 Kbits/sec should be envisionned. As a general rule, five (5) regional centres (Ontario, Québec, Maritime, Prairie, Pacific) should be included as super nodes in the future network and could comprise a total of up to eight (8) or nine (9) earth stations (Toronto, Montréal, Ottawa, Halifax, St. John (New Foundland), Winnipeg, Calgary, Hamilton, Vancouver).

Our suggestions for the corresponding choices of access methods according to the types of traffic can be summarized as follows.

(1) For transactions of short duration and random arrival (i.e., bursty traffic) whose maximum volume of traffic should fit within two (2) 64 Kbits/sec channels, we recommend Packet Switching combined with some sort of Demand Access or Reservation scheme (i.e., one of the protocols discussed in sections 4.3 and 4.4).

Typical applications that will fit this framework are:

- Over the counter queries.
- Real time time sharing computing and applications.
- On line inquiry report systems.
- Communicating word processors.
- Electronic Mail.

(2) For transactions of long duration and involving a continuous stream traffic, once established, our recommended access method would be Demand Access SCPC and would require one (1) terminal SCPC or MCPC (for rates larger than 64 Kbits/sec) per earth station (i.e., a maximum of probably ten channels).

Typical applications that will fit this pattern are:

- Bulk file transfer.
- Telephone and Video conferencing.
- Facsimile.
- In general, dialed transactions lasting over one minute and whose traffic requests are essentially timely and geographically distributed and independent.

(3) For continuously dense and regular traffic, the best approach is certainly fixed assignment Light Route TDMA with as many full duplex channels as connected pairs of nodes. A variant of this scheme could be Fixed Assignment SCPC which probably would be cheaper to implement at the present time.

5.4 Conclusions

In this report we have considered the basic issues associated with the implementation of a data communication network designed to fit the needs of the Government Telecommunication Agency and based on the use of a unique satellite channel bandwidth and power. The main objective of this study has been to present the basic available methods of access for different users of the satellite channel as well as the implications which result from such a use of this channel. More specifically, we have concentrated simultaneously on the classical techniques which are based on the use of frequency, time and code division access methods either on a fixed or demand assignment basis as well as the more recent schemes which use random or partially random access methods. Our final suggestions are in the form of several scenarios pertaining to the particular applications suggested by the Agency. Since no quantitative analysis has been carried out in the course of this contract, our final conclusion will be to recommend that further studies are carried out by which a more quantitative approach can be applied in order to rank order the remaining alternatives so as to suggest a final choice.

Further fundamental issues pertaining to the use of a satellite channel are concerned with the overall integrity associated with the transmitted information. In this respect, two appendices present a comprehensive account of the methods available and deemed suitable, on the one hand, to alleviate the problems of control of the transmission errors and, on the other, to maintain the required level of confidentiality within the system.

REFERENCES

[CCIT 80] International Telegraph and Telephone Consultative Committee (CCITT), Yelow Book, ITU Edition, Geneva, 1980.

[CCIT 81] CCITT Study Group XVIII, "Report of the Meeting of Working Party XVIII/3: Network Performance Objectives", CCITT Contribution COM XVIII, No. R5, Geneva, July 1981.

[DECI 82] Decina, M. and de Julio, U., "Performance of Integrated Digital Networks: International Standards", Conference Record, IEEE Int. Conf. on Com., pp. 2D.1.1-2D.1.6, 1982.

[GRUB 82] Gruber, J., R. Vickers and Cuddy, D., "Impact of Errors and Slips on Voice Service", Conference REcords, IEEE Int. Conf. on Com., pp. 2D.5.1-2D.5.7, 1982.

[LAM 78] Lam, S., "A New Measure for Characterizing Data Traffic", IEEE Trans. on Com., Vol. COM-26, pp. 137-140, Jan. 1978.

[SAST 84] Sastry, A.R.K., "Performance Objectives for ISDN's", IEEE Com. Magazine, Vol. 22, Jan. 1984.

[URIE 82] Urien, M. and Rault, M., "Errors and Jitter Performance of Digital Network", Conference Record, IEEE Int. Conf. on Com., pp. 2D.4.1-2D4.5, 1982.

APPENDIX A

ERROR CONTROL PROCEDURES

In any network related data transmission system, information originating at a source must be transmitted to a distant user through a noisy channel. Because of the fact that channel noise cannot be eliminated, no matter how sophisticated the modulation scheme can be, the transmitted signals do not arrive at the receiver exactly as transmitted and hence errors are made in conveying the information to the user. The required level of performance of a digital communication system depends greatly on the type of information it is designed to handle. For analog related data resulting from analog to digital conversion of voice or video signals, the level of performance is usually measured in terms of the Bit Error Rate (BER) which might be required to be less than 10^{-3}. However, in certain other situations such as computer to computer communication, where the performance requirement might be as high as not to miss more than one block out of 10^{10}, only schemes based on the use of an error detection and repetition procedure (so called ARQ scheme) can provide a sufficient level of protection in most situations. Additional constraints on the data may be such as not to tolerate variable delays during transmission. As a consequence, the users' requirements on data accuracy, rate and delays together with the transmission equipment own parameters such as transmitted power and bandwidth are key factors to be considered in the implementation of any error control procedure. From the standpoint of transmission through a satellite channel, a corresponding first approximation model for the transmission channel can be chosen as a discrete memoryless additive gaussian noise channel with a propagation delay (one hop) of slightly

over 250 msec. Assuming a binary input/output alphabet, the corresponding cross-over probability (Bit Error Rate) p is typically smaller than 10^{-3} and the model is the so called classical Binary Symmetrical Channel (BSC) as depicted in Figure A.1. When the cardinality q of the input/output alphabet is greater than two, typically a power of a prime number and, furthermore, first the alphabet is endowed with the structure of an additive group and, secondly, the noise is non signal dependent; the BSC generalizes to the q-SC (q-ary Symmetrical Channel). Finally, in certain applications where the decisions made by the demodulator are softened so as to take into account the likelihood associated with the demodulated symbol, the cardinality of the output alphabet is larger than the one associated with the input set leading, whenever the noise affects the symbols independently from one to the other, to the general Discrete Memoryless Channel (DMC) model. From a general standpoint, error control techniques can be divided into two main classes, namely Forward Error Control (FEC) and Automatic Repeat upon reQuest (ARQ) procedures. Furthermore, some hybrid schemes combining advantages of one particular technique so as to compensate for the disadvantages of the other have also been recently introduced. In this appendix we will review these concepts in the context of the satellite channel. The overall organization of the material will be as follows. First the basic properties of block codes will be considered so as to introduce the fundamental class of cyclic codes and the important subclass comprising the BCH and Reed-Solomon codes. Secondly, convolutional codes will be considered with a special emphasis on decoding techniques such as the Viterbi and Jelinek/Zigangirov algorithms. Finally, ARQ procedures will be analysed and the subject material will conclude on some remarks concerning the level of error performance to be expected from using any of the particular schemes.

A.1 Block Codes

In block coding, the information sequence which is to be protected against errors is first broken down into blocks of length k which are then mapped through a one-to-one relation into corresponding blocks of n channel symbols. If the input and output alphabets are identical (i.e., binary), the dimensionless quantity R = k/n measures the fraction of information symbol per channel symbol and we speak of an (n,k) block code such as depicted in Figure A.2. An example of a (4,2) block code over the binary alphabet is given in Figure A.3. It is readily seen that for this code an input sequence of 011110... will result into the encoded sequence 010100001111... Whenever errors occur during transmission, the received block contains different symbols at the position where those errors were introduced. If $\underline{r} = (r_1, r_2, \ldots, r_n)$ represents the output from the receiver corresponding to the input $\underline{c} = (c_1, c_2, \ldots, c_n)$, an error is said to have occured in position j provided r_j is different from c_j. The Hamming distance between $\underline{r}$ and $\underline{c}$ denoted by $d_H(\underline{r},\underline{c})$ is defined as the number of positions where $\underline{r}$ and $\underline{c}$ differ. Whenever the input/output alphabets are identical and endowed with the structure of a group, it is readily seen that, if we define the error vector $\underline{e} = \underline{c}-\underline{r}$, then $d_H(\underline{r},\underline{c}) = W_H(\underline{e})$ where $W_H(\underline{e})$ represents the Hamming weight of $\underline{e}$ (i.e., the number of positions where $\underline{e}$ is different from 0).

The minimum Hamming distance of the code $V = \{\underline{y}_1, \underline{y}_2, \ldots, \underline{y}_M\}$ is

$$d = \min_{\underline{y}_i \neq \underline{y}_j} d_H(\underline{y}_i, \underline{y}_j).$$

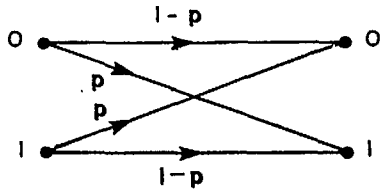


FIGURE A1 : BINARY SYMMETRIC CHANNEL

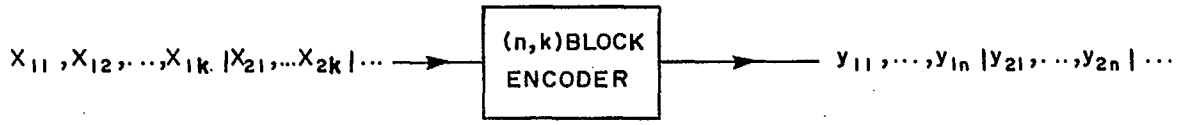


FIGURE A2 : BLOCK ENCODER

information 2-tuple	Codeword
00	1010
01	0101
10	1111
11	0000

FIGURE A3 : (4,2) BLOCK CODE

When a code C is used to communicate over a DMC having the same input and output alphabets, the code can correct any pattern of t or fewer errors and detect any pattern of t+s or fewer errors provided

$$d \geq 2t+s.$$

The most straightforward way to decode $\underline{r}$ is to compute $d_H(\underline{r},\underline{y}_i)$ $i = 1, 2,\ldots,M$ and choose that i which gives the minimum. Obviously, this procedure is only practical for small values of M.

In order to obtain codes with practical decoding procedures, it is necessary to impose some algebraic structure on the code itself. The basic properties of these structures are reviewed in the next section.

A.1.1 Algebraic Primer

A field is a non-empty set F with two binary operations + and . such that:

- $a+(b+c) = (a+b)+c$
- $a+b = b+a$
- There exists an element 0 such that $a+0 = 0+a$ for every $a \varepsilon F$
- For every $a \varepsilon F$ there exists an element $-a$ such that $a+(-a) = (-a)+a = 0$
- $a.(b.c) = (a.b).c$
- $a.b = b.a$
- There exists an element 1 such that $a.1 = 1.a = a$ for every a different from 0 in F
- For every a different from 0 in F there exists an element a^{-1} in F such that $a.a^{-1} = a^{-1}.a = 1$.

Familiar examples of fields are the fields of rational numbers, the field of real numbers, the field of complex numbers and the field of rational functions.

A field having only a finite number of elements is called a Galois Field and denoted GF(q), where q is the number of elements in the field. It may be shown that q must be a power of a prime and that for any such power there exists such a field. Hence there are fields of size 2, 4, 5, etc... but no fields of size 6. An integer m modulo a positive integer n is the remainder of m upon division by n (e.g., 3 modulo 2 is 1). The set of integers modulo a prime number p is a field. The smallest such field is GF(2) = {0,1} where 1 + 0 = 0 + 1 = 1, 0 + 0 = 1 + 1 = 0, and 1.1 = 1. The next largest field is GF(3) = {0,1,2} and its addition and multiplication tables appear in Figure A.4. A polynomial f(x) with coefficients in GF(p) is irreducible over GF(p) if the decomposition of f(x) as f(x) = a(x)b(x), where a(x),b(x) are polynomials over GF(p) implies that a(x) or b(x) is a constant. For example $1+x+x^2$ is irreducible over GF(2). For any prime p and any integer m > 0, there exists an irreducible polynomial f(x) over GF(p) of degree m. A polynomial a(x) modulo a polynomial b(x) is the remainder of a(x) upon division by b(x). The set of polynomials over GF(p) modulo an irreducible polynomial f(x) of degree m over GF(p) is a field of size p^m. Using therefore $f(x) = 1+x+x^2$ we can construct the field GF(4). Its elements are {0,1,x,1+x} and the addition and multiplication tables for GF(4) are as represented in Figure A.5. Operations in a field $GF(p^m)$ are in fact operations modulo a polynomial and it is possible to devise efficient circuits to carry out these operations [PETE 61].

If α (different from 0) is an element in GF(q), then the order of α is the smallest positive integer r such that $\alpha^r = 1$. It may

be shown that r divides q-1. There always exists an element α of order q-1 called a primitive element in which case $GF(q)= \{0,1,\alpha,\alpha^2,\ldots\alpha^{q-2}\}$. If $\alpha \in GF(p^m)$, then the polynomial f(x) over GF(p) of least degree such that $f(\alpha) = 0$ is called the minimal polynomial of α. A polynomial over GF(q) of degree r is primitive if it is irreducible and if it does not divide x^j-1 for $0<j<q^r-1$ (it is known to divide $x^{q^r-1}-1$). For example $1+x+x^2$ is a primitive polynomial over GF(2). If we use a primitive polynomial over GF(p) of degree m to construct $GF(p^m)$, then x will be a primitive element in $GF(p^m)$ and $GF(p^m)$ is precisely equal to the set $\{0,1,x,x^2,\ldots,x^{p^m-2}\}$. For further details on these matters see [PETE 61].

A group is a set V with a binary operation + such that:

- $a+(b+c) = (a+b)+c$
- There exists an element 0 in V such that $a+0 = 0+a = a$ for every a in V.
- For every a in V there exists an element -a such that $a+(-a) = (-a)+a = 0$.

If $a+b = b+a$ for every a and b in V then V is called an abelian (or commutative) group. A vector space W over a field F consists of an abelian group $\{W,+\}$ and a field $\{F,+,.\}$ such that for every vector v in W and for every scalar α in F there exists a vector αv such that:

- $\alpha(v+w) = \alpha v+\alpha w$
- $(\alpha+\beta)v = \alpha v+\beta v$
- $1.v = v$
- $(\alpha\beta)v = \alpha(\beta v)$

+	0	1	2
0	0	1	2
1	1	2	0
2	2	0	1

•	1	2
1	1	2
2	2	1

FIGURE A4 : MULTIPLICATION AND ADDITION TABLES FOR GF (3)

+	0	1	x	1+x
0	0	1	x	1+x
1	1	0	1+x	x
x	x	1+x	0	1
1+x	1+x	x	1	0

•	1	x	1+x
1	1	x	1+x
x	x	1+x	1
1+x	1+x	1	x

FIGURE A5 : ADDITION AND MULTIPLICATION TABLES FOR GF (4)

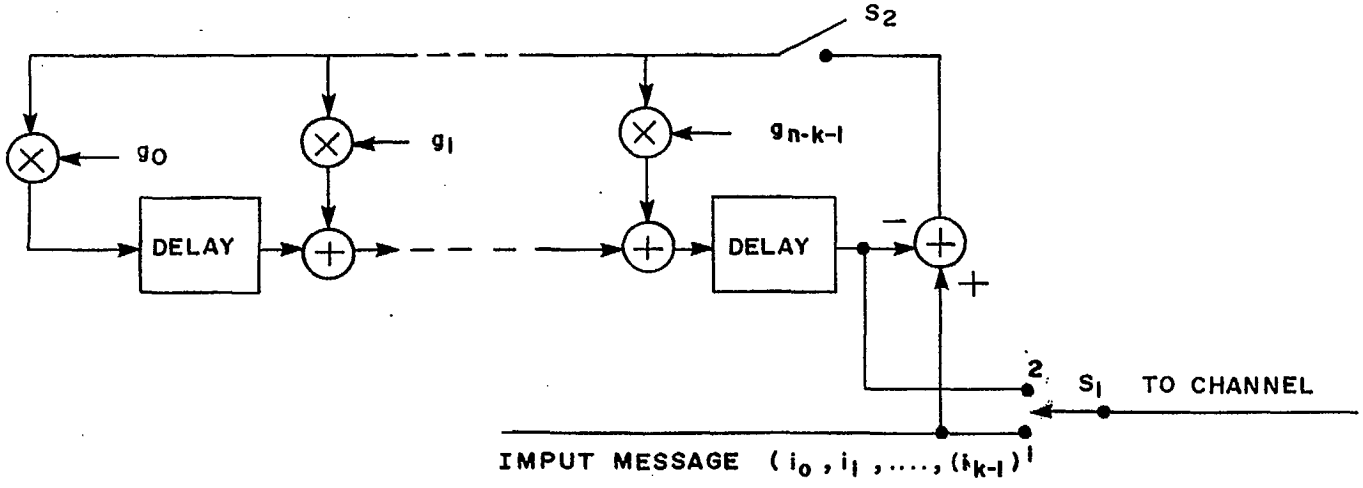


FIGURE A6: BASIC n-k CELLS SHIFT REGISTER ENCODING CIRCUIT FOR THE CYCLIC CODE WITH GENERATING POLYNOMIAL $g(x)=g_0+g_1x+\ldots+x^{n-k}$

⊕ GF (q) Adder

⊗ GF (q) Multiplier

If F is a field, $W = F^n = \{(\alpha_1,\alpha_2,\ldots,\alpha_n) \mid \alpha_i \varepsilon F\}$ is the set of all n-tuples over F and furthermore, if we define $\alpha(\alpha_1,\ldots,\alpha_n) = (\alpha\alpha_1,\alpha\alpha_2,\ldots,\alpha\alpha_n)$, then F^n is a vector space over F. A set of vectors $v_1,v_2,\ldots,v_n$ is <u>independent</u> if the identity $0 = \alpha_1 v_1 + \ldots + \alpha_n v_n$, $\alpha_i \varepsilon F$, implies that $\alpha_1 = \alpha_2 = \ldots = \alpha_n = 0$. An infinite set of vectors is independent provided all its finite subsets are independent. A set of vectors S <u>spans</u> W if every vector v in W can be expressed as a linear combination of vectors in S. A linearly independent set S which spans the vector space W is called a <u>basis</u> for W. A vector space W with a finite basis is called a finite dimensional vector space. The number of vectors in any basis is the same and is called the dimension of the space. For example $e_1 = (1,0,0,\ldots,0)$, $e_2 = (0,1,0,\ldots,0),\ldots, e_n = (0,0,\ldots,0,1)$ is a basis for F^n. A subset V of W is called a <u>subspace</u> if V is closed under addition and scalar multiplication. For example V = {(0000), (1111)} is a subspace of F^4 where F = GF(2). The dimension of V is 1. In general, the dimension of a subspace is less than or equal to the dimension of the containing space W.

A.1.2 Linear Codes

An (n,k) linear code over GF(q) is a k-dimensional subspace V of F^n. For example {(0000), (1010), (0101), (1111)} is a (4,2) binary linear code. If v is an (n,k) linear code over GF(q) and $g_1,g_2,\ldots,g_k$ represent a basis for V then:

$$G = \begin{vmatrix} g_1 \\ g_2 \\ . \\ . \\ g_k \end{vmatrix} \qquad (1.2.1)$$

is a generator matrix for the code. We also say that V is the row space of G. Conversely any full rank k by n matrix over GF(q) generates an (n,k,) linear code over GF(q). The codeword corresponding to the k-tuple x can be written as:

$$y = xG \tag{1.2.2}$$

As an example, the (4,2) binary code whose generating matrix is:

$$G = \begin{vmatrix} 1010 \\ 0101 \end{vmatrix}$$

comprises all the following 4-tuples (0000) = (00)G, (1010) = (10)G, (0101) = (01)G and (1111) = (11)G.

Note that an (n,k) linear code has many generator matrices in correspondence with all the different bases. An (n,k) linear code is furthermore called systematic if it has a generator matrix of the form

$$G = \{I_k \mid P\}, \tag{1.2.3}$$

where I_k = diagonal (1 1 ... 1) and P is a k by n-k matrix. When using such a G, x is encoded into

$$y = xG = (x,xP),$$

and the first k-tuple represents the unchanged information symbols while xP is referred to as the added n-k parity check. In practice, it can be desirable, although not necessary, to use a generator matrix in the systematic form. Two (n,k) linear codes V and V^1 are equivalent if

the words of V^1 can be obtained by permuting the coordinates of the words in V. For example the codes V = {(0000), (1010), (0101), (1111)} and V^1 = {(0000), (1100), (0011), (1111)} are equivalent. It is an easy matter to show that any linear code is equivalent to a systematic linear code. Furthermore, on a Discrete Memoryless channel, two equivalent codes will have the same probability of error. Such a statement is not necessarily true for decoders which are not maximum likelihood (or minimum distance) decoders.

If V is an (n,k) linear code, then the words of V can be viewed as the set of solutions of an homogeneous system of (n-k) linear equations of full rank in n unknowns. In matrix notation, this means that there exists an (n-k)xn matrix H called a parity check matrix such that

$$V = \{y \mid yH^T = 0\}.$$

If G is in systemactic form, $G = [I_k | P]$, then it is easy to verify that H can be taken as:

$$H = [-P^T \mid I_{n-k}]. \qquad (1.2.4)$$

As an example, if

$$G = \left|\begin{matrix}1010\\0111\end{matrix}\right|,$$

then

$$H = \left|\begin{matrix}1110\\0101\end{matrix}\right|.$$

The matrix H is a (n-k,n) full rank matrix which can be used to generate an (n,n-k) block code V^D called the dual code of V and with the property that any codeword of V is orthogonal to any codeword of V^D and vice-versa.

A.1.3 Syndrome Decoding of Linear Block Codes

Let V be an (n,k) linear code over GF(q) used to communicate over a discrete memoryless channel with input and output alphabets GF(q). If v is transmitted and r received, then the error pattern e is defined as:

$$e = r-v$$

and it follows that

$$r = v+e.$$

If G is a generator matrix for V then there exists an n-k by n full rank matrix H over GF(q) such that:

$$GH^t = 0. \qquad (1.3.1)$$

A matrix which satisfies this property can be easily found by using the parity check matrix as defined in the previous section. It then follows that $v \in V$ if and only if

$$vH^t = 0. \qquad (1.3.2)$$

Hence if $r = v + e$ is received we can compute the syndrome rH^t as:

$$rH^t = vH^t + eH^t = eH^t \qquad (1.3.3)$$

which depends only on the error pattern. Of course all the n-tuples v+e, $v \in V$ have the same syndrome. Hence to do syndrome decoding we can for each of the q^{n-k} syndromes store the most probable error pattern having this syndrome and then use the syndrome as an address to do table look-up decoding.

- Example:

Let $q = 2$, $n = 7$, and $k = 4$ and let V be the (7,4) linear code with parity check matrix

$$H = \begin{vmatrix} 1010101 \\ 0110011 \\ 0001111 \end{vmatrix}.$$

A generator matrix can be easily obtained as:

$$G = \begin{vmatrix} 1110000 \\ 1001100 \\ 0101010 \\ 1101001 \end{vmatrix}$$

If $x = (1010)$ is the information then the corresponding codeword is (1011010). If $r = (1001010)$ is received then the syndrome is calculated to be:

$$s = rH^t = (110).$$

For this particular example, and because of the nature of H, it is apparent that if no more than one digit is in error, then the decimal equivalent (from left to eight) of the syndrome gives the position of

the error. The syndrome (110) corresponds to 3 indicating a single error in position 3. This example illustrates the simplicity of correcting a single error using the well known Hamming code of length n = 7.

In syndrome decoding, in general, we have to store $q^{n-k} = q^{n(1-R)}$ error patterns so that this technique is only practical for high rate codes. However syndrome decoding is the basic error detection technique in use since a non zero syndrome value indicates do the receiver that the received n-tuple is not a codeword and hence cannot have been sent by the transmitter.

Given a linear block code V, one of the main parameters of V is its minimum distance which is easily shown to be the minimum Hamming weight over the set of all non zero codewords, i.e.

$$d_{min} = \min_{y \neq 0} W_H(y) \qquad (1.3.4)$$

where $y \in V$ and y is different from 0.

The fundamental role played by d is brought up in the following fundamental theorem.

Theorem A.1.3.1:

An (n-k) linear block code of minimum distance d can correct up to t errors and detect t+1, t+2,...,t+s errors if

$$d_{min} > 2t+s.$$

The basic properties associated with syndrome decoding can be summarized in the following proposition:

Proposition A.1.3.2:

Any (n-k) linear code V can correct the set of error patterns E_c (which is assumed to contain the all 0 sequencel) and detect the set of errors E_d if the elements of E_c have distinct syndromes (i.e., they must belong to distinct cosets as formed by the set of n-tuples having identical syndromes) and the elements of E_d have syndromes distinct from that of E_c.

In view of the results of the preceding fundamental theorem, the statement in the foregoing proposition implies furthermore the following properties:

- all the n-tuples of Hamming weight $[(d-1)/2]$, where the notation $[x]$ represents the integer part of x, belong to one and only one coset;

- no n-tuple of Hamming weight $\geq d_{min}$ can have a non-zero syndrome.

A.1.4 Cyclic Codes

The cyclic codes form the most important class of codes discovered so far. A cyclic code (more precisely a linear cyclic code) of length n is an (n,k) linear code V over GF(q) such that if $\underline{v} = (v_1, v_2, \ldots, v_n) \in V$ then so does its cyclic shift $(v_n, v_1, v_2, \ldots, v_{n-1})$. For example {0000,1010,0101,1111} is a cyclic code of length 4. In treating cyclic codes, it is convenient ot consider n-tuples as polynomials rather than vectors. Hence we associate with the n-tuple $(v_0, v_1, \ldots, v_{n-1})$ the polynomial $v_0+v_1x + \ldots+v_{n-1}x^{n-1}$. The above code then would be $\{0, 1+x^2, x+x^3, 1+x+x^2+x^3\}$. If V is an (n,k) cyclic code over GF(q)

and if g(x) is the monic polynomial (the highest order coefficient is 1) of least degree in V then it can be shown that (see [PETER 61]:

- $V = \{a(x)g(x) \mid \text{degree } a(x) < k\}$
- g(x) divides the polynomial x^n-1.

The polynomial g(x) will have degree n-k and is unique. It is called the generator polynomial of V. Conversely, if g(x) is a monic polynomial of degree n-k which divides x^n-1 then V as defined by the multiples of g(x) of degree less than n is an (n,k) cyclic code with generator g(x). For the previous example $g(x) = 1+x^2$. With cyclic codes the encoding consists in multiplying the information polynomial i(x) by the generator polynomial g(x). This operation is most easily done with shift registers, multipliers and adders on the Galois field GF(q).

To illustrate, we consider the encoding of a cyclic code with generator polynomial

$$g(x) = g_0 + g_1x + \ldots + g_{n-k-1}x^{n-k-1} + x^{n-k}.$$

Let $i(x) = i_0+i_1x+\ldots+i_{k-1}x^{k-1}$ be the information polynomial, and consider $i(x)x^{n-k}$. Let q(x) be the quotient resulting from the division of $i(x)x^{n-k}$ by g(x) and r(x) be the rest no that r(x) is a polynomial of degree less than n-k. It follows that the polynomial:

$$i(x)x^{n-k} - r(x) = q(x)g(x) \qquad (1.4.1)$$

is a multiple of g(x) and hence is a codeword. Clearly the corresponding n-tuple can be written as

$$(-r_0,-r_1,\ldots,-r_{n-k-1},i_0,i_1,\ldots,i_{k-1})$$

and is in systematic form provided the high order coefficients are sent first on the channel. The basic circuitry for the implementation of the foregoing encoding procedure is represented in Figure A.6. The whole encoding process requires n clock strokes. For the first k cycles, the switch S_1 is in position 1 and the contact S_2 is closed resulting in the transmission of the information symbols over the channel as well as the computation of the n-k parity check symbols in the shift register. The last n-k cycles are used to send these symbols over the channel by switching S_1 over to position 2 and opening the contact S_2.

- Example A.1.4.1:

We consider the binary (7,3) cyclic code generated by $g(x) = 1+x+x^2+x^4$ whose corresponding encoding circuit is represented in Figure A.7. Suppose that we wish to encode the information sequence (101). As the information bits are shifted in the shift register sequentially, the contents of the memory cells evolve as follows:

	r_0	r_1	r_2	r_3	Input
• Initial state	0	0	0	0	1
• First shift	1	1	1	0	0
• Second shift	0	1	1	1	1
• Third shift	0	0	1	1	

The corresponding encoded codeword is then (0011101).

Since g(x) is a factor of x^n-1, it follows that the polynomial $h(x) \triangleq (x^n-1)/g(x)$ which is of degree k and can be written as

$$h(x) = h_0 + h_1x + \ldots + h_{k-1}x^{k-1} + x^k$$

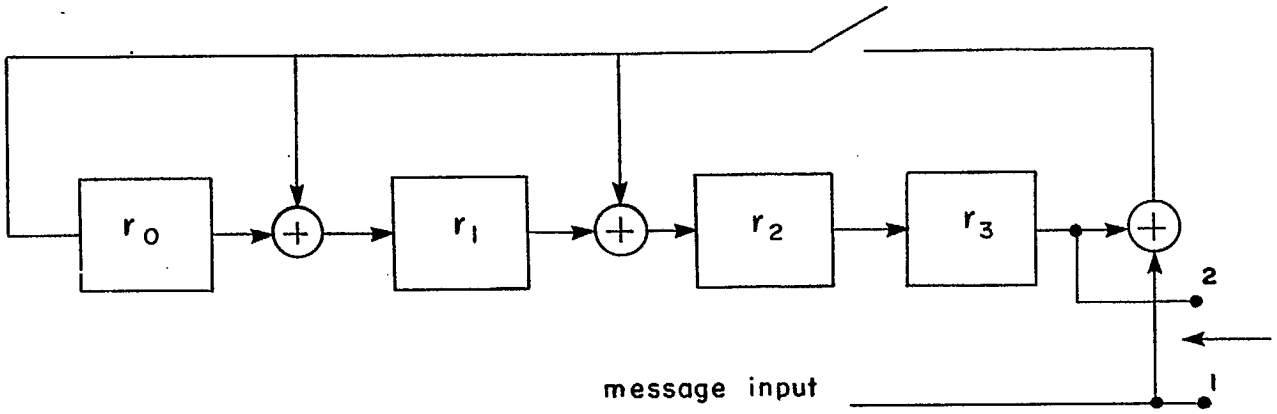


FIGURE A7 : ENCODER FOR THE CODE OF EXAMPLE A.4.1

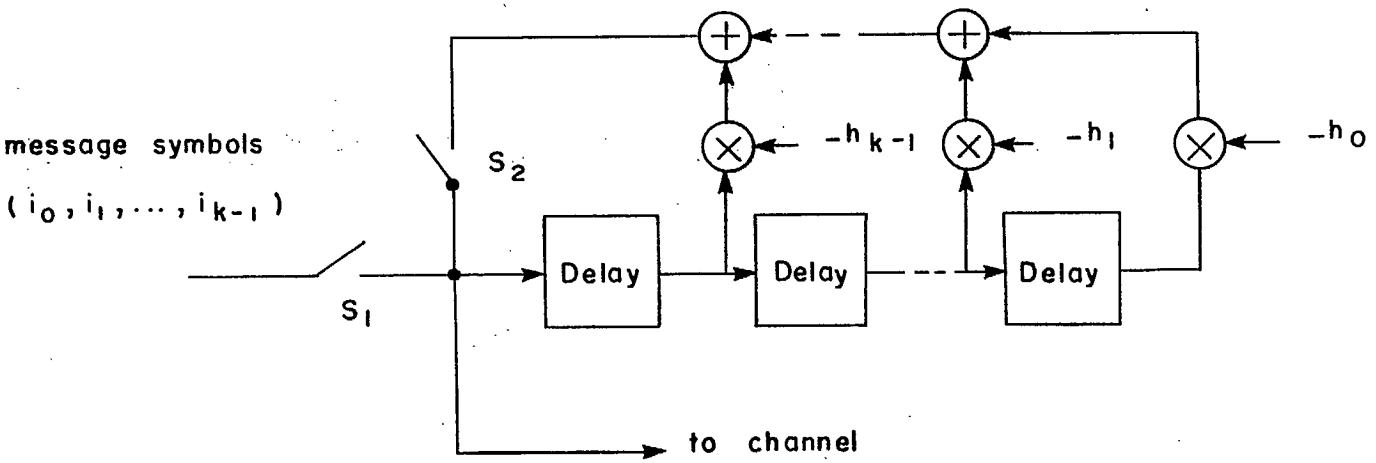


FIGURE A8 : BASIC k CELLS SHIFT REGISTER ENCODING CIRCUIT FOR A CYCLIC CODE WITH PARITY CHECK POLYNOMIAL

$h(x) = h_0 + h_1 x \ldots + x^k$

⊕ GF (q) Adder

⊗ GF (q) Multiplier

satisfies for every codeword c(x) in V to the relation $c(x)h(x) = 0 \bmod(x^n-1)$. If we write $c(x) = c_0+c_1x+\ldots c_{n-1}x^{n-1}$, it follows that the previous identity implies the following relationship between the n-tuple $(c_0,c_1,\ldots,c_{n-1})$ and the coefficients $(h_0,h_1,\ldots,h_{k-1})$

$$c_{n-k-j} = -\sum_{i=0}^{k-1} h_i\, c_{n-j-i} \qquad j = 1,2,\ldots,n-k. \qquad (1.4.2)$$

As a consequence provided the first k symbols of the codeword c are given as $c_{n-1} = i_{k-1}, c_{n-2} = i_{k-2}, \ldots, c_{n-k} = i_0$, the previous recurrence relations allow for the computation of the remaining symbols. This computational procedure is furthermore easily mechanized by using a k cells shift register as represented in Figure A.8. With S_1 closed and S_2 open, the k information symbols are sent over the channel and fed into the shift register. The contacts S_1 and S_2 are then respectively closed and open for the remaining n-k clock strokes while the parity symbols are computed according to the relation 1.4.2 and passed on to the channel.

- Example A.1.4.2:

Referring back to the code used in example A.1.4.1, the polynomial h(x) is easily found to be $h(x) = (x^7+1)/(x^4+x^2+x+1) = x^3+x+1$ resulting in the encoding circuit of Figure A.9. If it is desired to encode the message (101), the parity symbols are computed in four cycles as follows:

	P_0	P_1	P_2	Parity symbol
• Initial state	1	0	1	
• First cycle	1	1	0	1
• Second cycle	1	1	1	1
• Third cycle	0	1	1	0
• Fourth cycle	0	0	1	0

resulting, as required, in the encoded codeword (0011101).

The two basic circuits which have been considered for the encoding of a cyclic code require an n-k shift register and at most n-k Galois Field multipliers and adders for the one based on the use of the generating polynomial while a k shift register and no more than k multipliers and adders are needed when the parity chech polynomial is used. For high rate codes (i.e., the case $R > \frac{1}{2}$) the circuit based on the generating polynomial is obviously the most economical. Furthermore, it has the basic advantage of allowing for an easy implementation of the syndrome former (the basic ingredient of the decoder) at the receiving end.

At the decoder, it is indeed fairly easy to check whether or not a detectable error has occured, namely we divide the received polynomial $r(x) = c(x) + e(x)$ by $g(x)$ to obtain the remainder $s(x)$. If $s(x) = 0$ we assume that no errors have occured since the received n-tuple is a codeword; otherwise a detectable error has occured. Based on the foregoing discussion around the basic circuit of Figure A.6, it has been established that the content of the shift register is the rest of the division of $x^{n-k}i(x)$ by $g(x)$. As a consequence, the similar circuit which appears in figure A.10 computes the rest of the division of $r(x)$ (the received n-tuple) by $g(x)$, i.e., computes the syndrome, once the n received symbols have been shifted into the register.

An alternate description of cyclic codes in terms of the roots of the generating polynomial is fundamental in the construction of the BCH and Reed-Solomon codes to be considered in the next section. We now explicitly assume that n, the block length, and q, the alphabet size, are relatively prime. For $q = 2$, this means that n is odd.

Let then be an (n,k) cyclic code, n and q relatively prime, and let $g(x)$ be the generator of V. Now there exists a field F which

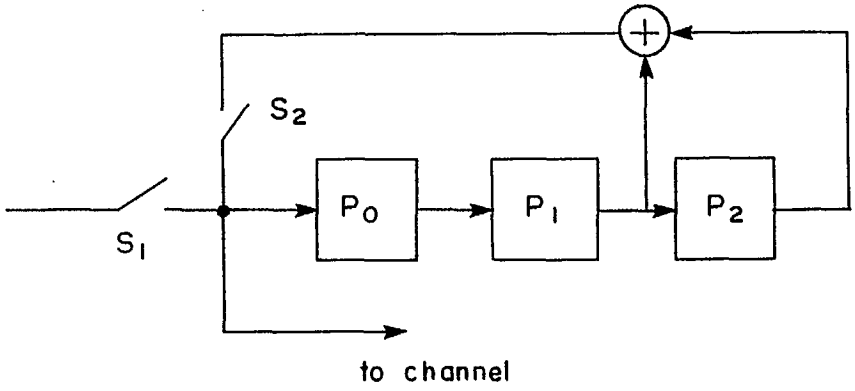


FIGURE A.9: ENCODER FOR THE CODE OF EXAMPLE A.4.2

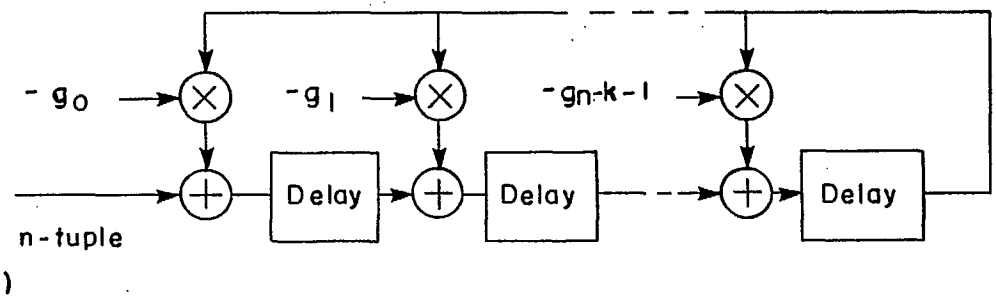


FIGURE A.10: BASIC SYNDROME FORMER FOR THE CYCLIC (n-k) CODE WITH GENERATING POLYNOMIAL $g(x) = g_0 + g_1 x + \ldots + x^{n-k}$

⊕ GF (q) Adder

⊗ GF (q) Multiplier

contains GF(q) and which contains the roots $\alpha_1,\alpha_2,\ldots,\alpha_r$, r = n-k, of g(x). Since n and q are relatively prime then g(x) has no repeated roots [BERL 68]. It now can easily be shown that $F(x) = f_0+f_1x+\ldots+f_{n-1}x^{n-1}$ belongs to V if and only if $\alpha_1,\alpha_2,\ldots,\alpha_r$ are roots of f(x). Conversely if $\alpha_1,\alpha_2,\ldots,\alpha_r$ are elements in an extension field of GF(q), then the polynomial of least degree over GF(q) having $\alpha_1,\alpha_2,\ldots,\alpha_r$ as roots will generate a cyclic code of some length n. The parity check matrix of a cyclic code can be easily expressed in terms of the r roots of g(x) $\alpha_1,\alpha_2,\ldots,\alpha_r$ in a suitable extension field $GF(q^m)$ of GF(q) as the rxn matrix

$$H = \begin{vmatrix} 1 & \alpha_1 & \alpha_1^2 \ldots \alpha_1^{n-1} \\ 1 & \alpha_2 & \alpha_2^2 \ldots \alpha_2^{n-1} \\ \cdot & & \\ \cdot & & \\ \cdot & & \\ 1 & \alpha_r & \alpha_r^2 \ldots \alpha_r^{n-1} \end{vmatrix} \qquad (1.4.3)$$

- Example A.1.4.3:

In GF(5), the polynomial $g(x) = (x-2)(x-4) = x^2+4x+3$ divides x^4-1 and generates a (4,2) cyclic code over GF(5). The roots of g(x) belong to GF(5) and are $\alpha_1 = 2$, $\alpha_2 = \alpha_1^2 = 4$ so that we can take as parity check matrix the 2x4 matrix of elements in GF(5).

$$H = \begin{vmatrix} 1 & 2 & 4 & 3 \\ 1 & 4 & 1 & 4 \end{vmatrix}$$

A.1.5 BCH and Reed-Soloman (R-S) Codes

The most important class of cyclic codes, from the standpoint of applications, are BCH codes discovered by Bose and Ray-Chaudhuri in 1960 and independently by Hocquenghen in 1959. To construct a BCH code of length n over GF(q) (with n and q relatively prime) and designed distance d we proceed as follows. Firs let m be the smallest positive integer such that n divides q^m-1 and let β be an element in GF(q^m) of order n. Let g(x) be the least common multiple of the minimum polynomials over GF(q) of $\beta,\beta^2,\ldots,\beta^{d-1}$; then g(x) will divide x^{n-1} and will therefore generate a cyclic code V. The minimum distance of V will be a least d. For a proof see [MCWI 78] or [BERL 68].

- Example A.1.5.1:

We will now construct a binary BCH code of length 15. Since $n = 15 = 2^4-1$ and $q = 2$, it follows that $m = 4$. Hence β is an element in GF(16) of order $n = 15$, i.e., a primitive element. It is known that $1+x+x^4$ is a primitive polynomial of degree 4 hence using it to construct GF(16) we may let β be one of its roots. The minimum polynomial of β is consequently $m_\beta(x) = 1+x+x^4$. Let $d = 5$ which implies that we need to find the minimal polynomial of β,β^2,β^3 and β^4. The elements β^2,β^4 have the same minimal polynomial as β. The minimal polynomial of β^3 is $(x-\beta^3)(x-\beta^6)(x-\beta^{12})(x-\beta^9)$ which may be computed (using the recursion $\beta^4 = 1+\beta$) to be $1+x+x^2+x^3+x^4$; hence $g(x) = (1+x+x^4)(1+x+x^2+x^3+x^4)$ which therefore generates a (15,7) binary cyclic code of minimum distance at least 5 (hence this code can correct any pattern of 2 or fewer errors).

The BCH codes have a practical decoding algorithm discovered by Berlekamp. The derivation of the properties of this algorithm is rather involved and, for further details, the interested reader is referred to Massey's paper [MASS 69] or the encyclopedic work of Mac Williams and Sloane [WCWI 78].

The basic step of the Berlekamp algorithms is the linear Feedback Shift register (LFSR) synthesis algorithm for finding the shortest linear feedback shift register $\{C(x),L\}$ that generates a sequence $s_0,s_1,\ldots,s_{n-1}$ of elements of GF(q). To illustrate, we consider first the sequence generated by the linear feedback shift register of length L and connection polynomial

$$C(D) = 1 + c_1 D + \ldots + c_{L-1} D^{L-1} + c_L D^L.$$

with D the unit delay operator, and represented on Figure A.11.

If $s_0,s_1,\ldots,s_{L-1}$ are given elements of GF(q), the circuit of Figure A.11 generates the unique sequence with Huffman D-transform

$$S(D) = s_0 + s_1 D + \ldots + s_j D^j \qquad (1.5.1)$$

whose elements satisfy the relation

$$s_{j+L} = - \sum_{m=1}^{L} c_m s^{j+L-m} \qquad j = 0,1,2,\ldots \qquad (1.5.2)$$

By multiplying (1.5.2) by D^{j+L}, summing over all j and using definition (1.5.1), the following identity is easily arrived at

$$S(D) = \frac{P(D)}{C(D)} \qquad (1.5.3)$$

with P(D) a polynomial of degree at most L-1 and whose coefficients are given in terms of the initial values $s_0, s_1, \ldots, s_{L-1}$ by:

$$P_i = \sum_{m=0}^{i} c_{i-m}\, s_m \qquad i = 0,1,\ldots,L-1. \tag{1.5.4}$$

With these preliminary remarks in mind, the LFSR synthesis algorithm of Berlekamp and Massey finds the smallest possible value of L together with an associated connection polynomial C(D) such that the output from the corresponding circuit of Figure A.11 matches a given sequence $s_0, s_1, \ldots s_n$ for $n = 0,1,2,\ldots$ The procedure can be outlined as follows:

- LFSR Synthesis for $\{s_0, s_1, \ldots, s_{N-1}\}$

Step 0: {Initialisation}

$C(D) \leftarrow 1$, $B(D) \leftarrow 1$, $L \leftarrow 0$, $b \leftarrow 1$, $x \leftarrow 1$, $n \leftarrow 0$.

Step 1: {Compute the discrepancy between the sequence and the output from the present LFSR}.

$$d \leftarrow s_n + \sum_{i=1}^{L} c_i s_{n-i}.$$

If $(d = 0)$, {no change needed}, $x \leftarrow x+1$, goto Step 2.
Else, {changes must be made},

If $(2L > n)$, {only the connection polynomial needs be changed}
$C(D) \leftarrow C(D) - d.b^{-1}.D^{x}.B(D)$,
$x \leftarrow x+1$, goto Step 2.

Else, {the length and connection polynomial of the shift register need be changed}
$T(D) \leftarrow C(D)$,
$C(D) \leftarrow C(D) - d.b^{-1}.D^{x}.B(D)$,
$B(D) \leftarrow T(D)$,
$L \leftarrow n+1-L, x \leftarrow 1, b \leftarrow d$.

Step 2: $n \leftarrow n+1$ {next iteration}, if $n = N$ Stop.

Else, goto Step 1.

NOTE: In general, several connection polynomials can be found as solutions of the LFSR synthesis algorithm. However, whenever $L \leq N/2$ it turns out that the solution is unique. Furthermore, the algorithm finds such a solution after the processing of at most 2L digits.

- Example A.1.5.1:

We consider GF(7) and the sequence $\underline{S} = (0,2,5,0,\ldots)$. The application of the LFSR algorithm yields in tabular form.

n	L	C(D)	x	B(D)	b	s_n	d
0	0	1	1	1	1	0	0
1	0	1	2	1	1	2	2
2	2	$1+5D^2$	1	1	2	5	5
3	2	$1+D+5D^2$	2	1	2	0	1
4	2	$1+D+D^2$	3	1	2	-	-

Furthermore, using the relations (1.5.4), it is readily found that $P(D) = 2D$.

Switching back to the decoding of BCH codes: let $\alpha, \alpha^2, \ldots \alpha^{2t}$ (i.e, t is the designed error correction capability of the code) be the roots of the generating polynomial where α is some n^{th} primitive root of unity. Assume that $v(x)$ represents the polynomial of degree n-1 associated with the transmitted codeword and that $y(x) = v(x)+e(x)$ is received, where $e(x)$ is the error vector

$$e(x) = \sum_{i=0}^{n-1} e_i x^{n-i-1}. \qquad (1.5.5)$$

Let us assume that there were τ actual errors, i.e., $e_{i_1}, e_{i_2}, \ldots, e_{i_\tau}$ are the only non zero elements of $e(x)$ (the random errors introduced during the transmission over the channel).

The corresponding syndrome is defined as the vector $\underline{S} = (S_1, S_2, \ldots, S_{2t})$ such that:

$$S_j \triangleq y(\alpha^j) = v(\alpha^j) + e(\alpha^j) = e(\alpha^j) \quad j = 1,2,\ldots,2t \qquad (1.5.6)$$

where the last identity proceeds from the fact that $v(\alpha^j) = 0$ for $j = 1,2,\ldots,2t$ since $v(x)$ is a multiple of the generating polynomial whose roots are α^j, $j = 1,2,\ldots,2t$. Using (1.5.5), (1.5.6) can be rewriten as:

$$S_j = \sum_{m=1}^{\tau} e_{i_m} (\alpha^j)^{n-i_m-1} \quad j = 1,2,\ldots,2t \qquad (1.5.7)$$

by multiplying S_j by D^{j-1} and summing over j, we furthermore get:

$$S(D) \triangleq \sum_{j=1}^{2t} S_j D^{j-1} = \sum_{m=1}^{\tau} e_{i_m} \alpha^{n-i_m-1} \sum_{j=0}^{2t-1} (\alpha^{n-i_m-1}.D)^j \qquad (1.5.8)$$

Using the fact that $\alpha^n = 1$ and $\sum_{j=0}^{2t-1} x^j = (1-x^{2t})/(1-x)$, (1.5.8) can be rewritten as:

$$S(D) = \sum_{m=1}^{\tau} \frac{\alpha^{-i_m-1} e_{i_m}}{(1-D\alpha^{-i_m-1})} \bmod D^{2t}. \qquad (1.5.9)$$

As a consequence and provided that at most t errors occured during the transmission, α^{i_m+1}, $1 \leq m \leq t$, are the <u>roots</u> of the <u>unique</u> connection polynomial C(D) which realizes the first 2t terms of the syndrome sequence S(D). Furthermore, the values of the corresponding errors are easily computed from the corresponding numerator polynomial P(D) through the identities

$$e_{i_m} = \frac{P(\alpha^{i_m+1})}{\alpha^{-i_m-1}} \left[\frac{1-D\alpha^{-1_m-1}}{C(D)}\right] \qquad \begin{matrix} m = 1,2,\ldots,t \\ D = \alpha^{i_m-1} \end{matrix} \qquad (1.5.10)$$

From the relation (1.5.9), it is readily seen that $C(D) = \prod_{m=1}^{\tau} (1-D\alpha^{-i_m-1})$. As a consequence, by taking the formal derivative of C(D) with respect to D, we have:

$$C'(D) = \sum_{m=1}^{t} \alpha^{-i_m-1} \prod_{\substack{j=1 \\ j \neq m}}^{t} (1-D\alpha^{-i_j-1})$$

which in turns implies

$$C'(\alpha^{i_m+1}) = -\alpha^{-i_m-1} \cdot \left[\frac{C(D)}{1-D\alpha^{-i_m-1}}\right]_{D=\alpha^{i_m+1}}$$

Substituting back in (1.5.10) yields

$$e_{i_m} = -\frac{P(\alpha^{i_m+1})}{C'(\alpha^{i_m+1})}. \qquad (1.5.11)$$

An easy procedure for finding the roots of C(D) is to substitute $\alpha, \alpha^2, \ldots \alpha^n$ in turn into C(D). If α^m is a root, an error has occured in the symbol labelled n-m and the corresponding error magnitude e_{i_m} is easily found through the use of relation (1.5.11). Since the high order symbols of any codeword are assumed to be received first, decoding can take place in real time by shifting the symbols one at a time and correcting them by checking if the corresponding power of α is a root (so called Chien searcher). In the affirmative, the error magnitude is evaluated and substracted from the received codeword. The coresponding procedure is schematically outlined in Figure A.12. It comprises three basic components.

- A syndrome computer which evaluates from the received codeword r(x) the coefficients $S_j = r(\alpha^j)$ $j = 1,2,\ldots,2t$.

- A Berlekamp-Massey LFSR synthesis algorithm implementation which computes C(D), the error location polynomial from the sequence S_j. $j = 1,2,\ldots,2t$.

- A Chien searcher combined with an error magnitude computer for the error localisation and correction.

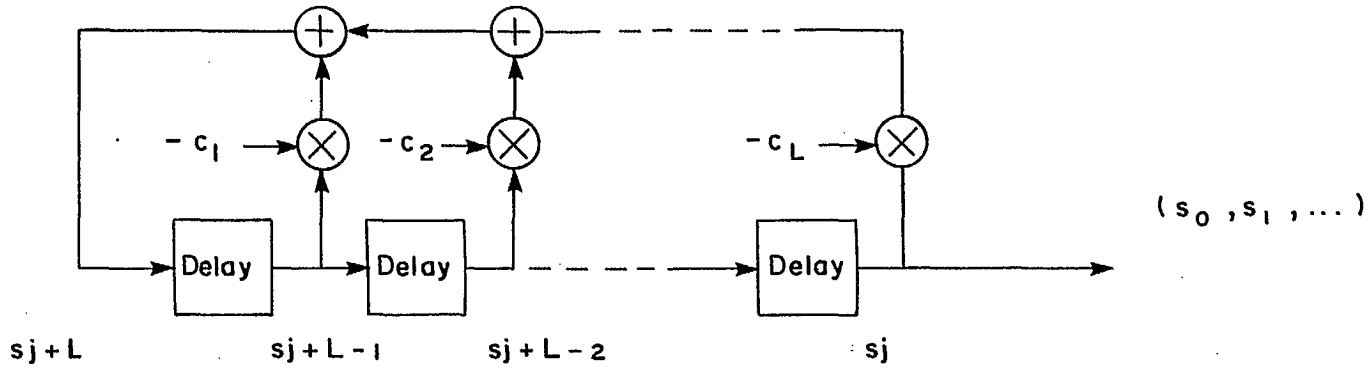


FIGURE A.11 : LINEAR FEEDBACK SHIFT REGISTER WITH CONNECTION POLYNOMIAL $1 + c_1 D + \ldots + c_L D^L$

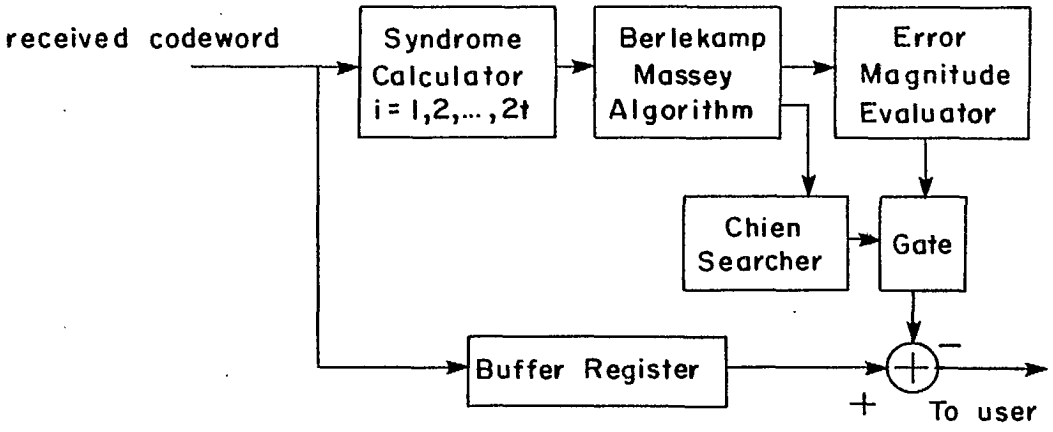


FIGURE A.12 : BASIC DECODING PROCEDURE FOR BCH CODES

• Example A.1.5.2:

We consider the 2 errors correcting BCH code over GF(7) of length 6 whose generator polynomial is $g(x) = (x-\alpha)(x-\alpha^2)(x-\alpha^3)(x-\alpha^4)$ where $\alpha = 5$ is a primitive 6th root of unity. We assume the received 6-tuple is $r = (0,0,2,0,3,0)$ whose corresponding polynomial is $r(x) = 2x^2 + 3x^4$.

The syndrome is found to be $\underline{S} = (0,2,5,0)$. Referring back to example A.1.5.1, we obtain the connection polynomial $C(D) = 1+D+D^2$ and the corresponding numerator polynomial $P(D) = 2D$. Using the Chien search, it is found that $\alpha^2 = 4$ and $\alpha^4 = 2$ are the roots of $C(D)$. As a consequence, two errors have occured in positions 4 and 2 of magnitude $e_4 = +3$ and $e_2 = +2$ respectively, so that the decoded codeword is found to be the all zero-tuple.

The types of codes we have been discussing so far were designed basically to correct random errors. However, on most real channels and particularly the satellite channel, the errors are not random. Many channels are "bursty" in the sense that errors tend to occur in burst. A burst of length b is an error vector whose only non-zero coordinates are among b consecutive digits, the first and last being non-zero. For example 0001010100---0 is a burst error of length 5. A multiple burst error is defined analogously. Now any cyclic code V can detect any burst of length n-k or less for the following reason. If $v(x) \in V$, $B(x)$ is a burst of length n-k or less and $r(x) = v(x)+B(x)$ is the received polynomial, then $r(x)$ is not divisible by $g(x)$ from the following argument. If $r(x)$ were divisible by $x^iB(x) = b_0 + b_1x + \ldots + b_{b-1}x^{b-1}$, since multiplication by x^i corresponds to i cyclic shifts, then $x^iB(x)$ belongs to V which implies that $x^iB(x)$ is divisible by $g(x)$. But, this is impossible (unless $B(x) = 0$) since the degree of $g(x)$ is n-k and $b \leq n-k$.

We now consider a class of codes capable of correcting multiple bursts of errors and which are very useful in practice. A Reed-Solomon code over GF(q) is a BCH code of length n = q-1. Hence to construct a Reed-Solomon code of designed distance d, we choose α to be a primitive element of GF(q) and form the generator polynomial $g(x) = (x-\alpha)(x-\alpha^2)\ldots(x-\alpha^{d-1})$. Note that the minimal polynomial of $\beta \in GF(q)$ over GF(q) is simply $(x-\beta)$. Since the degree of g(x) is d-1 we have that n-k = d-1 or k = n-d+1. Now d = n-k+1 and the minimum distance of the code is at least d. But for any (n,k) linear code it may be shown that the minimum distance is at most n-k+1; hence for the Reed-Solomon codes the minimum distance is in fact exactly n-k+1.

Furthermore, if $q = p^m$ and p a prime, then every element in GF(q) can be expressed as an m-tuple over GF(q). If we replace every coordinate of the vectors in an (n,k) Reed-Solomon code by an m-tuple we obtain a linear code over GF(q) with parameters $n = (p^m-1)m$ and $k = p^m-d$. The original code over GF(q) can correct any pattern of $t = [(d_{min}-1)/2]$ or fewer errors, hence can correct any burst of length $b \leq (t-1)m+1$. Of course, the code can also correct multiple bursts as long as they do not affect more than t of the original coordinates over GF(q). Best of all, such a code may be decoded using the Berlekamp algorithm for BCH code, as previously outlined.

- Example A.1.5.3:

Let $q = 2^4$. Using $1+x+x^4$ to generate GF(16) with α as a root of $1+x+x^4$, we obtain the code with generator $g(x) = (x-\alpha)(x-\alpha^2)(x-\alpha^3)(x-\alpha^4) = x^4+\alpha^{13}x^3+\alpha^6x^2+\alpha^3x+\alpha^{10}$. Using the fact that any element in GF(16) can be expressed as a linear combination of $1,\alpha,\alpha^2$ and α^3, the vector $(\alpha^{10},\alpha^3,\alpha^6,\alpha^{13},1,0,\ldots,0)$ translates into the binary vector (1110,0001,0011,1011,1000,0000,...) of overall lenght 60. The binary code has parameter n = 60, $k = 2^4-d = 16-5 = 11$. Since t = 2 the binary code can correct any burst of length less than or equal to 5.

A.2 Convolutional Codes

Whenever convolutional encoding is used, the transmitted data are first broken down into blocks of length k which are then mapped into n channel input letters as computed from the current and the past $m \geq 0$ input blocks. In practice the values of k and n ($k < n$) are small, typically chosen in the range of the integers {1,2,3,4,5}. Furthermore, the case m = 0 corresponds theoretically to the block code case although the values of k and n implied in these two applications remain extremely far apart. As in the block code case, the rate is defined as R = k/n (provided the input and output alphabets are identical) while the quantity m is usually referred to as the memory of the code (the related quantity K = (m+1)k is also called the input constraint length of the code. As an example, let us consider the k = 1, n = 2, m = 2 linear convolutional encoder represented on Figure A.13. Assuming the circuit initially at rest at time $u \leq 0$, the following equations can be used to compute the output.

$$y_u^{(1)} = x_{u-2} + x_u$$

$$y_u^{(2)} = x_{u-2} + x_{u-1} + x_u \qquad (2.1)$$

with $x_u = 0$ for u 0.

Hence if 1011010... is the input data stream, then the output stream, obtained by interleaving the two outputs, is 11,01,00,10,10,00,01,... Since the encoder puts out two bits for each incoming bit it follows that this encoding is accompanied by a two fold bandwith expansion. Such a bandwith expansion always accompanies coding schemes. The impulse response of the present encoder is obtained by exciting the encoder with the sequence 1,0,0,0,... to obtain the corresponding output

11,01,11,00,00,00,... Letting $G_0 = [11]$, $G_1 = [01]$, $G_2 = [11]$ and using the fact that the encoder is linear and time invariant it follows that if $y_u = [y_u^{(1)}, y_u^{(2)}]$, then

$$y_u = \sum_{j=0}^{2} x_{u-j} G_j \tag{2.2}$$

(i.e., the output is obtained as the convolution of the input and the impulse response). A general (n,k) convolutional encoder is characterized by k impulse responses, the i-th impulse response being obtained by exciting the encoder with the input (00--010--0, 00---0, 00--0,...) where the first k-tuple has a 1 in the i-th position. These impulse responses determine m+1k by n binary matrices $G_0, G_1, \ldots, G_m$ in such a way that

$$y_u = \sum_{j=0}^{m} x_{u-j} G_j. \tag{2.3}$$

This set of equations can also be written in matrix form as

$$[y_0, y_1, \ldots,] = [x_0, x_1, \ldots] G \tag{2.4}$$

where the semi-infinite matrix G is given by

$$G = \begin{vmatrix} G_0 & G_1 & \cdots & G_m & 0 & 0 & \cdots \\ 0 & G_0 & G_1 & \cdots & G_m & 0 & \cdots \\ 0 & 0 & G_0 & G_1 & \cdots & G_m & \cdots \end{vmatrix} \tag{2.5}$$

G is called the generator matrix of the encoder. The convolutional code generated by an (n,k) convolutional encoder is then the set of all output sequences obtained when the encoder is driven by all possible

rational (semi-infinite and eventually periodic) sequences. We remark that an (n,k) convolutional code has many encoders and in practice one would use a so called minimal encoder(*) as described by Forney [FORN 70]. For the encoder described by equation (2.3), if we define the state of the encoder at time u to be the content of the shift register

$$S_u = (x_{u-m}, x_{u-m+1}, \ldots, x_{u-1}).$$

then it becomes clear that the output at time u is determined by the state at time u and the input at time u. The number of states is 2^{km} and it is possible to describe the encoder by its state diagram. The state diagram is a directed graph with 2^{km} labelled nodes in which each node bears the label of one state. There is a directed branch from node S to node S^1 if there is an input which will drive the encoder from state S to state S^1 and S^1 is said to be a successor of S. Each node has exactly 2^k successors. Moreover, if S^1 is a successor of S and if the input digit x drives S into S^1 with the corresponding output y, then the branch from S to S^1 bears the label x/y. For the rate 1/2 encoder introduced earlier the state diagram is shown in Figure A.14. If we now label the branch x/y with Z^iW^j (the gain of the branch) where i and j are the Hamming weights of the output y and the input x respectively, and if we further split the 0-node into two nodes, one an input node and the other the terminal node, we obtain the signal flowchart for the encoder which is by definition the product of the gains of its constituent branches. An encoder is said to be catastrophic if there exists a closed path in its signal flowchart of gain 1 when W = 1. The encoder of the previous example is non-catastrophic. The property of

---

(*) Minimal encoders are characterized by the fact that they are feed-forward encoders which require a minimum number of memory cells in the corresponding obvious realization.

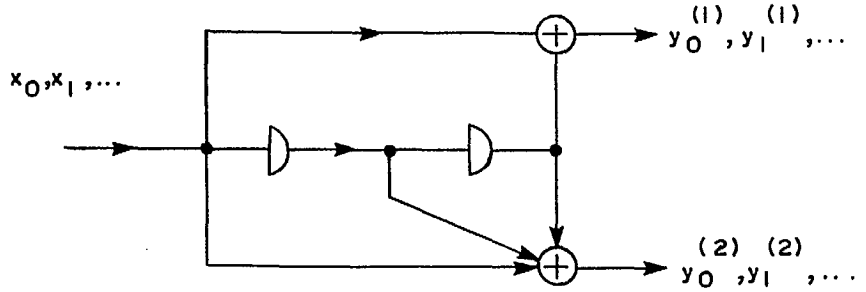
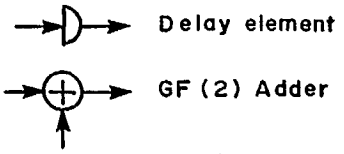


FIGURE A13: CONVOLUTIONAL ENCODER



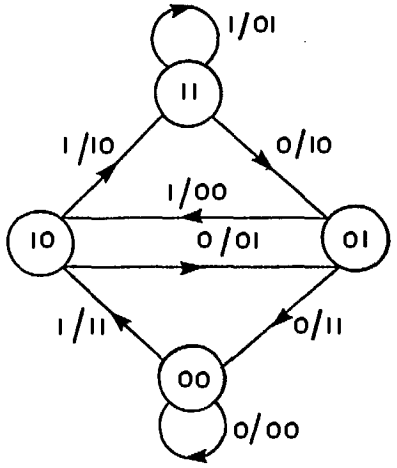


FIGURE A14: STATE DIAGRAM FOR A CONVOLUTIONAL ENCODER WITH m = 2

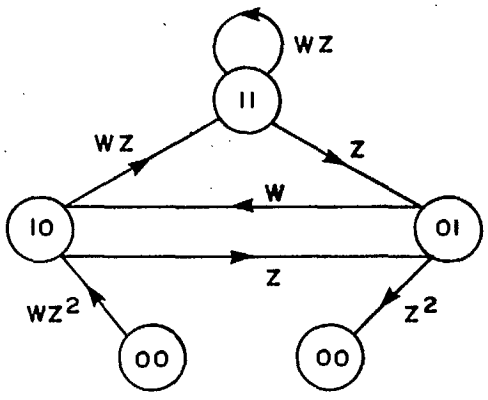


FIGURE A15: SIGNAL FLOWCHART FOR THE GRAPH OF FIGURE A14

being catastrophic is an undesirable property for it can lead to catastrophic error propagation in the decoder (see Massey and Sain [MASS2 69] and Forney [FORN 70]). If k = 1 and $G(D) = [g_1(D), g_2(D), \ldots, g_n(D)]$, then Massey and Sain [MASS2 69] have shown that the encoder G(D) is non-catastrophic if and only if the greatest common divisor of $g_1(D)$, $\ldots, g_n(D)$ is a power of D. For the previous example $C(D) = [1+D^2, 1+D+D^2]$ and it is easy to verify that the greatest common divisor between $1+D^2$ and $1+D+D^2$ is 1. A complete path in the signal flowchart is one which starts at the input node and terminates at the output node. The transmission gain (or transfer function) of the signal flowchart is the sum of the gains of all the complete paths. It is denoted by A(Z,W). For the signal flowchart of Figure A.16, we obtain by using, for example, Mason's rule

$$A(Z,W) = WZ^5/(1-2WZ). \tag{2.6}$$

In particular A(Z,1) is of the form

$$A(Z,1) = A_0+A_1Z+A_2Z^2+\ldots \tag{2.7}$$

with $A_0 = A_1 = \ldots = A_{d_f-1} = 0$ and A_{d_f} different from zero, where $d_f > 0$ is an important parameter of the code called the free distance. Furthermore, it should be clear that A_i in this case represents the number of codewords of Hamming weight i. For the previous example we have $d_f = 5$.

A.2.1 Tree, Trellis, and Viterbi Decoding

To obtain the tree corresponding to a convolutional encoder, we start with the root node which we label with the zero-state. Extending from the root node there will be 2^k branches, one for each possible information k-tuple, each branch being labelled with the new state

of the encoder. At level one in the tree there will be 2^k nodes and more generally at level j in the tree will be 2^{kj} nodes. The tree corresponding to the rate 1/2 encoder of the previous example is sketched in Figure A.16. A "1" directs the encoder up in the tree and a "0" directs the encoder down in the tree. Every path in the tree corresponds to a possible transmitted sequence (or codeword). Since the next state and the output digits of a convolutional encoder are determined by the previous state and present input, it is clear that the tree description is very redundant and we may, without losing anything, regroup together at any one level all the nodes bearing the same label. When we do this we obtain the trellis description of the encoder. For the above example we obtain Figure A.17. In practice the information digits are not fed continuously into the encoder but are fed in blocks of length L. These L information k-tuples are then encoded followed by m all zero k-tuples to return the encoder to the all zero state. This process is repeated for the next L information k-tuples, etc... In practice L is much greater than m so that the rate of the code is still very close to k/n. When this is done the trellis reconverges to the all zero state. For the previous example where L = 3, the resulting trellis appears in Figure A.18. Consider then using a convolutional code with parameters n,k,L and m to communicate over a binary input DMC. What is transmitted then is a complete path in the trellis. If $r = (r_0, r_1, \ldots, r_{L+m-1})$ is the sequence of received output branch blocks, then the task of a MLD (Maximum Likelihood Decoder) is to maximize the function

$$P[r/y], \text{ y a path in the trellis} \tag{2.8}$$

Equivalently we may maximize the logarithm of this function, i.e.,

$$\Gamma[y] = \sum_{i=0}^{L+m-1} \log P[r_i/y_i] = \sum_{i=0}^{L+m-1} \Gamma[y_i] \tag{2.9}$$

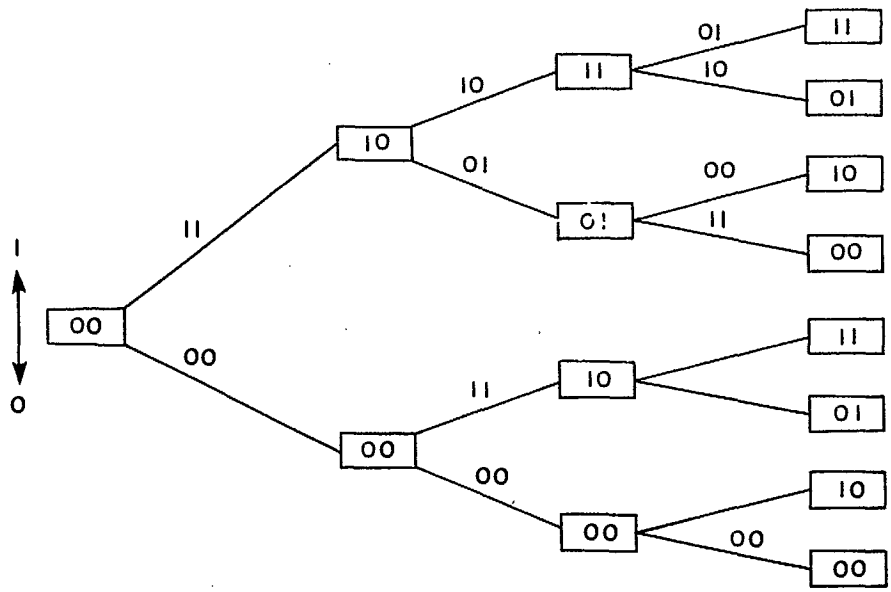


FIGURE A.16 : TREE DIAGRAM FOR A RATE 1/2 CODE

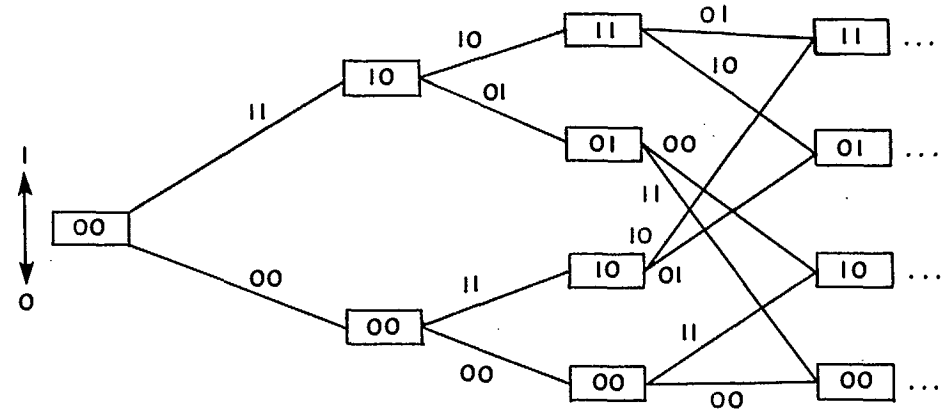


FIGURE A.17 : TRELLIS FOR RATE 1/2 CODE

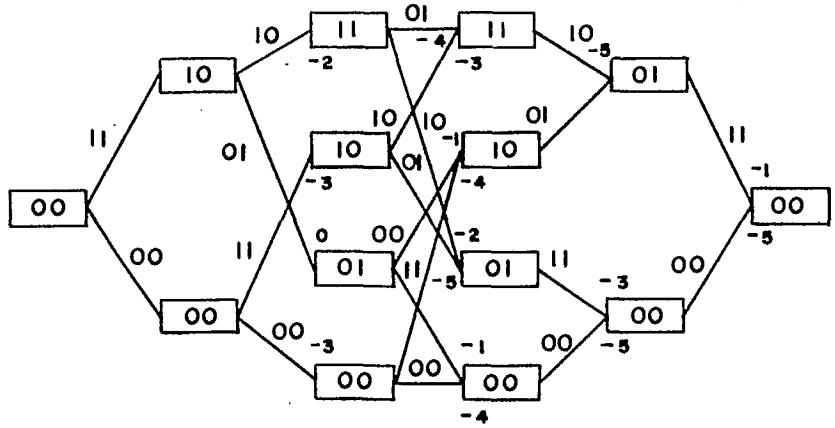


FIGURE A.18 : TRELLIS FOR CODE WITH L = 3, m = 2 , R = 1/2

where (2.9) defines the metric function $\Gamma[.]$. The brute force solution to this problem is to compute $\Gamma(y)$ for all 2^{Lk} codewords y. A far more efficient may of doing this was discovered by viterbi [VITE 67]. Viterbi decoding is predicated by the following simple principle (for a proof see [VITE 79]). Consider a path in the trellis and a second path which converges with the first at level j in the trellis. Let $y_{[o,j]}$ denote the first j+1 k-tuples of the first path and $y^1_{[o,j]}$ the corresponding digits of the second path in question. Then if

$$\Gamma[y_{[o,j]}] = \sum_{o}^{j} \Gamma[y_u] > \Gamma[y^1_{[o,j]}] = \sum_{o}^{j} \Gamma[y_u{}^1].$$

$y^1_{[o,j]}$ cannot be the first j+1 n-tuples of an optimum path. Consequently, all paths in the trellis having $y^1_{[o,j]}$ as a prefix can be discarded from further consideration in the search for an optimum path. This suggests the use of the following stratagem (known as the Viterbi decoding) to find an optimum path in the treillis.

First we form a stack consisting of all 2^{km} paths of length mn output digits with their corresponding metrics. (Note that there is no convergence of paths in the first m levels in the trellis). Next we extend all the paths in the initial stack and compute the associated metrics. The stack now has size $2^{k(m+1)}$.

For each node (2^{km} in number) at level m+1 we compare all the 2^k paths in the above stack which converge at that node and discard all except the one (or one) having the largest metric. The reduced stack now has size 2^{km} again.

We repeat the last two steps until the length of the paths is (L+m)n digits. At this point there will be only one path in the stack which will be an optimum path (the stack starts shrinking at level L+1).

It should be apparent that the practical implementation of the Viterbi decoder is limited by the size of the required stack which grows exponentially with memory m or with the constraint length K defined as:

$$K = (1+m)k \tag{2.10}$$

A.2.2 Sequential Decoding

For convolutional codes with a large constraint length Viterbi decoding is impractical. We can then turn to a class of decoding algorithms called sequential decoding algorithms of which the two most popular are the stack algorithm and the Fano algorithm. When speaking of sequential decoding it is best to take the tree description of the convolutional code. The object of the decoder is to find that path in the tree which is the most probable (or very nearly so) cause of the received sequence. A sequential decoder accomplishes this by exploring the tree and extending only one node at a time. Which node is extended is determined by a metric. Note at this point that a Viterbi decoder extends all the nodes in its stack at each step in the algorithm. Hence all extended paths in Viterbi decoding have the same length whereas in sequential decoding, the paths (or nodes) in the stack may have different lengths. Of course the idea is to extend a node which is more likely to lead to a "best" complete path in the tree. The metric to use was suggested by Fano [FANO 63] and later justified on theoretical background by Massey [MASS 72]. Let r be the received sequence and let $y = (y_0, y_1, \ldots, y_S)$ be a node in the tree; then the metric to associate with this node is:

$$L_f(y) = \sum_{j=0}^{S} \left(\log \frac{P(r_j/y_j)}{P(r_j)} - nR \right) \tag{2.11}$$

where R is the rate of the code and $P(r_j)$ is the probability of receiving r_j given 0 and 1 are equally likely to be transmitted i.e.,

$$P(r) = [P(r/0) + P(r/1)]/2$$

The metric of (2.11) is called the Fano metric. The idea then is to always extend the node of largest metric. The simplest way to do this is to keep all explored nodes in an ordered list of nodes of decreasing metric and then always extend the node at the top of the list. This is the "stack" algorithm [JELI 69], [ZIGA 66].

A very important quantity associated with any sequential decoding algorithm is the average number of computations C_0 per decoded bit or number of computations per decoded digit. A computation is by definition the extension of one node. In practice, one is allowed a maximum n_{max} computation per bit and we are then interested in the probability distribution

$$P = P(C_0 > n_{max}) \tag{2.12}$$

The quantity C_0 is of course a random variable depending on how noisy the channel is. The amount of time required to decode L bits is variable and so the information at the receiver must be buffered. The probability of buffer overflow is directly proportional to P. The quantity C_0 is known to be a Pareto random variable and is such that $P(C_0 > j)$ decreases only as a small negative power of j. For this reason P cannot be made extremely small with sequential decoding. One final comment is that the stack algorithm requires very little logical processing but requires a lot of storage. The Fano algorithm on the other hand displays the opposite behaviour in the sense that it uses very little storage but involves a lot of logical processing. In this algorithm, the decoder moves back and forth one step at a time in the tree

whereas the stack algorithm can jump several steps in a single extension. Whenever a node is extended, the metric at the new node is compared with a threshold T. According to how the metric compares with T, the decoder either moves forward or backward and adjusts the value of T accordingly. The threshold is modified in multiples of some quantization size Δ. The threshold at a node with metric V is said to be tight if $T \leq V < T+\Delta$. A node is visited either in a forward or backward move and a fictious node reflecting barrier preceding the root node is introduced with a metricinfinity which allows the algorithm to always bounce back in the forward mode from that node. Detailed steps of the Fano algorithm appear in the flowchart of Figure A.19, while the exact analysis of the Fano sequential decoding algorithm may be found in Gallager [GALL 68]. In particular it must be shown that the decoder never enters into a loop, i.e., it always reaches the end of the tree. Under certain circumstances, it may be shown (see [GEIS 73]) that the Fano algorithm finds the same path as the stack algorithm. In general, the limiting factor of the Fano algorithm is the amount of time it takes to decode a given received sequence. From a general standpoint, it may however be said that when the signal to noise ratio is large, implementations of the Fano algorithm are more efficient than the stack algorithm and further use a substantially smaller amount of storage.

A.3 Error Detection and ARQ Schemes

A typical communication system employing an ARQ error control scheme is shown in Figure A.20. In such systems, where block coding is used, a sufficient number of redundant symbols is used in order to achieve the required error detection capability. Basically, no error correction is performed by the decoder, but whenever an error is detected in a block, a retransmission of that block is requested through a reverse channel. A block is accepted by the user only after it appears

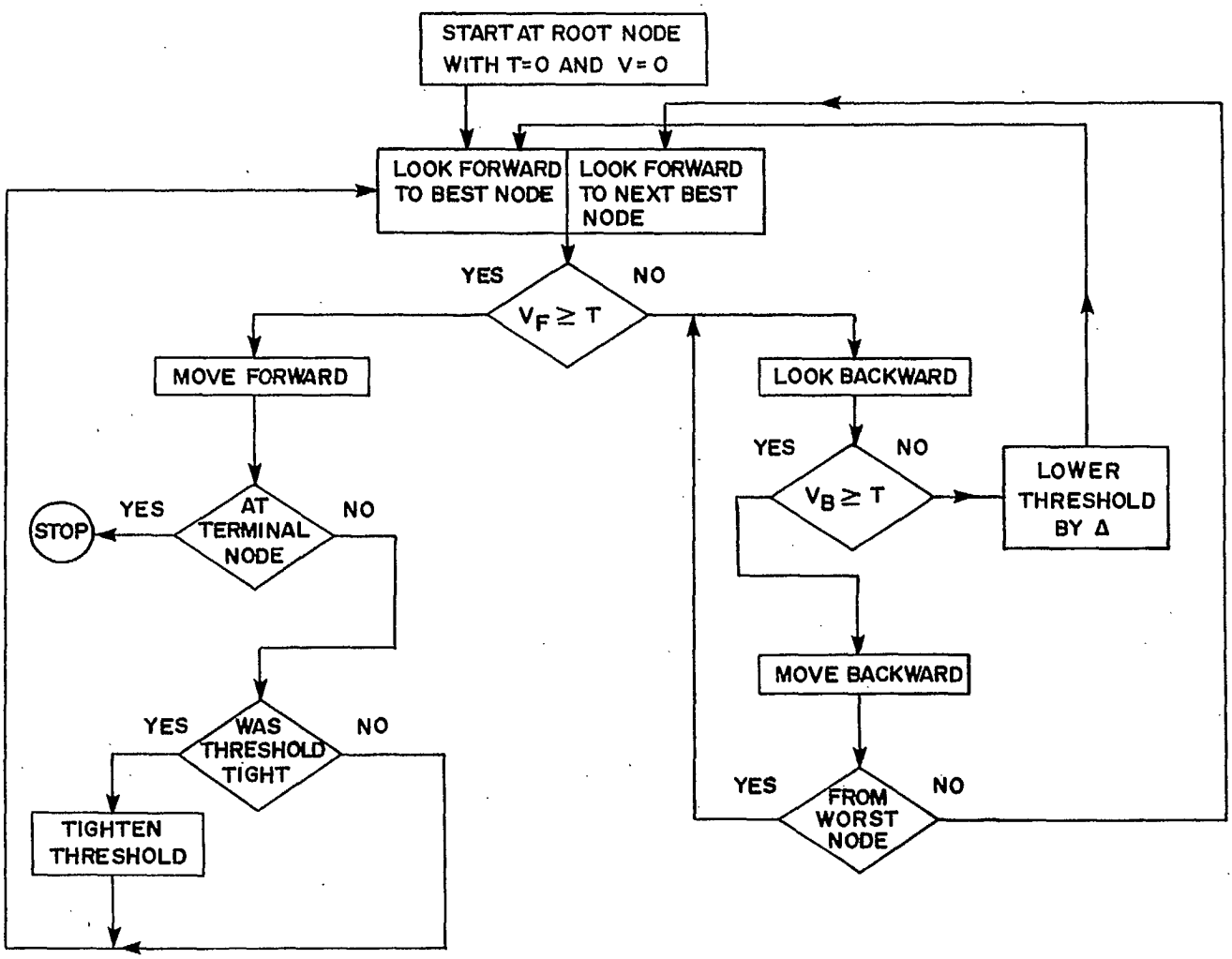


FIGURE 19: FLOWCHART OF THE FANO ALGORITHM FOR SEQUENTIAL DECODING

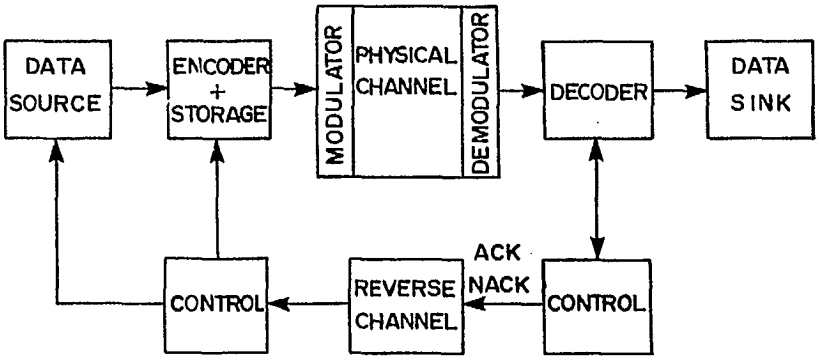


FIGURE A20: BLOCK DIAGRAM OF A DIGITAL COMMUNICATION LINK USING AN ARQ ERROR CONTROL SCHEME

to be error free. In this particular instance, the important measures of performance consist of the undetected error probability P_u which is typically very small ($P_u << 10^{-10}$) and the throughput efficiency of the system.

The principal advantages of an ARQ error control can be measured in terms of:

- Low undetected error rate.
- Effectiveness on most real channels.
- Moderate decoder cost and complexity.

The low undetected error probability is the consequence of the power of error detecting codes. Moreover, such systems are extremely robust in the sense that all information blocks delivered to the user can be accepted with equal confidence, even during periods of poor channel quality.

The information throughput of an ARQ system depends greatly on the number of retransmissions requested, thus on channel transmission characteristics as well as on the error detecting code.

The selection of a code is less difficult for an ARQ system than it is for an FEC system mainly because error detection codes are less sensitive to channel error patterns, in the sense that it does not matter very much how errors occur on the channel. Consequently the use of ARQ is effective on most channels whereas it should be noted that this property of ARQ systems is not shared in general by FEC systems.

Error detecting codes include the simple parity check block codes and both LRC/VRC type of codes as well as polynomial codes (also

known as Cyclic Redundancy Check or simply CRC codes). These polynomial codes which are based upon treating bit strings as representations of polynomials with coefficients of 0 and 1 can be encoded simply by shift registers in a fashion very similar to the encoding of cyclic codes. Apart from the CRC codes, any good error correcting codes (BCH codes) can be used for error detection. It can be noted at this point that the error detection capability of a given code (which measures the maximum weight of the errors which are at least detectable) is twice its error correcting capability. Besides some impressive error detection characteristics, ARQ systems are not without drawbacks. Among the main disadvantages are the following:

- A feedback control channel is required.
- The system operates with a variable decoding delay.
- The data source must be controllable or a suitable amount of storage must be provided to restrict the loss of data within tolerable limits.

The occurence of retransmission induces a decoding delay, which is measured by the time between the first arrival of a block at the decoder and its delivery to the user. Naturally a multiple transmission of the same block increases the decoding delay and hence reduces the throughput efficiency. In addition to the decoding lag, the round-trip propagation delay may be important and lowers further the information throughput. For a satellite channel, the round-trip delay is of the order of 500 msec so that the use of ARQ systems on this channel will drastically reduce the throughput unless suitable action is taken.

Finally during intervals of retransmission, the data source incoming bits must be stored in order to avoid a loss of information

bits at the encoder. Therefore the source must be somewhat controllable, and depending on the nature of the data source statistics this problem can only be overcome by the use of a suitable amount of storage.

In the forthcoming paragraphs, the throughout efficiencies associated with the classical ARQ schemes will be considered and compared in the context of their use on a satellite channel. The final paragraph will furthermore conclude by reviewing some recently introduced procedures which might also be worth of further considerations. As a measure of the effective use of the transmission channel (i.e, the percentage of useful channel uses), the throughput efficiency is certainly the key factor to be considered in comparing the different procedures. The final parameter to be considered in the last section measures the level of reliability of the data as delivered to the sink and is essentially independent of the repetition logic in use.

A.3.1 Stop and Wait ARQ Scheme

The simplest and most widely used detection-retransmission scheme is the stop-and-wait ARQ system. In this system, after the transmission of a block, the sending terminal waits for a positive or negative acknowledgement from the receiving terminal before proceeding to another block or retransmitting the same block.

Assuming that the effect of acknowledgement errors is negligible, the system throughput can be computed as

$$\eta = \frac{k(1-P_B)}{(\bar{n}+\bar{T}\bar{R}_s)} \text{ bits/symbol,} \qquad (3.1)$$

where the following quantities are defined as:

P_B = detectable block error probability
k = block size
n = number of information symbols per block
R_s = signalling speed on the channel in symbols/sec
T = round trip propagation delay + turnaround times (half duplex mode) + transmission time of the ACK/NACK message.

In order to explore the dependency between the system throughput η and the channel errors and system parameters, we assume that the channel is binary symmetrical with crossover probability p, and that the rate of the error detection code is constant and independent of the block size N. Thus assuming all errors are detected we have

$$P_B \cong 1 - (1-p)^n \cong 1-e^{-np} \tag{3.2}$$

and hence (3.1) can be written as

$$\eta = \frac{Rne^{-np}}{n + TR_s}$$

where $R = (k/n)$ is constant $\cong 1$. The block size n should then be chosen so as to maximize the throughput η. One then obtains, defining C as R_sT, the delay counted in channel symbols,

$$n_{opt} = (C/2)\left((1+(4/Cp))^{\frac{1}{2}} - 1\right) \tag{3.3}$$

or

$$\eta_{opt} = R\,\frac{n_{opt}e^{-n_{opt}p}}{n_{opt} + C}. \tag{3.4}$$

• Example A.3.1:

Assuming R_s = 50 Kbits/sec, T = 0.5 sec (corresponding to a two hops delay on a satellite channel), R = 1 and p = 10^{-3}, we obtain $n_{opt} \cong 963$ and η_{opt} = 1.42%. As a consequence, the channel is indeed very poorly used since only a little over one block out of a hundred yields a useful transmission so that the net transmission rate is only 500 x 1.42 = 710 bits/sec.

A.3.2 Continuous (GO-BACK N) ARQ Scheme

The continuous system transmits blocks of n bits consecutively and without any delay between blocks as long as positive acknowledgments are received at the sending terminal. Whenever a negative acknowledgment is received, the sending terminal essentially "backs-up" to the erroneous block and retransmits that block and all subsequent blocks in their natural order. Provided $C = TR_s$ represents the combined propagation and processing delay (counted in channel symbols) and N is defined as $\lceil C/n \rceil$ (the smallest integer $\geq C/n$). It follows that every time there is a repetition, the continuous ARQ logic "Goes-back" N blocks and proceeds forward from that point on. Using the same parameters definitions as in section A.3.1, it is easily found that the throughput efficiency can be computed as

$$\eta = \frac{R(1-P_B)}{(I+NP_B)}. \qquad (3.5)$$

On the BSC with crossover probability p, it is easy to prove that η can always be upper bounded as:

$$\eta \leq \eta^* = 1/(1+Cp). \qquad (3.6)$$

Optimization of η with respect to the block length n can be carried out asssuming that the parity check symbols form a block of constant length t = n-k. In this case, it is found by straightforward computation and assuming that n_{opt} as computed is such that the approximation $(1-p)^{n_{opt}} \cong e^{-n_{opt}p}$ is valid and furthermore 1/p C and 1/p

$$n_{opt} \cong t/2 + (t/p)^{1/2} \tag{3.7}$$

- Example A.3.2:

Using the same parameters values as in example A.3.1 and furthermore setting t = 16 (usual CRC parity bytes), it is found that the throughput efficiency is upper bounded as $\eta \leq \eta^* = 3.85\%$. The value of n_{opt} as obtained from (3.7) is $\eta_{opt} \cong 135$ and the corresponding throughput efficiency is $n_{opt} \cong 3.15\%$. This example illustrates the fact the bound of (3.6) is actually very tight.

A.3.3 Selective Repeat ARQ Scheme

In this mode, the procedure behaves essentially like the Go-Back N scheme except that, whenever a negative acknowledgment is received at the sending station, only the erroneous block is repeated. As expected, the throughput efficiency can be greatly enhanced through this modification and the corresponding expression is easily found to be

$$\eta = k(1-P_B)/n. \tag{3.8}$$

The fact that the above expression is independent of T (the "propagation" delay) renders the Selective Repeat scheme extremely attractive for satellite applications. However, the practical implementation of the logic remains rather complex [HACC 78]. Furthermore,

since the blocks are not delivered in their natural order at the sink, some means must be provided at the receiving end to rearrange them before delivery to their final destination. In this respect, Miller and Lin [MILL 81] have done an analysis on some variants of a Selective Repeat ARQ procedure involving a finite buffer receiver. Obviously, since a compromise must be made to handle the finite memory of the sink, the overall throughput degrades somewhat from the one given in expression (3.8) and, most importantly, the value of the propagation delay becomes again a key factor in the overall performance index.

A.3.4 Other Variants of ARQ Error Control Procedures

In order to palliate some of the inherent deficiencies associated with the classical ARQ error control procedures, different variants have recently been introduced in the literature. These variants come in two basic flavors.

- Schemes which alter or mix the steps of the underlying logics so as to marginally offset the basic impediments. One such sheme is the Selective-Repeat plus Go-Back-N (SR+GBN) ARQ. In this procedure, the transmitter operates in the Selective-Repeat mode for any block in the transmission buffer that has not been acknowledged up to v retransmissions, while the receiver stores those blocks that are successively received. Should that particular block not be acknowledged after γ attempts, the transmitter then switches to the Go-Back-N (GBN) retransmission mode so that new blocks are not sent. Retransmission starts with the current block and the N-1 succeeding blocks that were transmitted following the γ^{th} SR retransmission attempt for the currently stalled block.

Other ARQ schemes with such mixed modes of retransmission have been considered in the following references ([SAST 75], [MORR 78], [TOWS 79]). Although they are much simpler to implement than the S-R scheme and its finite receiver buffer version, they remain essentially less efficient with respect to the throughput efficiency.

- Hybrid ARQ schemes: from a general standpoint, it should be clear from the foregoing expressions giving the throughput efficiency associated with the classical ARQ schemes that a net improvement on the value of this parameter will be obtained by simply reducing the number of retransmission requests on the channel. Such a result can be simply obtained by improving on the raw bit error rate of the transmission channel by the use of Forward Error Correction. Such a scheme has been considered by the authors [CONA 78]. The suggested Forward Error Control procedure is based on the use of a convolutionnal code with Viterbi decoding. Performance curves have been computed showing the expected improvements using rate 1/2, 2/3 and 3/4 convolutional codes. Although it is possible to extend the range of useful throughput values to accomodate satellite delays, the price to be paid is in the form of an added bandwidth expansion on the high speed satellite channel (to compensate for the convolutional coding rate). To avoid this drawback, it was suggested to use the parity check for either error detection or error correction to prevent too many successive and costly repetitions. In such a scheme, whose name has been coined as a Hybrid II ARQ by Shu Lin [LIN 83], two codes are used for error detection and/or error correction in the transmitted blocks. The logic simply switches to error correction to avoid the buildup of

blocked information at the emitter and the receiver. Equivalently, this yields a better throughput value. However the global level of reliability, as compared with standard ARQ schemes and hybrid procedures using inner forward error correction, is somewhat reduced.

A.4 Reliability of Error Control Procedures

The level of reliability of any error control procedure is a measure of the confidence which can be attributed to the data as delivered to the sink. Depending on the error control procedure logic in operation (i.e, ARQ or FEC) and the code in use, it can be measured either by the undetected block error rate (P_{ue}) or block error rate (P_{BE}) when block codes are used in detection-only (ARQ) or correction-only (FEC) modes, or by the residual Bit Error Rate (BER) at the output of a transmission channel using convolutional encoding combined with real-time Viterbi decoding or feedback/definite decoding. The next two paragraphs concentrate on some important issues useful in the evaluation of these reliability parameters.

A.4.1 Reliability of block coded error control schemes

Given an (n,k) linear binary block code C, the key parameters in evaluating the error performance of this code are the minimum distance d_{min} which is the minimum Hamming weight of the non-zero codewords in C and the weight enumerating polynomial A(z) of C defined as

$$A(z) \triangleq \sum_{i=0}^{n} A_i z^i, \qquad (4.1)$$

where A_i represents the number of codewords in C of Hamming weight i. Although we always have $A_0 = 1$, $A_n = 1$ or 0 (depending on wether or not

the all-one codeword belongs to C) and furthermore $A(1) = \sum_{i=0}^{n} A_i = 2^k$, the evaluation of A(z) is in general very involved. In fact A(z) and d_{min} are not independent since d_{min} is the smallest positive index such that $A_{d_{min}} \neq 0$.

An important result is that the weight enumerating polynomials A(z) and B(z) of an (n,k) code C and its dual $C^{\perp}$ are related through the fundamental MacWilliam's identities which can be formulated by equating the coefficients of like powers of x on each side of the identity

$$2^k B(z) = (1+z)^n A((1-z)/(1+z)). \qquad (4.2)$$

• Example A.4.1: if n = 3 and k = 2, we obtain respectively:

$$4B_0 = A_0+A_1+A_2+A_3 = 4 \rightarrow B_0 = 1$$

$$4B_1 = 3A_0+A_1-A_2-3A_3$$

$$4B_2 = 3A_0-A_1-A_2+3A_3$$

$$4B_3 = A_0-A_1+A_2-A_3$$

The expression (4.2) is fundamental in evaluating the weight enumerating polynomial of high rate codes, such as used in detection-only ARQ schemes, from the corresponding function of the dual code. In fact the number of codewords in this last code is usually manageable to allow for systematic search. Based on error detection only, a block is assumed to be received correctly whenever it is a codeword. Hence, the undetected errors correspond to error patterns which are identical to codewords. As such, on a BSC with cross-over probability p, the undetected error probability can be expressed as

$$P_{ue}(p) = \sum_{i=0}^{n} A_i p^i (1-p)^{n-i} = (1-p)^n[(A(p/(1-p))-1]. \qquad (4.3)$$

Using (4.2), we can rewrite (4.3) with respect to the weight distribution polynomial of the dual code as

$$P_{ue}(p) = 2^{-(n-k)}B(1-2p)-(1-p)^n. \qquad (4.4)$$

Now it ts readily seen that for $p << \frac{1}{2}$

$$P_{ue}(p) \cong A_{d_{min}} (1-p)^n(p/(1-p))^{d_{min}} < 2^k(1-p)^n(p/(1-p))^{d_{min}} \qquad (4.5)$$

while if $p = \frac{1}{2}$, (4.4) yields the trivial value

$$P_{ue}(\tfrac{1}{2}) = 2^{-(n-k)}B_0-2^{-n} = 2^{-(n-k)}-2^{-n} < 2^{-(n-k)}. \qquad (4.6)$$

It is appealing to argue (unfortunately, falsely as will turn out later) that the case $p = \frac{1}{2}$ corresponds to the worst case (noisiest channel) so that $2^{-(n-k)}$ should be an upper bound on $P_{ue}(p)$ independently of the value of p and the particular code in use. The fact that such a plausibility argument is incorrect was first pointed out by Leung and Hellman [LEUN 76].

For our purposes, the following counter-example will be used to disprove the rather casually made foregoing claim.

- Example A.4.2:

Let us consider the (60,3) linear code formed by repeating 20 times each information bit in the input 3-tuple. This code is readily found to have the weight enumerating polynomial $A(z) = 1+3z^{20}+3z^{40}+z^{60}$. As a consequence it follows that we have:

$$P_{ue}(1/3) = (2/3)^{60}[3(1/2)^{20}+3(1/2)^{40}+(1/2)^{60}] \cong 7.78 \times 10^{-17}$$

while

$$P_{ue}(1/2) = 2^{-57}-2^{-60} \cong 6.07 \times 10^{-18} < P_{ue}(1/3)$$

To further the conclusions of this simple example, it turns out that even for small values of p, a "bad" code can give P_{ue} much larger than the "worst case" $2^{-(n-k)}$. However, by averaging over the ensemble of all (n,k) linear binary codes, it has been recently proved by Massey [MASS 78] that the average of the true worst case value of P_{ue}, denoted by $\overline{P}_{WC}$, satisfies the bound

$$\overline{P}_{WC} \leqslant n2^{-(n-k)}, \tag{4.7}$$

so that the right hand side of this relation can be taken confidently as the "worst case bound" for any reasonably good code.

In order to evaluate the performance of a block code in forward Error Correction code, we consider the behavior of a Maximum Likelihood Decoder. The fundamental result concerning the correction capability of a given code is contained in the following

Theorem:

There exists a decoder that corrects all patterns of t or fewer errors if and only if $t < d_{min}/2$, where d_{min} is the minimum distance of the code.

A consequence of this results is that if $t \triangleq [\frac{d_{min}-1}{2}]$, where $[x]$ represents the largest integer $\leqslant x$, the Block Error Rate P_{BE} of the transmitted blocks can be overbounded on the BSC as:

$$P_{BE} \leq \sum_{i=t+1}^{n} \binom{n}{i} p^i (1-p)^{n-i} = 1 - \sum_{i=0}^{t} \binom{n}{i} p^i (1-p)^{n-i} \qquad (4.8)$$

The expression on the right hand side of (4.8) is in general computationnally hard to evaluate. An overbound based on a geometric series gives a simpler, almost as tight bound as

$$P_{BE} \leq \binom{n}{t-1} (1-p)^n \left[\frac{p}{1-p}\right]^{t+1} \sum_{m=0}^{\infty} \left[\frac{p}{1-p}\right]^m \left[\frac{n-t-1}{t+2}\right]^m \qquad (4.9)$$

where we have used the following simple inequality

$$\binom{n}{i+m} \leq \binom{n}{i} \left(\frac{n-1}{i+1}\right)^m .$$

Evaluating the summation in (4.9) yields the final bound

$$P_{BE} \leq \binom{n}{t+1} (1-p)^n \left[\frac{p}{1-p}\right]^{t+1} \frac{1}{1-a} \qquad (4.10)$$

provided $a \triangleq \frac{p}{1-p} \frac{n-t+1}{t+2} < 1$.

Further simplification in the expression of (4.10) can be obtained by using the Stirling approximation to the factorial. The result yields

$$P_{BE} \leq \frac{2^{nH(\lambda)}}{(2\pi n\lambda(1-\lambda))^{1/2}} (1-p)^n \left[\frac{p}{1-p}\right]^{t+1} \frac{1}{1-a} \qquad (4.11)$$

provided a, as defined in (4.10), satisfies $a < 1$, $\lambda \triangleq \frac{t+1}{n} \leq \frac{1}{2}$ and $H(\lambda)$ is the binary entropy function $H(\lambda) = -\lambda \log_2 \lambda - (1-\lambda)\log_2(1-\lambda)$.

• Example A.4.3:

Recently a binary (55,16) Goppa code has been discovered whose minimum distance d_{min} is equal to 19. This code is the best known code in this range of block length and rate. Assuming a BSC with p =.01, the following upper bounds on P_{BE} are found using respectively (4.9), (4.10) and (4.11) 1.35×10^{-10}, 1.94×10^{-10}, 1.96×10^{-10}.

A weaker, yet simpler, bound on P_{BE} can be obtained by using the Chernoff bound on the tail of the binomial distribution. Carrying on the necessary algebra yields [VITE 79]:

$$P_{BE} \leq 2^{nH(\lambda)} (1-p)^n \left[\frac{p}{1-p}\right]^{t+1} \qquad (4.12)$$

with $\lambda \triangleq \frac{t+1}{n} > p$ and $H(\lambda)$ as defined in (4.11).

To illustrate, the value of the bound (4.12) computed according to the parameters used in example A.4.3 yields 1.34×10^{-9}.

In certain applications associated with error-correction, the parameter of interest becomes not the block error rate but rather the "output" Bit Error Rate (BER) which measures the fraction of output symbols which are erroneous in a very long sequence. The exact determination of the BER is quite involved in general. The difficulty of the analysis resides mostly in the fact that, not only BER is code dependent, but it is also a function of the particular encoder as well as of the decoding algorithm in use. For Maximum Likelihood types of decoding, it is however possible to bound simultaneously from above and from below BER by noticing that any block error results in at least one input error and at most k errors so that we always have, in this particular case:

$$\frac{1}{k} P_{BE} \leq BER \leq P_{BE}. \tag{4.13}$$

For a systematic cyclic code, whenever P_{BE} is small, the following approximation

$$BER \cong \frac{d_{min}}{n} P_{BE} \tag{4.14}$$

becomes very tight since, in this particular case, decoding errors occur almost always to the nearest neighbours causing d_{min} errors in the codeword bits.

A.4.2 Reliability of Convolutional Codes Using Viterbi Decoding

The evaluation of the error performance of convolutional codes will be restricted to determining a tight bound on the Bit Error Rate of a convolutionally encoded high speed transmission channel using Maximum Likelihood decoding such as could be implemented through a Viterbi decoder. This situation is of interest whenever the convolutional code is used as the inner code of a hybrid system in which case, provided Time Division Multiple Access is used to guarantee that errors remain independent at the output of the super channel, BER is the cross-over probability for the outer ARQ process. For further information on this particular situation, we refer the reader to [CONA 78]. At high signal-upon noise ratio, a tight upper bound on BER can be computed through the path enumerating function A(Z,W) introduced in section A.2. Referring to [VITE 79], it is found that

$$BER \leq \frac{1}{k} \frac{\delta A(Z,W)}{\delta W} \Bigg|_{\substack{W=1 \\ W=1}} \triangleq \sum_{j=d_f}^{+\infty} c_j \gamma^j \tag{4.15}$$

where $\gamma \triangleq \sum_{r \in Y} (\text{Prob}(r|0)\ \text{Prob}(x|1))^{1/2}$ is the characteristic of the two inputs (0,1) Discrete Memoryless Channel with output alphabet {Y} on which the convolutional code is used. If hard decision is in effect, then $\gamma = (2\ p(1-p))^{1/2}$ where p is the symbol error rate on the high speed channel. For the running example considered in Figure A.16, we find, assuming hard decision,

$$\text{BER} \cong \frac{p^5}{(1-2p)^2}. \tag{4.16}$$

To illustrate, if we assume binary PSK (Phase Shift Keying) modulation on an ideal gaussian channel (no symbol interference taken into account) at a signal-upon-noise ratio E_s/N_o = 6dB and hard decision, the value of p is found to be 2.6×10^{-3} whereas (4.16) yields BER $\cong 1.73 \times 10^{-5}$.

REFERENCES

[BERL 68] Berlekamp, E.R., "Algebraic Coding Theory", McGraw-Hill, 1968.

[CONA 78] Conan, J. and Haccoun, D., "High Speed Transmission of Reliable Date on Satellite Channels", Proceedings of the Fourth International Symposium on Digital Satellite Communication, ISDSC, pp. 296-301, Montréal, Canada, Oct. 1978.

[FANO 63] Fano, R.M., "A Heuristic Discussion of Probabilistic Decoding", IEEE Trans. on Information Theory, Vol. IT-9, pp. 64-74, April 1963.

[FORN 70] Forney, G.D., "Convolutional Codes 1: Algebraic Structure", IEEE Trans. on Information Theory, Vol. IT-16, No. 6, pp. 720-738, Nov. 1970.

[GALL 68] Gallager, Robert C., "Information Theory and Reliable Communication", John Wiley and Sons, 1968.

[GEIS 73] Geist, J.M., "Search Properties of Some Sequential Decoding Algorithms", IEEE Trans. on Information Theory, Vol. IT-19, pp. 519-526, July 1973.

[HACC 78] Haccoun, D., Conan, J. and Golly, G., "Node to Node Protocols on a High Speed Full-Duplex Satellite Link", Proc. Natl. Telecommun. Conf., Dallas, Texas, pp. 28.1.1-28.1.5, Dec. 1978.

[JELI 69] Jelinek, F., "A fast Sequential Decoding Algorithm Using a Stack", IBM J. Res. Dev., Vol. 13, pp. 675-685, Nov. 1969.

[LEUN 76] Leung, S.K., and Hellman, M.E., "Concerning a Bound on Undetected Error Probability", IEEE Trans. on Information Theory, Vol. IT-22, pp. 235-237, March 1976.

[LIN 83] Lin, S. and Costello, D.J. Jr., "Error Control Coding: Fundamentals and Applications", Prentice-Hall, 1983.

[MASS 69] Massey, J.L., "Shift-Register Synthesis and BCH Decoding" IEEE Trans. on Information Theory, Vol. IT-15, no. 7, pp. 122-127, Jan. 1969.

[MASS 69] Massey, J.L. and SAIN, M.K., "Inverses of Linear sequential Circuits", IEEE Trans. on Computers, Vol. AC-14, pp. 141-149, April 1969.

[MASS 78] Massey, J.L., "Coding Techniques for Digital Data Networks", Proceedings of the Information Theory and Systems, NTC-Fachberichte, Vol. 65, Berlin, Sept. 18-20 1978.

[MCWI 78] MacWilliams, F.J. and Sloane, N.J.A., "The Theory of Error Correcting Codes", The North-Holland publishing Co., 1978.

[MILL 81] Miller, M.J. and LIN, S., "The Analysis of Some Selective-Repeat ARQ Schemes with Finite Memory", IEEE Trans. on Communications, Vol. COM-29, pp. 1307-1315, Sept. 1981.

[MORR 78] Morris, J.M., "On Another Go-Back-N ARQ Technique for High Error Rate Conditions", IEEE Trans. on Communications, COM-26, pp. 187-189, Jan. 1978.

[PETE 61] Peterson, W.W., "Error Correcting Codes", M.I.T. Press, 1961.

[SAST 75] Sastry, A.R.K., "Improving Automatic-Repeat-Request (ARQ) Performance on Satellite Channels Under High Error Rate Conditions", IEEE Trans. on Communication, COM-23, pp. 436-439, April 1975.

[TOWS 79] Towsley, D., "The Stutter Go-Back-N ARQ Protocol", IEEE Trans. on Communications, COM-27, pp. 869-875, June 1979.

[VITE 67] Viterbi, A.J., "Error Bounds for Convolutional Codes and an Asymptotically Optimum Decoding Algorithm", IEEE Trans. on Information Theory, Vol. IT-13, pp. 260-269, April 1967.

[VITE 79] Viterbi, A.J. and Omura, J.K., "Principles of Digital Communication and Coding", McGraw-Hill, 1979.

[ZIGA 66] Zigangirov, K.Sh., "Some Sequential Decoding Procedures", Prob. Peredachi Inform., Vol. 2, pp. 13-25, 1966.

APPENDIX B

CRYPTOGRAPHY

In order to fully motivate the need for encryption, it should be said that back in the early days when corporations and universities had a single computer centre, achieving security was easy. All the organization had to do was to station a guard at the door of the computer room. The guard made sure that no one removed any tapes, disks or decks of cards from the room unless explicitly authorized to do so. With the advent of networking, the situation has changed drastically and, as a consequence, no one can anymore claim to be able to manually control the integrity of the millions of bits that daily move around between the computers of the world. Moreover, organizations have no means to make sure that their data are not secretly copied or tempered with by wiretap or any other means during their transfer towards the final destination. Worst of all and, probably, the most significant factor in motivating some in depth study of crytography in the framework of this study is the fact that, whenever satellite links are being used on the transission path, all data are available to anyone who cares to go through the trouble of erecting a small antenna and earth station to listen to the incoming waves. Clearly, in such a new environment some kind of encryption (also called encipherment) capability is needed to make sure that data reach their intended recipients without being intercepted and/or tempered with by intruders during the transmission.

If we go back in time, cryptography has a long and colorful history. However, our intent in this chapter is not to review all the steps in this long history. Rather, we will concentrate on the two latest developments of cryptography which will probably have the most significant impact in the future developments related to this study

namely the recent advent of the DES (Digital Encryption Standard) system as well as the introduction of PKS (Public Kay Crytosystems) as new and powerful means of circumventing the two major problems of key handling and authentication associated with standard cryptography. For a more complete and colorful historical presentation of the classical cryptographic techniques, the book by Kahn [KAHN 67] is highly recommanded.

B.1 A Review of Classical Cryptographic Techniques

The general model for a classical cryptographic system is depicted in Figure B.1. In this model, the messages to be encrypted, generally referred to as plaintext, are transformed by a cryptographic function E_k which is parameterized by a key K. The output of the encryption process, known as ciphertext or cryptogram, is then transmitted either through a messenger service or by electromagnetic means such as radio or guided transmission. In generel, it can be assumed that the enemy or wiretaper hears and accurately copies down the compplete ciphertext. However, unlike the intended recipient, he does not know what the key is and, as such, is essentially unable to easily decrypt the ciphertext. On certain occasions, the intruder is capable not only to listen to the communication channel (passive intruder), but he can also record messages and play them back later as many times as he so desires. On other instances, but he can even inject his own messages or ultimately modify, to his own advantages, legitimate messages before they reach their intended receiver (active intruder). From a general standpoint, the art of breaking ciphers is known as cryptalysis while the techniques used in devising ciphers is usually referred to as cryptography. Furthermore, the scientific activity dealing collectively with cryptography and cryptanalysis is known as crytology.

With possibly the exceptions of the earliest cryptographic schemes (i.e., as examples the ones invented by Caesar or Vigenère), one of the fundamental rule of cryptography should be based on the fact that one must assume that the cryptanalyst knows the general method of encryption used. The main reason behind such a hypothesis is basically due to the fact that the amount of effort necessary to invent, test, and install a new method every times the old technique is compromised or even thought to be compromised has always made it quite impratical to keep this component of a cryptographic scheme a secret. Furthermore, thinking it is secret when it is not does in fact create more harm than good. It is at this point that the role of the key becomes fundamental. The key consists usually of a short string of characters that selects one among many potential encryption functions. Furthermore, and in contrast to the general cryptographic algorithm or mechanism, which may not be changed very often, the key can be changed as often as desired. As a consequence the general model becomes one in which all internal mechanisms or algorithms are in fact known except for the internal parameters which can be changed at any time through the use of different keys.

On the other side of the coin, the cryptanalyst's problem has essentially three variations depending on the level of sophistication that can be attributed to the intruder. The less dangerous attack, from a standpoint of security, is the one dealing with the situation where the cryptanalyst has a quantity of ciphertext but no related plaintext. Typically, the solution of cryptograms which appear in newspapers falls into this category of problems. In order for an encryption system to be of any value, it should at least withstand such a ciphertext only attact. We reach one step higher in the level of sophistication whenever the cryptanalyst has access to matched pairs of ciphertext and plaintext. Such a situation is usually referred to as a

known plaintext attack on the enciphering scheme. Finally the third and possibly last level of sophistication on the part of the cryptanalyst must consider the situation in which the cryptanalyst has the ability to select the kind of plaintext he desires to encrypt. Such a problem, viewed from the point of security, is known as the chosen plaintext attack and can be considered to be the ultimate test in the evaluation of the level of vulnerability associated with a given encryption procedure. As an example which illustrates the advantage that such a situation can provide in the solution of a given cryptogram, consider the case of a typical newspaper cipher problem in which you are allowed to ask questions such as what is the ciphertext associated with the plaintext ABCD...?

Novices in the field of cryptography often assume that if a cipher system can withstand a ciphertext only attact, it must be secure. Such an assumption turns out ot be extremely naive taking into account the environment in which the intruder operates. In many situations, the cryptanalyst can make a good guess at what part of the plaintext might be. Such is the case around a computer facility where, for example, the first question many time-sharing systems inquire whenever you call them up is "PLEASE LOGIN". It should then be obvious that, equipped with some pairs of matched plaintext and ciphertext, the task of the cryptanalyst becomes much easier. The main conclusion to be draw from those simple remarks is that, in order to achieve a high level of security, the cryptographer should be extremely conservative and make sure that his system is unbreakable even if his opponent is able to encrypt any amount of plaintext.

Encryption procedures have historically been divided into two categories; namely substitution and transposition ciphers which we now describe in some details.

B.1.1 Substitution Ciphers

In a substitution cipher, each letter or group of letters is replaced by an other letter or group of letters to disguise it. The oldest cipher of this type is the Caesar cipher, attributed to Julius Caesar. In this method, a becomes D, b becomes E, c becomes F,..., and z becomes C. For example, the plaintext "attack" becomes "DWWDFN". (In examples, plaintext will be given in lower case letters while ciphertext will appear in upper case letters). A slight generalization of the Caesar cipher allows for the ciphertext alphabet to be shifted by k letters instead of three. In this case, the value of k becomes the key to the general method of circularly shifted alphabets. However, beware, the Caesar cipher may have fooled the carthaginians, but it has not fooled anyone since!

The next improvement in complexity is to have each of the symbols in the plaintext, say for the sake of simplicity the 26 letters map onto some other letter arrangement. For example:

Plaintext : abcdefghijklmnopqrstuvwxyz
Ciphertext : QWERTYUIOPASDFCHJKLZXCVBNM

Such a general system is know as a monoalphabetic substitution cipher with the key being the 26 letters string corresponding to the full alphabet. For the key given above, "attack" would be encrypted as "QZZQEA". At first glance, this might appear to be a safe system because, although the cryptanalyst knows the general encryption technique (i.e., monoalphabetic substitution), he has no idea of which one among the $26! \cong 4 \times 10^{26}$ possible keys is in use. In contrast with Caesar's cipher, trying all of them is not very promising (even at a speed of 1μs per solution, a computer would take 10^{13} years to complete

the search). Nonetheless, given a surprising small amount of ciphertext (i.e., under a ciphertext only attact) such a cipher can be fairly easily broken down.

The basic attack takes advantage of the statistical properties of natural languages. In English, for example, e is the most common letter, followed by t, a, o, n, i, etc... The most common two-letter combinations, or digrams, are: th, er, re and an. The most common three-letter combinations, also called trigrams, are: the, and, ion and ent. Consequently, a cryptanalyst trying to break a monoalphabetic cipher would start out by counting the relative frequencies of all letters in the ciphertext. Then he might tentatively assign the most common one to e and the next common one to t. He would then look at the trigrams to find one of the form tXe; which strongly suggests that X is h. Subsequently, if the pattern thYt occurs frequently, the Y probably stands for a. With this basic information on hand, he can then look for a frequently occuring trigram of the form aZW, which is most likely and. As seen from the foregoing discussion, it becomes easy, in general, for the cryptanalyst to build up a tentative plaintext letter by letter by making guesses at common letters, digrams and trigrams.

A general countermeasure to render the cryptanalyst's job more difficult consists in smoothing out the frequencies of the letters appearing in the ciphertext, so the letters e, t, etc... do not stand out so clearly. One way of attaining this result is to introduce multiple cipher alphabets, to be used in rotation for substitution. Such a scheme is known as a polyalphabetic cipher. The simplest example of polyalphabetic cipher is the Vigenère cipher which uses a square matrix of 26 Caesar's alphabets (cyclic alphabets). The first row, called row A, is ABC...XYZ. The next row, referred to as B, is BCD...YZA. Finally the last row, called Z, is ZAB...WXY.

Similarly to the monoalphabetic case, these ciphers also have a key. However, instead of being a string of 26 distinct characters, the key is usually a short word or phrase, such as COOKIEMONSTER. To encrypt a message, the key is written repeatedly above the plaintext. such as for example

COOKIEMONSTERCOOKIEMONSTER...
itwasnicetoseeyouagaininsu...

The key above each plaintext letter tells us which row to use for encryption. For example, the i is encrypted using the Caesar's alphabet of row C, and so on. It should be clear that any plaintext letter will be represented by different letters in the ciphertext depending on its position in the plaintext. Furthermore, digrams and trigrams will also be enciphered differently according to their appearance order.

A more powerful polyalphabetic cipher can be constructed by using arbitrary monoalphabetic cipher for the row of the square matrix. However, in this case, not only should the running key be memorized, but the monoalphabetic ciphers table as well. However, as is the case in general, the longer the key, the more difficult it is to maintain its secrecy.

Although unquestionably much better than the monoalphabetic cipher, polyalphabetic ciphers can also be broken down easily under ciphertext only attack, provided that the cryptanalyst has a sufficient amount of ciphertext. The basic trick is to guess the key length. Assumming a key length of k letters, we can rearrange the ciphertext in rows of k letters. Provided the key length estimate is correct, it follows that each column will have been encrypted according to the same monoalphabetic cipher and can be attacked as such.

The next step up in complexity for the cryptographer is to use a key which is at least as long as the plaintext, so that the afore mentionned mode of attack becomes useless. This scheme yields in fact a truly unbreakable cipher which is known as a one time key or running key cryptosystem. In the binary case, such a scheme can be simply constructed by first choosing a random binary sequence, then converting the plaintext to a binary form by using, for example, its ASCII representation and finally computing, as the ciphertext, the Exclusive OR of the two sequences. Deciphering the plaintext is then most easily performed by applying the Exclusive OR operator on the received ciphertext and the binary key to obtain the plaintext. The validity of this deciphering scheme follows from the fact that, in the binary case Exclusive Oring of a sequence with itself always results into the zero sequence.

The fundamental theorem on the existence of ideal secrecy systems is due to Shannon [SHAN 49]. As a matter of fact, it turns out that the demonstration of this result is based on a constructive method which makes use of a one time key. It should be clear, however, that the running key cipher is quite impractical. The first reason follows from the fact that since the key cannot be memorized, both parties must carry a written copy with them. As a consequence, it follows that a highly secure channel must be provided for the delivery of keys. Furthermore, since the key is as long as the plaintext, the same number of plaintext characters can be transmitted on this highly secure channel without the use of a cipher system!!!

As a conclusion to this paragraph and before considering another class of ciphering algorithms, namely Transposition ciphers, we will make the following remarks. Although perfectly secure systems do exist, they require, in general, an enormous amount of secret key information to be exchanged. As a consequence, since compression of

information prior to encryption will reduce the key length, data compression will always result in enhancement of data security. Furthermore, Shannon's original ideas on cryptography were also quite in advance of their time since he already suggested that a more useful and practical approach to cryptography should be built on concepts of "computational complexity" (to use the modern phrasiology) as opposed to information theoretic concepts. It turns out that these concepts constitute the basic theoretical background on which Public Cryptosystems are conceived.

B.1.2 Transposition Ciphers

Substitution ciphers essentially preserve the order of the plaintext symbols but disguise them. Transposition ciphers, on the other hand, reorder the letters but do not disguise them. Figure B.2 represents a common transposition cipher of the columnar transposition type. The logic of the procedure operates as follows. The cipher is keyed by a word or phrase without any repeated letters. In the running example of Figure B.2, MEGABUCK is the key. The purpose of the key is to number the columns: column 1 being under the key letter closest to the start of the alphabet and so on. The plaintext is then written horizontally, as a series of rows. The ciphertext is then read out by column, starting with the column whose key letter is the lowest in the ordered list.

To break a transposition cipher, the cryptanalyst must first be aware that he is dealing with a transposition cipher. This is most easily checked by computing the frequency of the letter E, T, A, O, I, N, etc... and verifying that they fit the normal pattern for plaintext. If so, the cipher must be clearly a transposition cipher, because in such ciphers every letter represents itself in the ciphertext. The

next step is to make a guess at the number of columns. In many cases, a probable word or phrase may be guessed at from the context. For example, suppose that our cryptanalyst suspected the phrase million dollars to occur somewhere in the plaintext. Observe that the digrams MO, IL, LA, IR and OS occur in the ciphertext as the result of this phrase wrapping around. The ciphertext letter O follows the ciphertext letter M (i.e., they are vertically adjacent in column 4) because they are separated in the probable phrase by a distance equal to the key length. If a key of length 7 had been used instead, the diagrams MD, IO, LL, LL, IA, OR and NS would have occured instead. In fact, it is readily seen that for each key length a different pattern of digrams appears in the ciphertext. By hunting for the various possibilities, the cryptanalyst can often determine the key length.

The remaining step is to order the columns. When the number of columns, k, is small, each of the k(k-1) columns pairs can be examined to see if its digram frequencies match that for English plaintext. The pair with the best match is assumed to be correctly positioned. Now, each remaining column is tentatively tried as the successor to this pair. The column whose digrams and trigrams frequencies give the best match is tentatively assumed to be correct. The predecessor column is found in the same way. The entire process is continued until a potential ordering is found. Chances are that the plaintext will be recognizable at this point (e.g., if million occurs, it is clear what the error is).

Some transposition ciphers accept a fixed-length block of input and produce a fixed-length block of output. These ciphers can be completely described by just giving a list telling the order in which the characters are to be output. For example, the cipher of Figure B.2 can be seen as a 64 character block cipher. Its output is 4, 12, 20,

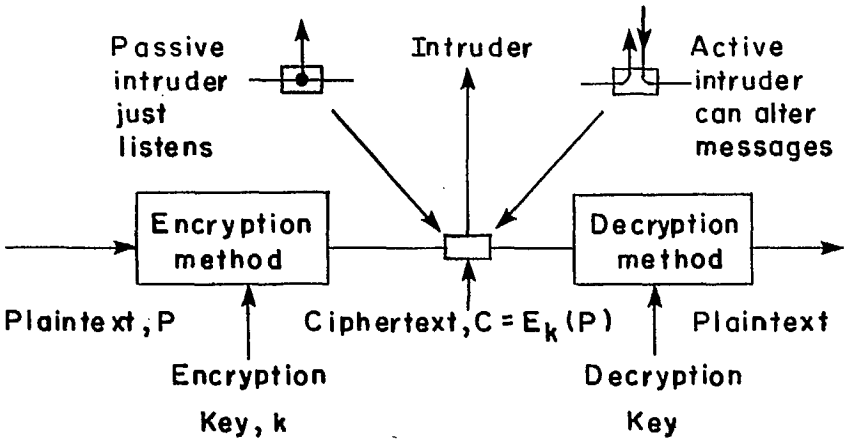


FIGURE B.1 : GENERAL ENCRYPTION MODEL

M	E	G	A	B	U	C	K
7	4	5	1	2	8	3	6
p	l	e	a	s	e	t	r
a	n	s	f	e	r	o	n
e	m	i	l	l	i	o	n
d	o	l	l	a	r	s	t
o	m	y	s	w	i	s	s
b	a	n	k	a	c	c	o
u	n	t	s	i	x	t	w
o	t	w	o	a	b	c	d

Plaintext:

pleasetransferonemilliondollarsto myswissbankaccountsixtwotwo

Ciphertext:

AFLLSKSOSELAWAIATOOSSCTCLNMOMANT ESILYNTWRNNTSOWDPAEDOBUOERIRICXB

FIGURE B2 : A TRANSPOSITION CIPHER

28, 36 44, 52, 60, 5, 13, ..., 62. In order words, the fourth input character, A, is the first to be output, followed by the twelfth, F, and so on.

B.2 The Data Encryption Standard (DES)

Following the growing concern in recent years about privacy and secrecy both within the governmental agencies and private companies, it was decided by the Federal Government of the United States to investigate means of implementing standards for the protection of the privacy of data stored in the Federal data banks. Although geared initially for applications within the government, there is no doubt that the ensuing Data Encryption Standard will have a great impact in the private industry as well.

Since its adoption in 1977 as Federal Information Processing 46 [FIPS 77], DES has in fact enjoyed great success with manufacturers, making a sweeping change in the availability of cryptographic products. Where before these were available only from a few specialized companies, now approximately a dozen major computer and electronics manufacturers are producing equipment ranging from individual DES chips to stand alone link encryption units and to cryptographically protected automated teller machines. For example, Intel (Chip 8294), Motorola (Chips MGD 8080 and DSM 6800), Rockwell (Chip CR-300), Western Digital (Chip DE 20001/2) and Fairchild (Chip 9414) have all designed hardware for public sale except IBM which restricts its use internally.

B.2.1 Description of the DES

Probably the simplest point of view is to look at the DES algorithm as a sophisticated substitution cipher operating over an

alphabet of 2^{64} letters. Referring to Figure B.3, an input block of 64 bits which constitutes a letter in this alphabet, is replaced by a new letter, the output block. From a mathematical standpoint, the DES procedure is then simply a permutation operation on letters of this very large alphabet. The key (64 bits including 8 parity bits) selects one of 2^{56} possible simple substitution ciphers.

In a little more detail, the algorithm consists of 16 iterations of a standard building block as shown in Figure B.4, where each building block takes 48 bits of key and uses them to transform its 64 bits of input into 64 bits of output. Aside from the different key bits used, each of these blocks is identical. The output of each building block becomes the input to the next building block. There is a transposition of the two halves of the final output to make decryption easier (see Figure B.5). The initial permutation IP is of no cryptographic significance.

The interior of the standard building block is shown in Figure B.6. The right-hand 32 bits are copied unchanged to the left-hand 32 bits of the output of the block. The right-hand 32 bits are spread out and scrambled into 48 bits with the help of the E-table and summed modulo 2 (Exclusive OR) with the 48 bits of key. The resulting 48 bits are passed through a nonlinear transformation called the S-boxes to form 32 bits. The expansion from 32 bits back to 48 bits performed by the E-table and followed by a compression back to 32 bits by the S-boxes is the crucial part of the entire algorithm. Not only does it permit 48 bits of key to be used at this stage, but it also implies that a single bit change in the input to the building block will cause many bits to change in the output, and in a way that is hard to predict.

The S-boxes are actually a set of eight transformations, each of which transforms six bits of input into four bits of output. In order to further spread the effect of the transformations evenly over the whole 32 bits, the output of the S-boxes is permuted by the so-called P-table. The 32 bit output of the P-table is summed (by Exclusive OR) with the 32 bits of left-hand input to the block and the result is the 32 bits of right-hand output of the box. The action of the box can be represented by the relations

$$L' = R$$
$$R' = L + f(K,R)$$

Notice that the value of f(K,R) can be expressed also in terms of the output of this block as f(K,L') and this is what makes decryption possible. To decrypt, the same algorithm is used but the keys used in the standard building blocks are used in the reverse order.

Key selection is done by a peripheral and rather uninteresting scheme that reuses each of the original 56 bits about 14 times so that the resulting procedure which, by its original design, had the possibility of 16 x 48 bits of key now has only 56 bits of key. For a complete algorithmic description of the DES procedure, suitable for machine implementation, the reader is referred to Tanenbaum [TANE 81, pages 398-399]. The U.S. National Bureau of Standards has specified four general modes of operation for the use of DES [FIPS 81]. Depending on the particular application on hand, one mode may be more appropriate than the others. The four prescribed modes are as an Electronic Code Book (ECB mode), a Cipher Block Chaining scheme (CBC mode), a Cipher Feedback procedure (CF mode) and finally as the output of the Output Feedback Mode (OF mode). This mode, which is illustrated in Figure B.7, is particularly interesting for applications where data are transmitted

continuously on a medium which is the subject of transmission errors. The reason for choosing such a mode results from the fact that it is not subject to error propagation effects (i.e., a single transmission error will not result in multiple errors when decrypted). This can be seen by observing, from Figure B.7, that a bit received in error over the transmission medium will affect only one bit of the Exclusive OR gate and, as such, will not affect the DES feedback path. This becomes important for voice, facsimile and video applications where transmissions occur in real time and are not protected by error recovery schemes such as ARQ (Automatic Repeat Query) or FEC (Forward Error Correction). The other modes, on the other hand, have the serious disadvantages of causing an additional degradation in the user's bit error rate due to the error propagation effect of the inverse DES encryption function. Nevertheless, the Output Feedback mode has one potential disadvantage over the other modes in the form of an added sensitivity to jitter, bit loss and clock slippages. However, in the Time Division Environment which will be used most certainly for the application under study, the clock stability at the earth station is believed to be steady enough so as to make this problem of marginal concern at the present time.

B.2.2 The DES Controversy

At the time the National Bureau of Standards advertised in the Federal register for anyone interested to submit proposals for a Data Encryption Standard, IBM responded with a version of their Lucifer scheme [SMIT 71] that had been developed in about 1970. The scheme submitted was substantially identical in design to Lucifer with seemingly minor changes. The big change, as it turned out, was that the key size was reduced from 128 bits in Lucifer to 56 bits in the scheme submitted to NBS.

No other schemes were submitted that NBS felt worth pursuing and the IBM scheme was proposed for adoption. The proposal was advertised in the Federal Register in March 1975. Comments were invited from within the government and from the public.

The proposal as published was misleading in a number of ways. In particular, the proposal spoke of a key length of 64 bits and it required careful reading to discover that 8 of these bits were parity check bits to be thrown away by the algorithm. More important is the fact that the proposal also completely ignored any possibility of cryptanalysis.

The only substantive public comment on the proposal was from M. Hellman and W. Diffie of Stanford Unversity. They observed that the key length of 56 bits was uncomfortably short and that exhaustive key search seemed to be a possibility. They suggested that the key length be increased at least to 64 bits and if possible to 128 bits as was the case with the Lucifer case.

Diffie and Hellman were largely ignored by NBS. As a result, and in order to get a wider hearing, they published a letter in the Communications of the ACM in early 1976 [DIF1 76].

There was somewhat of an uproar as a result and finally NBS had to accept the fact that there is such a thing as cryptanalysis and that somehow or other the Hellman-Diffie questions had to be answered.

Two workshops were held by NBS to "answer the criticism", as they put it. The first consisted mainly of hardware specialists and was held on August 30, 1976. This workshop consisted of experts who were well qualified to evaluate the feasibility of the cryptanalytic

scheme that had been proposed by Hellman and Diffie, and they decided that the criticism was of no concern and that their scheme was not implementable. On the other hand, many of the participants in this workshop had a strong financial interest in the success of the proposed DES scheme. In fact, they represented manufacturers who had already started development and were quite naturally hesitant to make any changes. Of course, they also stood to lose if the credibility of the DES was to be destroyed.

The second workshop was mainly composed of software experts with no financial stake in the outcome and their conclusions were not so optimistic. The second workshop ended without a consensus, but the overall outcome of the two workshops was to highlight the following points:

(1) a great deal of public noise about possible weakness of the DES,

(2) the conclusion that the Hellman-Diffie scheme was not practical at present or within a few years, but nevertheless that:

(3) the key length provided no safety margin*.

---

* It is still believed by some people, although strongly denied by both IBM and the NBS, that the National Security Agency was mostly instrumental in keeping the key length small. Furthermore, the fact that the basic ideas behind the design of the S-boxes are still classified leaves a certain doubt about the possibility that some hidden tricks might make it easy to decrypt DES, given the right information, so as to further weaken the strength of the DES in view of certain people.

It also turned out that some of the design principles used by IBM were classified and could not be divulged to the members of the workshop. This revelation made discussion of the proposed standard somewhat more difficult.

The weight of the evidence was that the effective lifetime of the standard, if adopted, would be little more than 10 years. The standard was adopted in early 1977 and became effective in July of that year.

The upshot is that, doubtless, anyone with sufficient time and money can read messages encrypted with the DES, but the amount of time and money is exceedingly large, well beyond the means available to the local burglar or even the most skilled embezzlers. As a mater of fact, the best schemes proposed to date to crack the DES still require multimillion dollar investments.

B.3 Public Key Crypto Systems (PKS)

One of the most important goal in conventional cryptographic systems is to maintain the secrecy of the key. In fact, the prerequisite to conventional cryptography is the existence of a secure channel along which keys can be transmitted. The security can then be extended to other channels of higher bandwidth and smaller delay by encrypting the message sent on them. However, in order to develop large scale and secure telecommunications networks, new approaches must be put forward. The reason can be seen mainly from the "key distribution problem" that would occur if we consider a totally switched network with n users for which it is necessary to provide $n(n-1)/2$ distinct keys. Even for a moderate size network of say $n = 1000$ users, this would amount to managing 50,000 keys, clearly a rather shaky and certainly extremely costly

enterprise. In this particular context, it turns out that the newly proposed cryptographic technique which is now known as Public Key Cryptography might provide some relief. Furthermore, such procedures provide also some added advantages over conventional schemes such as digital signature and authentication capability.

A public key cryptosystem is one in which the conversion of plaintext to ciphertext and, vice-versa, the conversion of ciphertext to plaintext are done using different keys. Furthermore, given one of the keys, it is just as difficult to discover the other as it would be to discover the plaintext given only a sample of the ciphertext. This separation of the keys for encrypting and decrypting makes it possible to disclose one (the public key) while retaining the other (the secret key).

Because the public key can be revealed without compromising the secret key, the process of providing suitable keys to the sender and the receiver ("key distribution") can be made both freer and more secure. Public key cryptosystems also make possible a new form of authentication called "digital signature". A message that has been encrypted with a secret key could only have been created by the holder of that secret key. The identity of the creator can, however, be verified by anyone who has the corresponding public key. This property (creatability by only one person but recognizability by many) allows the digital signature to play much the same role in electronic communication that a written signature plays in paper communication.

Public key cryptography was discovered in the spring of 1975 and the first paper on the subject appeared in June 1976 [DIF2 76]. In the intervening four years three major approaches, drawn from different areas of mathematics, have been found for implementing it.

B.3.1 One-Way-Trapdoor Functions And Complexity

The Public cryptosystems all make use of one-to-one functions which are difficult to invert unless some side information (so called Trapdoor information) is used to help in the process of inversion.

Loosely speaking, we can define a function $f:A \to B$ as a one-way function, provided:

(1) f is one-to-one,

(2) f(a) is easy to compute for every $a \in A$,

(3) f^{-1} is difficult to compute* for almost all b in the range B of f.

Furthermore, whenever the domain A and the range B of f are identical, we consider f as a one-way permutation.

As a rather weak example, from the standpoint of complexity, of a one-way function we can consider the function $f_M:a \to b = Ma$, where M is a non singular n x n square matrix. Any basic algorithm to compute b requires a complexity of about n^2 operations. However, matrix inversion requires about n^3 operations so the one-way ratio in complexity is only of the order of n for this particular example.

---

* The therminology "easy" and "difficult" to compute for f and the "inverse" of f should be understood as follows: an "easy" to compute algorithm applying to f is known to the individual trying to find a procedure to compute the inverse of f. We also defer for later the definition of complexity.

The added property which makes a one-way function into a one-way-trapdoor function can be summarized as follows:

(4) There exists some side information (the "trapdoor" information) which, whenever known, makes $f^{-1}(b)$ easy to compute for all b in the range of f.

As an example of a one-way trapdoor function, consider the enciphering function associated with any conventional cryptosystem that is secure against a chosen plaintext attack. Clearly this function is a one-way-trapdoor function and the trapdoor for this particular case is the secret key.

An interesting example which will also be of value in the sequel is furnished by logarithms in finite Galois Field with 2^m elements (hereafter noted as $GF(2^m)$). Now recall that if α is a root of a primitive polynomial of degree m whose coefficients belong to GF(2) (i.e., α is a primitive element of $GF(2^m)$), then any non zero element β in $GF(2^m)$ can be uniquely written as

$$\beta = \alpha^i \text{ for some integer i such that } 0 \leq i \leq 2^m - 1 \qquad (3.1.1)$$

which can also be rewritten as

$$i \overset{\Delta}{=} \log_\alpha \beta \qquad (3.1.2)$$

by defining i as the logarithm with respect to α of the element β. Also $\beta \in GF(2^m)$ has a unique representation as a binary m-tuple

$$\beta = (\beta_1, \beta_2, \ldots, \beta_m) \qquad \beta_i \in GF(2) = \{0,1\} \qquad (3.1.3)$$

with respect to the basis $(\alpha^0,\alpha,\alpha^2,\ldots,\alpha^{m-1})$ of $GF(2^m)$ seen as a vector space of dimension m over GF(2), while the integer i has the radix two form

$$i = (a_{m-1},\ldots,a_0) \text{ with } a_i \in \{0,1\}, \text{ and } i = \sum_{i=0}^{m} a_i 2^i. \quad (3.1.4)$$

The table which appears in Figure B.8 represents the set of all non zero elements β of GF(16) and their logarithms with respect to primitive elements α which are roots of the polynomial X^4+X+1. i.e., for example:

$$\beta = \alpha^{11} = \alpha^4\times\alpha^4\times\alpha^3 = (\alpha+1)\ (\alpha+1)\alpha^3$$

$$= \alpha^5+\alpha^3 = \alpha\alpha^4+\alpha^3 = \alpha(\alpha+1)+\alpha^3$$

$$= \alpha^3+\alpha^2+\alpha = \Rightarrow \alpha = (0111)$$

while $\log_\alpha\beta = 11 = \Rightarrow \log_\alpha\beta = (1011)$.

The following conjecture has been made about the logarithmic function as defined above. In general, given $\beta = \alpha^i$, it is very difficult to find $i = \log_\alpha\beta$. Equivalently, the inverse function $\log_\alpha^{-1}(\bullet)$ or $\exp_\alpha(\bullet)$ is a one-way permutation. (The best procedure known is that, given α^i, one cannot improve on simply guessing the value j and computing α^j until at last $\alpha^j = \alpha^i$). For further details, we refer the reader to the work of Pohlig and Hellman [POHL 78] where it is suggested to use logarithms in GF(p), where p is a large prime instead of 2^m. However, the principles do carry on. Moreover, it has been recently shown by Adelman [ADLE 80] that it is "fairly easy" to take logarithms in GF(p). He estimated about 2.6 days on a fast computer with a p value requiring of the order of 200 bits.

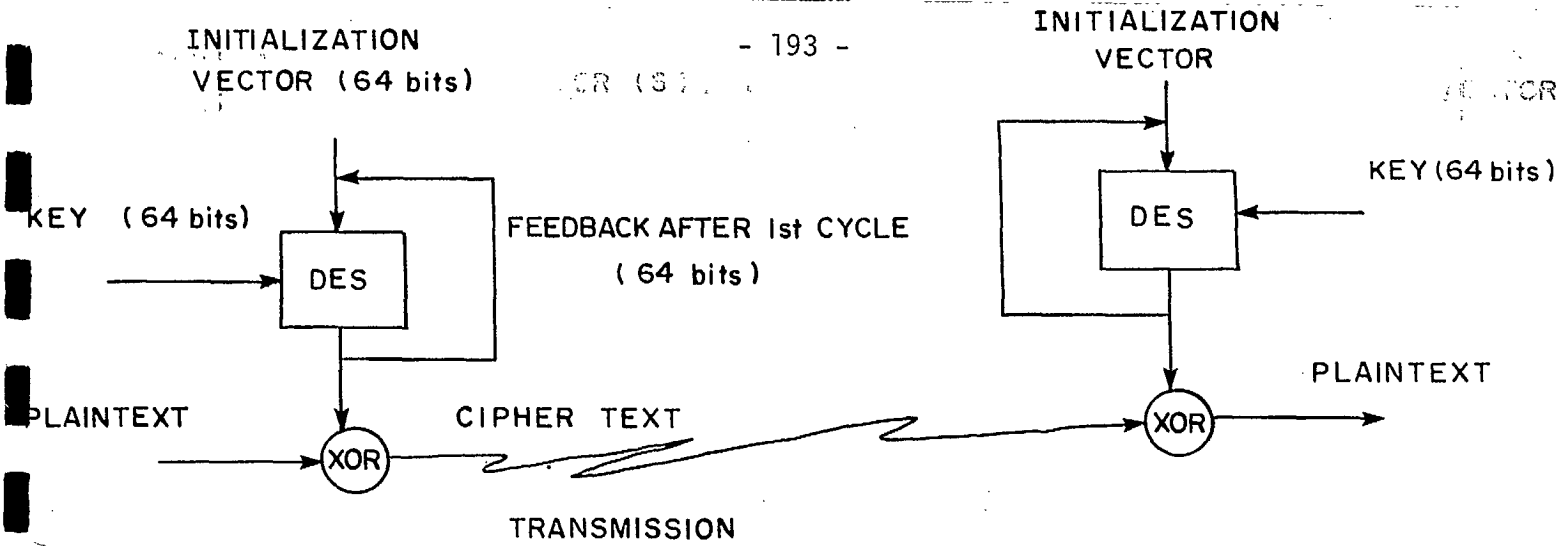


FIGURE B.7 : OUTPUT FEEDBACK MODE USING DES

β		$\log_{\alpha} \beta$
$\alpha^0 = 1$	(1000)	(0000)
$\alpha^1 = \alpha$	(0100)	(0001)
α^2	(0010)	(0010)
α^3	(0001)	(0011)
$\alpha^4 = \alpha + 1$	(1100)	(0100)
$\alpha^5 = \alpha^2 + \alpha$	(0110)	(0101)
$\alpha^6 = \alpha^3 + \alpha^2$	(0011)	(0110)
$\alpha^7 = \alpha^3 + \alpha + 1$	(1101)	(0111)
$\alpha^8 = \alpha^2 + 1$	(1010)	(1000)
$\alpha^9 = \alpha^3 + \alpha$	(0101)	(1001)
$\alpha^{10} = \alpha^2 + \alpha + 1$	(1110)	(1010)
$\alpha^{11} = \alpha^3 + \alpha^2 + \alpha$	(0111)	(1011)
$\alpha^{12} = \alpha^3 + \alpha^2 + \alpha^7 + 1$	(1111)	(1100)
$\alpha^{13} = \alpha^3 + \alpha^2 + 1$	(1011)	(1101)
$\alpha^{14} = \alpha^3 + 1$	(1001)	(1110)

FIGURE B.8 : LOGARITHMS TABLE OF THE NON ZERO ELEMENTS OF GF (16)

A newly developped area of research is probably going to be of prime importance as a source of one-way trapdoor functions, namely the theory of complexity. A function is said to belong to the complexity class P (for polynomial) if it can be determined by a deterministic automaton (for example a Turing Machine) in polynomial time when measured with respect to a parameter variable which is related to the size of the problem. One might think of this class of function as the class of easily determined function although it is more proper to say that any function which does not belong to the class P must be hard to compute for at least some inputs. There exist some problems which are known not to be in class P [AHO 74, pp. 404-425]. For example, many problem which arise in engineering cannot be solved in polynomial time unless they are run on a computer with an unlimited degree of parallelism. These problems may or may not belong to the class P, but they belong to the class NP (for nondeterministic polynomial) of problems solvable in polynomial time on a nondeterministic computer (i.e., one with an unlimited degree of parallelism). More specifically, any potential solution to an NP problem can be checked in polynomial time on a deterministic automaton. Clearly all P problems belong to the class NP, although it is not known whether or not the containment is strict. Among the problems known to be solvable in NP time, but not in P time, are versions of the traveling salesman problem, the satisfiability problem for propositional calculus, the knapsack problem, the graph coloring problem as well as many scheduling and optimisation problems. Furthermore, Karp has identified a subclass of the NP problems called NP-complete, with the additional property that if any one of them is in the class P, then so are all the other NP problems. As such, the class of NP-complete problems regroups in a sense the most "difficult to solve" types of problems. As a matter of fact, all the examples afore mentioned are known to be NP-complete. It should be clear that, in

view of the definition of the class of NP-complete functions, any function in this category should be a good candidate for a one-way trapdoor function. Examples of applications in the context of cryptography will be given very shortly.

B.3.2 Diffie-Hellman-Merkle Public Key Crypto System (PKS)

In 1976, Diffie and Hellman proposed a new revolutionary technique for data encryption which they have termed a Public Crypto-System (PKS) to stress the fact that part of the key can be made public, simplifying considerably the distribution problem [DIF3 76]. Furthermore, using this procedure, it becomes relatively simple to authentify messages and provide for an easy solution to the "digital signature" problem. Similar ideas appear in a paper by Merkle published somewhat later in the Communications of the A.C.M. [MER1 78].

Given a network with M users, a Public Crypto System for such system consists of the following entities and procedures:

(1) A one-way trapdoor function $f_i : A_i \rightarrow B_i$ for user i together with a trapdoor Z_i known only to user i.

(2) User i publishes E_i, an easy to compute algorithm for f_i in a public directory available to all M users.

(3) Whenever user j wishes to send information securely to user i, he encodes it with E_i; i.e., he sends the cryptogram

$$Y = f_i(X) = E_i(X)$$

corresponding to its plaintext X.

(4) User i decrypts such a Y by using the easy to compute deciphering algorithm D_i which he, and only he, can find from his trapdoor Z_i.

$$D_i(Y) = f_i^{-1}(Y) = f_i^{-1}[f_i(X)] = X$$

Furthermore, such a public cryptosystem provides for an easy authentication or "digital signature" scheme as we proceed to explain. In this case, we assume that $A_i = B_i = A$ for all i and signed messages can be transmitted through the network as follows.

- Public Key Authentication (Digital Signature) Scheme

(1) User j, wishing to send the signed message X to user i sends

$$Y = E_i[D_j(X)]$$

(2) User i decrypts such Y as

$$E_j[D_i(Y)] = E_j[D_i[E_i[D_j(X)]]]$$
$$= E_j[D_j(X)]$$

However, since $A_j = B_j$, it must be the case that $E_j = D_j^{-1}$ so that the last relation yields X, the plaintext.

(3) User i knows that only user j could have sent this message X to him, since $D_j(X)$ is simply the message X with user j's signature so that no other user can forge j's signature on a message.

Based on the previous discussion about logarithms on finite fields and assuming the conjecture that $\log_{\alpha}^{-1}(\bullet)$ is a one-way trapdoor function, Diffie and Hellman have proposed the following public cryptosystem assuming a system with M users.

(1) User i randomly chooses an integer X_i, $0 \leq X_i \leq p^m - 1$ and computes $Y_i = \alpha^{X_i}$ (α a primitive element in $CF(p^m)$, p a prime). Furthermore, each user publishes the corresponding value Y_i which is made available in a public directory file.

(2) Whenever users i and j wish to communicate securely, they use a conventional cryptosystem with the key

$$Z_{ij} = \alpha^{X_i X_j}$$

NOTES: The following points can be made.

A - It is very easy for users i and j to compute the value Z_{ij} since

$$Z_{ij} = (\alpha^{X_i})^{X_j} = Y_i^{X_j}$$
$$= (\alpha^{X_j})^{X_i} = Y_j^{X_i}$$

B - However, provided the conjecture is true, it is very difficult for any other user to find Z_{ij}.

C - The main public directory requires only M entries.

D - Tampering with the public directory must be impossible.

E - It is suggested by Diffie and Hellman to use m = 1 and p a large prime number.

B.3.3 The Rivest-Shamir-Adleman (RSA) Algorithm

Following the pioneer work of Hellman and Diffie, Rivest, Shamir and Adleman [RIVE 78] at MIT were the first to propose a probably implementable scheme using the Public Cryptosystem concept. The procedure for obtaining the one way trapdoor function is based on deep concepts associated with number theory.

First, we recall that any positive integer $n \in Z^+$ is a prime number provided it is only divisible by 1 and by itself. Furthermore, any non prime number $n \in Z^+$ has a unique (up to a reordering of the factors) decomposition in prime factor

$$n = \prod_{i=1}^{k} p_i^{e_i} \tag{3.3.1}$$

where p_i are prime numbers and $e_i \in Z^+$.

If n_1 and n_2 are two integers, their greatest common divisor (gcd) is the largest integer $\gcd(n_1,n_2)$ which divides both n_1 and n_2. If n_1 and n_2 have no common prime factors, their gcd is 1 and they are called relatively prime. The $\gcd(n_1,n_2)$ is easily computed through an effective computational procedure known as the Euclid's algorithm which furthermore finds integers a and b such that $\gcd(n_1,n_2) = an_1 + bn_2$.

The Euler's Totient function of any integer $n \geq 1$ is defined as the quantity which satisfies

$$\phi(n) = \begin{cases} 1 & \text{if } n = 1 \\ \text{number of positive integers smaller and relatively prime to n whenever } n \geq 2. \end{cases}$$

It is readily shown that the function $\phi(n)$ can be evaluated as

$$\phi(p) = p-1 \quad \text{if } p \text{ is prime} \tag{3.3.2}$$

and

$$\phi(n) = \prod_{i=1}^{k} p_i^{e_i-1}(p_i-1) = n \prod_{i=1}^{k} (1 - \frac{1}{p_i}) \tag{3.3.3}$$

whenever n has the factorization given by (3.3.1). In particular,

$$\phi(p_1p_2) = (p_1-1)\ (p_2-2) \tag{3.3.4}$$

so that, for example, we have $\phi(35) = \phi(7\times5) = 6\times4 = 24$.

In studying the set Z^+ of positive integers, it is often useful to partition the finite set of integers Z^+ modulo $m(m \in Z^+)$ by defining the equivalence relation

$$n_1 = n_2(\text{mod } n) <=> n_1 - n_2 \text{ is divisible by } m \tag{3.3.5}$$

If $R_m(n)$ denotes the remainder of the division of n by m then (3.3.5) implies also

$$n_1 = n_2(\text{mod } n) <=> R_m(n_1) = R_m(n_2) \tag{3.3.6}$$

It is obvious to check that $R_m(n)$ enjoys the following properties

$$\begin{aligned} &(1)\quad R_m(n_1+n_2) = R_m(R_m(n_1) + R_m(n_2)) \\ &(2)\quad R_m(n_1,n_2) = R_m(R_m(n_1)R_m(n_2)) \end{aligned} \tag{3.3.7}$$

As a consequence, Z_m^+ (the set Z^+ reduced modulo m) is in general a ring with respect to the basic addition and multiplication operations defined in (3.3.7). However, whenever m is a prime number p, Z_p^+ becomes a field (i.e., every non zero element in Z_p^+ has a multiplicative inverse) which is called a prime Galois field (GF(p)). The reason for this fact can be found in the following theorem due to Fermat.

- Fermat's theorem: if p is a prime, then any integer n satisfies

$$n^p = n(\text{mod } p)$$

In a nut shell, if we exclude n = 0, the above relation shows that every non zero element n in GF(p) has n^{p-2} as its multiplicative inverse.

Whenever m is not a prime, the generalization of the foregoing result can be summarized in the fundamental.

- Euler's theorem: if n and m are relatively prime, then

$$n^{\phi(m)} = 1(\text{mod } m)$$

Based on this theoretical background, the RSA cryptosystem can be described as follows.

- RSA Cryptosystem for M users:

(1) User i randomly generates two large primes, p_i and q_i, and forms their product $m_i = p_i q_i$. Furthermore, he chooses a large integer e_i, $e_i \leqslant m_i$, relatively prime to $\phi(m_i)$. In the process of verifying that $\gcd(e_i, \phi(m_i)) = 1$, user i has also found easily a "secret" integer d_i (trapdoor information) by applying the Euclid's algorithm to e_i and $\phi(m_i)$ such that

$$1 = ae_i + b\phi(m_i)$$

and (3.3.8)

$$d_i = R_{\phi(m_i)}(a)$$

NOTE: the value of $\phi(m_i) = (p_i-1)(q_i-1)$ is easily determined by user i knowing $m_i = p_i q_i$. It is however extremely difficult to find $\phi(m_i)$ from the knowledge of the value of m_i alone.

(2) User i publishes the pair (m_i, e_i) as found in step 1) in the public directory available to all users.

(3) Whenever user j wants to send a message to user i, he first converts the message to a string of integer value $\{X_k\}$, $0 \leq X_k \leq m_i$, and $\gcd(X_k, m_i) = 1$: (this latter condition occurs for almost all the possible values of X_k) and transmits the sequence of cryptograms $\{Y_k\}$ with

$$Y_k = R_{m_i}(X_k^{e_i})$$

(4) Whenever user i receives Y_k, he decrypts it as

$$X_k = R_{m_i}(Y_k^{d_i})$$

where d_i is a defined in (3.3.8).

A posteriori verification that the decryption works, as it is claimed above, proceeds from the following steps.

First the relation $1 = ae_i + b\phi(m_i)$ implies

$$1 = R_{\phi(m_i)}(ae_i) = R_{\phi(m_i)}(R_{\phi(m_i)}(a)e_i) = d_i e_i \bmod(\phi_{m_i})$$

where the last right identify follows from (3.3.8). This last relation implies that

$$e_i d_i = Q\phi(m_i) + 1 \text{ for some quotient } Q.$$

It then follows that

$$Y_k^{d_i} = R_{m_i}(X_k^{e_i d_i}) = R_{m_i}((X_k^{Q\phi(m_i)})X_k)$$

$$= R_{m_i}(R_{m_i}(X_k^{Q\phi(m_i)})X_k).$$

However, from the Euler's theorem $X_k^{\phi(m_i)} = 1 \bmod (m_i)$ so that

$$R_{m_i}((X_k^{Q\phi(m_i)}) = R_{m_i}(R_{m_i}(X_k^{(\phi(m_i))^Q})) = 1 \Rightarrow Y_k^{d_i} = R_{m_i}(X_k) = X_k$$

as claimed in (3.3.9).

We now conclude our introduction to the RSA algorithm by the following remarks and simple running example.

(1) The operations of encryption and decryption are functionally identical and require at most $2\log_2(m_i)$ multiplicaitons modulo m_i. Without loss in generality, let us consider the encryption procedure and let $2^{\ell-1} \leq m_i \leq 2^{\ell}$ so that $\ell \cong \log_2 m_i$, and $0 \leq e_i \leq m_i$. Consequently, e_i has the binary expansion

$$e_i = \sum_{j=0}^{\ell-1} b_j 2^j \qquad b_j \in \{0,1\}$$

It then follows that

$$X^{e_i} = \prod_{j=0}^{\ell-1} (X^{2^j})^{b_j} \Rightarrow R_{m_i}(X^{e_i}) = R_{m_i}\left[\prod_{j=0}^{\ell-1} (R_{m_i}(X^{2^j}))^{b_j}\right]$$

It is readily seen that $R_{m_i}(X^2)$, $R_{m_i}(X^4)$, ..., $R_{m_i}(X^{2^{\ell-1}})$ can be found by ℓ successive squaring of $R_{m_i}(X)$ modulo m_i. Furthermore, depending on the values of the binary vector $\{b_j\}$, at most ℓ subsequent multiplications are required yielding $2\ell \cong 2 \log_2 m_i$ as an upperbound on the total number of multiplications.

(2) The key difficulty in finding d_i (the trapdoor information) is in the determination of $\phi(m_i) = (p_i-1)(q_i-1)$. It appears, at the present time, to be equivalent to the difficulty in factoring a large number $m_i = p_i q_i$ into prime factors. Such a problem is still considered to be an extremely difficult one, mostly because of the lack of success in developping efficient procedures for its solutions in the past two centuries.

(3) As a simple example, assume that the letters {A,B,C,..Z} are encoded as the integers {1,2,...,26}. Let p = 3, q = 11 so that m = 33, $\phi(m)$ = 20. A suitable value for e is 3, so that d can be found by solving the equation 3d = 1 (mod 20) which yields d = 7. The forthcoming table summarizes the operations of the algorithm for the encoding of the plaintext "TONI".

PLAINTEXT			CIPHERTEXT	DECRYPTION		
symbolic	numeric	X^3	$Y=X^3 \pmod{33}$	Y^7	$Y^7 \pmod{33}$	symbolic
T	20	8000	14	105413504	20	T
O	15	3375	9	4782969	15	O
N	14	2744	5	78125	14	N
I	9	729	3	2187	9	I

Obviously, the reader should be cautioned about the fact that the above example is of no cryptographic value. The values of p and q should require of the order of 100 digits and they should be of slightly different length. It is expected that a Monte Carlo type of search would require of the order of log p operations to obtain a prime of the order of p while the best factorization algorithms for the integer n have complexity of order $\sqrt{n}$.

B.3.4 The Merkle-Hellman Trapdoor Knapsack Public Crypto System

In 1978 Merkle and Hellman [MER2 78] have presented an algorithm based on the knapsack problem and which is conjectured to be a Public Cryptosystem. The knapsack problem is in fact conceptually deceptively simple. Given a knapsack into which some subset of a group of N available objects, whose weights $\{a_i / i = 1,N\}$ are known in advance, are to be placed, and given Y the total weight of the knapsack, find the "choice" vector $X^N \triangleq [x_1,x_2,\ldots,x_N]$, $x_i \in \{0,1\}$, such that $x = 1$ if the i^{th} object is in the knapsack and which satisfies

$$Y = \sum_{i=1}^{N} a_i x_i \qquad (3.4.1)$$

Such a problem is known to be NP-complete since it requires at most N additions to check whether a solution satisfies (3.4.1),

whereas it takes 2^N such verifications to exhaust all possible solutions (an exceedingly large number for values of N of the order of 200). NP-complete problems are known to be hard to solve in a worst case sense. In fact, the best known general methods of solution for the knapsack problem require time proportional to $2^{N/2}$. However, not all knapsack problems are hard to solve. For example if the weight set is $\{1,2,10,15,31,61\}$ and Y = 74, the solution is found by inspection to yield

$$x_6 = 1 \Rightarrow Y = 74-61 = 13$$

$$x_5 = 0,\ x_4 = 0,\ x_3 = 1 \Rightarrow Y = 13-10 = 3$$

$$x_2 = 1 \Rightarrow Y = 3-2 = 1$$

$$x_1 = 1 \Rightarrow X^6 = (1,1,1,0,0,1)$$

In fact, this partiular knapsack belongs to a subclass of "easy" to solve knapsacks which we now define formally as

- Definition: The knapsack $\{a_1,a_2,\ldots,a_N\}$ is easy to solve provided it satisfies

$$a_i \geq \sum_{j=1}^{i-1} a_j \qquad i = 2,\ldots,N.$$

Easy knapsacks can always be solved in exactly N steps as follows. Starting from the value j = N, we set $x_j = 1$ whenever $a_j \leq Y$ and substract a_j from Y. Otherwise we set $x_j = 0$. The procedure is then repeated by decrementing j by 1 until the final value 1 is reached. The proposed Trapdoor knapsack cryptosystem then proceeds as follows.

- Merkle-Hellman Trapdoor Knapsack Public Key Cryptosystem

(1) Each user, say user i, randomly chooses two relatively prime large integers w and m such that $w < m$.

In the course of verifying that gcd(w,m) = 1, user i also finds integers c and d such that

$$1 = cw + dm$$

so that $R_m(cw) = 1$ (i.e., c is the multiplicative inverse of w modulo m).

(2) User i then randomly selects the "easy" knapsack $\{a'_1, a'_2, \ldots, a'_N\}$ with the added property that

$$\sum_{i=1}^{N} a'_i < m$$

and compute $\{a_1, a_2, \ldots, a_N\}$ with

$$a_i = c\, a'_i \bmod m \qquad i = 1, \ldots, N \tag{3.4.3}$$

(3) User i then displays the "hard" knapsack $\{a_1, a_2, \ldots, a_N\}$ into the public directory available to all users.

(4) To encrypt his plaintext X, intended for user i, user j partitions X into a sequence of blocks of N bits $(x_1, x_2, \ldots, x_N)$ and transmits the corresponding sequence of knapsack weights

$$Y = \sum_{i=1}^{N} a_i x_i$$

(5) To decrypt the message, user i simply solves the easy knapsack $Y' = Yc \bmod m$ to obtain the corresponding plaintext.

A posteriori verification of the validity of the decryption scheme proceeds as follows.

$$Y' = Yc \bmod m = R_m(Yc) = R_m(c \sum_{i=1}^{N} a_i x_i)$$

In view of (3.4.3), this last relation yields, as required.

$$Y' = R_m(c \sum_{i=1}^{N} a_i' w x_i) = R_m(R_m(cw) \sum_{i=1}^{N} a_i' x_i)$$

$$= R_m(\sum_{i=1}^{N} a_i' x_i) = \sum_{i=1}^{N} a_i' x_i$$

where use has been made of

$$\sum_{i=1}^{N} a_i' x_i \leqslant \sum_{i=1}^{N} a_i' < m$$

We now conclude this chapter by the following remarks.

(1) The level of confidentiality of the above cryptographic scheme can be considerably enhanced by iterating the process several times with different values for N [SHAM 80].

(2) One of the major problems with the scheme is that the public file containing the keys (the hard knapsack) must be exceedingly large, so that, in order to avoid storing the whole knapsack weights, it might be suitable that user j asks for a

copy of the file $\{a_k^i/k=1,\ldots,N\}$ whenever he wants to communicate with user i. We refer the reader to the original paper [MER2 78] for some alternatives that might provide relief in the key diffusion problem.

REFERENCES

[ADLE 80] Adleman, L., "A Subexponential Algorithm for the Discrete Logarithm Problem with Application to Cryptography", Department of Mathematics and Laboratory for Computer Science, MIT, 1980.

[AHO 74] Aho, A.V., Hopcroft, J.E. and J.D. Ullman, "The Design and Analysis of Computer Algorithms, Addison-Wesley, Reading, MA., 1974.

[DIF1 76] Diffie, W. and Hellman, M.E., "A Critique of the Proposed Data Encryption Standard", Comm. A.C.M., Vol. 19, pp. 164-165, Mar. 1976.

[DIF2 76] Diffie, W. and Hellman, M.E., "Multiuser Cryptographic Techniques", National Computer Conference Proceedings, New-York, June 7-10, pp. 109-112, 1976.

[DIF3 76] Diffie, W. and Hellman, M.E., "New Directions in Cryptography", IEEE Trans. on Information Theory, Vol. IT-22, pp. 644-654, Nov. 1976.

[FIPS 77] U.S. Federal Information Proceessing Standards Publication 46, "Data Encryption Standard", National Bureau of Standards, Department of Commerce, Jan. 1977.

[FIPS 81] U.S. Federal Information Proceessing Standards Publication 81, "DES Modes of Operation", National Bureau of Standards, Department of Commerce, Dec. 1980.

[KAHN 67] Kahn, D., "The Codebreakers, The Story of Secret Writing", MacMillan, New-York, 1967.

[MER1 78] Merkle, R.C., "Secure Communication Over an Insecure Channel", Comm. A.C.M., Vol. 21, pp. 294-299, April 1978.

[MER2 78] Merkle, R.C. and Hellman, M.E., "Hiding Information and Signatures in Trapdoor Knapsack", IEEE Trans. on Information Theory, Vol. IT-24, pp. 525-530, Sept. 1978.

[POHL 78] Pohlig, S.C. and Hellman, M.E., "An Improved Algorithm for Computing Logarithms Over GF(p) and its Cryptographic Significance", IEEE Trans. on Information Theory, Vol. IT-24, pp. 106-110, Jan. 1978.

[RIVE 78] Rivest, R.L., Shamir, A. and Adleman, L., "On Digital Signature and Public Key Cryptosystems", Comm. A.C.M., Vol. 21, pp. 120-126, Feb. 1978.

[SHAM 80] Shamir, A. and Zippel, R., "On the Security of the Merkle-Hellman Cryptographic Scheme", IEEE Trans. on Information Theory, Vol. IT-26, pp. 339-340, May 1980.

[SHAN 49] Shannon, C., "Communication Theory of Secrecy Systems", Bell Systems J., Vol. 28, pp. 656-715, Oct. 1949.

[SMIT 71] Smith, J.L., "The Design of Lucifer, a Cryptographic Device for Data Communications", Res. Rep. RC-3326, IBM, 1971.

[TANE 81] Tanenbaum, A.S., "Computer Networks", Prentice-Hall, Englewood Cliffs, N.J., 1981.